1) Field survey (attacks/overflights)
2) Lab experiment
3) Supplement
PANELS
TABLES
SESSION INFO

Code to 'Acoustic and visual stimuli combined promote stronger responses to aerial predation in fish'

Juliane Lukas

4/14/2021

1) Field survey (attacks/overflights)

Data Input

```r
#read in the data
library(readr)
birds <- read_delim("../data/DATA_SI_BirdActivityField.csv" ,",", escape_double = FALSE, trim_ws = TRUE,
                    col_types = cols(`Bird_entry` = col_skip(), `Bird_exit` = col_skip()), na = "NA")
#data wrangling
library(tidyverse)
```

```
## ── Attaching packages ───────────────
```

```
## ✓ ggplot2 3.3.2     ✓ dplyr   1.0.2
## ✓ tibble  3.0.3     ✓ stringr 1.4.0
## ✓ tidyr   1.1.2     ✓ forcats 0.5.0
## ✓ purrr   0.3.4
```

```
## ── Conflicts ────────────────────────
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
birds$fsampling<-plyr::mapvalues(factor(birds$TimeID), from = c("1", "2", "3"), to = c("morning","midday","afternoon"))
birds$fday<-factor(birds$DayID)
# birds$PresenceTime_s[birds$PresenceTime_s==0]<-1
# summaryBy(PresenceTime_s~BirdSpecies, data=birds, FUN=c(length,sum, min,median,max,mean,sd))

#tally all bird obs per Day/Sampling
  meta.bird_df<-plyr::ddply(birds,~fday:fsampling,plyr::summarise,fsampling=first(fsampling), fday=first(fday),
                    attacks=sum(NAttacks, na.rm=TRUE), overflights=sum(Overflight, na.rm=TRUE))
  meta.bird_df<-gather(meta.bird_df, type, event, attacks:overflights, na.rm = FALSE, convert = FALSE)
  meta.bird_df$ftype<- factor(meta.bird_df$type, levels = c("overflights","attacks"))
  meta.bird_df
```

```
##    fday:fsampling fsampling fday        type event       ftype
## 1       1:morning   morning    1     attacks    14     attacks
## 2        1:midday    midday    1     attacks    90     attacks
## 3     1:afternoon afternoon    1     attacks    52     attacks
## 4       2:morning   morning    2     attacks     1     attacks
## 5        2:midday    midday    2     attacks    49     attacks
## 6     2:afternoon afternoon    2     attacks    41     attacks
## 7       3:morning   morning    3     attacks    23     attacks
## 8        3:midday    midday    3     attacks    67     attacks
## 9     3:afternoon afternoon    3     attacks    43     attacks
## 10      4:morning   morning    4     attacks     2     attacks
## 11       4:midday    midday    4     attacks    35     attacks
## 12    4:afternoon afternoon    4     attacks    67     attacks
## 13      1:morning   morning    1 overflights     8 overflights
## 14       1:midday    midday    1 overflights    19 overflights
## 15    1:afternoon afternoon    1 overflights    11 overflights
## 16      2:morning   morning    2 overflights    13 overflights
## 17       2:midday    midday    2 overflights    17 overflights
## 18    2:afternoon afternoon    2 overflights     4 overflights
## 19      3:morning   morning    3 overflights    16 overflights
## 20       3:midday    midday    3 overflights    15 overflights
## 21    3:afternoon afternoon    3 overflights     5 overflights
## 22      4:morning   morning    4 overflights     7 overflights
## 23       4:midday    midday    4 overflights    25 overflights
## 24    4:afternoon afternoon    4 overflights    16 overflights
```

Modeling

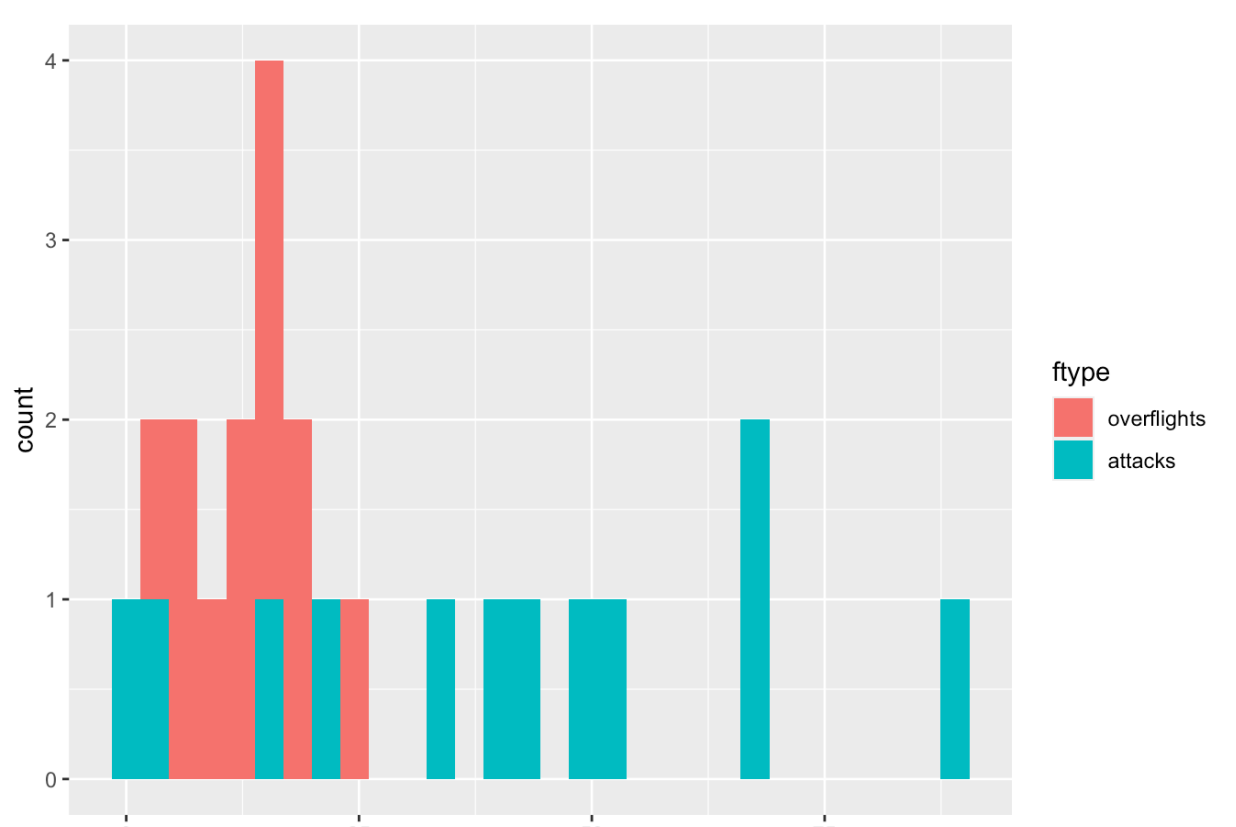
Bird disturbances - GLM

```r
#data descriptive: events per sampling
  doBy::summaryBy(event~fsampling|ftype, data=meta.bird_df, FUN=c(length,min,median,max,mean,sd))
```

```
##   fsampling       ftype event.length event.min event.median event.max
## 1   morning overflights            4         7         10.5        16
## 2   morning     attacks            4         1          8.0        23
## 3    midday overflights            4        15         18.0        25
## 4    midday     attacks            4        35         58.0        90
## 5 afternoon overflights            4         4          8.0        16
## 6 afternoon     attacks            4        41         47.5        67
##   event.mean  event.sd
## 1      11.00  4.242641
## 2      10.00 10.488088
## 3      19.00  4.320494
## 4      60.25 23.767976
## 5       9.00  5.597619
## 6      50.75 11.842719
```

```r
  ggplot2::ggplot(meta.bird_df, aes(event, fill = ftype)) + geom_histogram() #+ facet_wrap(~fsampling)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
####### GLMM #######
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

```r
  ## model fitting ##
        #fixed effect structure
        birds_gm.f0 <- glm.nb(event~1, data=meta.bird_df)
        birds_gm.samp <- glm.nb(event~fsampling, data=meta.bird_df)
        birds_gm.type <- glm.nb(event~ftype, data=meta.bird_df)
        birds_gm.2way <- glm.nb(event~ftype+fsampling, data=meta.bird_df)
        birds_gm.inter <- glm.nb(event~ftype*fsampling, data=meta.bird_df)

        drop1(birds_gm.inter, test="Chisq")
```

```
## Single term deletions
##
## Model:
## event ~ ftype * fsampling
##                 Df Deviance    AIC    LRT Pr(>Chi)
## <none>               29.221 185.24
## ftype:fsampling  2   44.998 197.02 15.777 0.000375 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
        bbmle::AICctab(birds_gm.f0, birds_gm.samp, birds_gm.type, birds_gm.inter,birds_gm.2way,
               base=TRUE, weights=TRUE,delta=TRUE,sort=TRUE)
```

```
##                AICc  dAICc df weight
## birds_gm.inter 194.2   0.0 7  0.8691
## birds_gm.2way  198.4   4.1 5  0.1095
## birds_gm.type  201.9   7.7 3  0.0186
## birds_gm.samp  206.0  11.7 4  0.0025
## birds_gm.f0    210.3  16.0 2  <0.001
```

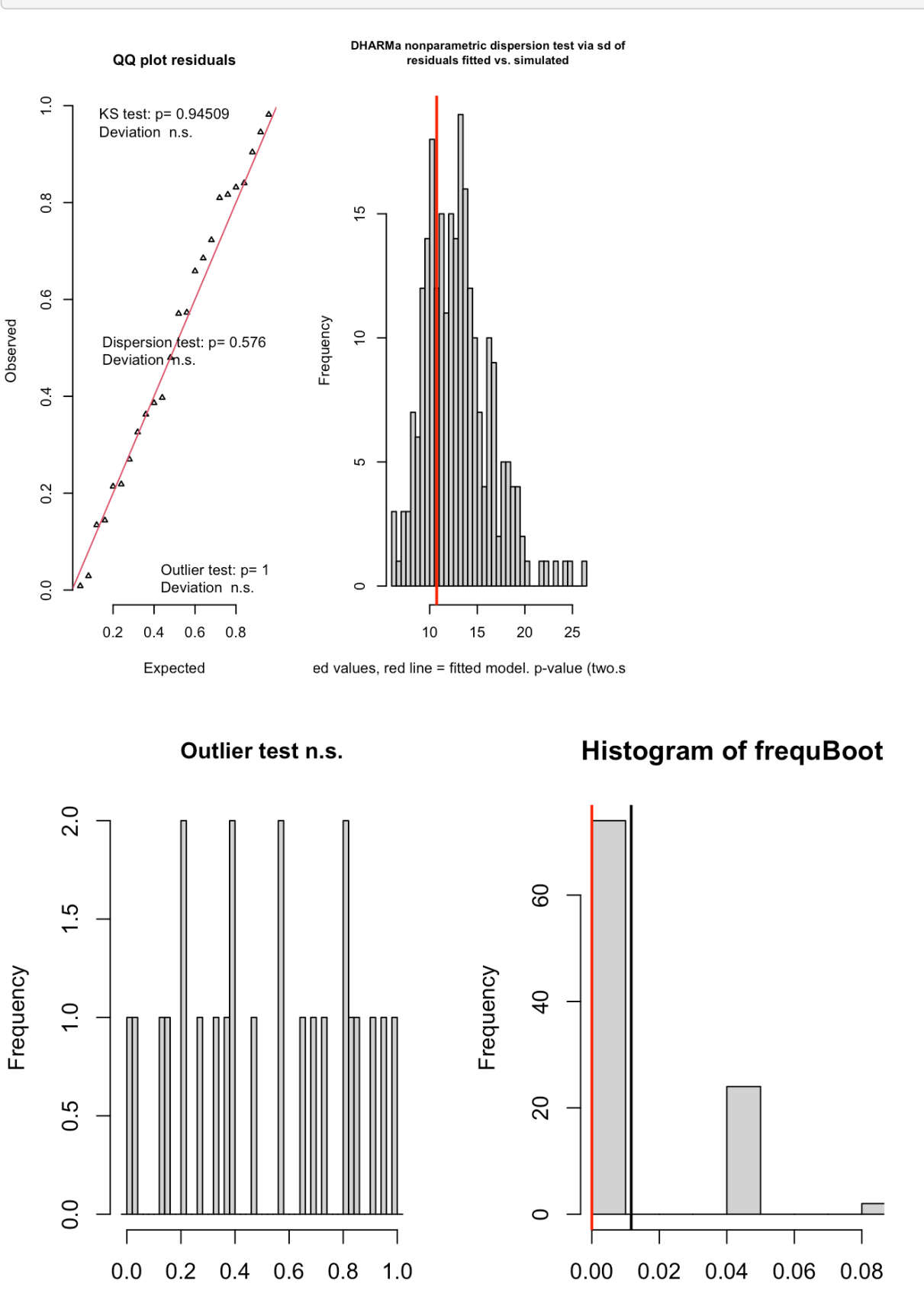
```r
        summary(birds_gm<-birds_gm.inter)
```

```
##
## Call:
## glm.nb(formula = event ~ ftype * fsampling, data = meta.bird_df,
##     init.theta = 7.284773703, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7532  -0.6772  -0.1246   0.6856   2.0845
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     2.39790    0.23884  10.040  < 2e-16 ***
## ftypeattacks                   -0.09531    0.34112  -0.279  0.77994
## fsamplingmidday                 0.54654    0.32330   1.691  0.09093 .
## fsamplingafternoon             -0.20067    0.34517  -0.581  0.56099
## ftypeattacks:fsamplingmidday    1.24937    0.44979   2.778  0.00547 **
## ftypeattacks:fsamplingafternoon 1.82500    0.46659   3.911 9.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(7.2848) family taken to be 1)
##
##     Null deviance: 110.184  on 23  degrees of freedom
## Residual deviance:  29.221  on 18  degrees of freedom
## AIC: 187.24
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  7.28
##           Std. Err.:  3.51
##
##  2 x log-likelihood:  -173.24
```

```r
  ## model diagnostics ##
        library(DHARMa)
```

```
## This is DHARMa 0.3.3.0. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa') Note: Syntax of plotResiduals has changed in 0.3.0, see ?plotResiduals for details
```

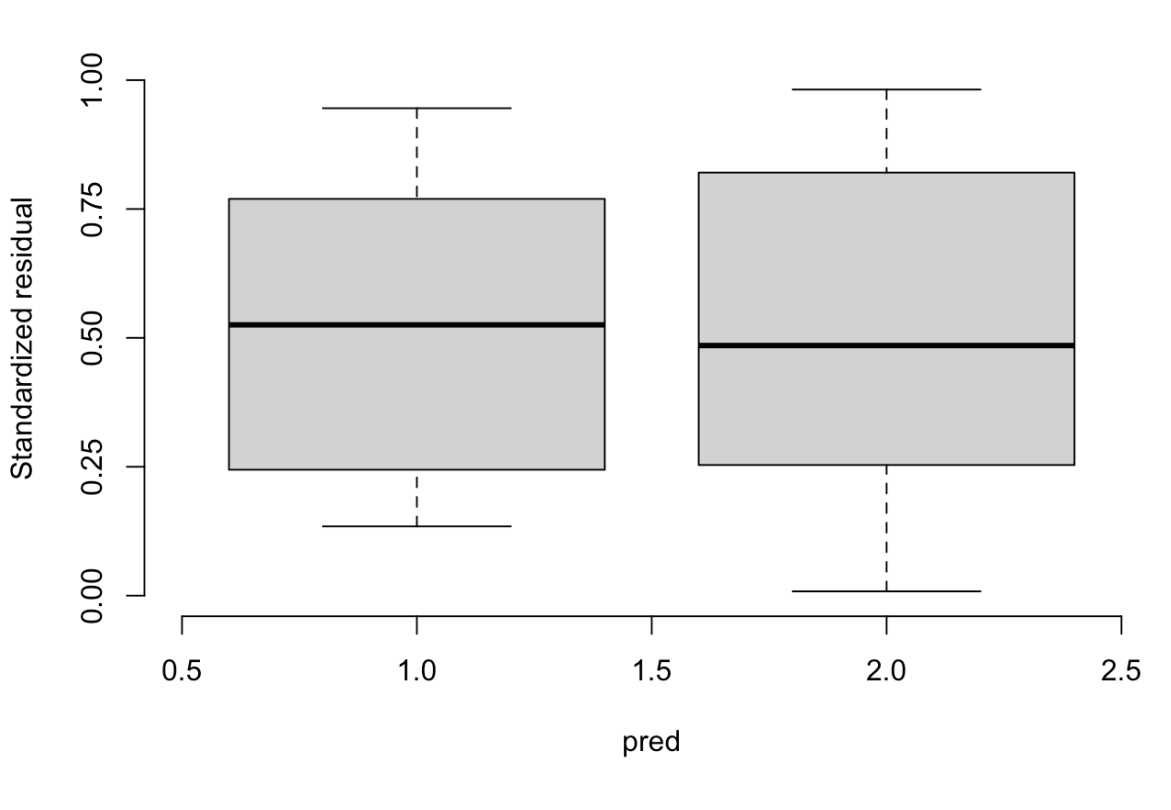
```r
        birds_gm.sim <- simulateResiduals(birds_gm)
        testResiduals(birds_gm.sim, plot=T)
```



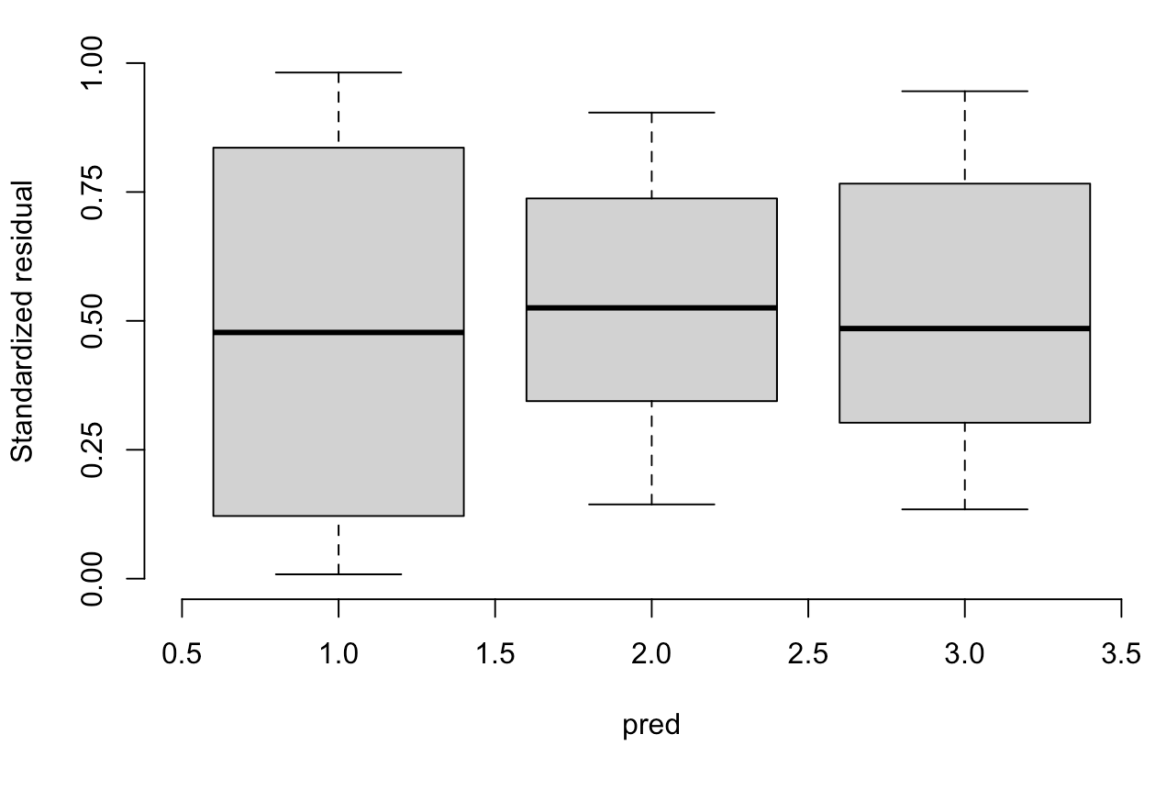
```
## $uniformity
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.10139, p-value = 0.9451
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.82763, p-value = 0.576
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutput
## outliers at both margin(s) = 0, observations = 24, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.00000000 0.04166667
## sample estimates:
## outlier frequency (expected: 0.0116666666666667 )
##                                                 0
```

```
## $uniformity
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.10139, p-value = 0.9451
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.82763, p-value = 0.576
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutput
## outliers at both margin(s) = 0, observations = 24, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.00000000 0.04166667
## sample estimates:
## outlier frequency (expected: 0.0116666666666667 )
##                                                 0
```

```r
        plotResiduals(birds_gm.sim, meta.bird_df$ftype)
```



```r
        plotResiduals(birds_gm.sim, meta.bird_df$fsampling)
```



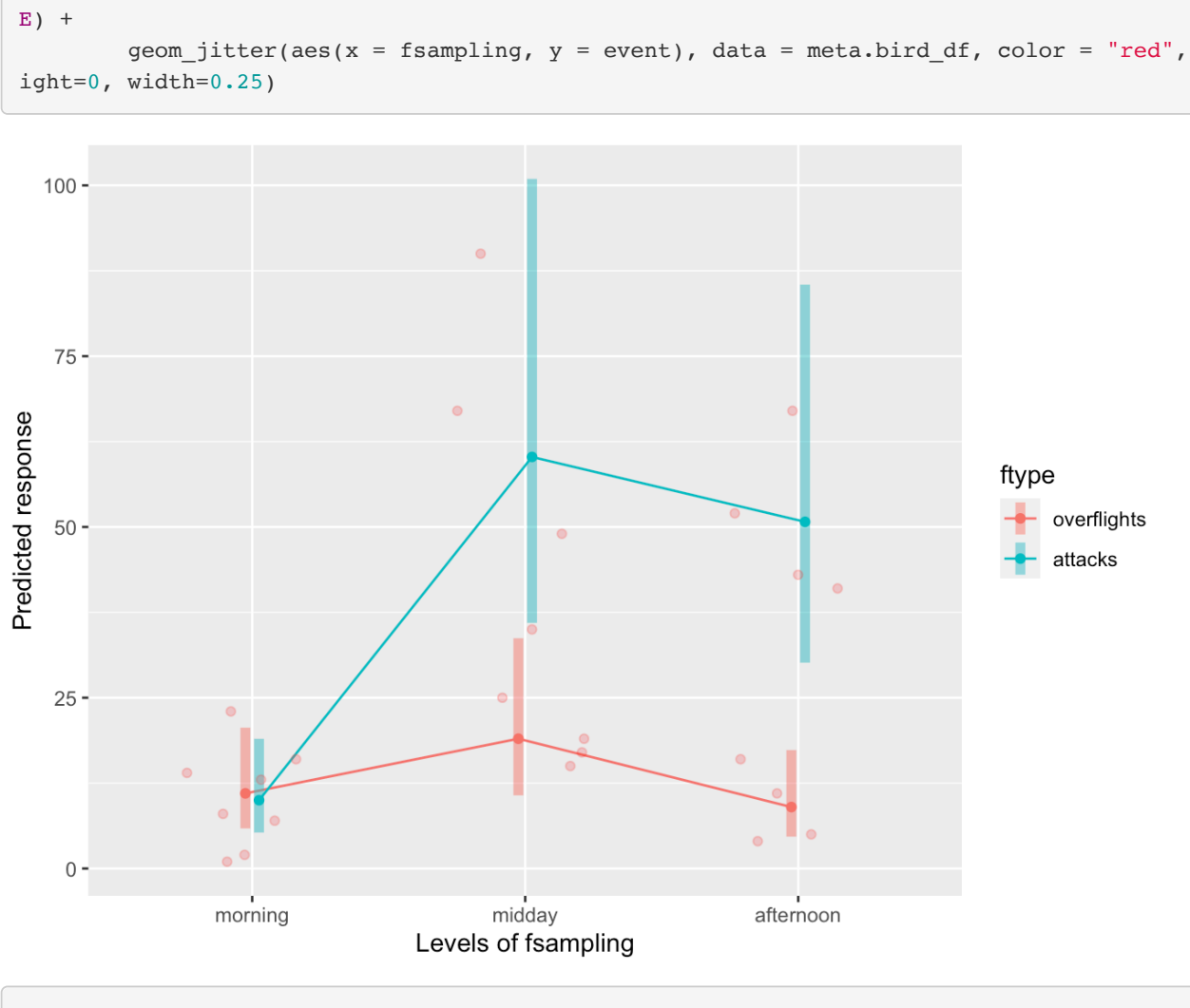
```r
  ## model interference ##
        library(emmeans)
        joint_tests(birds_gm)
```

```
##  model term      df1 df2 F.ratio p.value
##  ftype             1 Inf  25.607 <.0001
##  fsampling         2 Inf  13.581 <.0001
##  ftype:fsampling   2 Inf   7.917 0.0004
```

```r
        (birds_gm_EMMs<-emmip(birds_gm, ftype ~ fsampling, type = "response", adjust = "mvt", CIs = TRUE, plotit=FALSE))
```

```
##         ftype fsampling  yvar        SE  df       LCL       UCL        tvar
## 1 overflights   morning 11.00  2.627260 Inf  5.867562  20.62185 overflights
## 2     attacks   morning 10.00  2.435532 Inf  5.268429  18.98099     attacks
## 3 overflights    midday 19.00  4.139910 Inf 10.709303  33.70901 overflights
## 4     attacks    midday 60.25 11.816918 Inf 35.960781 100.94504     attacks
## 5 overflights afternoon  9.00  2.242715 Inf  4.671771  17.33818 overflights
## 6     attacks afternoon 50.75 10.053659 Inf 30.133990  85.47034     attacks
##        xvar
## 1   morning
## 2   morning
## 3    midday
## 4    midday
## 5 afternoon
## 6 afternoon
```

```r
        emmip(birds_gm, ftype ~ fsampling, type = "response", adjust = "mvt", CIs = TRUE, plotit=TRUE) +
        geom_jitter(aes(x = fsampling, y = event), data = meta.bird_df, color = "red", alpha=0.2, height=0, width=0.25)
```

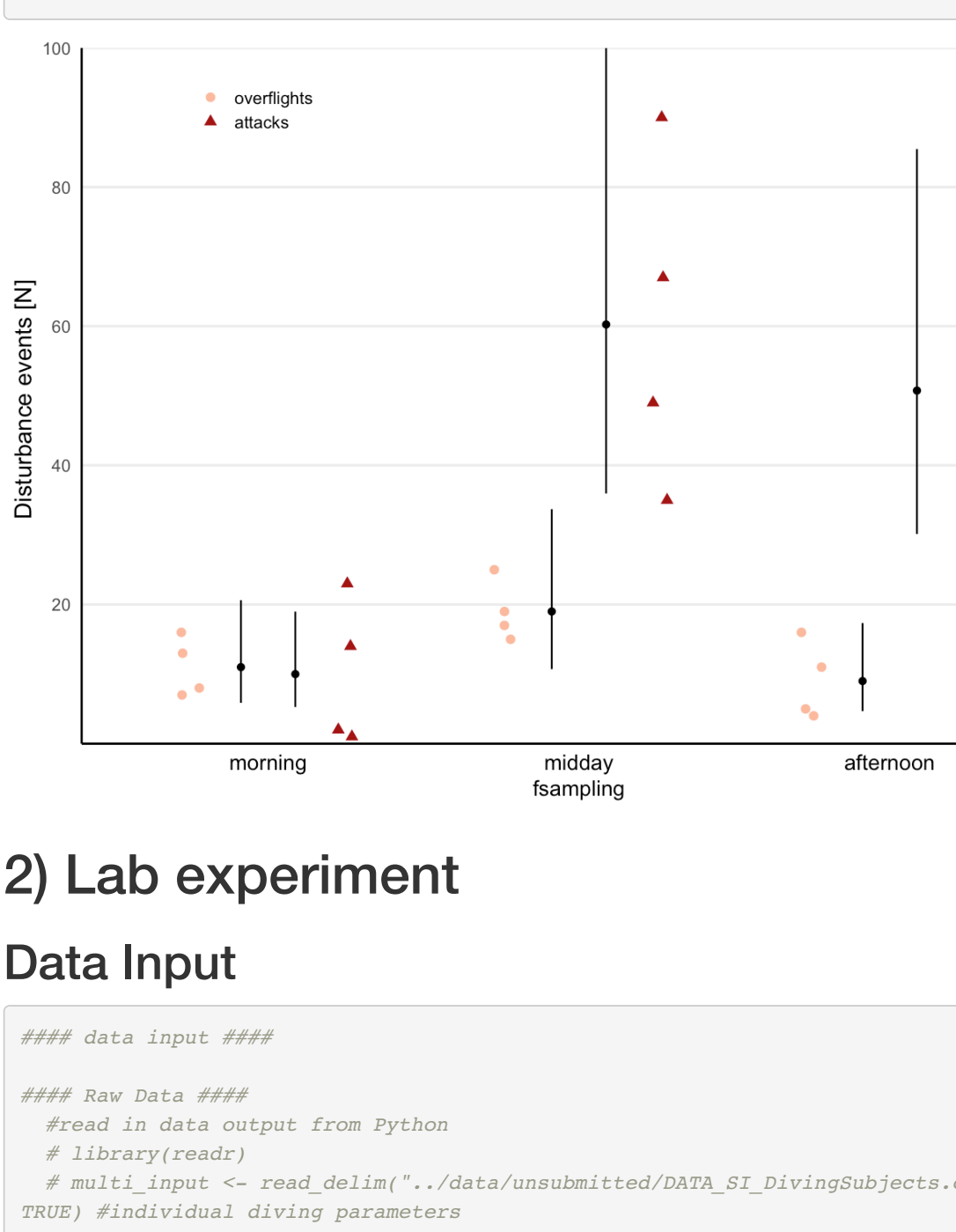


```r
        emmeans(birds_gm, pairwise ~ ftype|fsampling, adjust="mvt", type = "response")
```

```
## $emmeans
## fsampling = morning:
##  ftype       response    SE  df asymp.LCL asymp.UCL
##  overflights     11.0  2.63 Inf      6.45      18.8
##  attacks         10.0  2.44 Inf      5.80      17.2
##
## fsampling = midday:
##  ftype       response    SE  df asymp.LCL asymp.UCL
##  overflights     19.0  4.14 Inf     11.67      30.9
##  attacks         60.2 11.82 Inf     38.86      93.4
##
## fsampling = afternoon:
##  ftype       response    SE  df asymp.LCL asymp.UCL
##  overflights      9.0  2.24 Inf      5.15      15.7
##  attacks         50.8 10.05 Inf     32.59      79.0
##
## Confidence level used: 0.95
## Conf-level adjustment: mvt method for 2 estimates
## Intervals are back-transformed from the log scale
##
## $contrasts
## fsampling = morning:
##  contrast              ratio     SE  df z.ratio p.value
##  overflights / attacks 1.100 0.3752 Inf  0.279  0.7799
##
## fsampling = midday:
##  contrast              ratio     SE  df z.ratio p.value
##  overflights / attacks 0.315 0.0924 Inf -3.937  0.0001
##
## fsampling = afternoon:
##  contrast              ratio     SE  df z.ratio p.value
##  overflights / attacks 0.177 0.0565 Inf -5.433  <.0001
##
## Tests are performed on the log scale
```

Data Visualization

```r
##Plots
library(ggplot2)
## Plot raw data and mean_95CI
        set.seed(47); (p_birds<-  #set seed to make it reproducible
        ggplot(meta.bird_df, aes(x = fsampling, y = event, group=ftype)) +
        geom_jitter(aes(color= ftype, shape=ftype),
                    size=1.75, position = position_jitterdodge(jitter.height=0, jitter.width=0.15, dodge.width=1)) +
        geom_pointrange(data=birds_gm_EMMs, aes(x=fsampling, y=yvar, ymin = LCL, ymax = UCL, group=ftype),
                        position = position_dodge(0.35), color="black", size=0.4, fatten=1.2) +
        scale_color_manual(values = c("#FCBBA1", "#A50F15")) +
        scale_y_continuous(limits = c(0, 100), breaks=c(20,40,60,80,100),
                           oob = scales::squish, expand = c(0, 0), ) + # force through 0,0 and squish instead of censoring out-of-bound points
        theme_minimal()+
        theme(plot.title = element_blank(), panel.background = element_blank(),
              panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
              legend.position = c(0.175, 0.92), legend.text=element_text(size=8), legend.key.size = unit(0.4, "cm"),
              axis.line = element_line(colour = "black"),
              axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 10, color="black"),
              axis.title.y =element_text(size=11), axis.title.x =element_text(size=10)) +
        labs(y="Disturbance events [N] ", color=NULL, shape=NULL))
```



2) Lab experiment

Data Input

```r
#### data input ####

#### Raw Data ####
  #read in data output from Python
  # library(readr)
  # multi_input <- read_delim("../data/unsubmitted/DATA_SI_DivingSubjects.csv", delim=",", trim_ws = TRUE) #individual diving parameters
  # multi_coll<-read_delim("../data/unsubmitted/DATA_SI_DivingCollective.csv", delim=",", trim_ws = TRUE) #collective diving parameters
  # multi_diving<-read_delim("../data/unsubmitted/DATA_SI_DivingOverview.csv", delim=",", trim_ws = TRUE) #experimental design details
  #
  # #set up variables
  # library(tidyverse)
  # multi_df<-lapply(list(multi_input, multi_coll, multi_diving),function(df){
  #               df$fgroup<-factor(df$run)
  #               df$trial<-as.numeric(df$trial)
  #               df$fstimulus<-recode_factor(as.factor(df$stimulus), 'visual'="visual", 'sound'="acoustic", 'both'="bimodal",
  #                                            .default = levels(as.factor(df$stimulus)), .ordered=FALSE); #recode/order stimulus
  #         df})
  # #apply function to existing dataframes
  # multi_input<-multi_df[[1]]
  # multi_coll<-multi_df[[2]]
  # multi_diving<-multi_df[[3]]
  #
  # #percentage of untracked dives
  # sum((multi_diving$Nfish_group-as.numeric(multi_diving$Nfish_tracked)), na.rm=TRUE)/sum(multi_diving$Nfish_group, na.rm=TRUE)

#### Group Means ####
  # library(tidyverse)
  #   multi_groupmean<-plyr::ddply(multi_input,~fgroup:as.factor(trial),plyr::summarise,
  #                     fgroup=first(fgroup), trial=first(trial), fstimulus=first(droplevels(fstimulus)), N_fish=sum(!total_time==0,na.rm=TRUE),
  #                     mean_dive_depth= mean(-dive_depth,na.rm=TRUE),
  #                     mean_dive_speed_max= mean(dive_speed_max,na.rm=TRUE),
  #                     mean_dive_time= mean(dive_time,na.rm=TRUE),
  #                     mean_total_time= mean(total_time,na.rm=TRUE))

#### Merge dataframes ####
  # multi_mean<-merge(multi_groupmean, multi_coll[c("fgroup", "trial", "depth_coeffvar_aligned", "order_aligned")],
  #                   by=c("fgroup","trial"))
  # multi_mean<-merge(multi_mean, multi_diving[c("fgroup", "trial", "Nfish_group", "DiveRate")],
  #                   by=c("fgroup","trial"))
  # names(multi_mean)[names(multi_mean) == "depth_coeffvar_aligned"] <- "cohesion"
  # names(multi_mean)[names(multi_mean) == "order_aligned"] <- "polarization"
  # multi_mean$fstimulus<-recode_factor(multi_mean$fstimulus, 'visual'="visual", 'acoustic'="acoustic", 'bimodal'="bimodal",
  #                                            .default = levels(multi_mean$fstimulus), .ordered=FALSE)
  # multi_mean <- multi_mean[order(multi_mean$'fgroup:as.factor(trial)'), ] %>% #order dataframe by group:trial
  #    dplyr::mutate_if(is.numeric, round, digits=3) %>%
  #    dplyr::group_by(fgroup, fstimulus)  %>% droplevels()
  #
  #write.csv(multi_mean,"../data/DATA_SI_DivingGroupmeans.csv", row.names = FALSE)

#library(readr)
multi_mean<-read_delim("../data/DATA_SI_DivingGroupmeans.csv", delim=",", trim_ws = TRUE)
```

```
## Parsed with column specification:
## cols(
##   fgroup = col_double(),
##   trial = col_double(),
##   fgroup_trial = col_character(),
##   fstimulus = col_character(),
##   N_fish = col_double(),
##   Nfish_group = col_double(),
##   DiveRate = col_double(),
##   mean_dive_depth = col_double(),
##   mean_dive_speed_max = col_double(),
##   mean_dive_time = col_double(),
##   mean_total_time = col_double(),
##   cohesion = col_double(),
##   polarization = col_double()
## )
```

```r
multi_mean$fgroup<-factor(multi_mean$fgroup)
multi_mean$fgroup_trial<-factor(multi_mean$fgroup_trial)
multi_mean$fstimulus<-recode_factor(as.factor(multi_mean$fstimulus), 'visual'="visual", 'acoustic'="acoustic", 'bimodal'="bimodal",
                                    .default = levels(as.factor(multi_mean$fstimulus)), .ordered=FALSE); #recode/order stimulus

#### descriptives ####
    aggregate(multi_mean$mean_dive_depth~multi_mean$fstimulus, FUN=length)
```

```
##   multi_mean$fstimulus multi_mean$mean_dive_depth
## 1               visual                         43
## 2             acoustic                         48
## 3              bimodal                         49
```

```r
    xtabs(~fstimulus+fgroup, data=multi_mean)
```

```
##           fgroup
## fstimulus   1  2  3  4  5
##   visual    7 10  8 10  8
##   acoustic  9 10 10 10  9
##   bimodal   9 10 10 10 10
```

```r
    xtabs(~fstimulus+trial, data=multi_mean)
```

```
##           trial
## fstimulus  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##   visual   1 0 2 2 2 2 1 2 1  2  1  3  2  1  0  2  1  0  3  1  1  1  1  3  0  1
##   acoustic 2 3 1 2 1 2 4 0 0  1  3  2  1  1  3  2  3  1  2  1  1  2  0  2  2  4
##   bimodal  2 2 2 1 2 1 0 3 2  2  1  0  2  2  2  1  0  4  0  2  3  2  4  0  2  0
##           trial
## fstimulus  27 28 29 30
##   visual    2  2  2  1
##   acoustic  0  1  1  0
##   bimodal   3  1  1  2
```

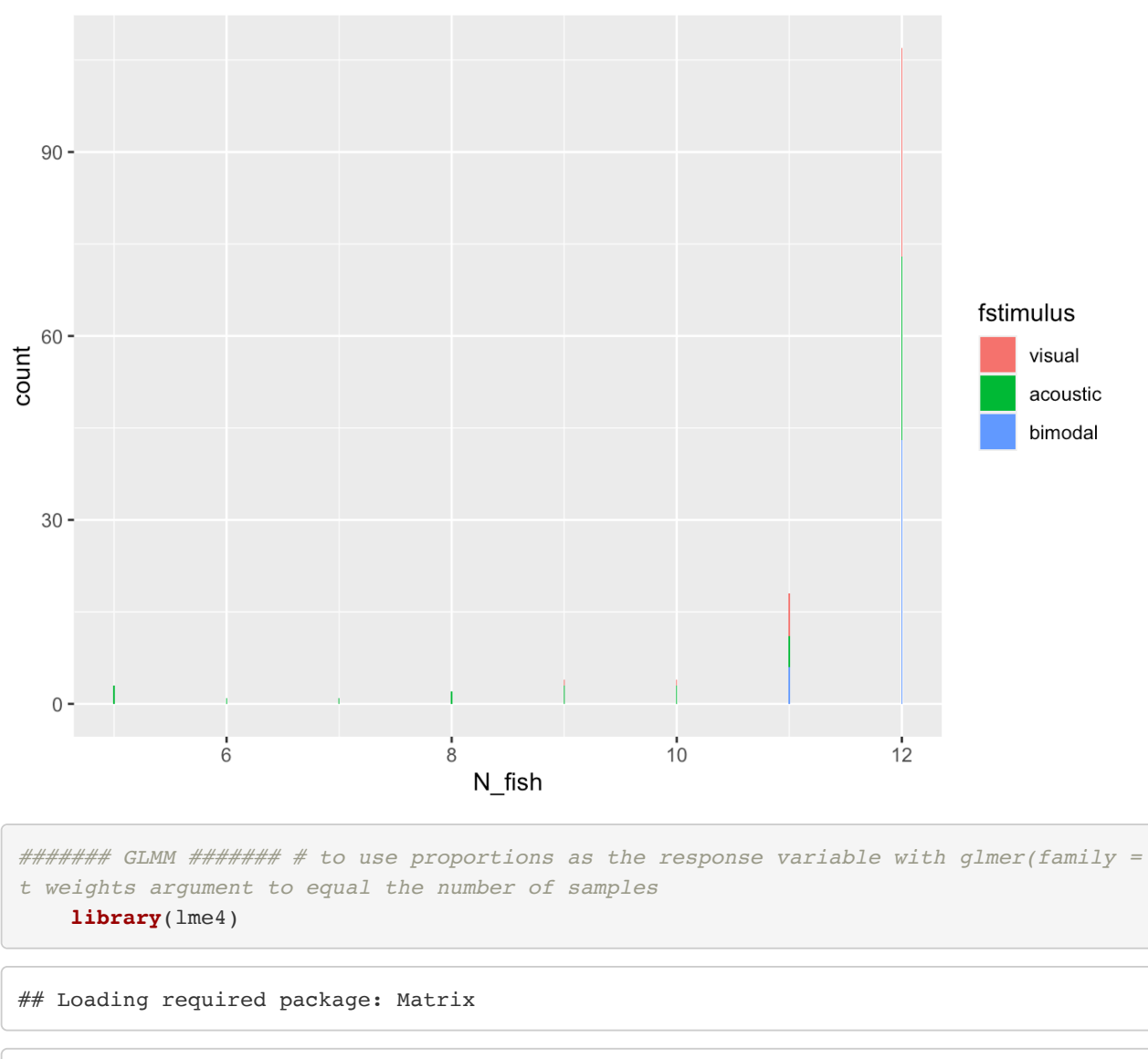
Modeling

Responsiveness - GLMER

```r
mean(multi_mean$DiveRate,na.rm=TRUE)
```

```
## [1] 0.9601429
```

```r
ggplot2::ggplot(multi_mean, aes(N_fish, fill = fstimulus)) + geom_histogram(binwidth = 0.01) #+ facet_wrap(~fstimulus)
```



```r
####### GLMM ####### # to use proportions as the response variable with glmer(family = binomial), set weights argument to equal the number of samples
    library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
    library(tidyverse)

  #random effect structure: varying Intercept best represents exp. design
  #fixed effect structure
      diving_gm.f0<- glmer(DiveRate~1 + (1|fgroup), weights=Nfish_group,
                          family=binomial(), data=drop_na(multi_mean, DiveRate))
```

```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!
```

```r
      diving_gm.stimulus<- glmer(DiveRate~fstimulus + (1|fgroup), weights=Nfish_group,
                          family=binomial(), data=drop_na(multi_mean, DiveRate))
```

```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!
```

```r
      diving_gm.trial<- glmer(DiveRate~trial + (1|fgroup), weights=Nfish_group,
                          family=binomial(), data=drop_na(multi_mean, DiveRate))
```

```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!
```

```r
      diving_gm.inter <- glmer(DiveRate~fstimulus*trial + (1|fgroup), weights=Nfish_group,
                          family=binomial(), data=drop_na(multi_mean, DiveRate))
```

```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!
```

```r
      diving_gm.2way <- update(diving_gm.inter, . ~ . -fstimulus:trial)
```

```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!
```

```r
      bbmle::AICctab(diving_gm.f0, diving_gm.stimulus, diving_gm.trial, diving_gm.inter,diving_gm.2way,
y,
        base=TRUE, weights=TRUE,delta=TRUE,sort=TRUE)
```

```
##                    AICc  dAICc df weight
## diving_gm.stimulus 185.6   0.0 4  0.677
## diving_gm.2way     187.7   2.1 5  0.234
## diving_gm.inter    189.6   4.1 7  0.089
## diving_gm.f0       272.5  86.9 2  <0.001
## diving_gm.trial    274.6  89.0 3  <0.001
```

```
    # F tests do not work for GLM(M)s (F tests are based on uncertainty of the estimate of the res
idual variance, but GLMs don't estimate residual variance)
    # instead, LRT-tests with dropping parameters, see http://bbolker.github.io/mixedmodels-misc/g
lmmFAQ.html#model-summaries-goodness-of-fit-decomposition-of-variance-etc.
    drop1(diving_gm.2way,test="Chisq")
```

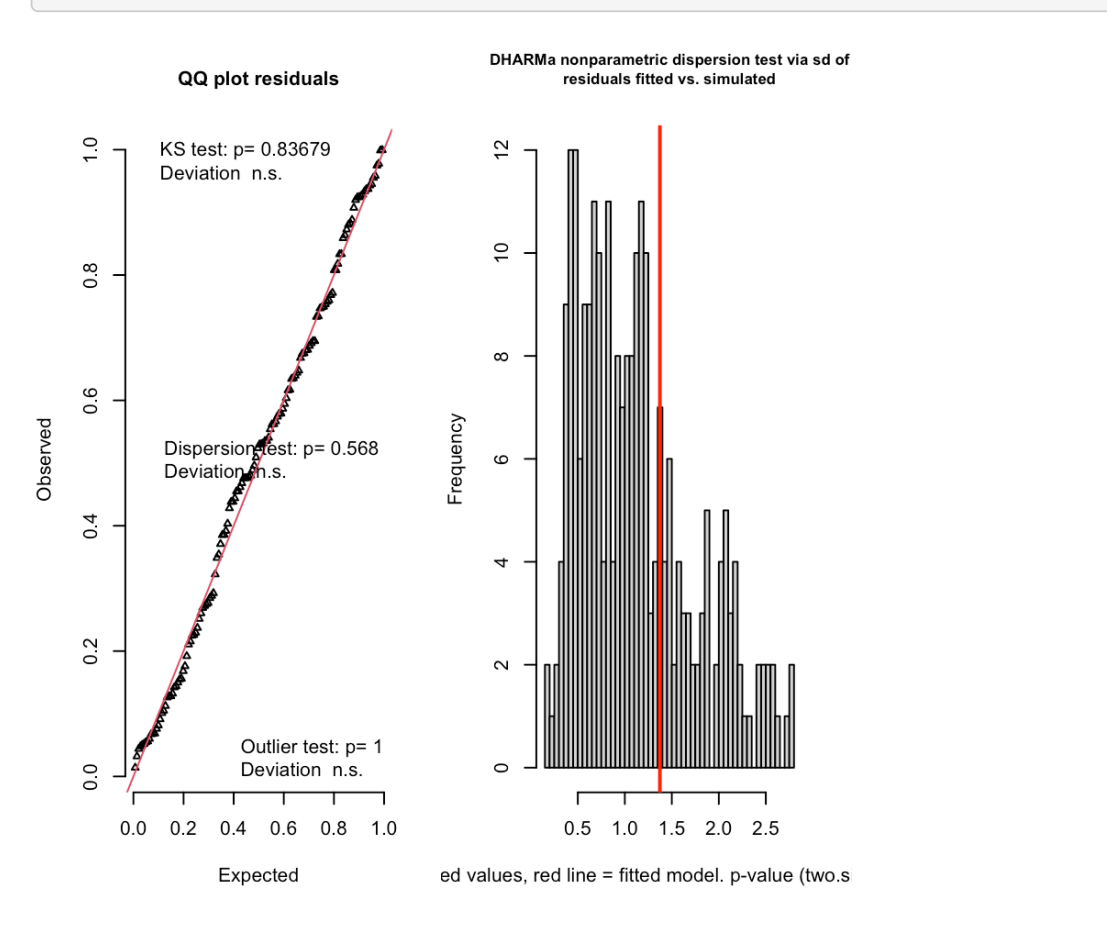
```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!

## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!
```

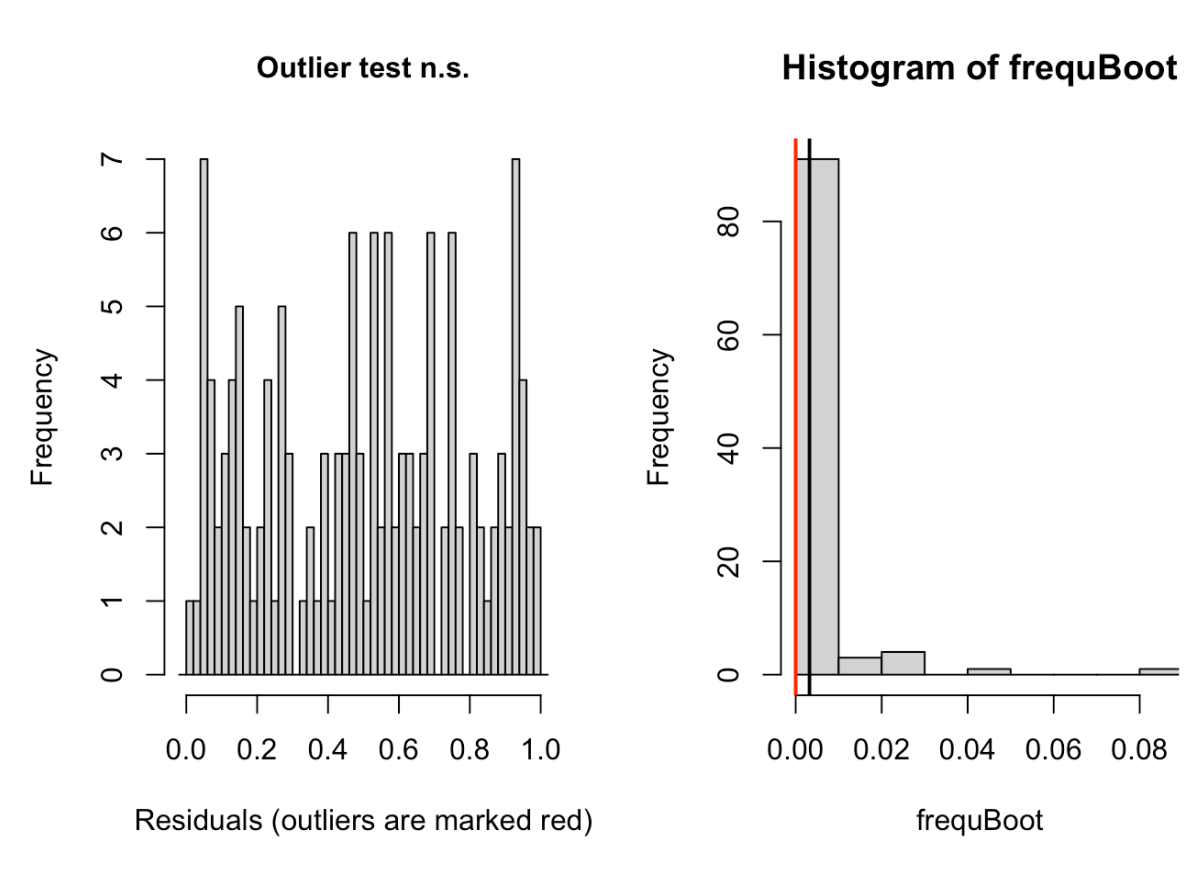
```
## Single term deletions
##
## Model:
## DiveRate ~ fstimulus + trial + (1 | fgroup)
##           npar    AIC    LRT Pr(Chi)
## <none>         187.24
## fstimulus    2 274.40 91.162  <2e-16 ***
## trial        1 185.26  0.024   0.876
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
    ## model diagnostics ##
    summary(diving_gm_final <- diving_gm.2way)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: DiveRate ~ fstimulus + trial + (1 | fgroup)
##    Data: drop_na(multi_mean, DiveRate)
## Weights: Nfish_group
##
##      AIC      BIC   logLik deviance df.resid
##    187.2    201.9    -88.6    177.2      135
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -9.1292  0.0685  0.1598  0.4839  2.5818
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  fgroup (Intercept) 3.304    1.818
## Number of obs: 140, groups:  fgroup, 5
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        5.67807    1.00135   5.670 1.42e-08 ***
## fstimulusacoustic -2.54139    0.47467  -5.354 8.60e-08 ***
## fstimulusbimodal   0.75869    0.73003   1.039    0.299
## trial              0.00262    0.01656   0.158    0.874
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fstmlsc fstmlsb
## fstimlscstc -0.433
## fstimlsbmdl -0.272  0.581
## trial       -0.230 -0.012  -0.008
```

```
    library(DHARMa)
    diving_gm_final.sim <- simulateResiduals(diving_gm_final) #use simulated data from the fitted
model to calculate an empirical cumulative density function
    testResiduals(diving_gm_final.sim, plot=T)
```

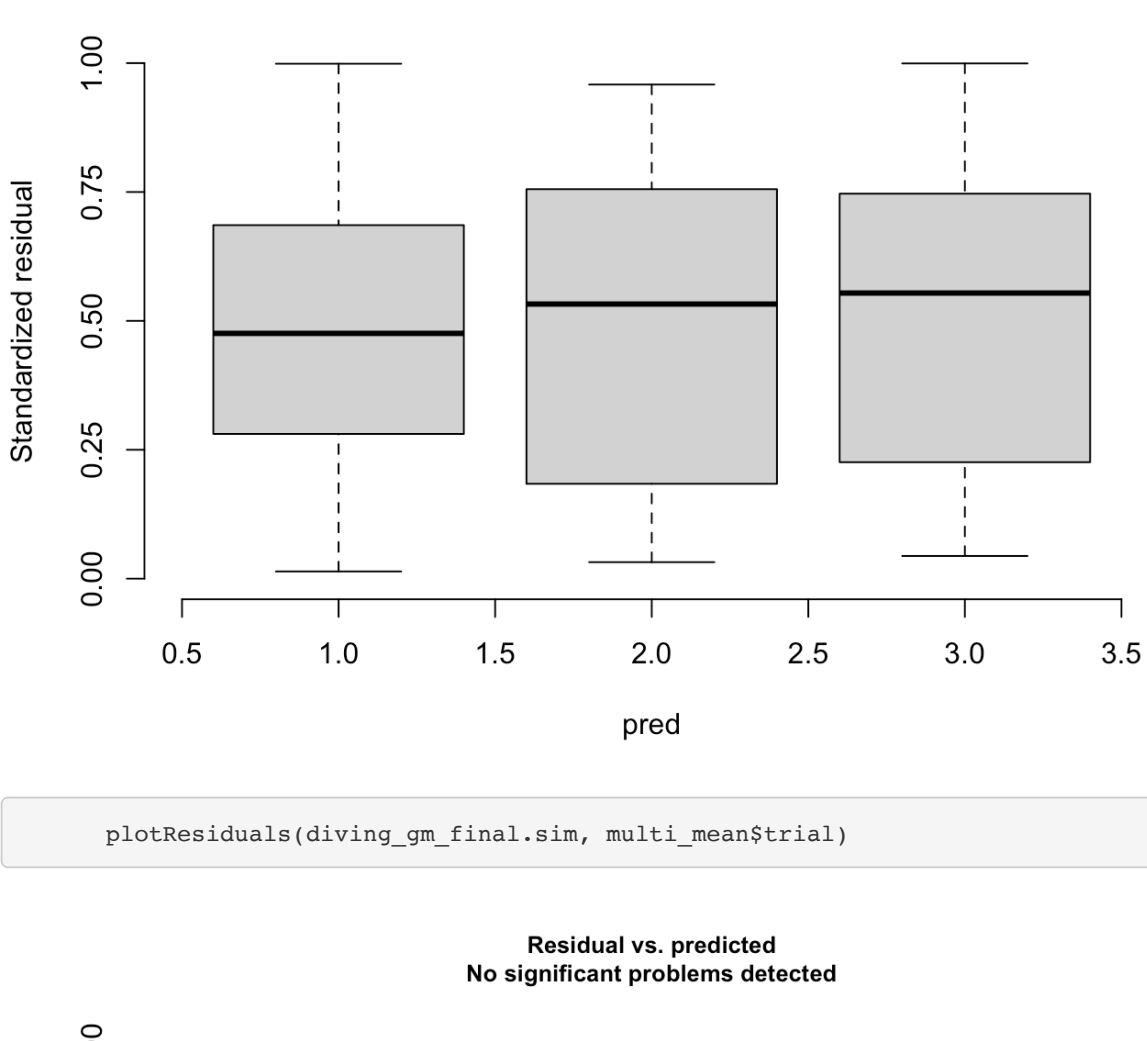




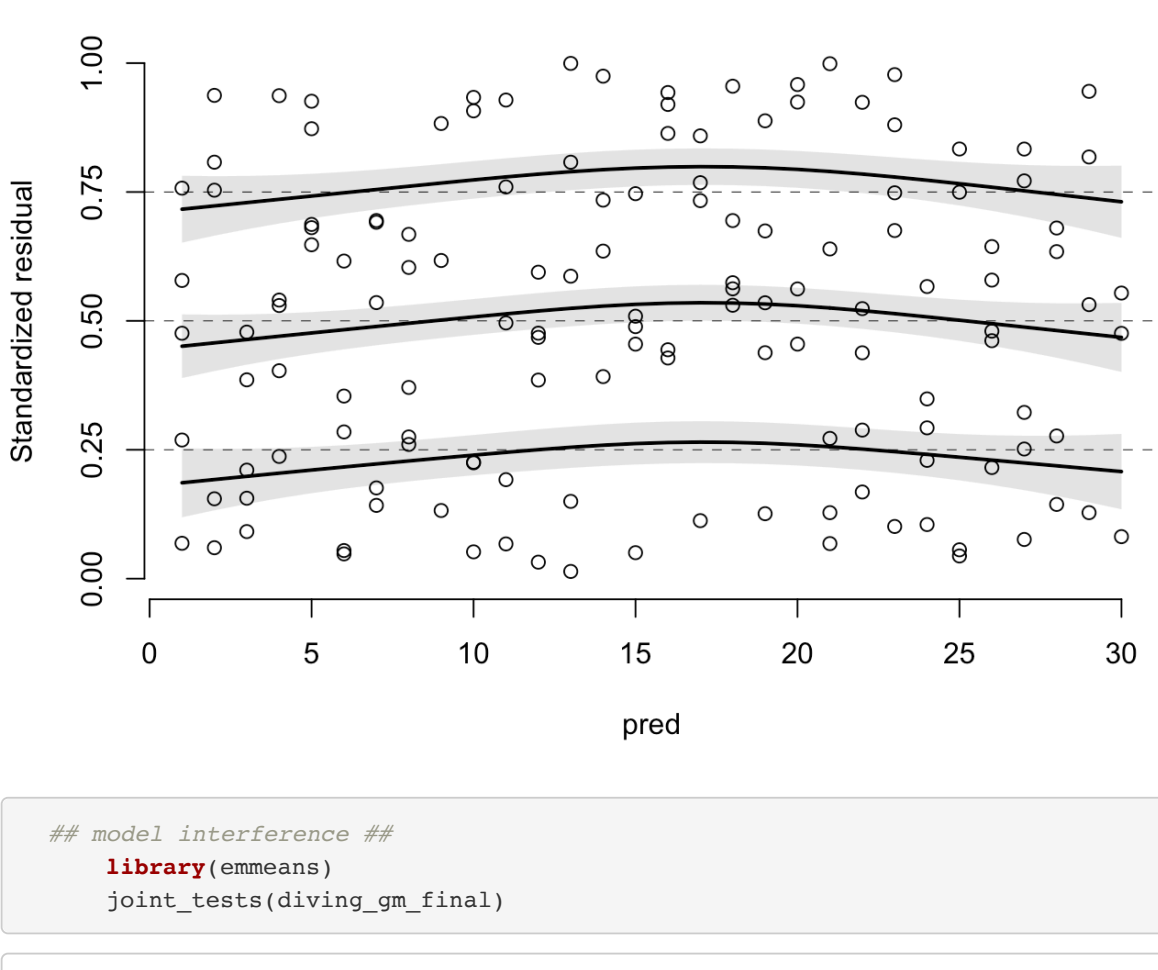
```
## $uniformity
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.052397, p-value = 0.8368
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 1.2246, p-value = 0.568
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutput
## outliers at both margin(s) = 0, observations = 140, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.00000000 0.02857143
## sample estimates:
## outlier frequency (expected: 0.00321428571428571 )
##                                                  0
```

```
## $uniformity
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.052397, p-value = 0.8368
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 1.2246, p-value = 0.568
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutput
## outliers at both margin(s) = 0, observations = 140, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.00000000 0.02857143
## sample estimates:
## outlier frequency (expected: 0.00321428571428571 )
##                                                  0
```

```
    plotResiduals(diving_gm_final.sim, multi_mean$fstimulus)
```



```
    plotResiduals(diving_gm_final.sim, multi_mean$trial)
```



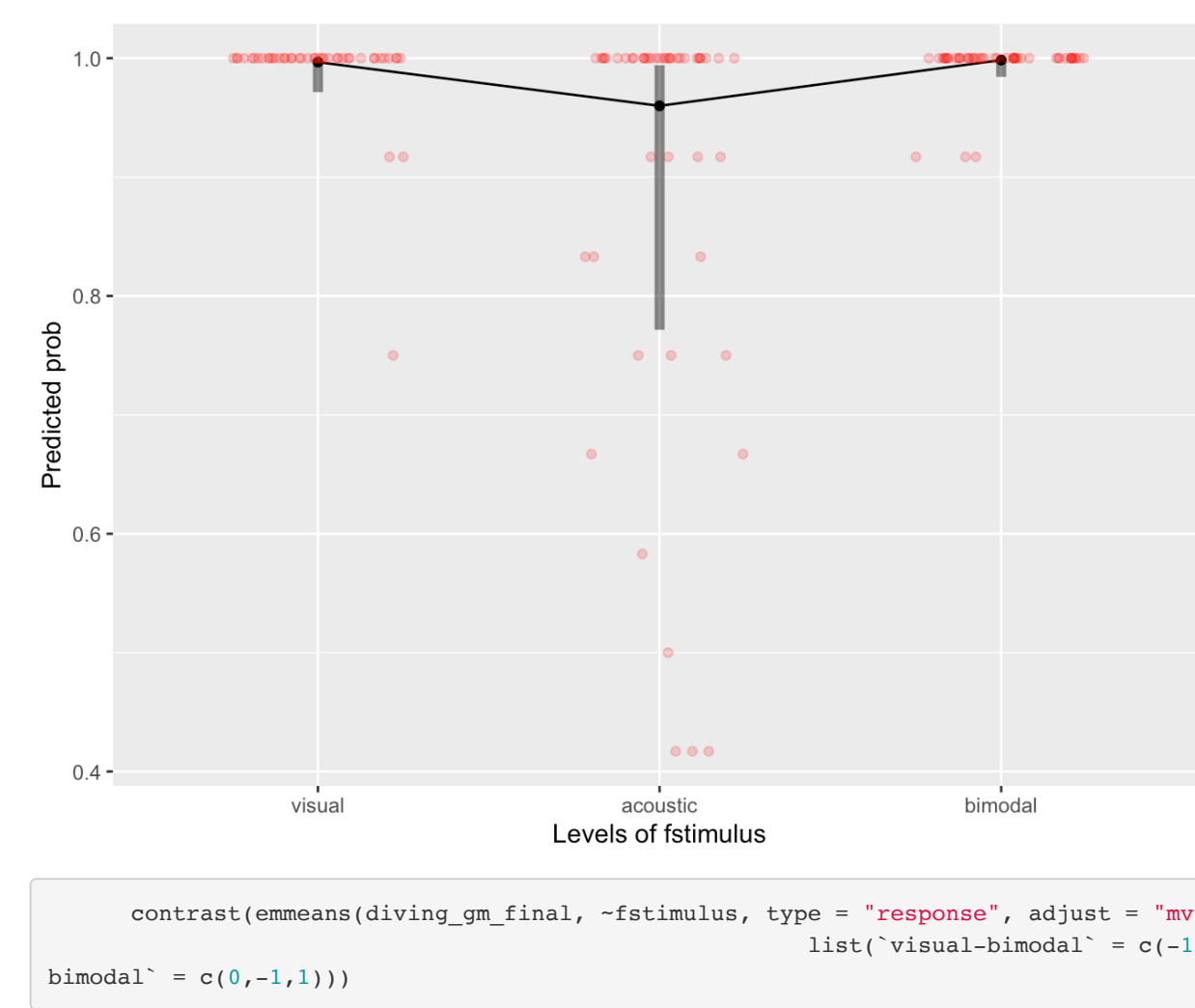
```
    ## model interference ##
    library(emmeans)
    joint_tests(diving_gm_final)
```

```
##  model term df1 df2 F.ratio p.value
##  fstimulus    2 Inf  27.347 <.0001
##  trial        1 Inf   0.025 0.8743
```

```
    (mean_respo_EMMs <- emmip(diving_gm_final, ~fstimulus, type = "response", adjust = "mvt", CIs
= TRUE, plotit=FALSE))
```

```
##   fstimulus      yvar          SE  df       LCL       UCL tvar     xvar
## 1    visual 0.9967233 0.003182976 Inf 0.9715849 0.9996306    1   visual
## 2  acoustic 0.9599304 0.033546407 Inf 0.7721232 0.9941308    1 acoustic
## 3   bimodal 0.9984629 0.001597115 Inf 0.9843660 0.9998508    1  bimodal
```

```
    emmip(diving_gm_final, ~fstimulus, type = "response", adjust = "mvt", CIs = TRUE, plotit=TRUE)
+
      geom_jitter(aes(x = fstimulus, y = DiveRate), data = multi_mean, color = "red", alpha=0.2, h
eight=0, width=0.25)
```



```
    contrast(emmeans(diving_gm_final, ~fstimulus, type = "response", adjust = "mvt"),
                                                     list(`visual-bimodal` = c(-1,0,1), `acoustic-
bimodal` = c(0,-1,1)))
```

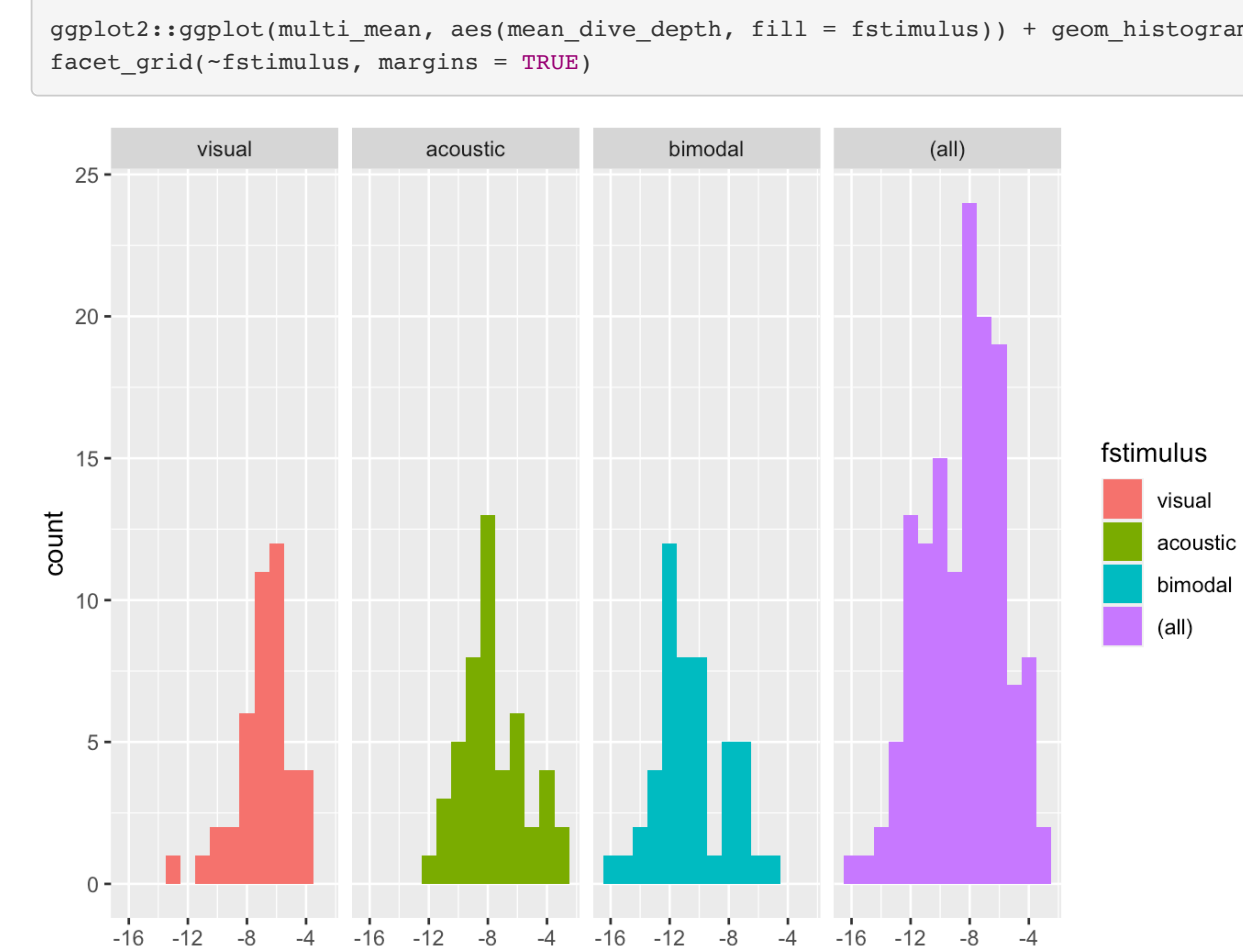
```
##  contrast         odds.ratio    SE  df z.ratio p.value
##  visual-bimodal         2.14  1.56 Inf 1.039   0.2987
##  acoustic-bimodal      27.11 16.16 Inf 5.536   <.0001
##
## Tests are performed on the log odds ratio scale
```

Dive depth - LMER

```
mean(multi_mean$mean_dive_depth,na.rm=TRUE)
```

```
## [1] -8.389629
```

```
ggplot2::ggplot(multi_mean, aes(mean_dive_depth, fill = fstimulus)) + geom_histogram(binwidth = 1) +
facet_grid(~fstimulus, margins = TRUE)
```



```
    ####### LMM #######
    library(lme4)
    library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```
    library(tidyverse)

    #random effect structure: varying intercept best represents exp. design
    #fixed effect structure
      mean_depth.f0<-lmer(mean_dive_depth~1 + (1|fgroup), REML=FALSE, data=drop_na(multi_mean, mean_
dive_depth))
      mean_depth.stimulus<-lmer(mean_dive_depth~fstimulus + (1|fgroup), REML=FALSE, data=drop_na(mul
ti_mean, mean_dive_depth))
      mean_depth.trial<-lmer(mean_dive_depth~trial + (1|fgroup), REML=FALSE, data=drop_na(multi_mean
, mean_dive_depth))
      mean_depth.inter<-lmer(mean_dive_depth~fstimulus*trial + (1|fgroup), REML=FALSE, data=drop_na(
multi_mean, mean_dive_depth))
      mean_depth.2way <- update(mean_depth.inter, . ~ . -fstimulus:trial)

      bbmle::AICctab(mean_depth.f0, mean_depth.stimulus, mean_depth.trial, mean_depth.inter,mean_dep
th.2way,
            base=TRUE, weights=TRUE,delta=TRUE,sort=TRUE)
```

```
##                     AICc  dAICc df weight
## mean_depth.stimulus 562.8   0.0 5  0.661
## mean_depth.2way     564.5   1.7 6  0.287
## mean_depth.inter    567.9   5.1 8  0.052
## mean_depth.f0       657.8  95.0 3  <0.001
## mean_depth.trial    659.3  96.5 4  <0.001
```

```
    drop1(mean_depth.2way) #test significance of fixed effects
```

```
## Single term deletions using Satterthwaite's method:
##
## Model:
## mean_dive_depth ~ fstimulus + trial + (1 | fgroup)
##           Sum Sq Mean Sq NumDF  DenDF F value Pr(>F)
## fstimulus 396.76 198.380     2 134.98 73.3276 <2e-16 ***
## trial       1.41   1.408     1 135.01  0.5204 0.4719
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
    ## model diagnostics ##
    summary(mean_depth.final <- update(mean_depth.2way, REML=TRUE))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: mean_dive_depth ~ fstimulus + trial + (1 | fgroup)
##    Data: drop_na(multi_mean, mean_dive_depth)
##
## REML criterion at convergence: 558.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2038 -0.4715  0.0281  0.5974  2.9039
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fgroup   (Intercept) 2.456    1.567
##  Residual             2.767    1.663
## Number of obs: 140, groups:  fgroup, 5
##
## Fixed effects:
##                   Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)       -6.45466    0.78751   5.84682  -8.196 0.000203 ***
## fstimulusacoustic -0.89834    0.35005 131.99146  -2.566 0.011395 *
## fstimulusbimodal  -3.92578    0.34795 131.99706 -11.283  < 2e-16 ***
## trial             -0.01168    0.01649 132.01193  -0.708 0.479974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fstmlsc fstmlsb
## fstimlscstc -0.251
## fstimlsbmdl -0.233  0.530
## trial       -0.322  0.052  -0.009
```

```
    performance::check_model(mean_depth.final)
```

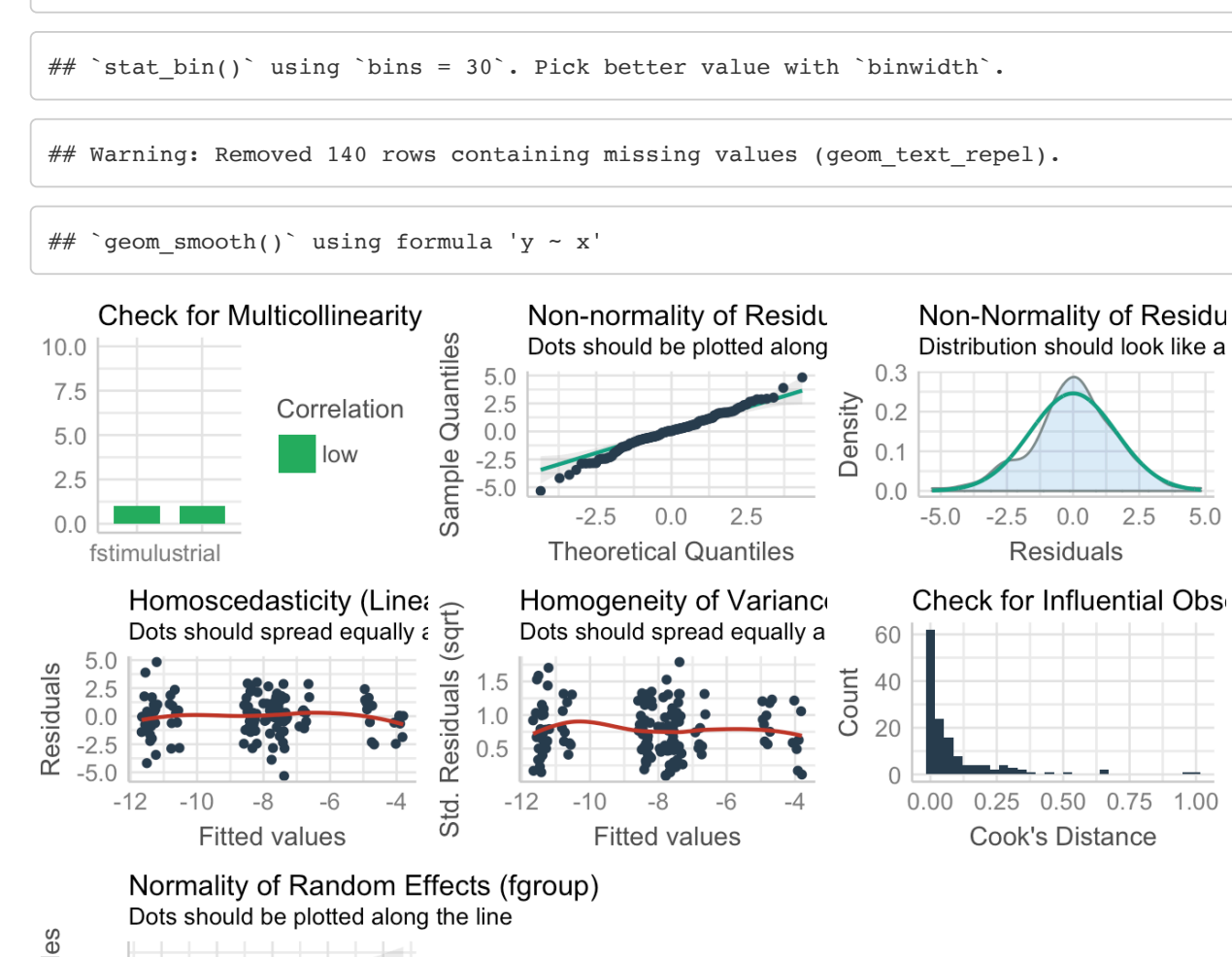
```
## Loading required namespace: qqplotr
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

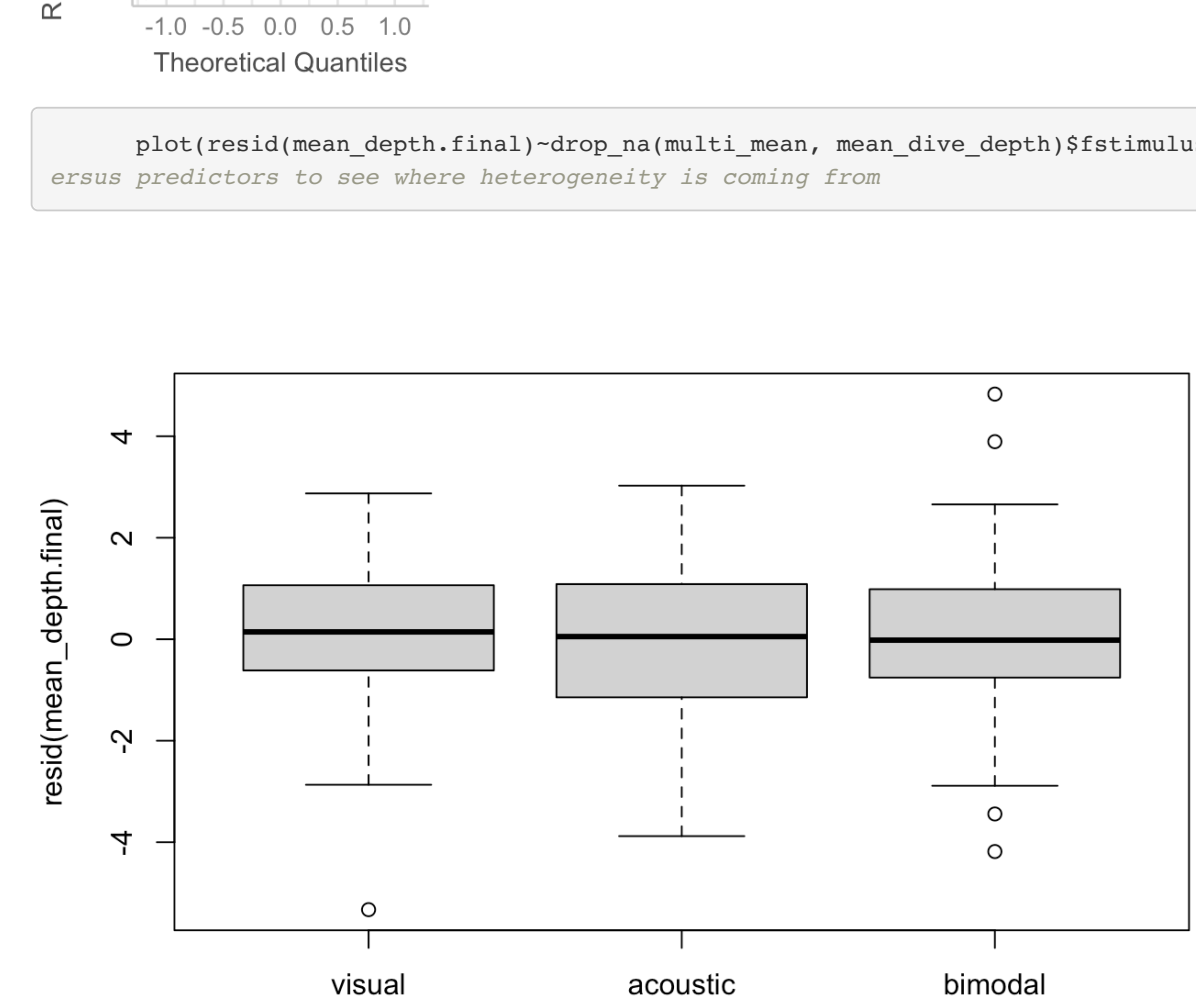
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 140 rows containing missing values (geom_text_repel).
```

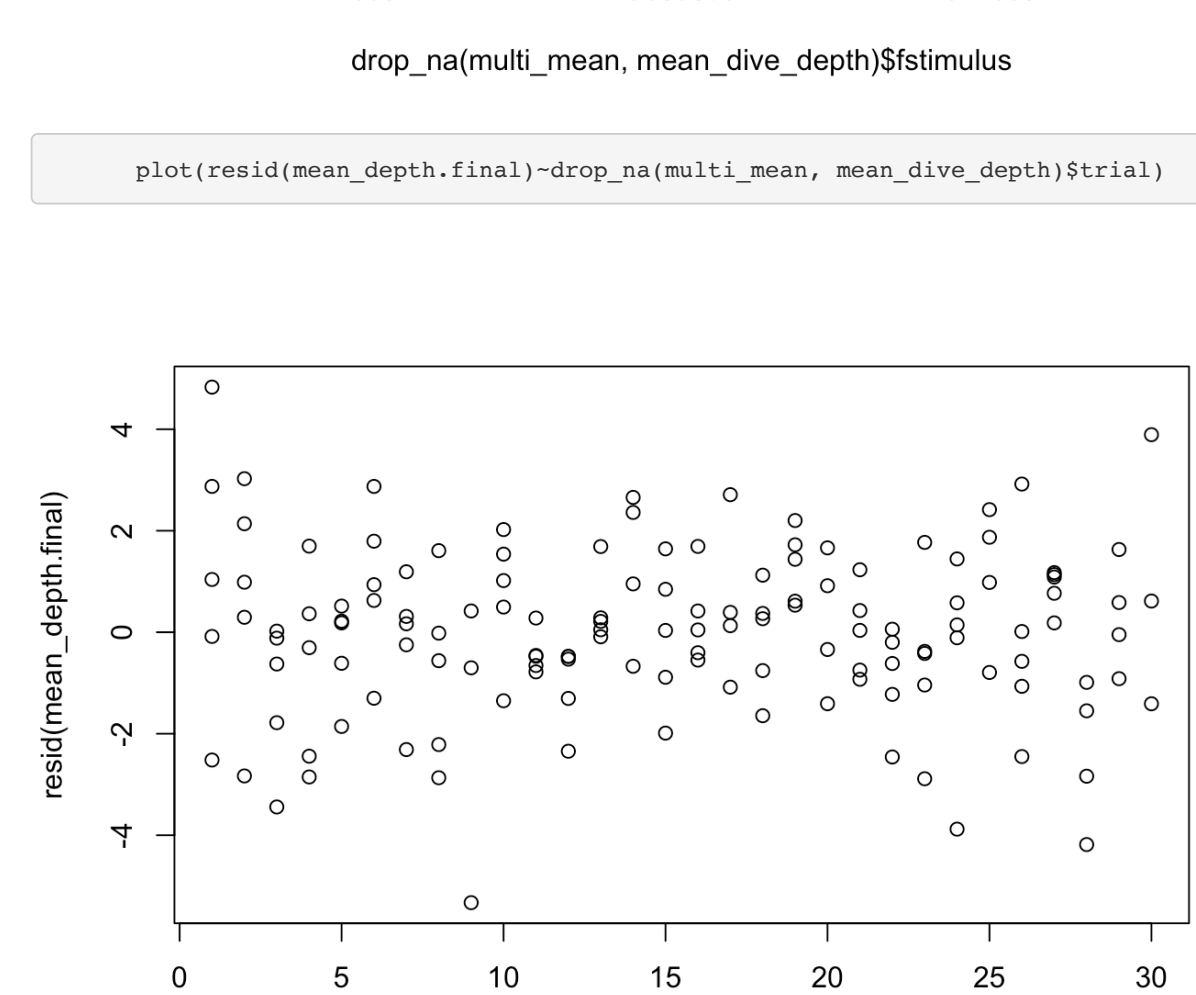
```
## `geom_smooth()` using formula 'y ~ x'
```



```
    plot(resid(mean_depth.final)~drop_na(multi_mean, mean_dive_depth)$fstimulus) #plot residuals v
ersus predictors to see where heterogeneity is coming from
```



```
    plot(resid(mean_depth.final)~drop_na(multi_mean, mean_dive_depth)$trial)
```



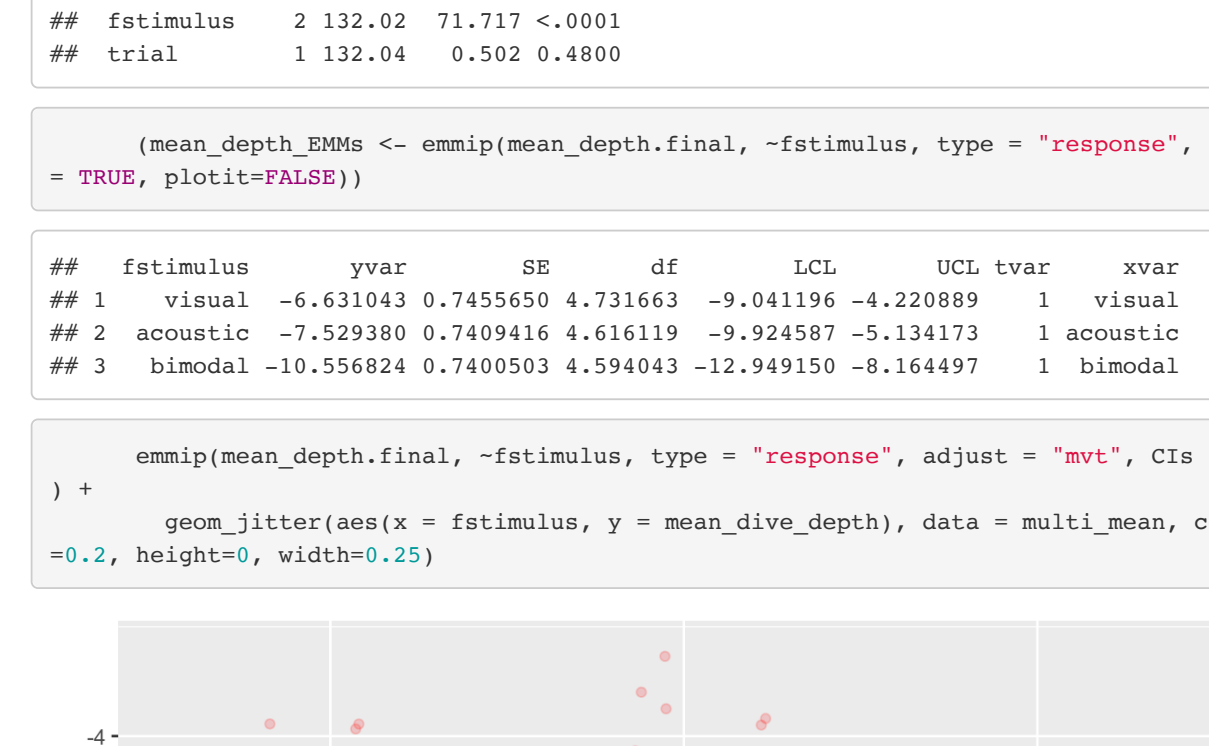
```
    ## model inference ##
    library(emmeans)
    joint_tests(mean_depth.final)
```

```
##  model term df1    df2 F.ratio p.value
##  fstimulus    2 132.02  71.717 <.0001
##  trial        1 132.04   0.502 0.4800
```

```
    (mean_depth_EMMs <- emmip(mean_depth.final, ~fstimulus, type = "response", adjust = "mvt", CIs
= TRUE, plotit=FALSE))
```

```
##   fstimulus       yvar        SE       df        LCL       UCL tvar     xvar
## 1    visual  -6.631043 0.7455650 4.731663  -9.041196 -4.220889    1   visual
## 2  acoustic  -7.529380 0.7409416 4.616119  -9.924587 -5.134173    1 acoustic
## 3   bimodal -10.556824 0.7400503 4.594043 -12.949150 -8.164497    1  bimodal
```

```
    emmip(mean_depth.final, ~fstimulus, type = "response", adjust = "mvt", CIs = TRUE, plotit=TRUE
) +
      geom_jitter(aes(x = fstimulus, y = mean_dive_depth), data = multi_mean, color = "red", alpha
=0.2, height=0, width=0.25)
```



```
    contrast(emmeans(mean_depth.final, ~fstimulus, type = "response", adjust = "mvt"),
                                                     list(`visual-bimodal` = c(-1,0,1), `acoustic-
bimodal` = c(0,-1,1)))
```

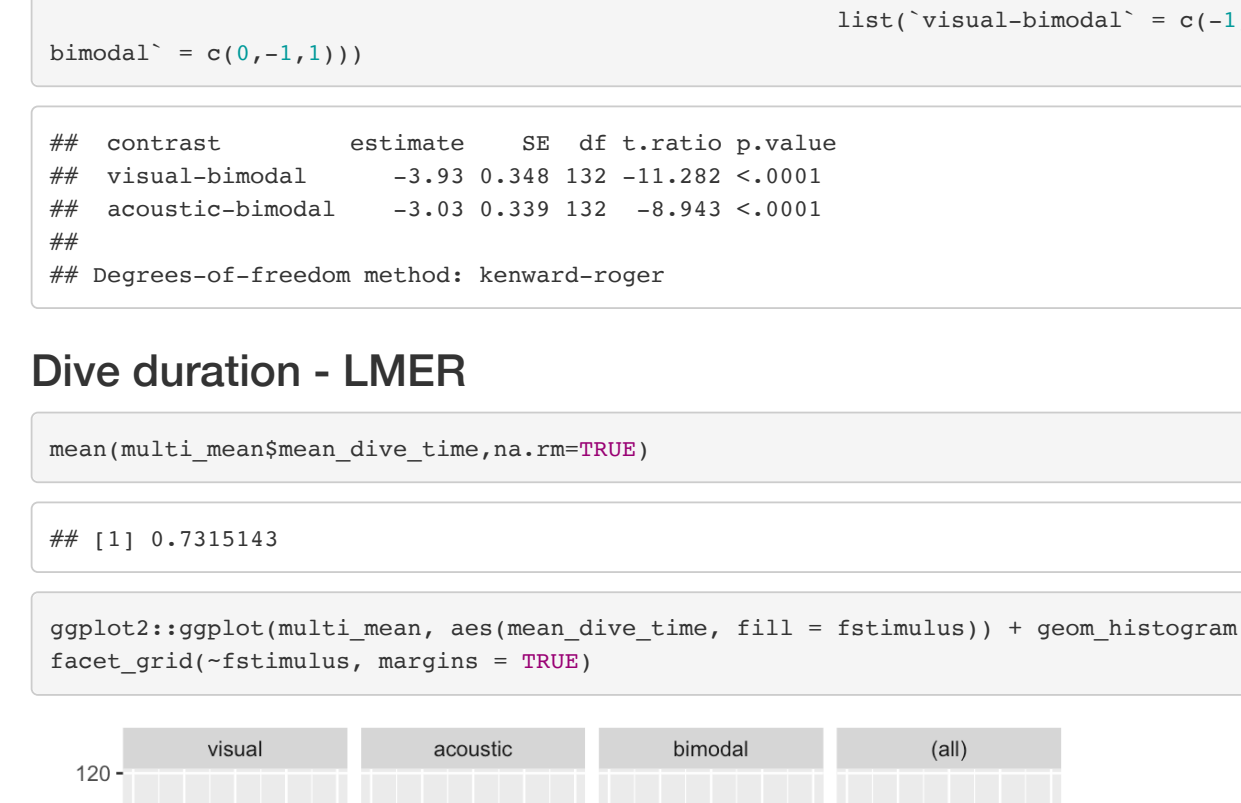
```
##  contrast         estimate    SE  df t.ratio p.value
##  visual-bimodal      -3.93 0.348 132 -11.282 <.0001
##  acoustic-bimodal    -3.03 0.339 132  -8.943 <.0001
##
## Degrees-of-freedom method: kenward-roger
```

Dive duration - LMER

```
mean(multi_mean$mean_dive_time,na.rm=TRUE)
```

```
## [1] 0.7315143
```

```
ggplot2::ggplot(multi_mean, aes(mean_dive_time, fill = fstimulus)) + geom_histogram(binwidth = 1) +
facet_grid(~fstimulus, margins = TRUE)
```



```
    ####### LMM #######
    library(lme4)
    library(lmerTest)
    library(tidyverse)

    #random effect structure: varying intercept best represents exp. design
    #fixed effect structure
      mean_time.f0<-lmer(mean_dive_time~1 + (1|fgroup), REML=FALSE, data=drop_na(multi_mean, mean_di
ve_time))
      mean_time.stimulus<-lmer(mean_dive_time~fstimulus + (1|fgroup), REML=FALSE, data=drop_na(multi
_mean, mean_dive_time))
      mean_time.trial<-lmer(mean_dive_time~trial + (1|fgroup), REML=FALSE, data=drop_na(multi_mean,
mean_dive_time))
      mean_time.inter<-lmer(mean_dive_time~fstimulus*trial + (1|fgroup), REML=FALSE, data=drop_na(mu
lti_mean, mean_dive_time))
      mean_time.2way <- update(mean_time.inter, . ~ . -fstimulus:trial)

      bbmle::AICctab(mean_time.f0, mean_time.stimulus, mean_time.trial, mean_time.inter,mean_time.2w
ay,
            base=TRUE, weights=TRUE,delta=TRUE,sort=TRUE)
```

```
##                     AICc  dAICc df weight
## mean_time.stimulus -161.7   0.0 5  0.57
## mean_time.2way     -160.2   1.5 6  0.27
## mean_time.inter    -159.1   2.6 8  0.16
## mean_time.f0        -50.6 111.1 3  <0.001
## mean_time.trial     -48.7 113.0 4  <0.001
```

```
    drop1(mean_time.2way) #test significance of fixed effects
```

```
## Single term deletions using Satterthwaite's method:
##
## Model:
## mean_dive_time ~ fstimulus + trial + (1 | fgroup)
##            Sum Sq Mean Sq NumDF  DenDF F value Pr(>F)
## fstimulus 2.87264 1.43632     2 134.98 91.7450 <2e-16 ***
## trial     0.01032 0.01032     1 135.04  0.6592 0.4183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
    ## model diagnostics ##
    summary(mean_time.final <- update(mean_time.2way, REML=TRUE))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: mean_dive_time ~ fstimulus + trial + (1 | fgroup)
##    Data: drop_na(multi_mean, mean_dive_time)
##
## REML criterion at convergence: -145.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2178 -0.7570 -0.1376  0.7066  2.4336
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fgroup   (Intercept) 0.006965 0.08345
##  Residual             0.016010 0.12653
## Number of obs: 140, groups:  fgroup, 5
##
## Fixed effects:
##                    Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)        0.549257   0.046257   7.947494  11.874 2.45e-06 ***
## fstimulusacoustic  0.207131   0.026628 132.000530   7.779 1.86e-12 ***
## fstimulusbimodal   0.353890   0.026468 132.011543  13.371  < 2e-16 ***
## trial             -0.001014   0.001254 132.040262  -0.808     0.42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fstmlsc fstmlsb
## fstimlscstc -0.325
## fstimlsbmdl -0.302  0.530
## trial       -0.417  0.052  -0.009
```

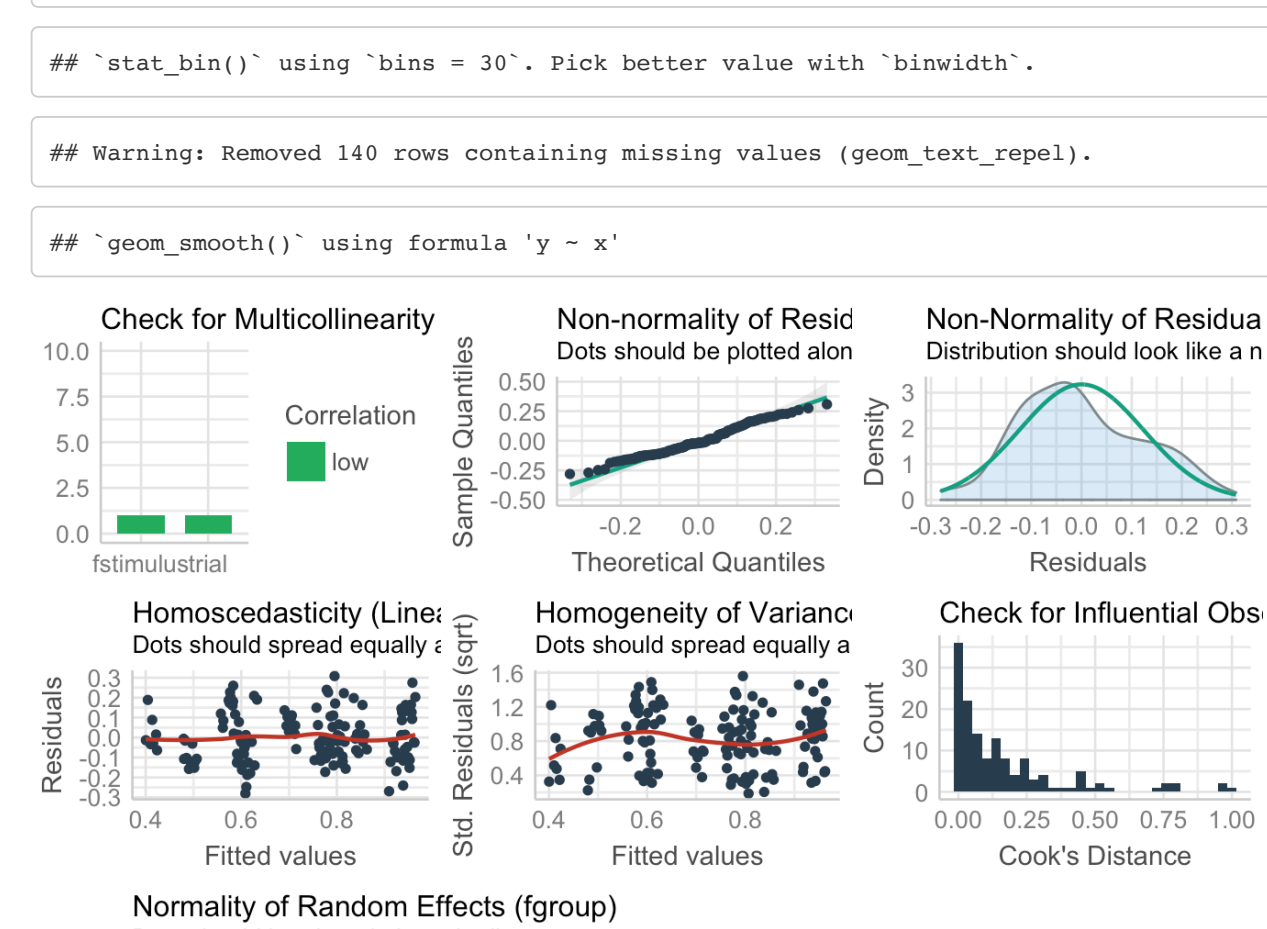
```
    performance::check_model(mean_time.final)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

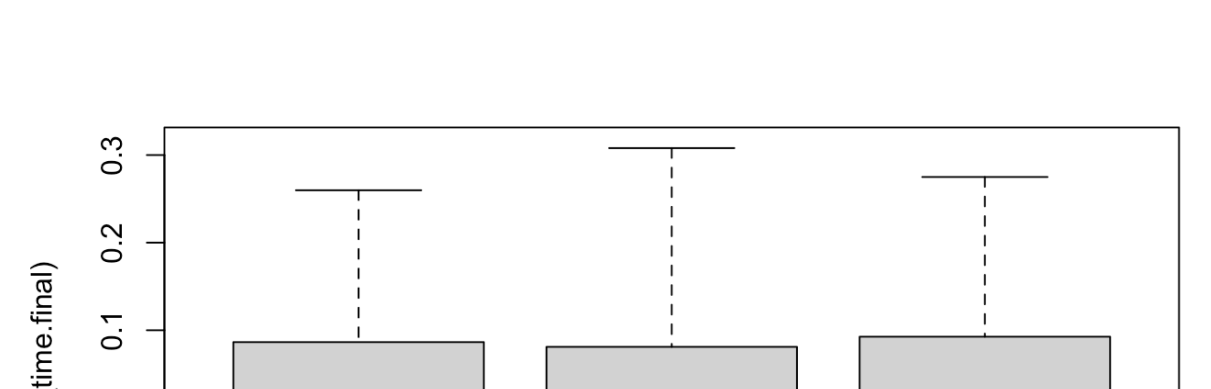
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

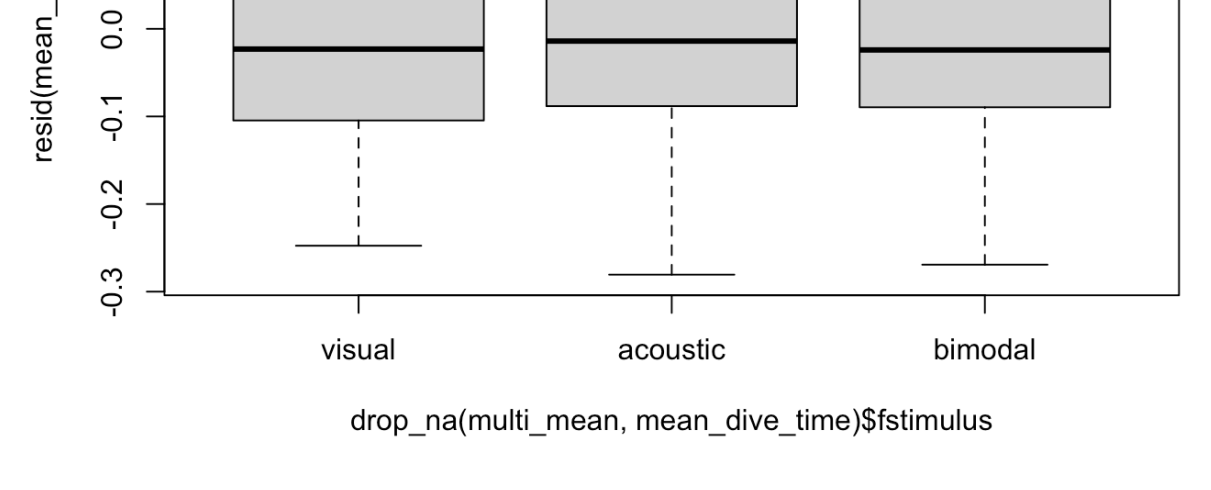
```
## Warning: Removed 140 rows containing missing values (geom_text_repel).
```

```
## `geom_smooth()` using formula 'y ~ x'
```

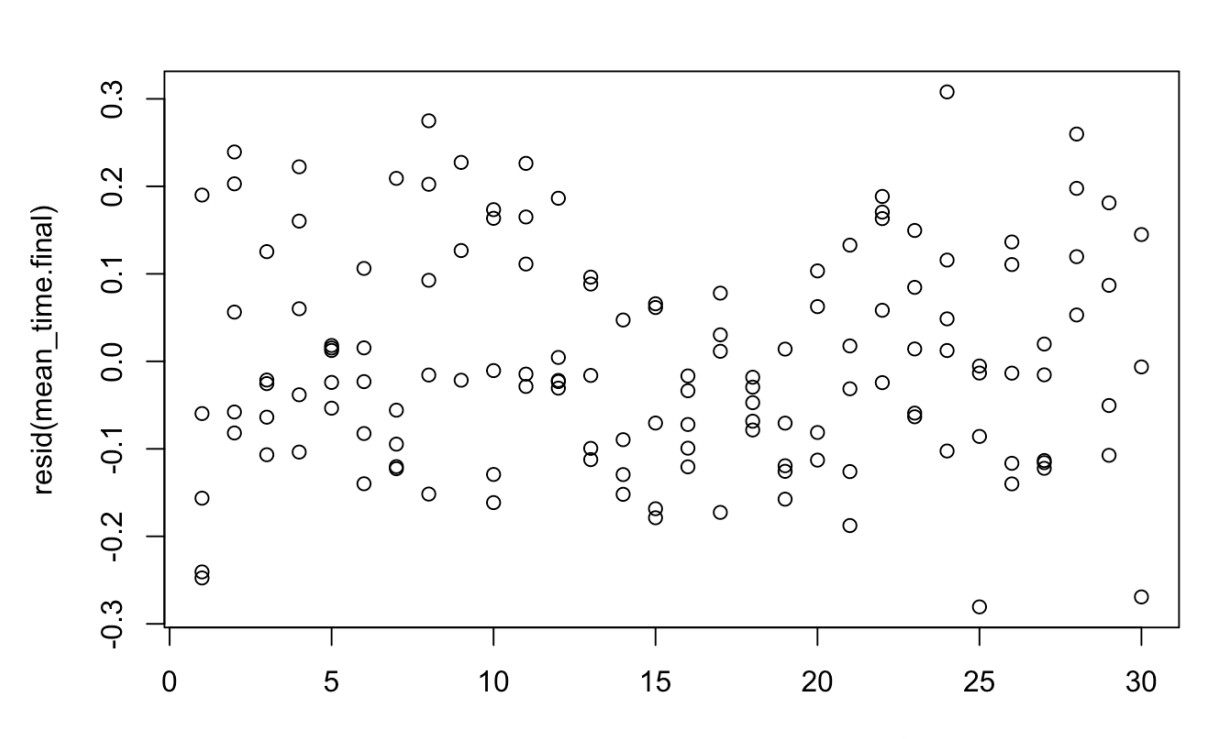


```
    plot(resid(mean_time.final)~drop_na(multi_mean, mean_dive_time)$fstimulus) #plot residuals ver
sus predictors to see where heterogeneity is coming from
```





```
    plot(resid(mean_time.final)~drop_na(multi_mean, mean_dive_time)$trial)
```



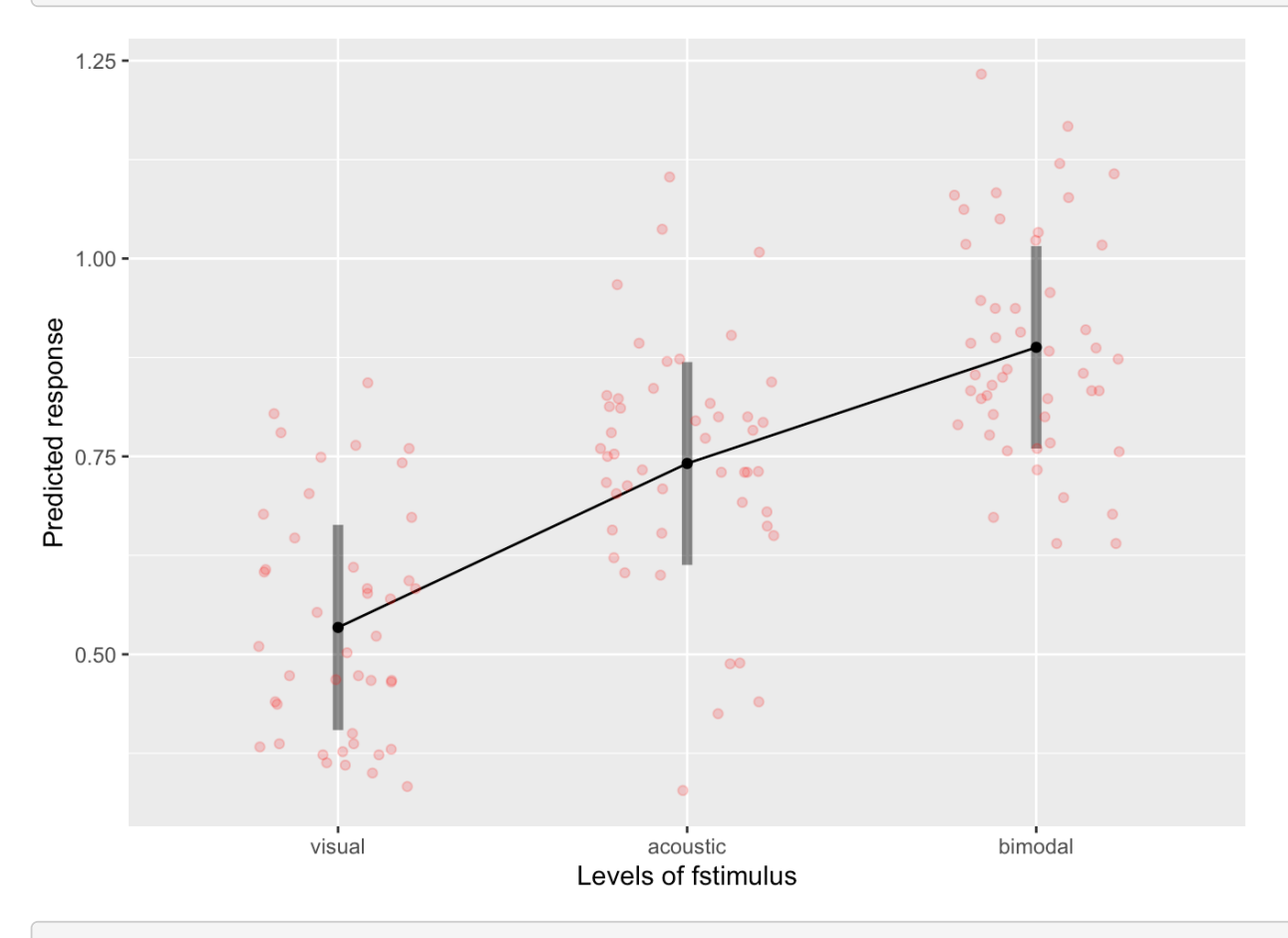
```
  ## model interference ##
    library(emmeans)
    joint_tests(mean_time.final)
```

```
##  model term df1    df2 F.ratio p.value
##  fstimulus    2 132.03  89.709 <.0001
##  trial        1 132.07   0.653 0.4204
```

```
    (mean_time_EMMs <- emmip(mean_time.final, ~fstimulus, type = "response", adjust = "mvt", CIs =
TRUE, plotit=FALSE))
```

```
##   fstimulus      yvar         SE       df       LCL       UCL tvar     xvar
## 1    visual 0.5339471 0.04204322 5.489001 0.4041975 0.6636968    1   visual
## 2  acoustic 0.7410782 0.04156753 5.247914 0.6127966 0.8693599    1 acoustic
## 3   bimodal 0.8878372 0.04147555 5.202026 0.7598394 1.0158349    1  bimodal
```

```
    emmip(mean_time.final, ~fstimulus, type = "response", adjust = "mvt", CIs = TRUE, plotit=TRUE)
+
      geom_jitter(aes(x = fstimulus, y = mean_dive_time), data = multi_mean, color = "red", alpha=
0.2, height=0, width=0.25)
```



```
    contrast(emmeans(mean_time.final, ~fstimulus, type = "response", adjust = "mvt"),
                                                  list(`visual-bimodal` = c(-1,0,1), `acoustic-
bimodal` = c(0,-1,1)))
```

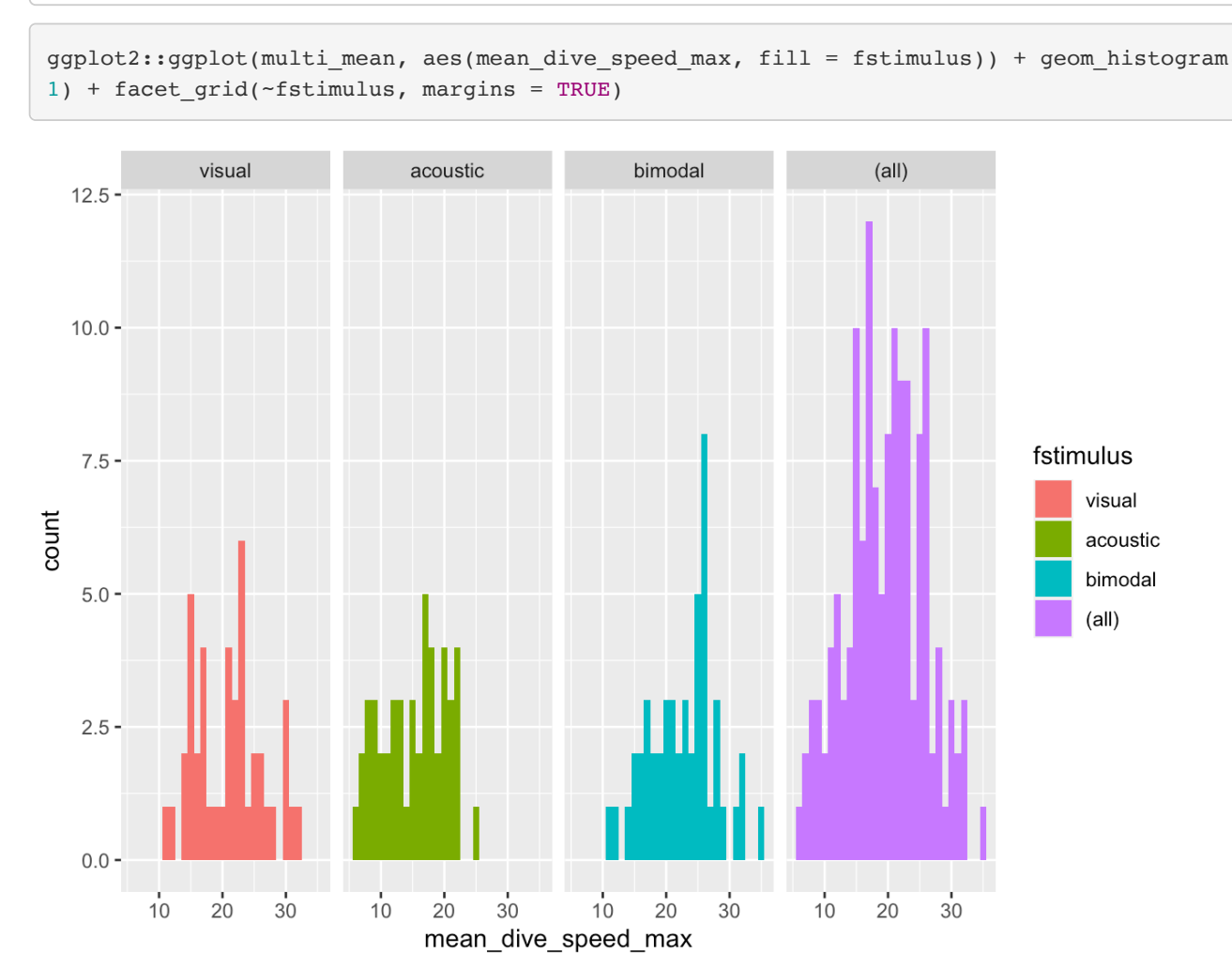
```
##  contrast         estimate     SE  df t.ratio p.value
##  visual-bimodal      0.354 0.0265 132 13.369  <.0001
##  acoustic-bimodal    0.147 0.0258 132  5.699  <.0001
##
## Degrees-of-freedom method: kenward-roger
```

Max dive speed - LMER

```
mean(multi_mean$mean_dive_speed_max,na.rm=TRUE)
```

```
## [1] 19.58499
```

```
ggplot2::ggplot(multi_mean, aes(mean_dive_speed_max, fill = fstimulus)) + geom_histogram(binwidth =
1) + facet_grid(~fstimulus, margins = TRUE)
```



```
####### LMM #######
    library(lme4)
    library(lmerTest)
    library(tidyverse)

    #random effect structure: varying Intercept best represents exp. design
    #fixed effect structure
      mean_speed_max.f0<-lmer(mean_dive_speed_max~1 + (1|fgroup), REML=FALSE, data=drop_na(multi_mea
n, mean_dive_speed_max))
      mean_speed_max.stimulus<-lmer(mean_dive_speed_max~fstimulus + (1|fgroup), REML=FALSE, data=dro
p_na(multi_mean, mean_dive_speed_max))
      mean_speed_max.trial<-lmer(mean_dive_speed_max~trial + (1|fgroup), REML=FALSE, data=drop_na(mu
lti_mean, mean_dive_speed_max))
      mean_speed_max.inter<-lmer(mean_dive_speed_max~fstimulus*trial + (1|fgroup), REML=FALSE, data=
drop_na(multi_mean, mean_dive_speed_max))
      mean_speed_max.2way <- update(mean_speed_max.inter, . ~ . -fstimulus:trial)

      bbmle::AICctab(mean_speed_max.f0,mean_speed_max.stimulus,mean_speed_max.trial,mean_speed_max.i
nter,mean_speed_max.2way,
              base=TRUE, weights=TRUE,delta=TRUE,sort=TRUE)
```

```
##                          AICc  dAICc df weight
## mean_speed_max.2way      797.6   0.0 6  0.739
## mean_speed_max.stimulus 800.4   2.8 5  0.182
## mean_speed_max.inter    802.0   4.5 8  0.079
## mean_speed_max.trial    869.6  72.0 4  <0.001
## mean_speed_max.f0       872.4  74.8 3  <0.001
```

```
      drop1(mean_speed_max.2way) #test significance of fixed effects
```

```
## Single term deletions using Satterthwaite's method:
##
## Model:
## mean_dive_speed_max ~ fstimulus + trial + (1 | fgroup)
##            Sum Sq Mean Sq NumDF  DenDF F value  Pr(>F)
## fstimulus 1458.13  729.06     2 135.01 51.3242 < 2e-16 ***
## trial       72.09   72.09     1 135.03  5.0746 0.02589 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
    ## model diagnostics ##
      summary(mean_speed_max.final <- update(mean_speed_max.2way, REML=TRUE))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: mean_dive_speed_max ~ fstimulus + trial + (1 | fgroup)
##    Data: drop_na(multi_mean, mean_dive_speed_max)
##
## REML criterion at convergence: 784.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3314 -0.7076  0.0602  0.7056  3.1841
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fgroup   (Intercept) 15.52    3.939
##  Residual             14.53    3.812
## Number of obs: 140, groups:  fgroup, 5
##
## Fixed effects:
##                   Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)        19.30638    1.94452   5.54899   9.929 9.69e-05 ***
## fstimulusacoustic  -5.52112    0.80214 132.01196  -6.883 2.14e-10 ***
## fstimulusbimodal    2.01659    0.79733 132.01661   2.529   0.0126 *
## trial               0.08402    0.03779 132.02901   2.224   0.0279 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fstmlsc fstmlsb
## fstimlscstc -0.233
## fstimlsbmdl -0.216  0.530
## trial       -0.299  0.052  -0.009
```

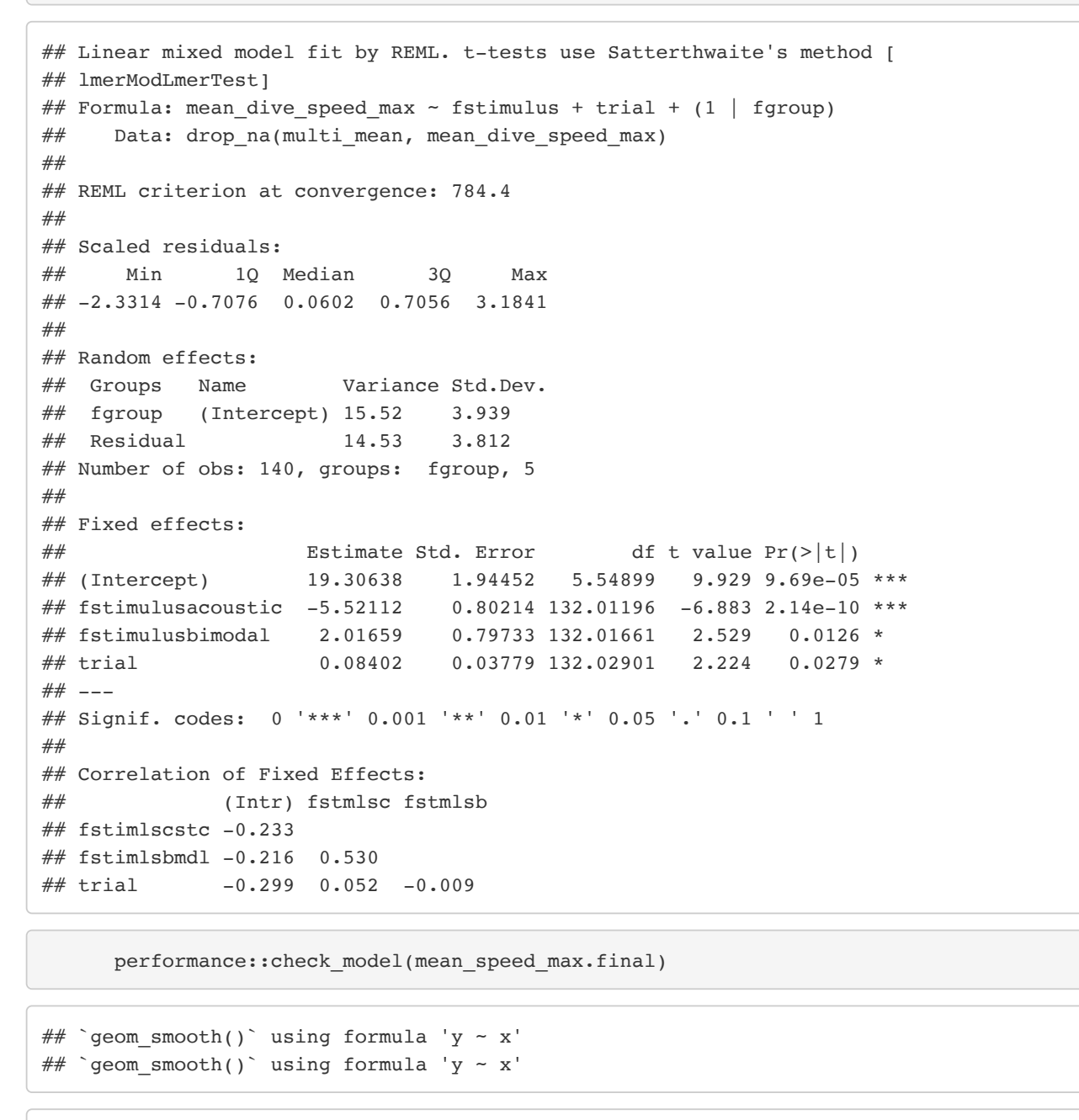
```
      performance::check_model(mean_speed_max.final)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

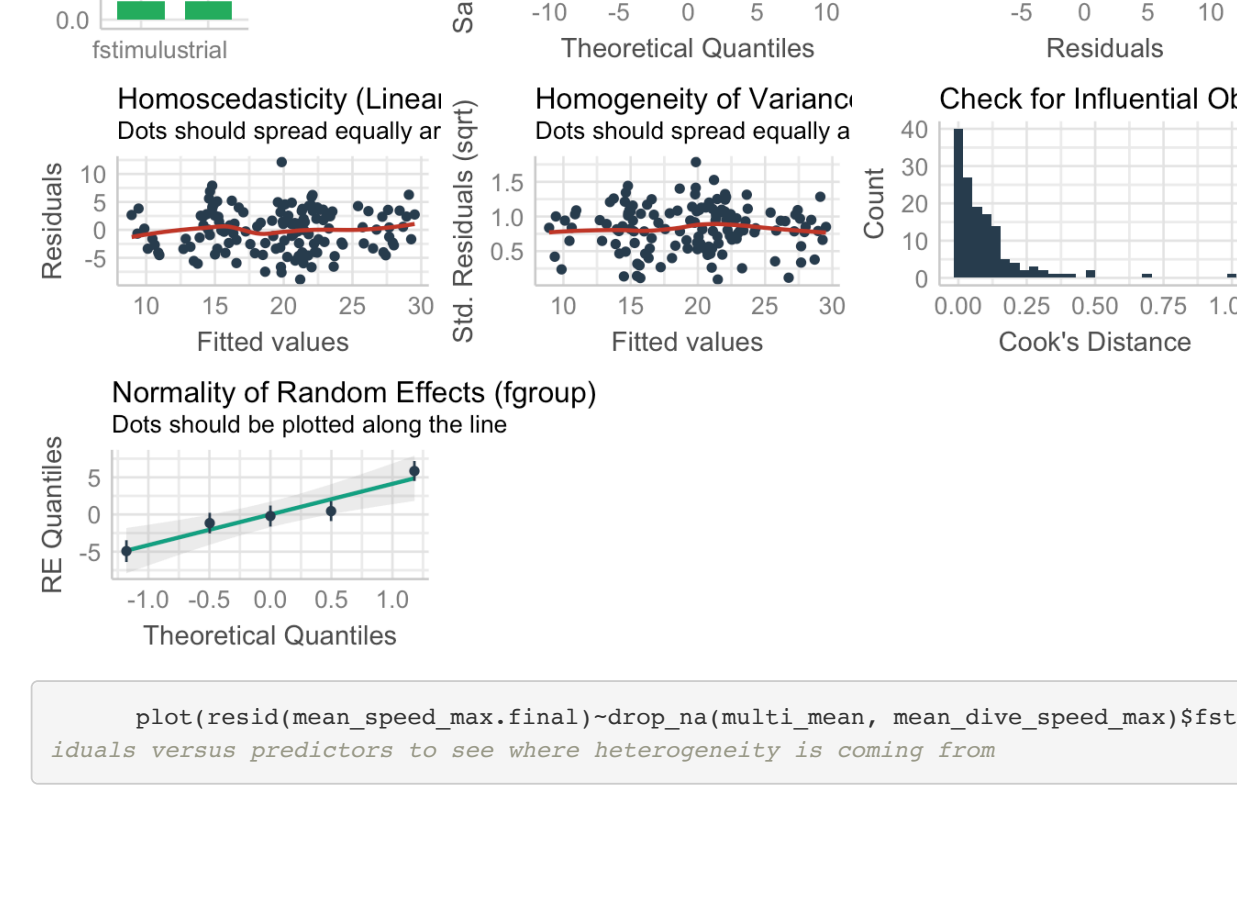
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 140 rows containing missing values (geom_text_repel).
```

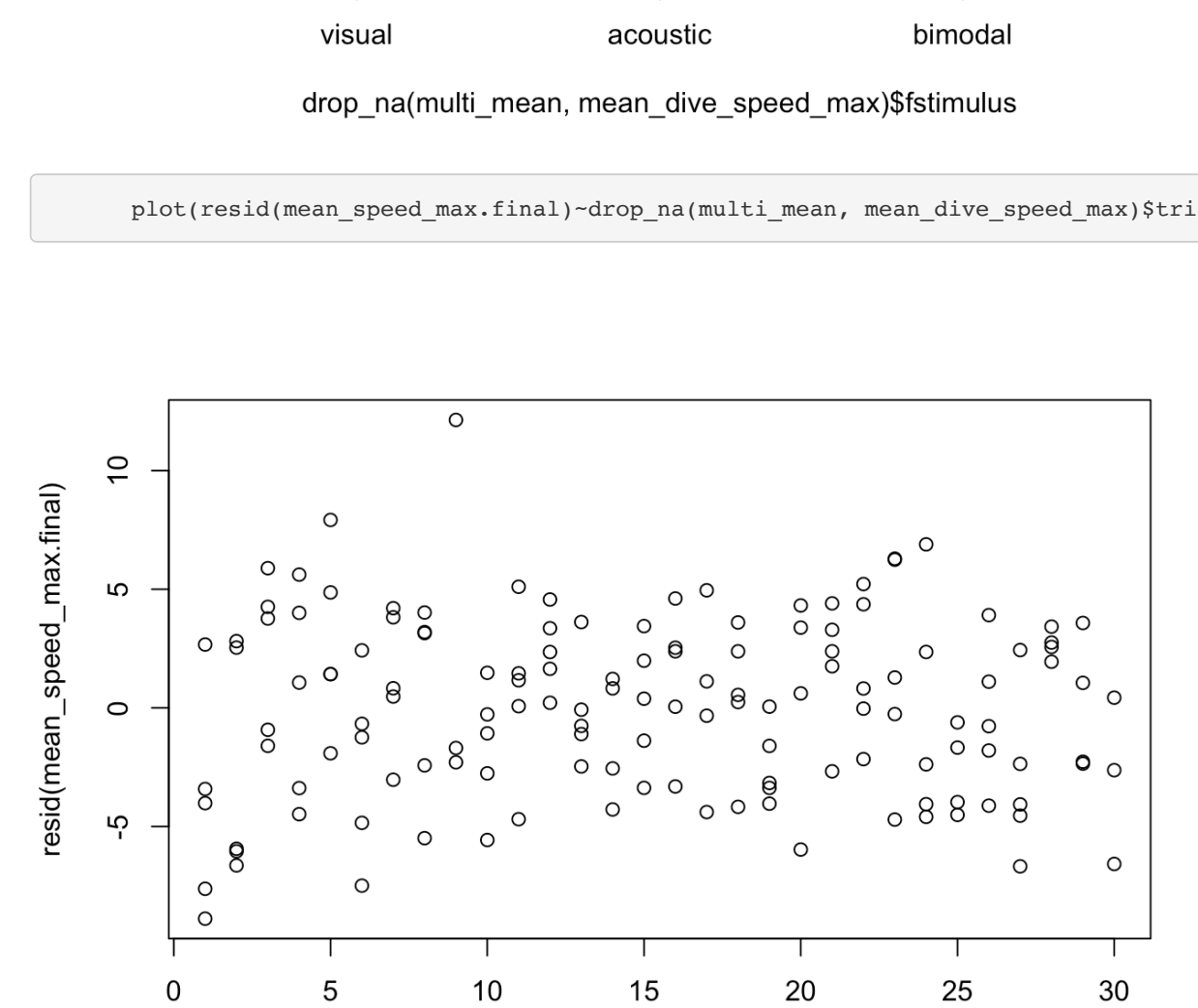
```
## `geom_smooth()` using formula 'y ~ x'
```



```
      plot(resid(mean_speed_max.final)~drop_na(multi_mean, mean_dive_speed_max)$fstimulus) #plot res
iduals versus predictors to see where heterogeneity is coming from
```



```
      plot(resid(mean_speed_max.final)~drop_na(multi_mean, mean_dive_speed_max)$trial)
```



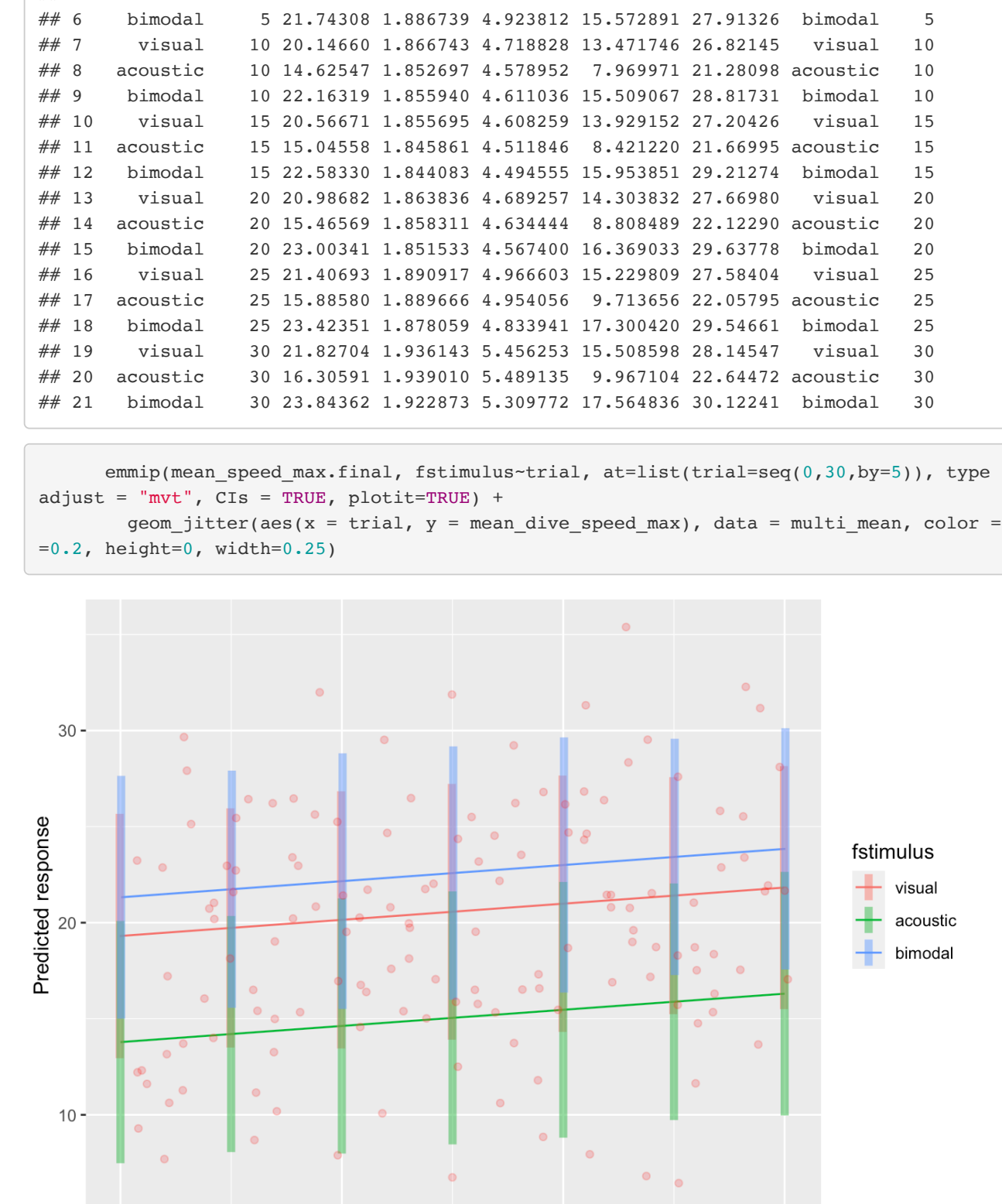
```
  ## model predictions ##
      library(emmeans)
      joint_tests(mean_speed_max.final)
```

```
##  model term df1    df2 F.ratio p.value
##  fstimulus    2 132.01  50.178 <.0001
##  trial        1 132.03   4.944 0.0279
```

```
      (mean_speed_max_EMMs = emmip(mean_speed_max.final, fstimulus~trial, at=list(trial=seq(0,30,by=
5)), type = "response", adjust = "mvt", CIs = TRUE, plotit=FALSE))
```

```
##    fstimulus trial     yvar       SE       df       LCL      UCL     tvar xvar
## 1     visual     0 19.30638 1.944525 5.551896 12.940180 25.67258   visual    0
## 2   acoustic     0 13.78526 1.922825 5.309588  7.495847 20.07466 acoustic    0
## 3    bimodal     0 21.32297 1.935576 5.451009 14.982488 27.66345  bimodal    0
## 4     visual     5 19.72649 1.896642 5.027487 13.518473 25.93451   visual    5
## 5   acoustic     5 14.20537 1.878608 4.839783  8.054334 20.35640 acoustic    5
## 6    bimodal     5 21.74308 1.886739 4.923812 15.572891 27.91326  bimodal    5
## 7     visual    10 20.14660 1.866743 4.718828 13.471746 26.82145   visual   10
## 8   acoustic    10 14.62547 1.852697 4.578952  7.969971 21.28098 acoustic   10
## 9    bimodal    10 22.16319 1.855940 4.611036 15.509067 28.81731  bimodal   10
## 10    visual    15 20.56671 1.855695 4.608259 13.929152 27.20426   visual   15
## 11  acoustic    15 15.04558 1.845861 4.511846  8.421220 21.66995 acoustic   15
## 12   bimodal    15 22.58330 1.844083 4.494555 15.953851 29.21274  bimodal   15
## 13    visual    20 20.98682 1.863836 4.689257 14.303832 27.66980   visual   20
## 14  acoustic    20 15.46569 1.858311 4.634444  8.808489 22.12290 acoustic   20
## 15   bimodal    20 23.00341 1.851533 4.567400 16.369033 29.63778  bimodal   20
## 16    visual    25 21.40693 1.890917 4.966603 15.229809 27.58404   visual   25
## 17  acoustic    25 15.88580 1.889666 4.954056  9.713656 22.05795 acoustic   25
## 18   bimodal    25 23.42351 1.878059 4.833941 17.300420 29.54661  bimodal   25
## 19    visual    30 21.82704 1.936143 5.456253 15.508598 28.14547   visual   30
## 20  acoustic    30 16.30591 1.939010 5.489135  9.967104 22.64472 acoustic   30
## 21   bimodal    30 23.84362 1.922873 5.309772 17.564836 30.12241  bimodal   30
```

```
      emmip(mean_speed_max.final, fstimulus~trial, at=list(trial=seq(0,30,by=5)), type = "response",
adjust = "mvt", CIs = TRUE, plotit=TRUE) +
        geom_jitter(aes(x = trial, y = mean_dive_speed_max), data = multi_mean, color = "red", alpha
=0.2, height=0, width=0.25)
```



```
      contrast(emmeans(mean_speed_max.final, ~fstimulus, type = "response", adjust = "mvt"),
                                                  list(`visual-bimodal` = c(-1,0,1), `acoustic-
bimodal` = c(0,-1,1)))
```

```
##  contrast         estimate    SE  df t.ratio p.value
##  visual-bimodal       2.02 0.797 132 2.529   0.0126
##  acoustic-bimodal     7.54 0.776 132 9.717   <.0001
##
## Degrees-of-freedom method: kenward-roger
```

```
      emtrends(mean_speed_max.final, ~fstimulus, var="trial", type = "response", adjust = "mvt")
```

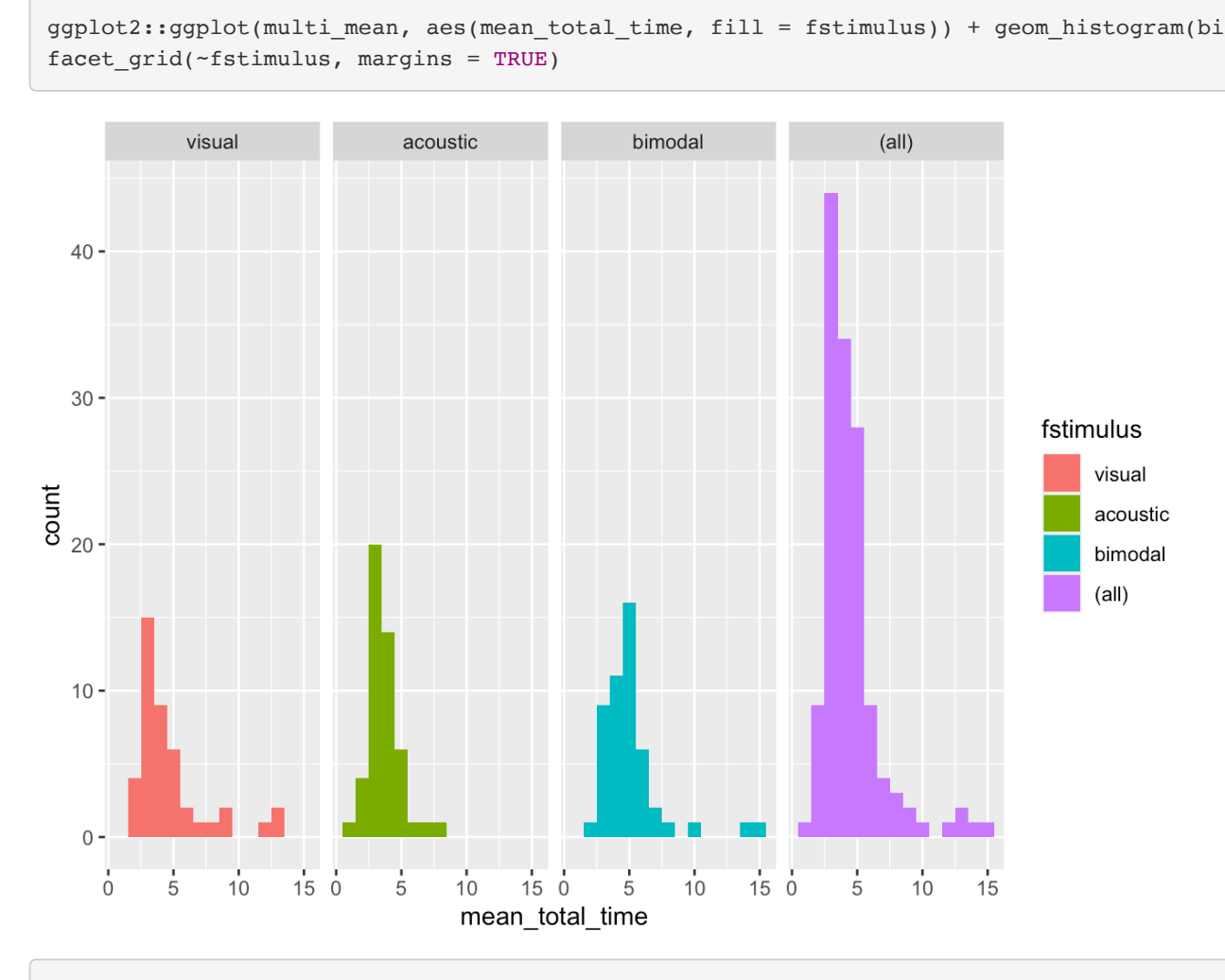
```
##  fstimulus trial.trend     SE  df lower.CL upper.CL
##  visual          0.084 0.0378 132  0.00926    0.159
##  acoustic        0.084 0.0378 132  0.00926    0.159
##  bimodal         0.084 0.0378 132  0.00926    0.159
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: mvt method for 3 estimates
```

Total dive duration - LMER

```
mean(multi_mean$mean_total_time,na.rm=TRUE)
```

```
## [1] 4.531779
```

```
ggplot2::ggplot(multi_mean, aes(mean_total_time, fill = fstimulus)) + geom_histogram(binwidth = 1) +
facet_grid(~fstimulus, margins = TRUE)
```



```
####### LMM #######
    library(lme4)
    library(lmerTest)
    library(tidyverse)

    #random effect structure: varying Intercept best represents exp. design
    #fixed effect structure
      mean_tottime.f0<-lmer(log(mean_total_time)~1 + (1|fgroup), REML=FALSE, data=drop_na(multi_mean
, mean_total_time))
      mean_tottime.stimulus<-lmer(log(mean_total_time)~fstimulus + (1|fgroup), REML=FALSE, data=drop
_na(multi_mean, mean_total_time))
      mean_tottime.trial<-lmer(log(mean_total_time)~trial + (1|fgroup), REML=FALSE, data=drop_na(mul
ti_mean, mean_total_time))
      mean_tottime.inter<-lmer(log(mean_total_time)~fstimulus*trial + (1|fgroup), REML=FALSE, data=d
rop_na(multi_mean, mean_total_time))
      mean_tottime.2way <- update(mean_tottime.inter, . ~ . -fstimulus:trial)

      bbmle::AICctab(mean_tottime.f0, mean_tottime.stimulus, mean_tottime.trial, mean_tottime.inter,
mean_tottime.2way,
              base=TRUE, weights=TRUE,delta=TRUE,sort=TRUE)
```

```
##                        AICc  dAICc df weight
## mean_tottime.2way       98.8   0.0 6  0.859
## mean_tottime.inter     102.6   3.8 8  0.126
## mean_tottime.stimulus 106.9   8.2 5  0.014
## mean_tottime.trial     119.7  21.0 4  <0.001
## mean_tottime.f0        124.6  25.8 3  <0.001
```

```
      drop1(mean_tottime.2way) #test significance of fixed effects
```

```
## Single term deletions using Satterthwaite's method:
##
## Model:
## log(mean_total_time) ~ fstimulus + trial + (1 | fgroup)
##           Sum Sq Mean Sq NumDF  DenDF F value   Pr(>F)
## fstimulus 2.7219   1.361     2 134.99  13.901 3.24e-06 ***
## trial     1.0540   1.054     1 135.03  10.765 0.001316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
    ## model diagnostics ##
      summary(mean_tottime.final <- update(mean_tottime.2way, REML=TRUE))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: log(mean_total_time) ~ fstimulus + trial + (1 | fgroup)
##    Data: drop_na(multi_mean, mean_total_time)
##
## REML criterion at convergence: 105.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.49846 -0.60221 -0.05505  0.46545  2.47183
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fgroup   (Intercept) 0.0709   0.2663
##  Residual             0.1001   0.3164
## Number of obs: 140, groups:  fgroup, 5
##
## Fixed effects:
##                    Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)         1.584977   0.137300   6.356136  11.544 1.69e-05 ***
## fstimulusacoustic  -0.200113   0.066591 132.000159  -3.005  0.00318 **
## fstimulusbimodal    0.133801   0.066191 132.007101   2.021  0.04526 *
## trial              -0.010187   0.003137 132.025456  -3.247  0.00148 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fstmlsc fstmlsb
## fstimlscstc -0.274
## fstimlsbmdl -0.254  0.530
## trial       -0.351  0.052  -0.009
```

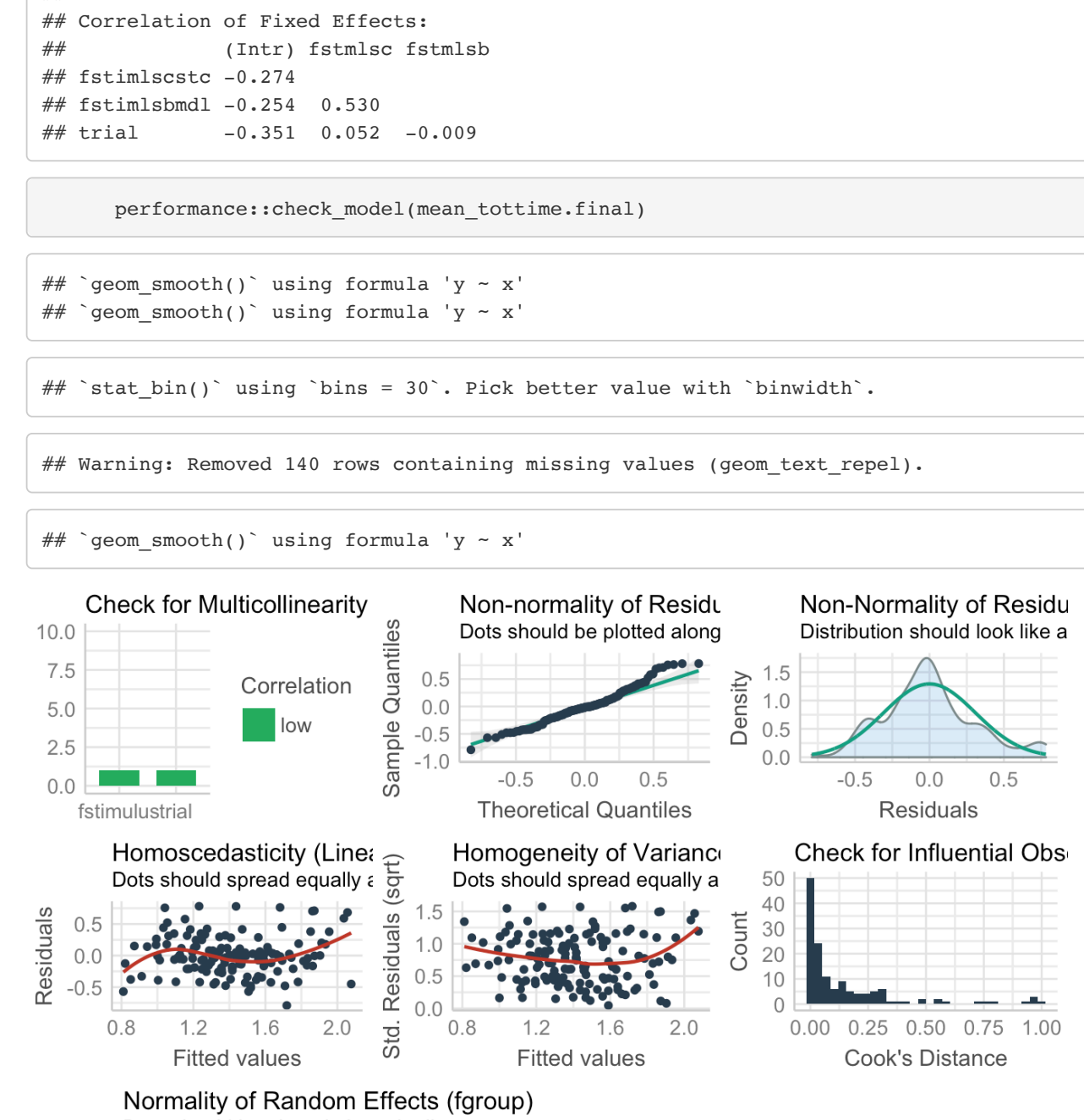
```
      performance::check_model(mean_tottime.final)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

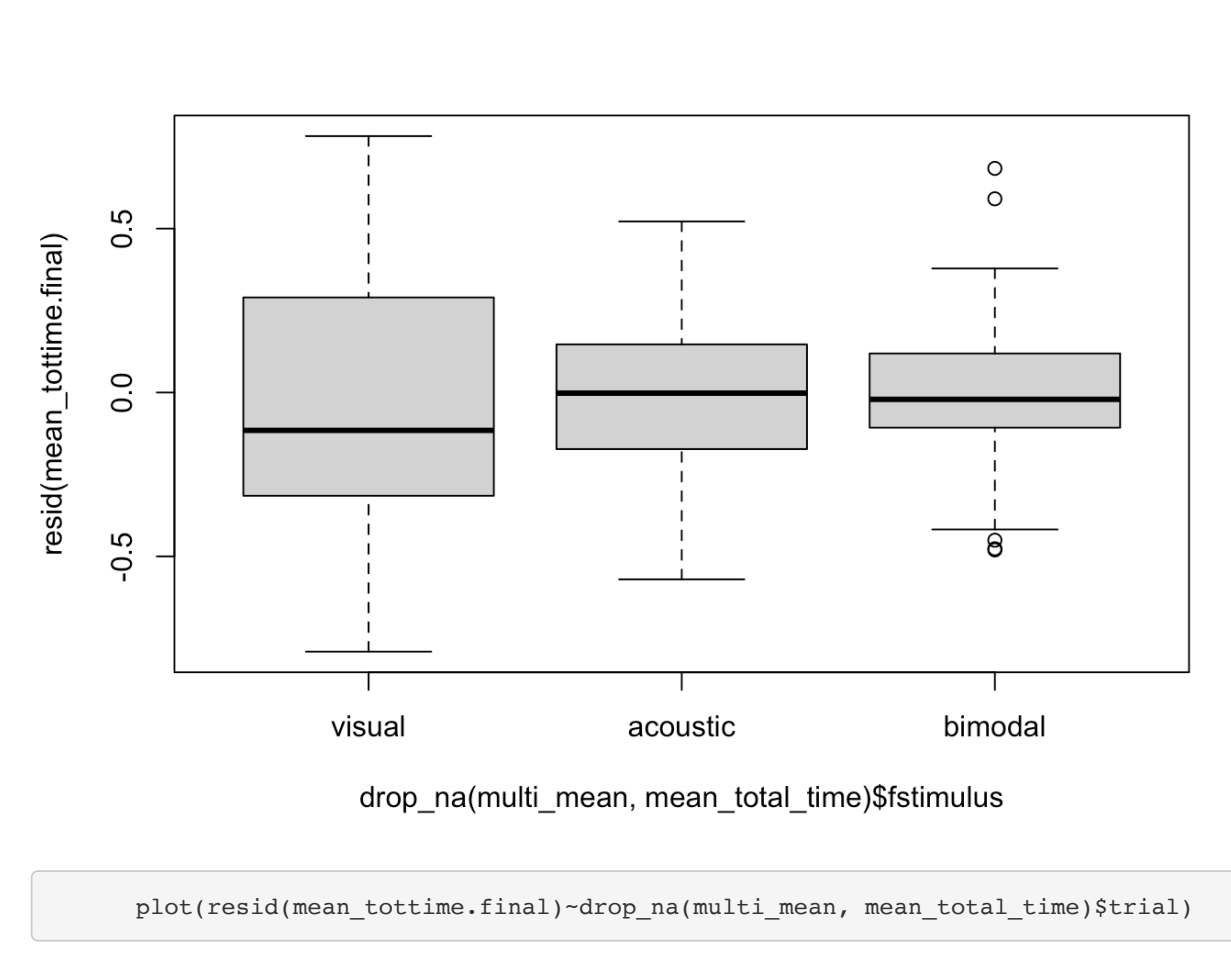
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 140 rows containing missing values (geom_text_repel).
```

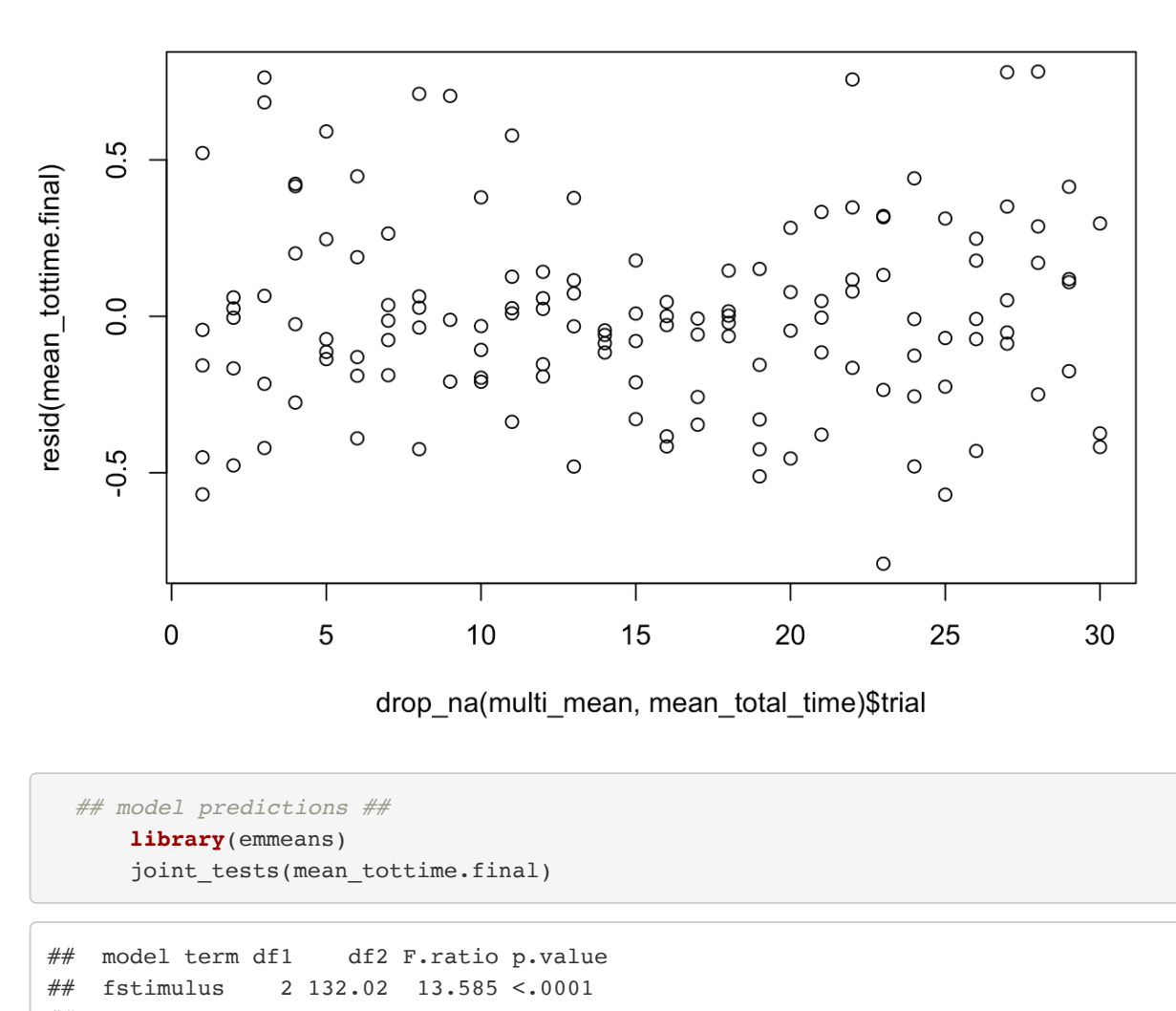
```
## `geom_smooth()` using formula 'y ~ x'
```



```
      plot(resid(mean_tottime.final)~drop_na(multi_mean, mean_total_time)$fstimulus) #plot residuals
versus predictors to see where heterogeneity is coming from
```



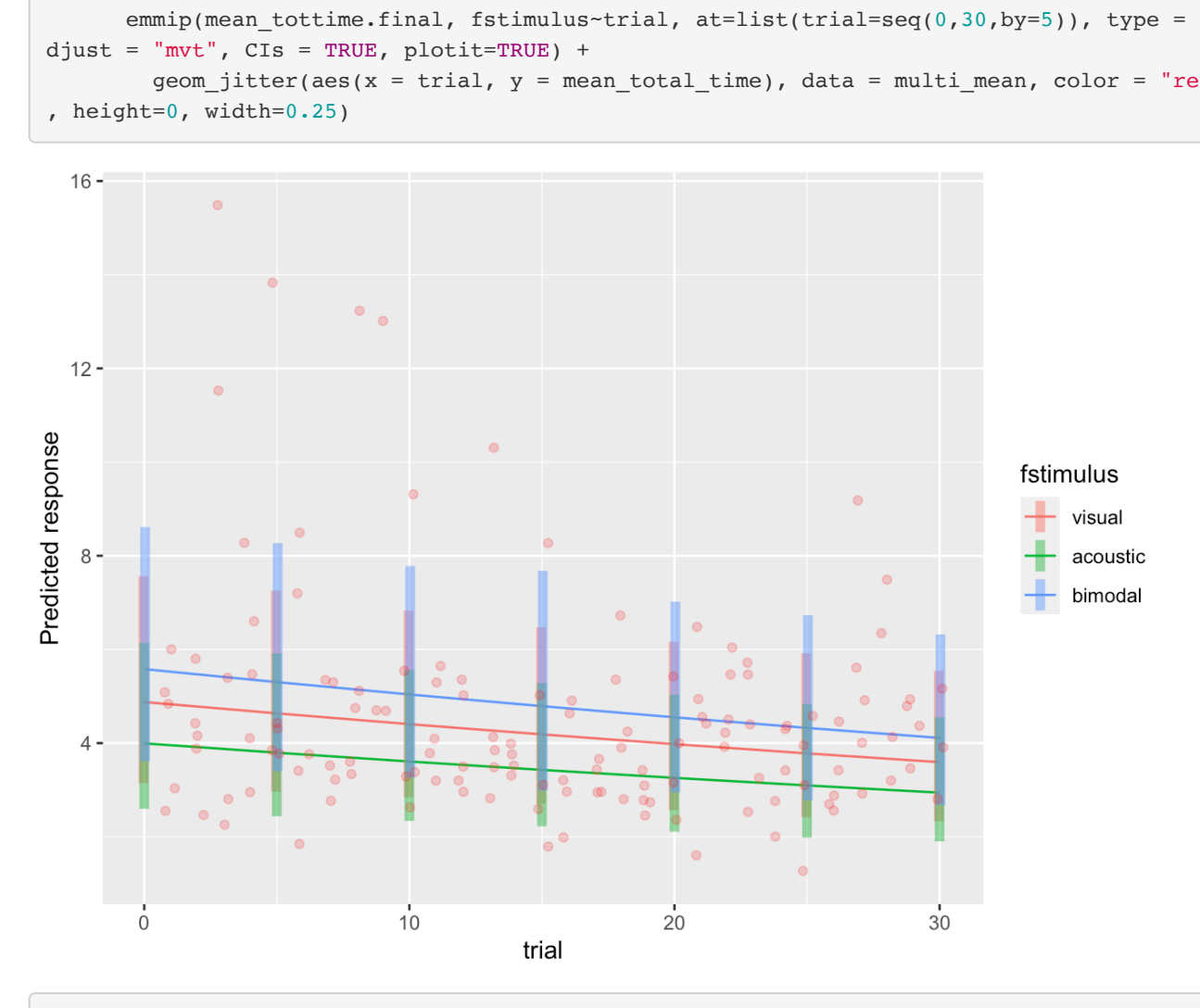
```
      plot(resid(mean_tottime.final)~drop_na(multi_mean, mean_total_time)$trial)
```



```
  ## model predictions ##
      library(emmeans)
      joint_tests(mean_tottime.final)
```

```
##  model term df1    df2 F.ratio p.value
##  fstimulus    2 132.02  13.585 <.0001
##  trial        1 132.05  10.544 0.0015
```

```
      mean_tottime_EMMs <- emmip(mean_tottime.final, fstimulus~trial, at=list(trial=seq(0,30,by=5)),
type = "response", adjust = "mvt", CIs = TRUE, plotit=FALSE)
      emmip(mean_tottime.final, fstimulus~trial, at=list(trial=seq(0,30,by=5)), type = "response", a
djust = "mvt", CIs = TRUE, plotit=TRUE) +
        geom_jitter(aes(x = trial, y = mean_total_time), data = multi_mean, color = "red", alpha=0.2
, height=0, width=0.25)
```



```
      contrast(emmeans(mean_tottime.final, ~fstimulus, type = "response", adjust = "mvt"),
                                                  list(`visual-bimodal` = c(-1,0,1), `acoustic-
bimodal` = c(0,-1,1)))
```

```
##  contrast         ratio     SE  df t.ratio p.value
##  visual-bimodal    1.14 0.0757 132 2.021   0.0453
##  acoustic-bimodal  1.40 0.0899 132 5.185   <.0001
##
## Degrees-of-freedom method: kenward-roger
## Tests are performed on the log scale
```

```
      emtrends(mean_tottime.final, ~fstimulus, var="trial", type = "response", adjust = "mvt")
```

```
##  fstimulus trial.trend      SE  df lower.CL upper.CL
##  visual        -0.0102 0.00314 132  -0.0164 -0.00398
##  acoustic      -0.0102 0.00314 132  -0.0164 -0.00398
##  bimodal       -0.0102 0.00314 132  -0.0164 -0.00398
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: mvt method for 3 estimates
```

Data Visualization

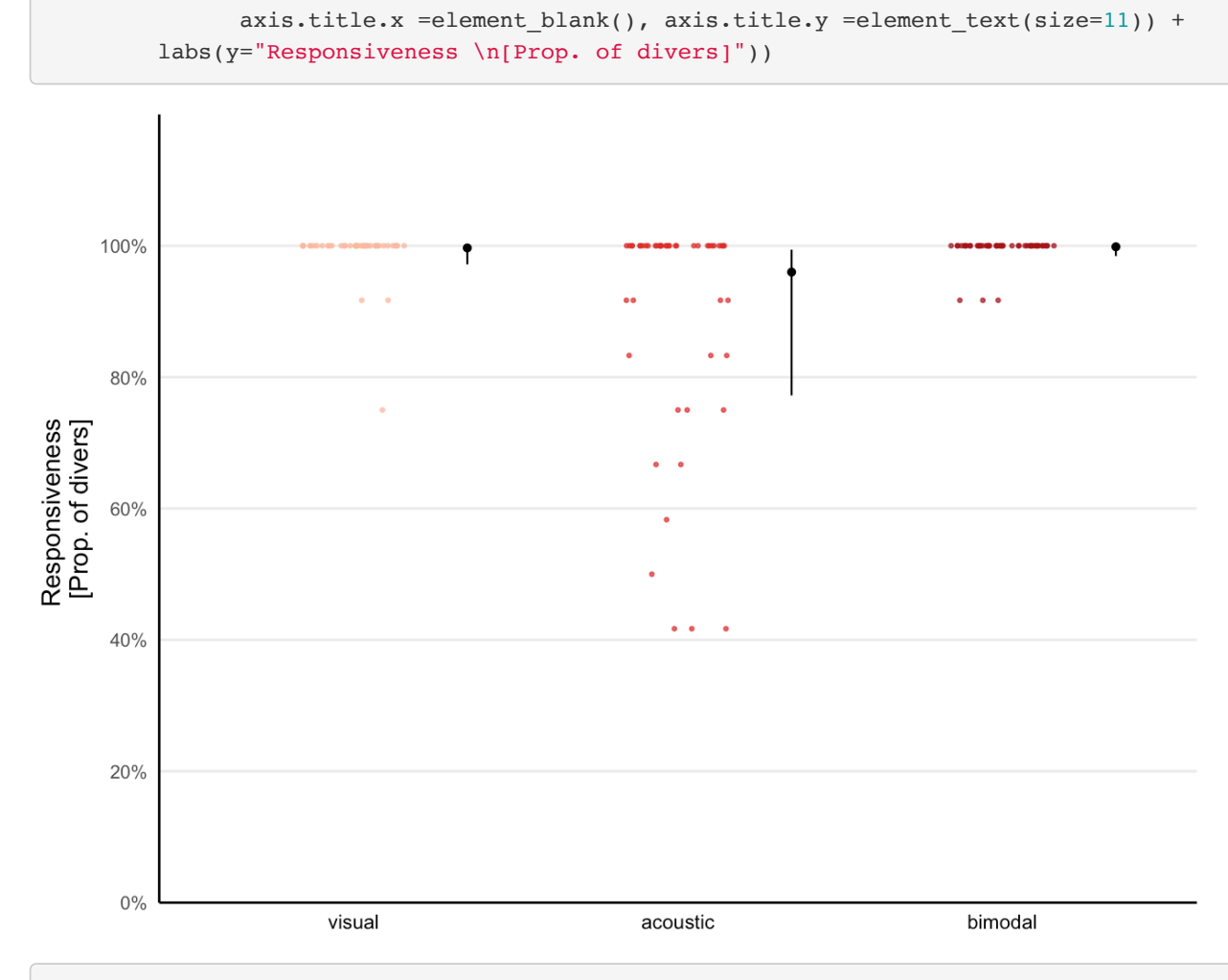
```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
```

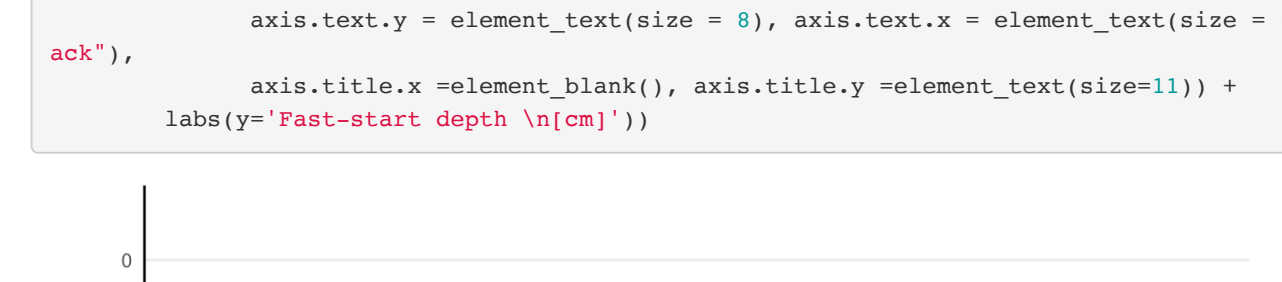
```
## The following object is masked from 'package:purrr':
##
##     discard
```

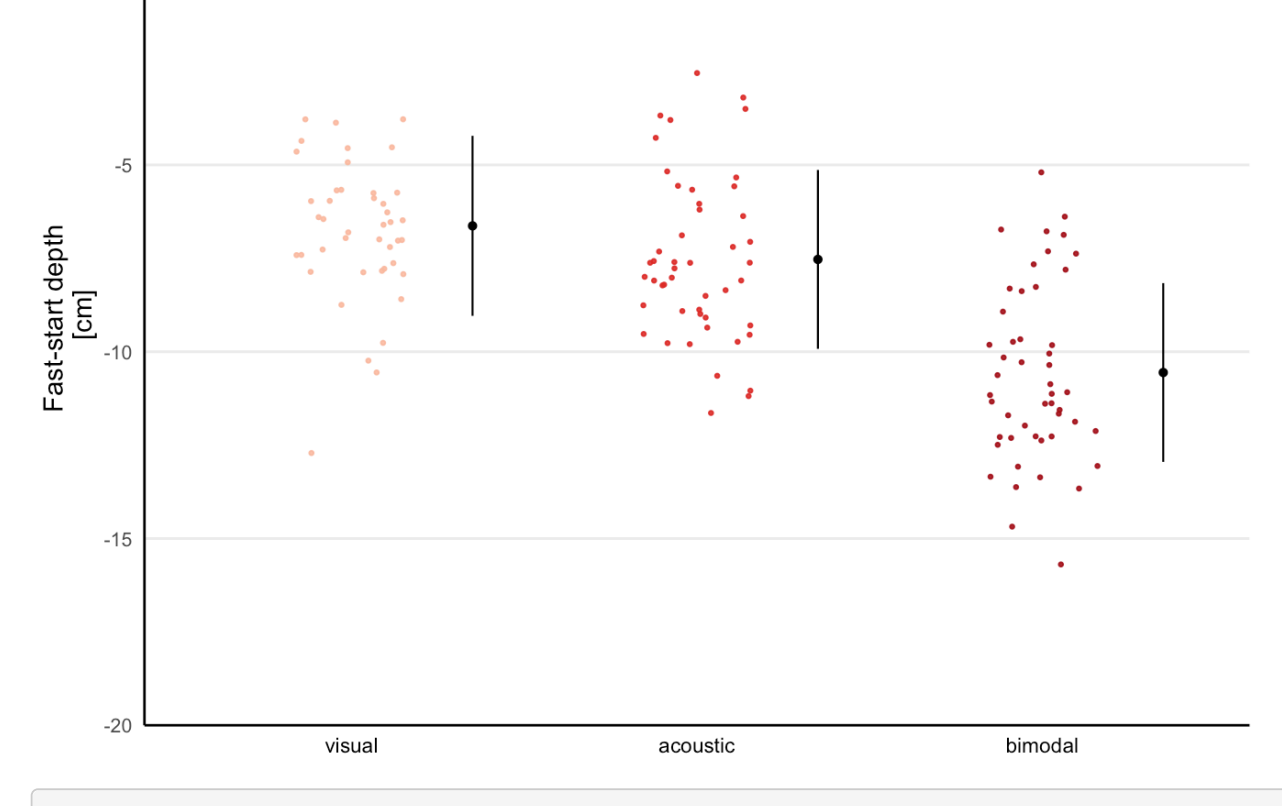
```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
(p_response<-ggplot() +
        geom_jitter(data=multi_mean, aes(x=fstimulus, y=DiveRate, color=fstimulus),
                size=0.5, alpha=0.7, position = position_jitterdodge(jitter.height = 0, jitter.wi
dth = 0.8)) +
        geom_pointrange(data=mean_respo_EMMs, aes(x=fstimulus, y=yvar, ymin = LCL, ymax = UCL),
                    position = position_nudge(0.35), color="black", size=0.4, fatten=1.2) +
        scale_color_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
        scale_y_continuous(limits = c(0, 1.2), breaks = seq(0, 1.1, by = 0.2), labels = percent, oob
= scales::squish, expand = c(0, 0)) +
        theme_minimal() +
        theme(plot.title = element_blank(), panel.background = element_blank(),
              panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
              legend.position = "null", axis.line = element_line(colour = "black"),
              axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 8.5, color="bl
ack"),
              axis.title.x =element_blank(), axis.title.y =element_text(size=11)) +
        labs(y="Responsiveness \n[Prop. of divers]"))
```

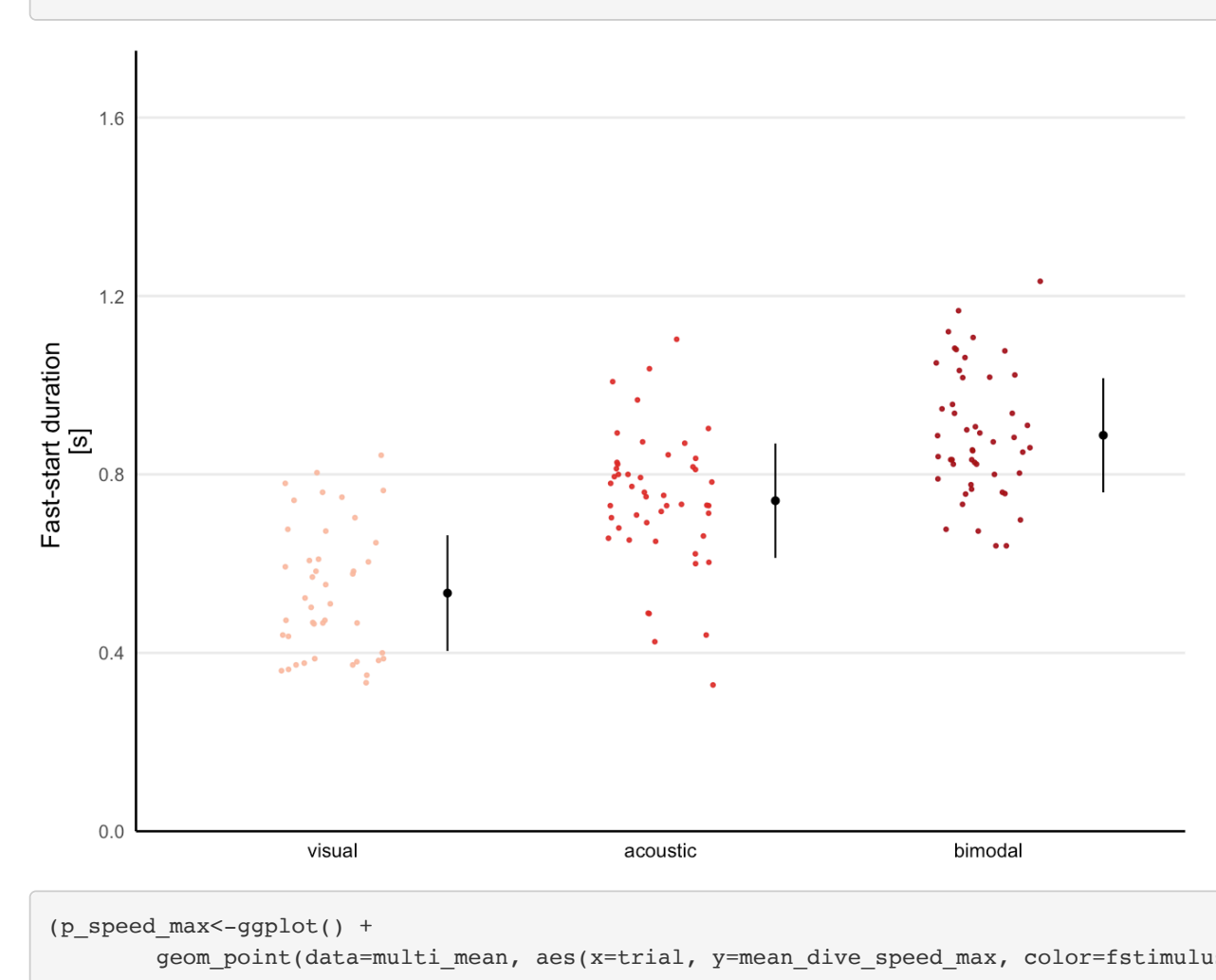


```
(p_depth<-ggplot() +
        geom_jitter(data=multi_mean, aes(x=fstimulus, y=mean_dive_depth, color=fstimulus),
                size=0.5, alpha=0.9, position = position_jitterdodge(jitter.height = 0, jitter.w
idth = 0.8)) +
        geom_pointrange(data=mean_depth_EMMs, aes(x=fstimulus, y=yvar, ymin = LCL, ymax = UCL),
                    position = position_nudge(0.35), color="black", size=0.4, fatten=1.2) +
        scale_color_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
        scale_y_continuous(limits = c(-20, 2), oob = scales::squish, expand = c(0, 0)) +
        theme_minimal() +
        theme(plot.title = element_blank(), panel.background = element_blank(),
              panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
              legend.position = "null", axis.line = element_line(colour = "black"),
              axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 8.5, color="bl
ack"),
              axis.title.x =element_blank(), axis.title.y =element_text(size=11)) +
        labs(y='Fast-start depth \n[cm]'))
```

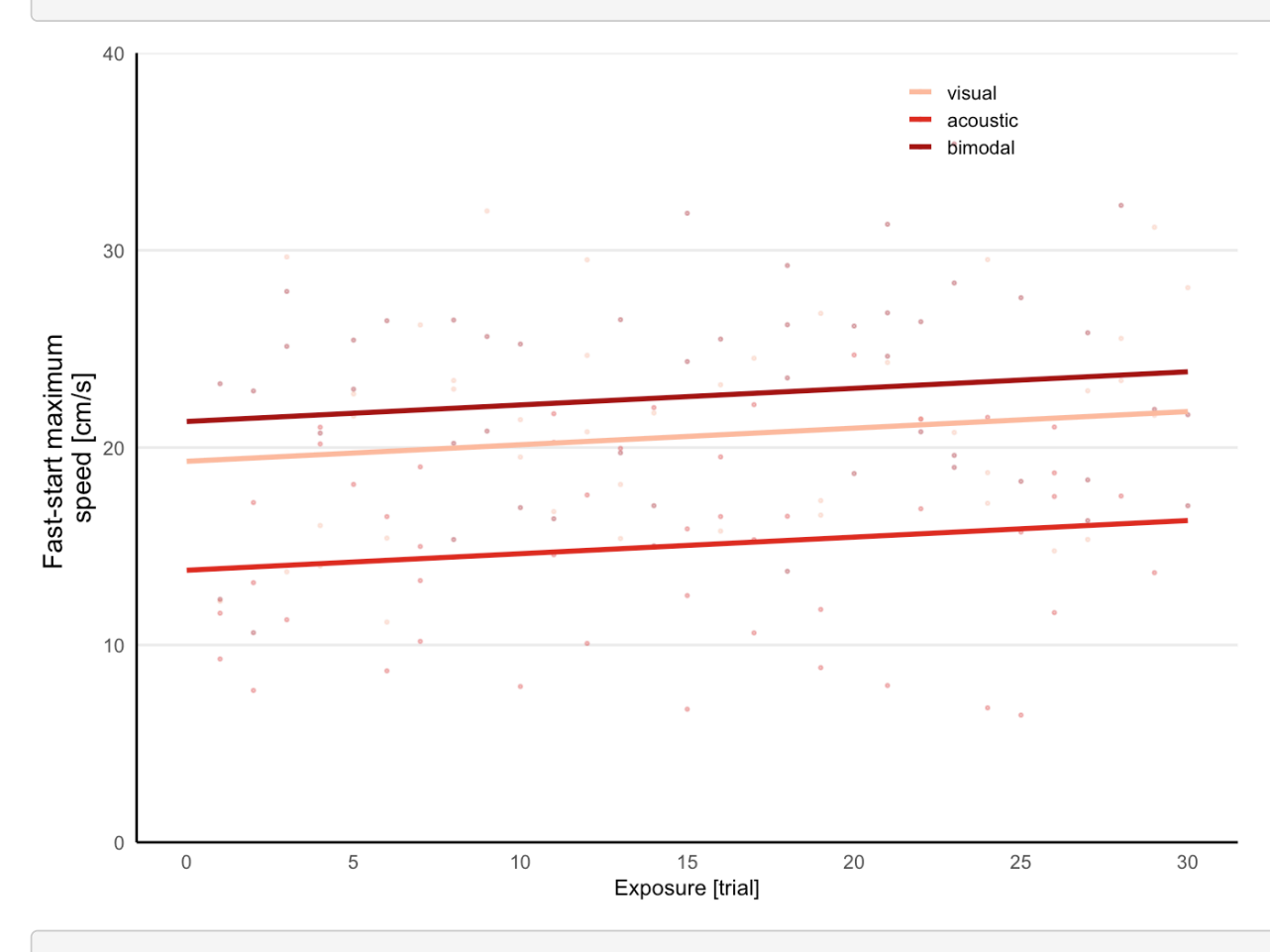




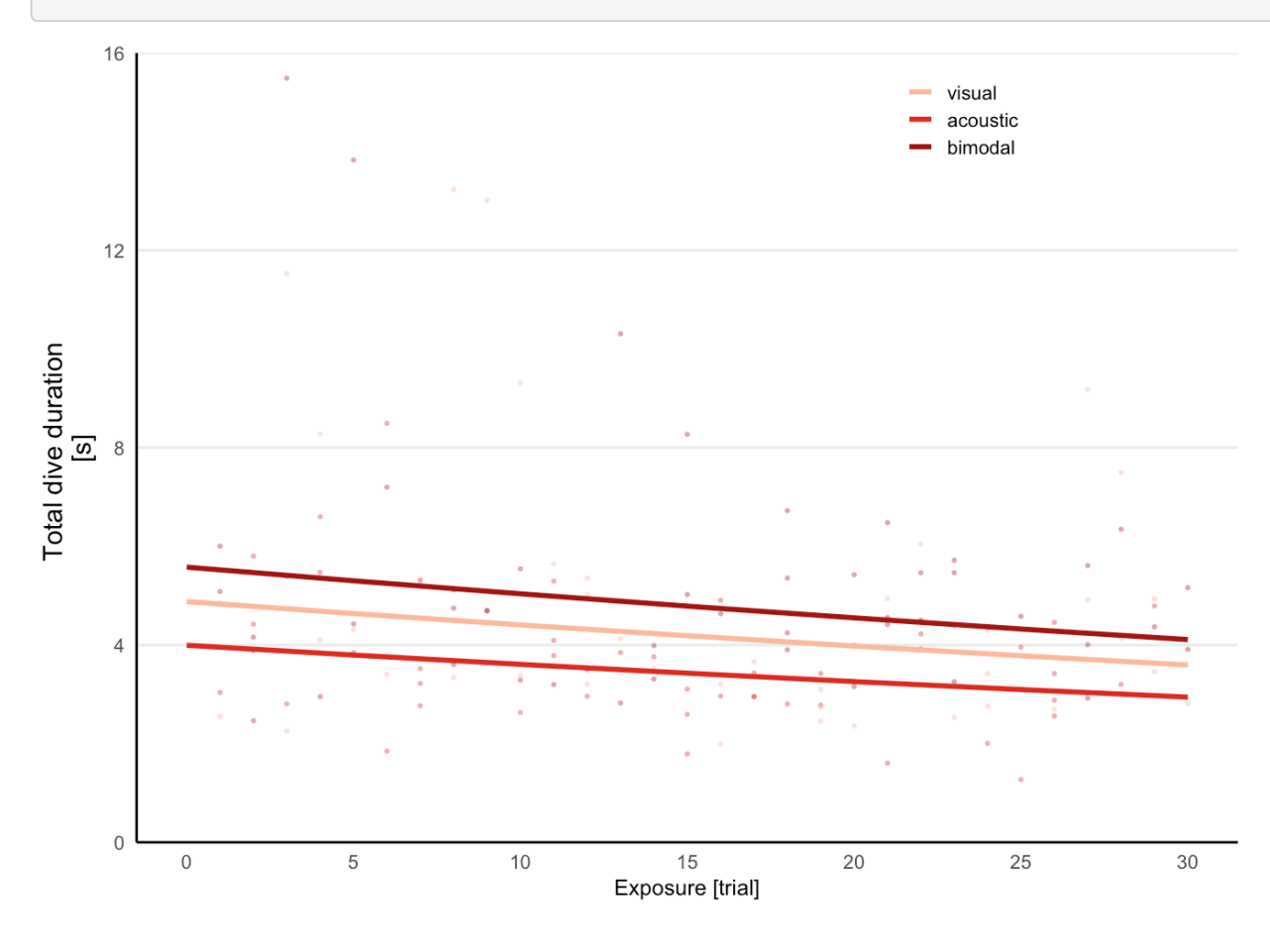
```
(p_time<-ggplot() +
        geom_jitter(data=multi_mean, aes(x=fstimulus, y=mean_dive_time, color=fstimulus),
                    size=0.5, alpha=0.9, position = position_jitterdodge(jitter.height = 0, jitter.w
idth = 0.8)) +
        geom_pointrange(data=mean_time_EMMs, aes(x=fstimulus, y=yvar, ymin = LCL, ymax = UCL),
                        position = position_nudge(0.35), color="black", size=0.4, fatten=1.2) +
        scale_color_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
        scale_y_continuous(limits = c(0, 1.75), breaks = seq(0, 1.75, by = 0.4), oob = scales::squis
h, expand = c(0, 0)) +
        theme_minimal() +
        theme(plot.title = element_blank(), panel.background = element_blank(),
              panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
              legend.position = "null", axis.line = element_line(colour = "black"),
              axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 8.5, color="bl
ack"),
              axis.title.x =element_blank(), axis.title.y =element_text(size=11)) +
        labs(y='Fast-start duration \n[s]'))
```



```
(p_speed_max<-ggplot() +
        geom_point(data=multi_mean, aes(x=trial, y=mean_dive_speed_max, color=fstimulus),
                   size=0.4, alpha=0.3) +
        # geom_ribbon(data=mean_speed_max_EMMs, aes(x=trial, y=yvar, ymin = LCL, ymax = UCL, group=f
stimulus, fill=fstimulus),
        #             size=0.6, alpha=0.1) +
        geom_line(data=mean_speed_max_EMMs, aes(x=trial, y=yvar, group=fstimulus, color=fstimulus),
size=1) +
        scale_color_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
        scale_fill_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
        scale_x_continuous(breaks = seq(0, 33, by = 5)) +
        scale_y_continuous(limits = c(0, 40), oob = scales::squish, expand = c(0, 0)) +
        theme_minimal() +
        theme(plot.title = element_blank(), panel.background = element_blank(),
              panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
              legend.position = c(0.75,0.94), legend.text=element_text(size=8), legend.key.size = un
it(0.4, "cm"),
              axis.line = element_line(colour = "black"),
              axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 8),
              axis.title.y =element_text(size=11), axis.title.x =element_text(size=9)) +
        labs(y='Fast-start maximum \nspeed [cm/s]', x='Exposure [trial]', color="", fill=""))
```



```
(p_tottime<-ggplot() +
        geom_point(data=multi_mean, aes(x=trial, y=mean_total_time, color=fstimulus),
                   size=0.4, alpha=0.3) +
        # geom_ribbon(data=mean_tottime_EMMs, aes(x=trial, y=yvar, ymin = LCL, ymax = UCL, group=fst
imulus, fill=fstimulus),
        #             size=0.6, alpha=0.1) +
        geom_line(data=mean_tottime_EMMs, aes(x=trial, y=yvar, group=fstimulus, color=fstimulus), si
ze=1) +
        scale_color_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
        scale_fill_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
        scale_x_continuous(breaks = seq(0, 33, by = 5)) +
        scale_y_continuous(limits = c(0, 16), oob = scales::squish, expand = c(0, 0)) +
        theme_minimal() +
        theme(plot.title = element_blank(), panel.background = element_blank(),
              panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
              legend.position = c(0.75,0.94), legend.text=element_text(size=8), legend.key.size = un
it(0.4, "cm"),
              axis.line = element_line(colour = "black"),
              axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 8),
              axis.title.y =element_text(size=11), axis.title.x =element_text(size=9)) +
        labs(y='Total dive duration \n[s]', x='Exposure [trial]', color="", fill=""))
```



3) Supplement

Field - Diving

```
library(readr)
df_dives <- read_delim("../data/DATA_SI_DivingField.csv",",", escape_double = FALSE, trim_ws = TRUE)
```

```
## Parsed with column specification:
## cols(
##   Site = col_double(),
##   Sampling_date = col_character(),
##   Sampling_time = col_time(format = ""),
##   Trial = col_double(),
##   H2S_µmolL_Culumber2016 = col_double(),
##   Ox_mgL_Culumber2016 = col_double(),
##   Depth_cm = col_double(),
##   Distance_cm = col_double(),
##   Turbidity_ntu = col_double(),
##   Video_name_50fps = col_character(),
##   Dichte = col_double(),
##   Frame_dive = col_double(),
##   Frame_surface = col_double(),
##   Divetime_s = col_double()
## )
```

```
library(tidyverse)
df_dives<-df_dives %>%
        dplyr::select(Divetime_s, Site, Trial) %>%
        subset(!is.na(Divetime_s)) #exclude sites for which we did not quantify DiveTimes
#set up variables
df_dives$fsite<-plyr::mapvalues(as.factor(df_dives$Site), from=c("1", "4", "5", "9", "10", "11"), to
=c("1", "2", "3", "4", "5", "6"))
df_dives$trial<-as.numeric(ordered(as.factor(df_dives$Trial), levels=c("1", "2", "3", "4", "5", "6",
"7", "8", "9", "10", "11", "12")))

#data descriptives:
  doBy::summaryBy(Divetime_s~fsite, data=df_dives, FUN=c(length,min,median,max,mean,sd))
```

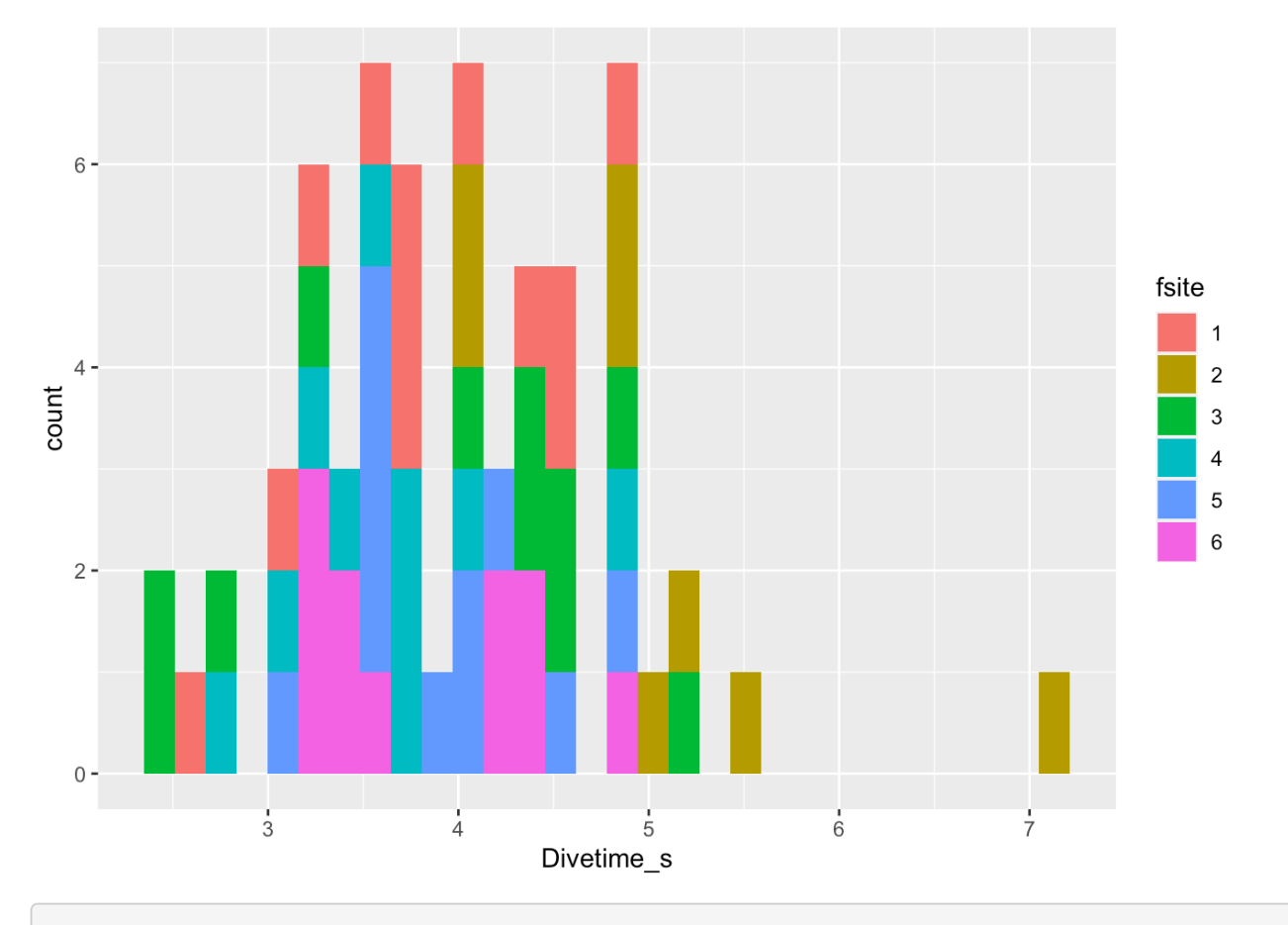
```
## # A tibble: 6 x 7
##   fsite Divetime_s.leng… Divetime_s.min Divetime_s.medi… Divetime_s.max
##   <fct>            <dbl>          <dbl>            <dbl>          <dbl>
## 1 1                   12            2.6              3.8            4.8
## 2 2                    8            4                4.9            7.1
## 3 3                   11            2.4              4.4            5.2
## 4 4                   10            2.7              3.6            4.8
## 5 5                   11            3                3.9            4.8
## 6 6                   11            3.2              3.6            4.8
## # … with 2 more variables: Divetime_s.mean <dbl>, Divetime_s.sd <dbl>
```

```
  mean(df_dives$Divetime_s,na.rm=TRUE)
```

```
## [1] 3.969841
```

```
  ggplot2::ggplot(df_dives, aes(Divetime_s, fill = fsite)) + geom_histogram() #+ facet_wrap(~fsampli
ng)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
####### LMM #######
    library(lme4)
    library(lmerTest)
    library(tidyverse)

    #fixed effect structure
      DiveTime.m1<-lmer(Divetime_s~trial + (1|fsite), REML=FALSE, data=drop_na(df_dives, Divetime_s)
)
      drop1(DiveTime.m1, test="Chisq")
```

```
## Single term deletions using Satterthwaite's method:
##
## Model:
## Divetime_s ~ trial + (1 | fsite)
##       Sum Sq Mean Sq NumDF  DenDF F value  Pr(>F)
## trial 1.8619  1.8619     1 57.378  3.6092 0.06249 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
    ## model diagnostics ##
      summary(DiveTime.final <- update(DiveTime.m1, REML=TRUE))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Divetime_s ~ trial + (1 | fsite)
##    Data: drop_na(df_dives, Divetime_s)
##
## REML criterion at convergence: 151.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.46681 -0.61788  0.02609  0.70543  3.00305
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fsite    (Intercept) 0.1725   0.4153
##  Residual             0.5238   0.7237
## Number of obs: 63, groups:  fsite, 6
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  4.31812    0.25707 13.53723  16.797 1.87e-10 ***
## trial       -0.05457    0.02948 56.44259  -1.851   0.0694 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## trial -0.661
```

```
      performance::check_model(DiveTime.final)
```

```
## Not enough model terms in the conditional part of the model to check for multicollinearity.
```

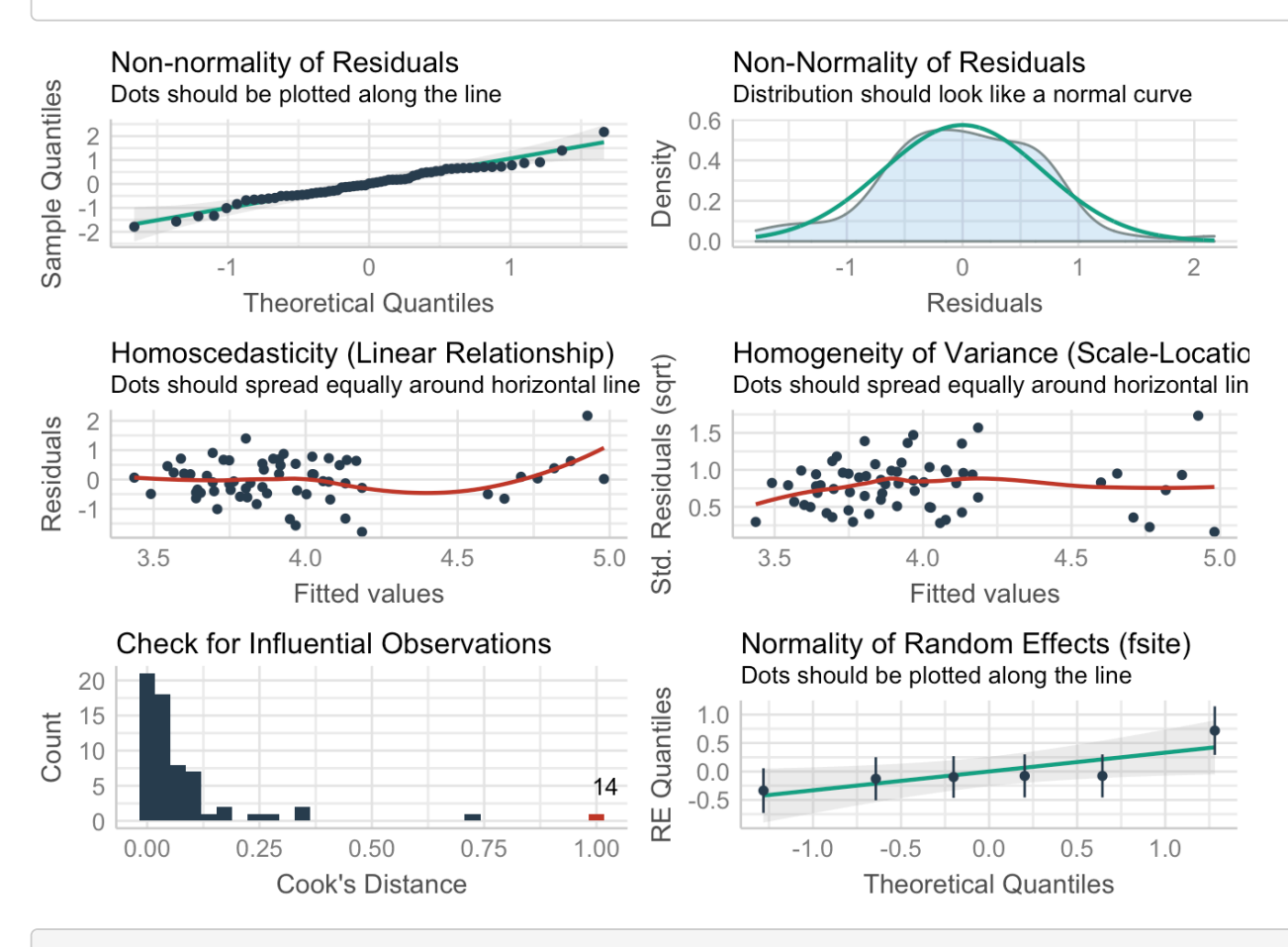
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## `geom_smooth()` using formula 'y ~ x'
```

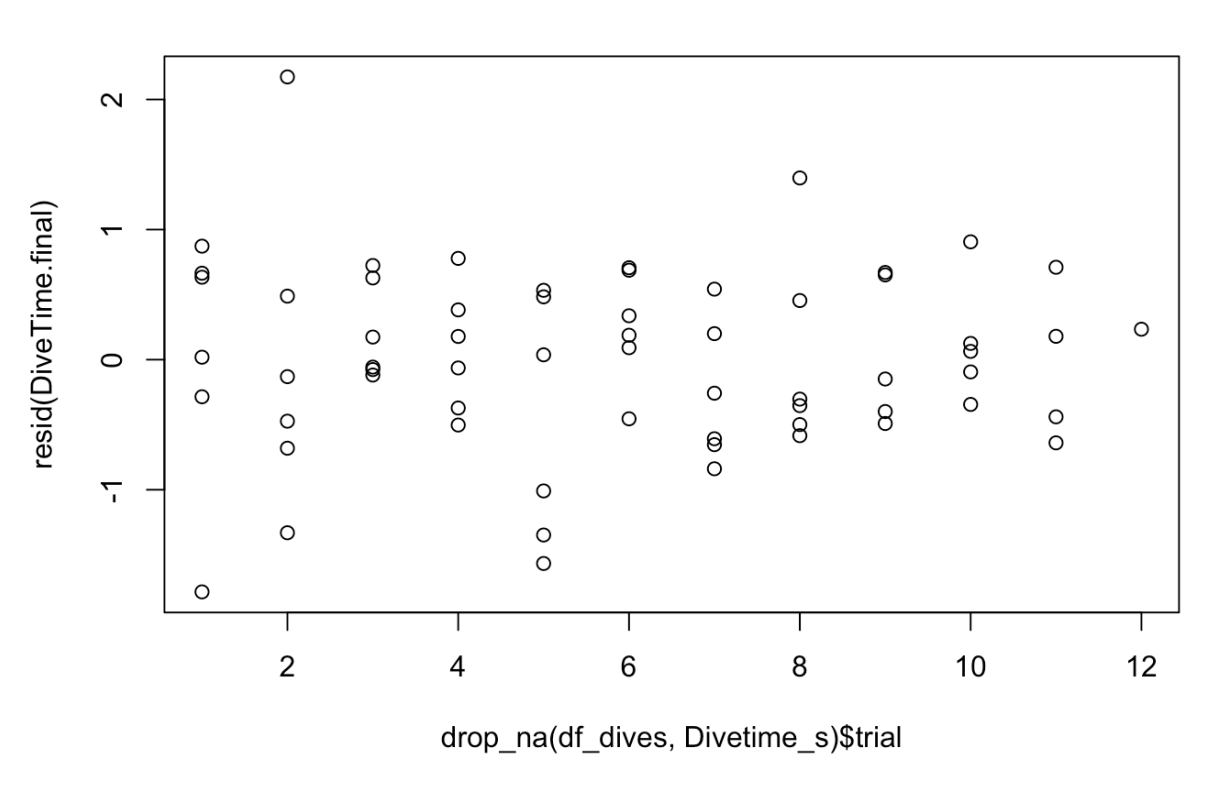
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 62 rows containing missing values (geom_text_repel).
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
      plot(resid(DiveTime.final)~drop_na(df_dives, Divetime_s)$trial) #plot residuals versus predict
ors to see where heterogeneity is coming from
```



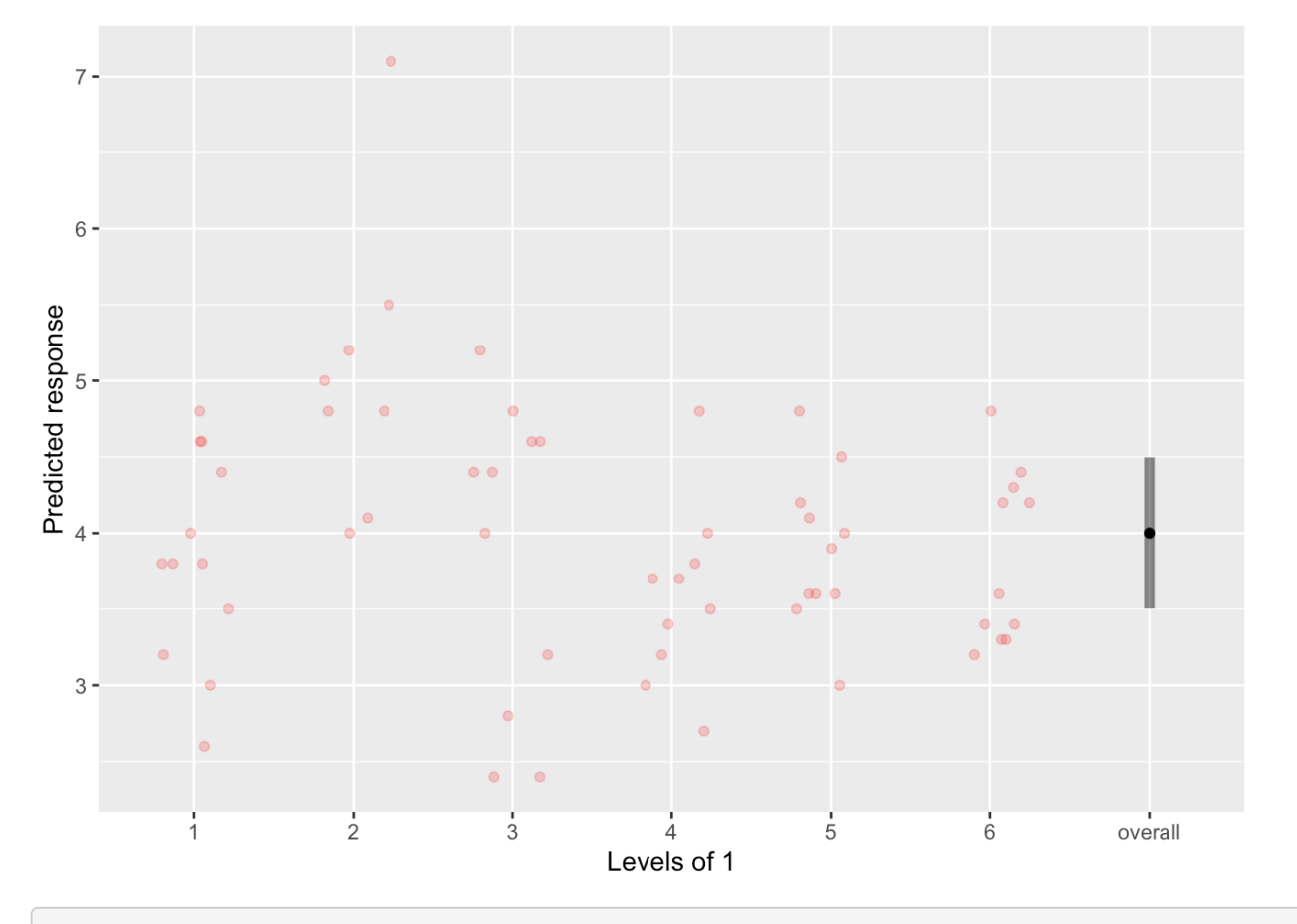
```
    ## model predictions ##
      library(emmeans)
      joint_tests(DiveTime.final)
```

```
##  model term df1   df2 F.ratio p.value
##  trial        1 56.94   3.403 0.0703
```

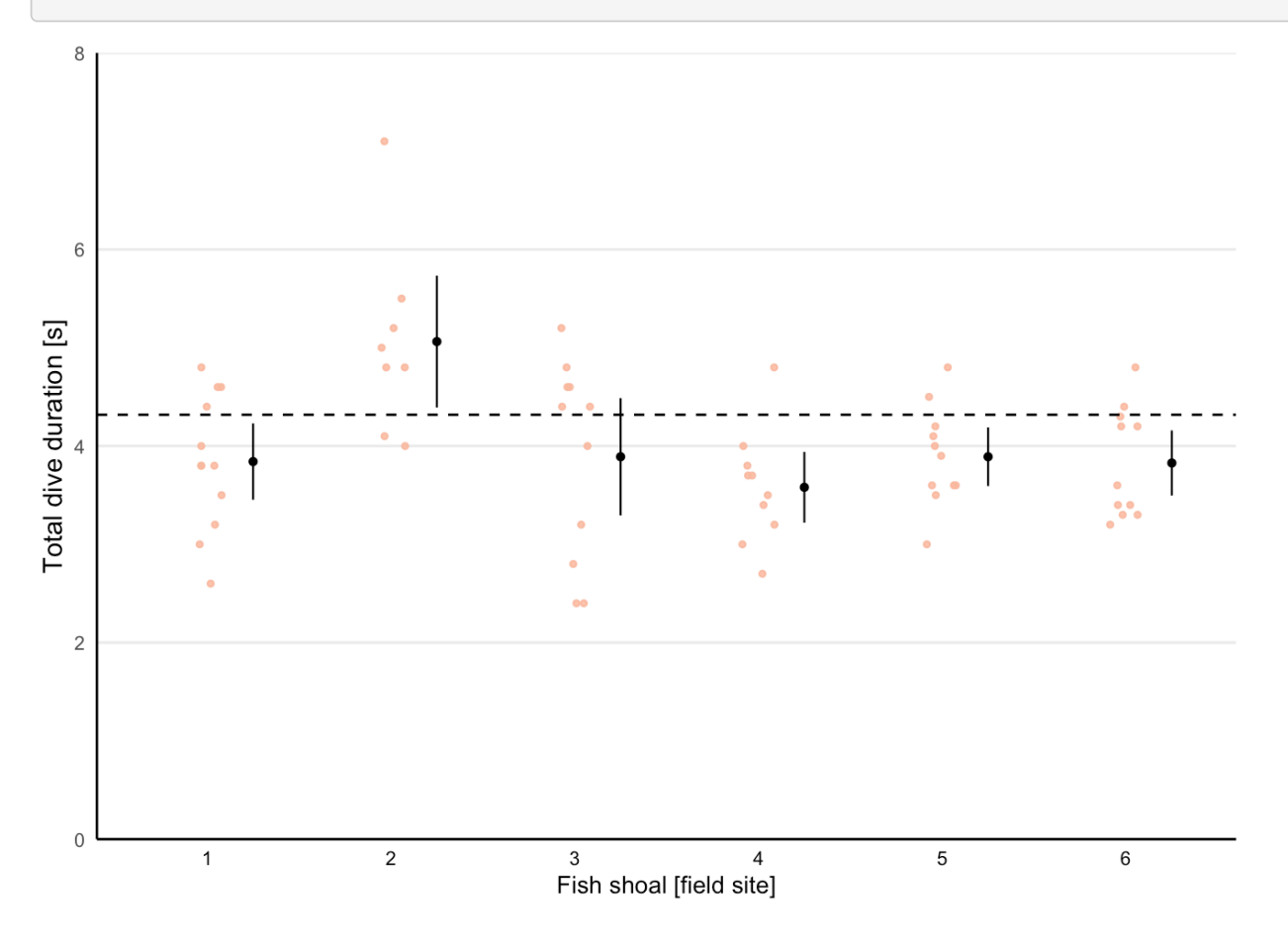
```
      emmip(DiveTime.final, ~1, type = "response", adjust = "mvt", CIs = TRUE, plotit=TRUE) +
        geom_jitter(aes(x = fsite, y = Divetime_s), data = df_dives, color = "red", alpha=0.2, heigh
t=0, width=0.25)
```

```
## Suggestion: Add 'at = list(1 = ...)' to call to see > 1 value per group.
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



```
## Plot raw data and mean_sdl
library(ggplot2)
(p_natDiving<-ggplot(data=df_dives, aes(x = fsite, y = Divetime_s, fill = fsite)) +
        #annotate('ribbon', x = c(-Inf, Inf), ymin = 4.32-0.26, ymax = 4.32+0.26, fill = 'grey92') +
        geom_jitter(size=0.9, alpha=0.9, color="#FCBBA1", position = position_jitterdodge(jitter.hei
ght=0, jitter.width=0.7, dodge.width=0)) +
        geom_hline(aes(yintercept = 4.31812), color="black", linetype = "dashed") +
        stat_summary(fun.data = mean_se, fun.args = list(mult = 1.96),
                     geom = "pointrange", position = position_nudge(0.25), color="black" , size=0.4,
fatten=1.2) +
        scale_y_continuous(limits = c(0, 8), breaks = seq(0, 8, by = 2), oob = scales::squish, expan
d = c(0, 0)) +
        theme_minimal()+
        theme(plot.title = element_blank(), panel.background = element_blank(),
              panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
              legend.position = "null", axis.line = element_line(colour = "black"),
              axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 8, color="blac
k"),
              axis.title.y =element_text(size=11), axis.title.x =element_text(size=10)) +
        labs(y="Total dive duration [s]", x='Fish shoal [field site]'))
```



Lab - Body length

```
#read in the data
  library(readr)
  multi_BL <- read_delim("../data/DATA_SI_DivingBodysize.csv", delim=",", trim_ws = TRUE)
```

```
## Parsed with column specification:
## cols(
##   groupID = col_character(),
##   group = col_double(),
##   trial = col_double(),
##   measurement = col_double(),
##   SL_cm = col_double(),
##   Temp_C = col_double(),
##   Temp_C_after = col_double(),
##   Ox_mgL = col_double(),
##   Ox_mgL_after = col_double()
## )
```

```
  multi_BL$fgroup<-recode_factor(as.factor(multi_BL$group),
                                '1'="group 1", '2'="group 2", '3'="group 3",
                                 '4'="group 4", '5'="group 5",
                                .default = levels(as.factor(multi_BL$group)), .ordered=FALSE);
  multi_BL$ftrial<-factor(multi_BL$trial)
  multi_BL$fobs<-paste(multi_BL$fgroup, multi_BL$ftrial)

#### Group Means ####
  library(tidyverse)
  multi_BL_groupmean<-plyr::ddply(multi_BL,~fobs,plyr::summarise,
                   fgroup=first(fgroup), ftrial=first(ftrial), n=length(SL_cm),
                   mean_SL_cm= mean(SL_cm,na.rm=TRUE), min_SL_cm= min(SL_cm,na.rm=TRUE), max_SL_cm=
max(SL_cm,na.rm=TRUE))
  round(mean(multi_BL_groupmean$mean_SL_cm), digits = 1)
```

```
## [1] 15
```

```
#### ANOVA ####
  library(lme4)
  summary(m1<-lm(mean_SL_cm~fgroup, data=multi_BL_groupmean))
```

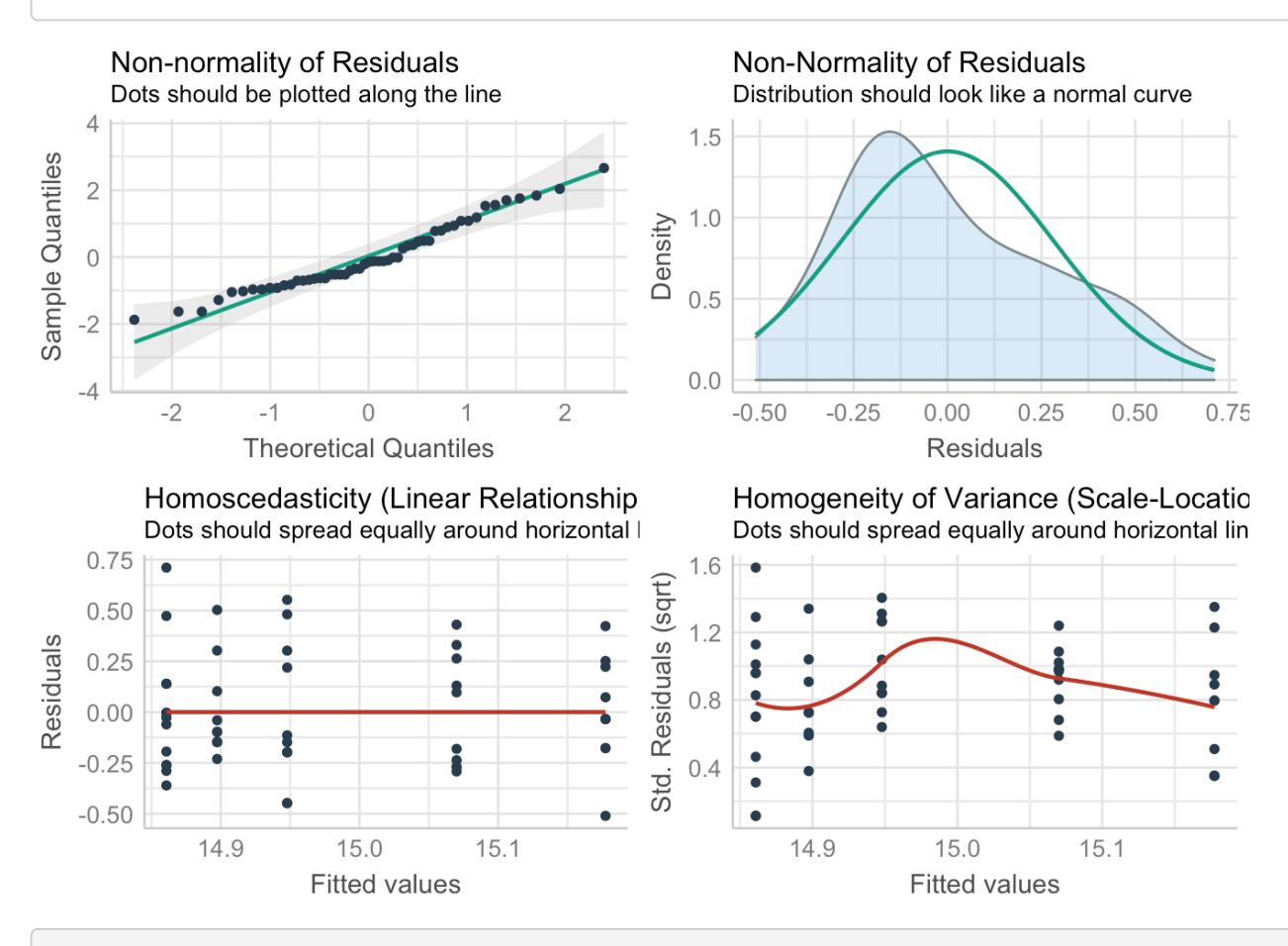
```
##
## Call:
## lm(formula = mean_SL_cm ~ fgroup, data = multi_BL_groupmean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51071 -0.19786 -0.04024  0.21881  0.71062
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    14.94786    0.09327 160.264   <2e-16 ***
## fgroupgroup 2 -0.05048    0.13190  -0.383   0.7037
## fgroupgroup 3 -0.08705    0.12406  -0.702   0.4863
## fgroupgroup 4  0.12214    0.13190   0.926   0.3591
## fgroupgroup 5  0.22952    0.13190   1.740   0.0883 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2949 on 48 degrees of freedom
## Multiple R-squared:  0.1494, Adjusted R-squared:  0.07849
## F-statistic: 2.107 on 4 and 48 DF,  p-value: 0.09441
```

```
  performance::check_model(m1)
```

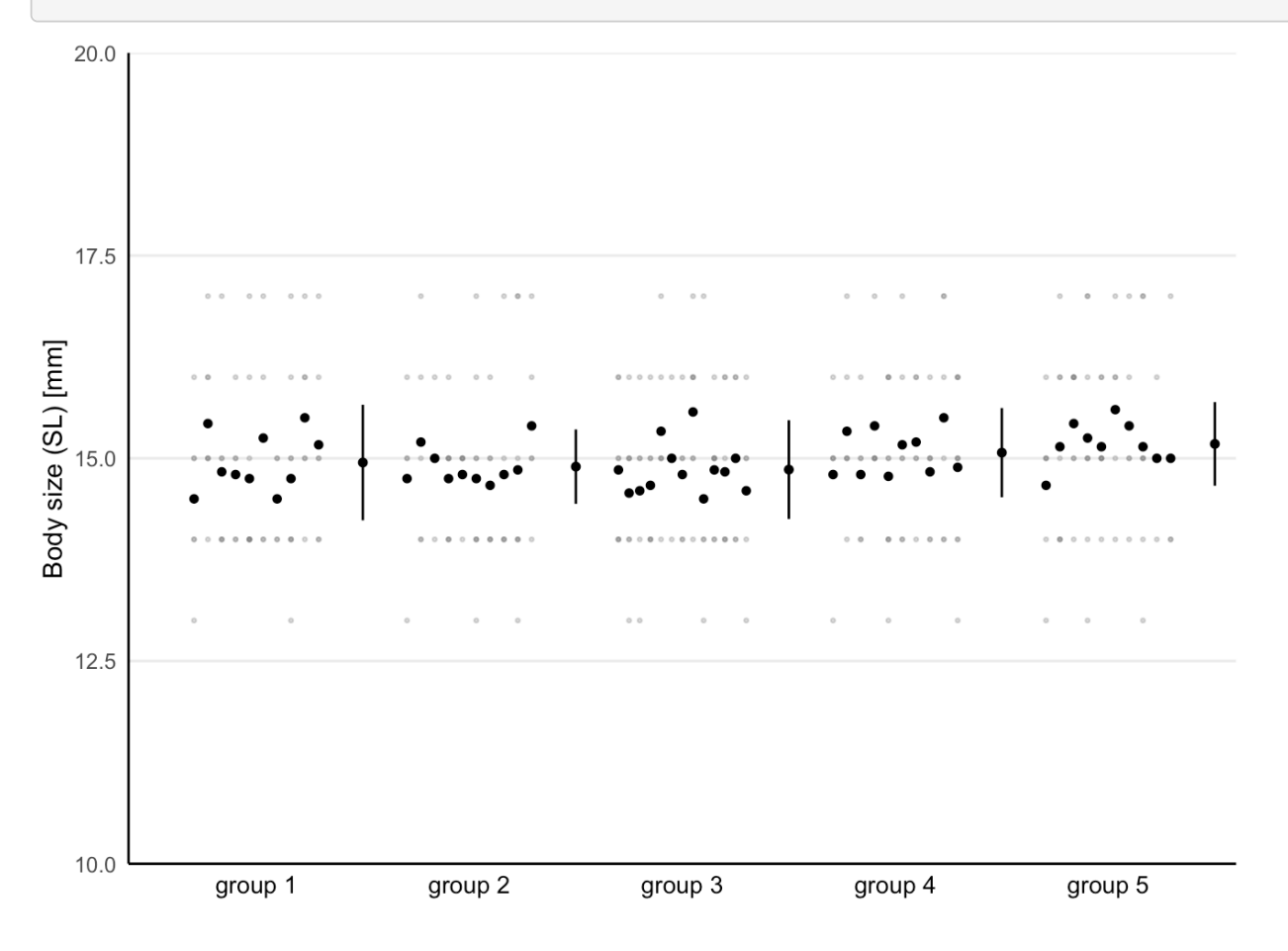
```
## Not enough model terms in the conditional part of the model to check for multicollinearity.
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
#### Plots ####
  library(ggplot2)
  (p_size<-ggplot() +
        geom_point(data= multi_BL, aes(x=fgroup, y=SL_cm, group=ftrial),
                   color="grey50", alpha=0.3, stat = 'identity', size=0.5, position = position_dodge
(0.65)) +
        geom_point(data= multi_BL_groupmean, aes(x=fgroup, y=mean_SL_cm, group=ftrial),
                   color="black", stat = 'identity', size=1.2, position = position_dodge(0.65)) +
        stat_summary(data=multi_BL_groupmean, aes(x=fgroup, y=mean_SL_cm),
                     fun.data = mean_sdl, fun.args = list(mult = 1.96),
                      geom = "pointrange", position = position_nudge(0.5), color="black", size=0.5,
fatten=1.2) +
        scale_color_brewer(palette="Reds") +
        scale_y_continuous(limits = c(10, 20),
                           oob = scales::squish, expand = c(0, 0), ) + # force through 0,0 and squ
ish instead of censoring out-of-bound points
        theme_minimal()+
        theme(plot.title = element_blank(), panel.background = element_blank(),
                panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
                legend.position = "null", axis.line = element_line(colour = "black"),
                axis.text.y = element_text(size = 9), axis.text.x = element_text(size = 10, colour =
"black"),
                axis.title.y =element_text(size=11), axis.title.x =element_blank(),
                strip.text.x = element_text(size=10, colour="black"), strip.background = element_rec
t(fill="lightgray")) +
        labs(y="Body size (SL) [mm]"))
```



Lab - Ox/Temp

```
## Dataset ox/temp
  library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
  multi_bygroups<-ddply(multi_BL,~fgroup,summarise,fgroup=first(fgroup),ox=mean(c(Ox_mgL, Ox_mgL_aft
er)), temp=mean(c(Temp_C, Temp_C_after)))

#### extract variables ####
  round(mean(multi_bygroups$temp), digits = 1)
```

```
## [1] 28.8
```

```
  round(range(c(multi_BL$Temp_C, multi_BL$Temp_C_after)), digits = 1)
```

```
## [1] 26.8 29.9
```

```
  round(mean(multi_bygroups$ox), digits = 1)
```

```
## [1] 0.6
```

```
  round(range(c(multi_BL$Ox_mgL, multi_BL$Ox_mgL_after)), digits = 1)
```

```
## [1] 0.3 1.0
```

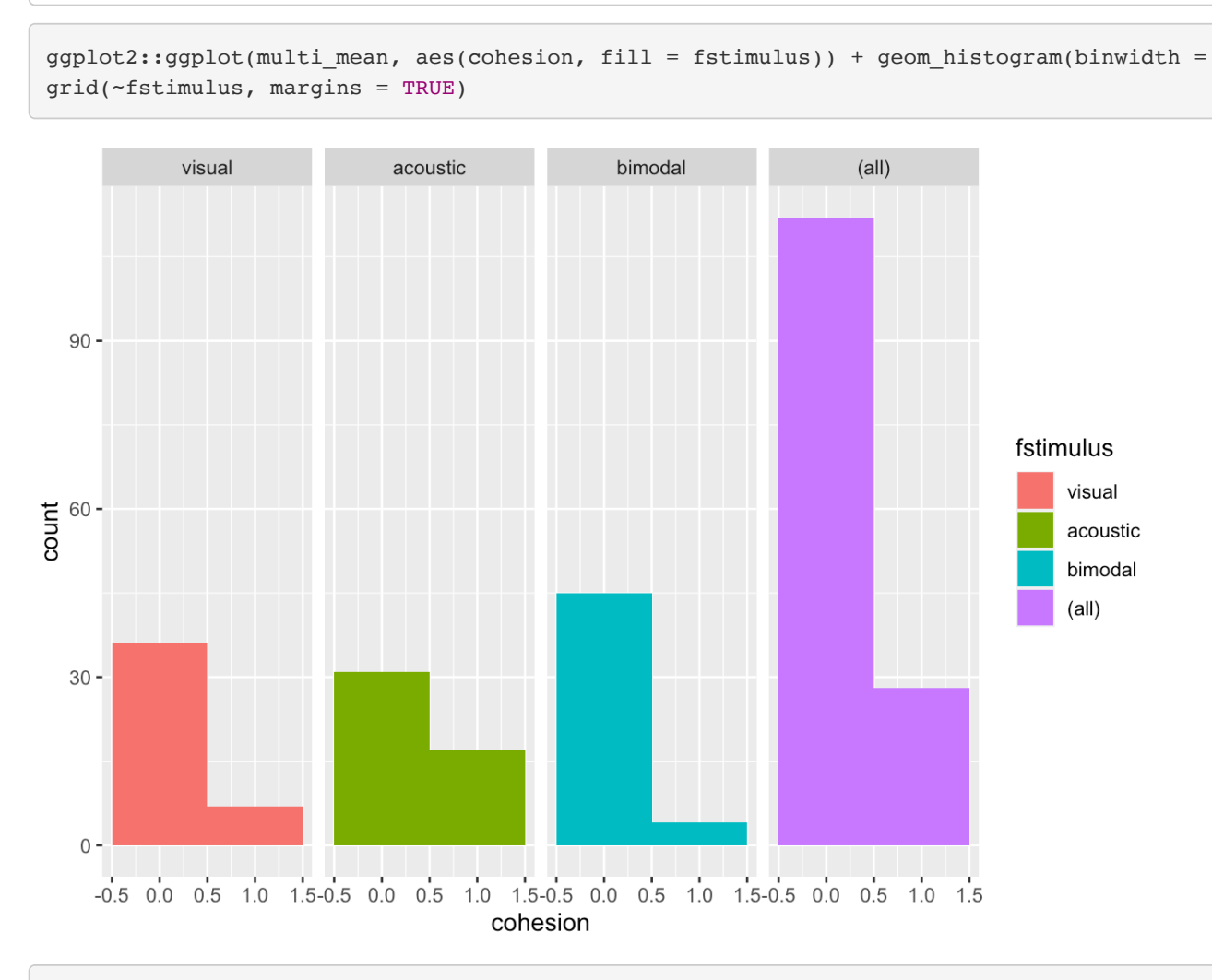
```
# No statistical testing: only 1 measurement per group!
```

Lab - Cohesion

```
mean(multi_mean$cohesion,na.rm=TRUE)
```

```
## [1] 0.4093071
```

```
ggplot2::ggplot(multi_mean, aes(cohesion, fill = fstimulus)) + geom_histogram(binwidth = 1) + facet_
grid(~fstimulus, margins = TRUE)
```



```
####### LMM #######
    library(lme4)
    library(lmerTest)
    library(tidyverse)

  #random effect structure: varying Intercept best represents exp. design
  #fixed effect structure
      coll.cov.f0<-lmer(log(cohesion)~1 + (1|fgroup), REML=FALSE, data=drop_na(multi_mean, cohesion)
)
      coll.cov.stimulus<-lmer(log(cohesion)~fstimulus + (1|fgroup), REML=FALSE, data=drop_na(multi_m
ean, cohesion))
      coll.cov.trial<-lmer(log(cohesion)~trial + (1|fgroup), REML=FALSE, data=drop_na(multi_mean, co
hesion))
      coll.cov.inter<-lmer(log(cohesion)~fstimulus*trial + (1|fgroup), REML=FALSE, data=drop_na(mult
i_mean, cohesion))
      coll.cov.2way <- update(coll.cov.inter, . ~ . -fstimulus:trial)

      drop1(coll.cov.2way) #test significance of fixed effects
```

```
## Single term deletions using Satterthwaite's method:
##
## Model:
## log(cohesion) ~ fstimulus + trial + (1 | fgroup)
##            Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## fstimulus 2.10698 1.05349     2 134.97  7.9454 0.0005468 ***
## trial     0.04618 0.04618     1 135.03  0.3483 0.5560910
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
      bbmle::AICctab(coll.cov.f0, coll.cov.stimulus, coll.cov.trial, coll.cov.inter, coll.cov.2way,
             base=TRUE, weights=TRUE,delta=TRUE,sort=TRUE)
```

```
##                   AICc  dAICc df weight
## coll.cov.stimulus 137.1   0.0 5  0.6432
## coll.cov.2way     139.0   1.8 6  0.2568
## coll.cov.inter    140.9   3.8 8  0.0963
## coll.cov.f0       148.2  11.1 3  0.0025
## coll.cov.trial    149.6  12.5 4  0.0012
```

```
      ## model diagnostics ##
      summary(coll.cov.final <- update(coll.cov.2way, REML=TRUE))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: log(cohesion) ~ fstimulus + trial + (1 | fgroup)
##    Data: drop_na(multi_mean, cohesion)
##
## REML criterion at convergence: 145.3
##
## Scaled residuals:
```

```
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.21413 -0.71827 -0.09853  0.69273  2.31307
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fgroup   (Intercept) 0.05926  0.2434
##  Residual             0.13559  0.3682
## Number of obs: 140, groups:  fgroup, 5
##
## Fixed effects:
##                     Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)        -0.985884   0.134822   7.913896  -7.313 8.76e-05 ***
## fstimulusacoustic   0.189065   0.077491 131.993237   2.440    0.016 *
## fstimulusbimodal   -0.102693   0.077026 132.004222  -1.333    0.185
## trial              -0.002106   0.003650 132.032871  -0.577    0.565
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fstmlsc fstmlsb
## fstimlscstc -0.325
## fstimlsbmdl -0.301  0.530
## trial       -0.417  0.052  -0.009
```

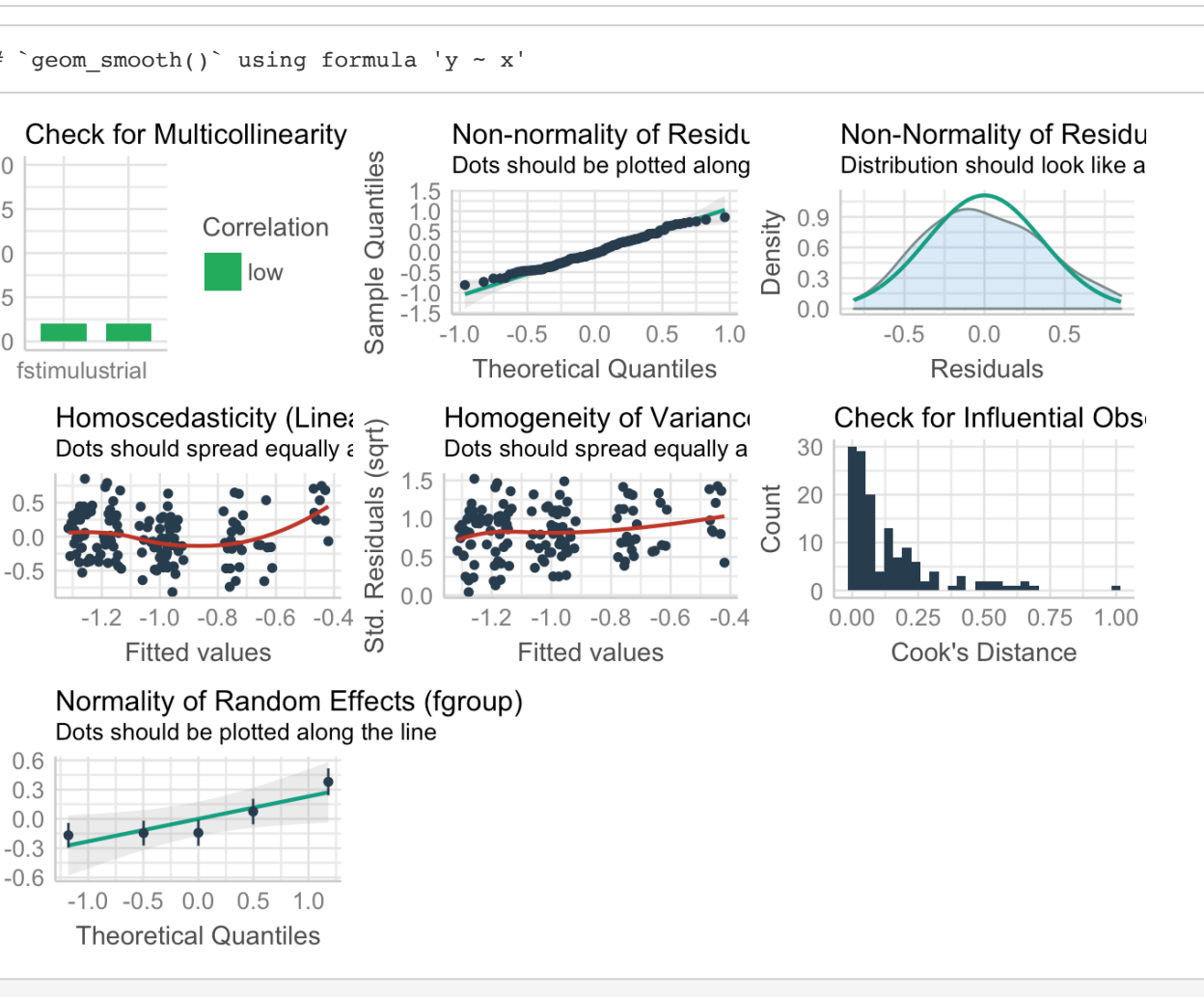
```
    performance::check_model(coll.cov.final)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

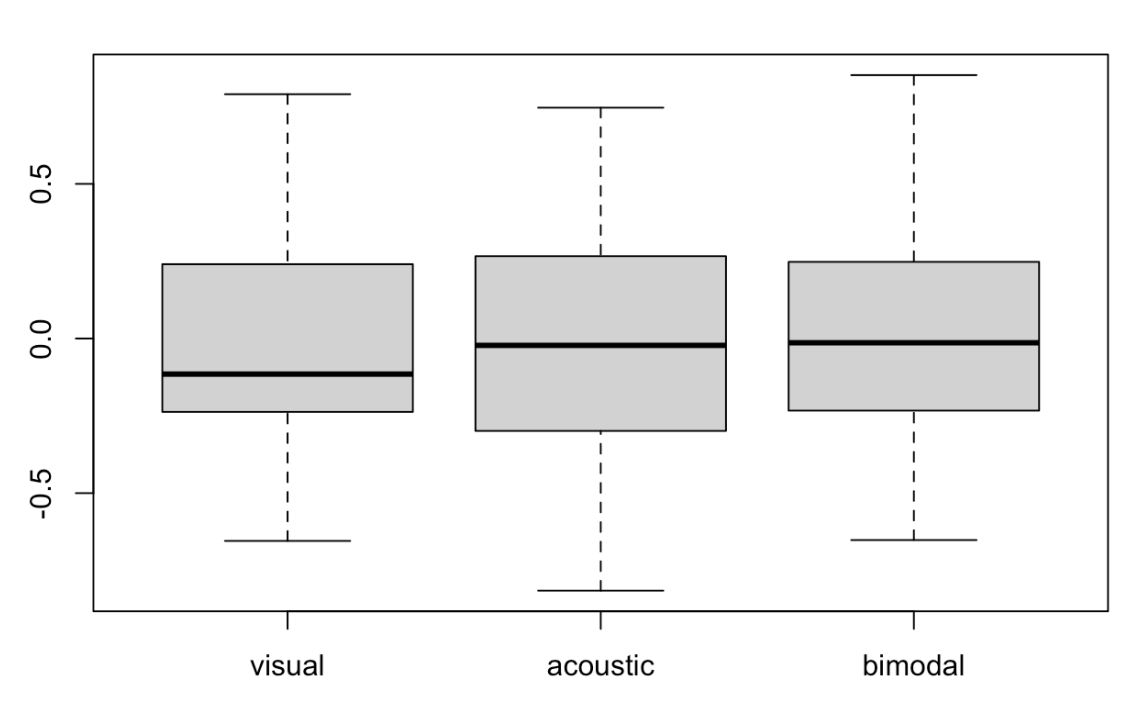
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 140 rows containing missing values (geom_text_repel).
```

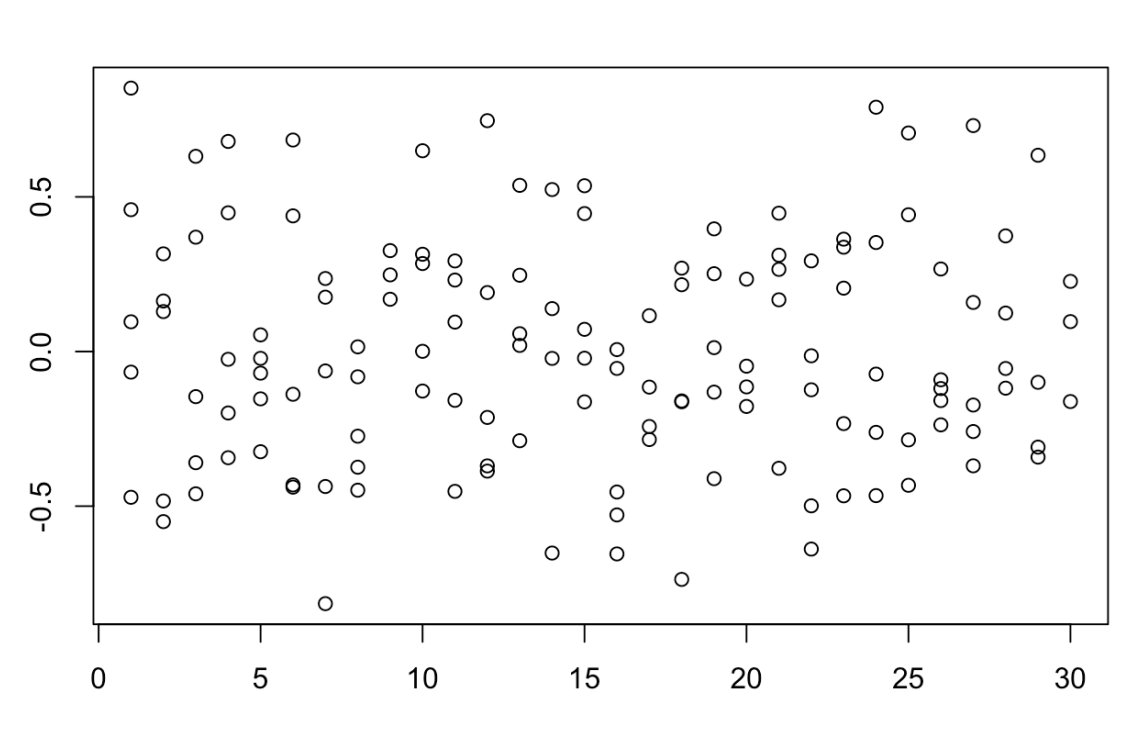
```
## `geom_smooth()` using formula 'y ~ x'
```



```
    plot(resid(coll.cov.final)~drop_na(multi_mean, cohesion)$fstimulus) #plot residuals versus pre
dictors to see where heterogeneity is coming from
```



```
    plot(resid(coll.cov.final)~drop_na(multi_mean, cohesion)$trial)
```



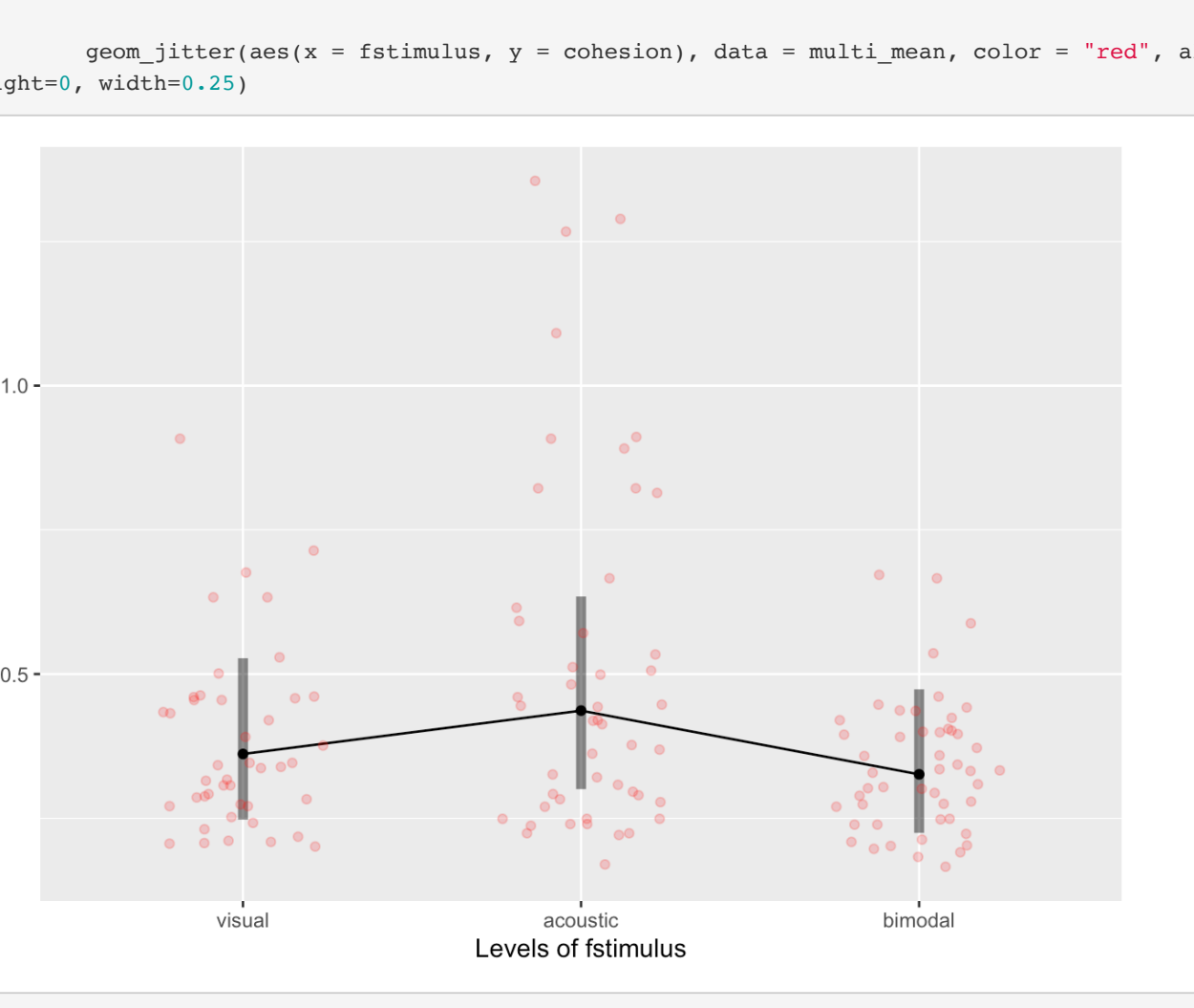
```
  ## model predictions ##
    library(emmeans)
    joint_tests(coll.cov.final)
```

```
##  model term df1    df2 F.ratio p.value
##  fstimulus    2 132.03   7.765 0.0006
##  trial        1 132.07   0.333 0.5651
```

```
    (coll.cov_EMMs <- emmip(coll.cov.final, ~fstimulus, type = "response", adjust = "mvt", CIs = T
RUE, plotit=FALSE))
```

```
##   fstimulus      yvar         SE       df       LCL       UCL tvar     xvar
## 1    visual 0.3614318 0.04430406 5.482108 0.2475643 0.5276727    1   visual
## 2  acoustic 0.4366528 0.05292125 5.242186 0.3003657 0.6347782    1 acoustic
## 3   bimodal 0.3261575 0.03944235 5.196518 0.2245431 0.4737562    1  bimodal
```

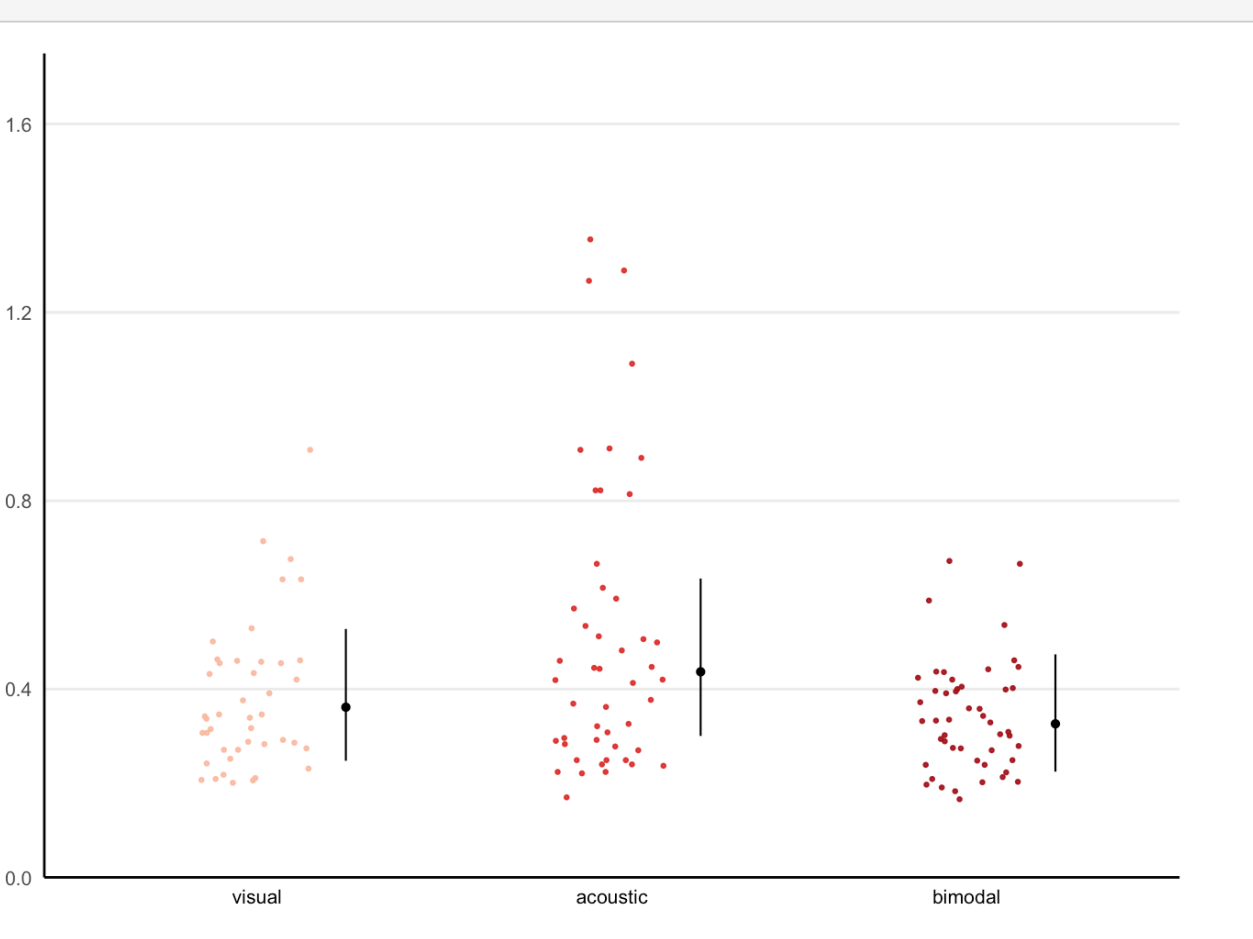
```
    emmip(coll.cov.final, ~fstimulus, type = "response", adjust = "mvt", CIs = TRUE, plotit=TRUE)
+
      geom_jitter(aes(x = fstimulus, y = cohesion), data = multi_mean, color = "red", alpha=0.2, h
eight=0, width=0.25)
```



```
    contrast(emmeans(coll.cov.final, ~fstimulus, type = "response", adjust = "mvt"),
                                                list(`visual-bimodal` = c(-1,0,1), `acoustic-
bimodal` = c(0,-1,1)))
```

```
##  contrast         ratio     SE  df t.ratio p.value
##  visual-bimodal   0.902 0.0695 132 -1.333  0.1848
##  acoustic-bimodal 0.747 0.0560 132 -3.893  0.0002
##
## Degrees-of-freedom method: kenward-roger
## Tests are performed on the log scale
```

```
    ## plot ##
    library(ggplot2)
    library(scales)
    (p_coh<-ggplot() +
      geom_jitter(data=multi_mean, aes(x=fstimulus, y=cohesion, color=fstimulus),
                size=0.5, alpha=0.9, position = position_jitterdodge(jitter.height = 0, jitter.w
idth = 0.8)) +
      geom_pointrange(data=coll.cov_EMMs, aes(x=fstimulus, y=yvar, ymin = LCL, ymax = UCL),
                position = position_nudge(0.25), color="black", size=0.4, fatten=1.2) +
      scale_color_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
      scale_y_continuous(limits = c(0, 1.75), breaks = seq(0, 1.75, by = 0.4),oob = scales::squish
, expand = c(0, 0)) +
      theme_minimal() +
      theme(plot.title = element_blank(), panel.background = element_blank(),
            panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
            legend.position = "null", axis.line = element_line(colour = "black"),
            axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 8, color="blac
k"),
            axis.title.x =element_blank(), axis.title =element_text(size=11)) +
      labs(y='Cohesion'))
```

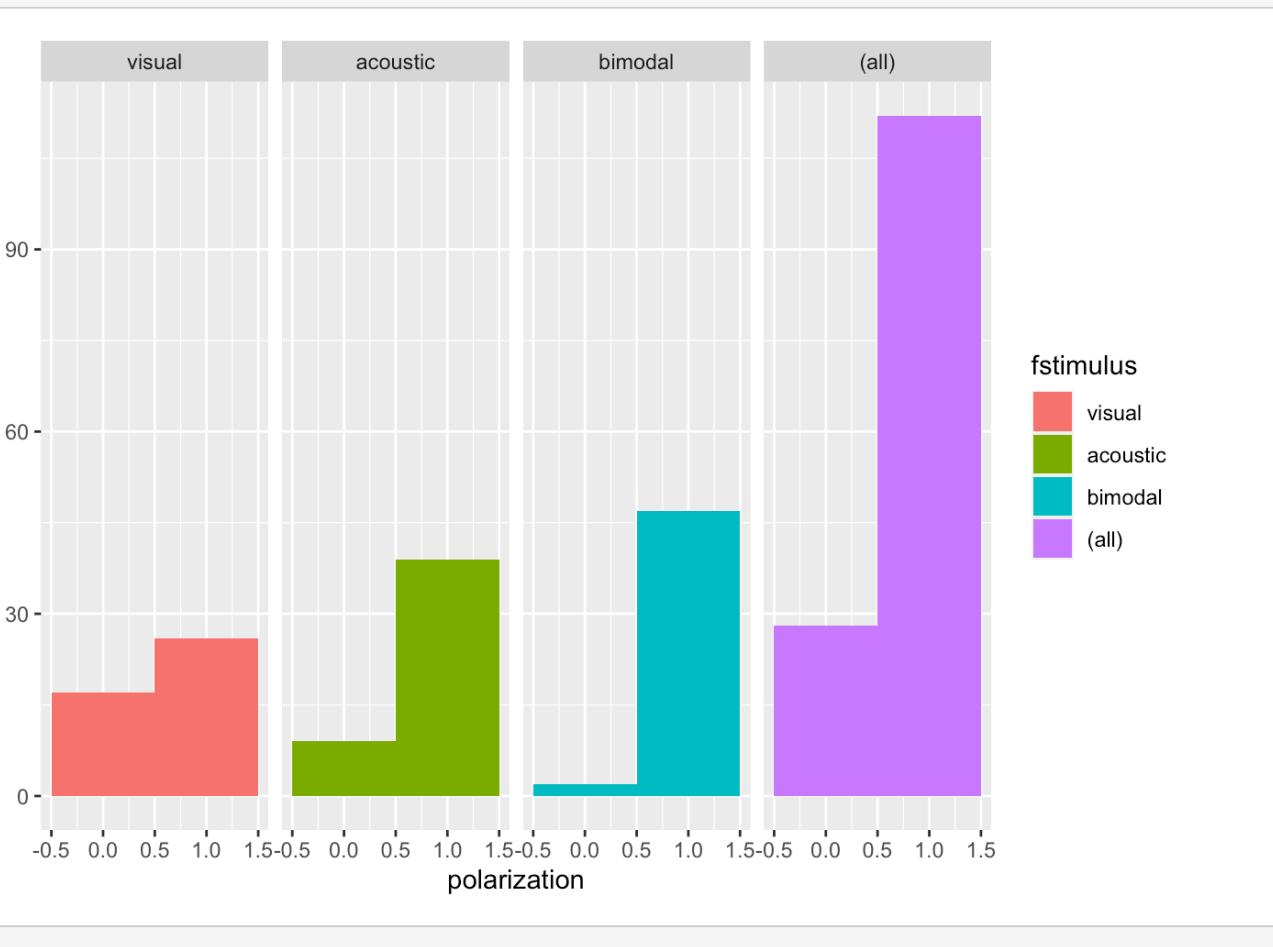


Lab - Polarization

```
mean(multi_mean$polarization,na.rm=TRUE)
```

```
## [1] 0.5926429
```

```
ggplot2::ggplot(multi_mean, aes(polarization, fill = fstimulus)) + geom_histogram(binwidth = 1) + fa
cet_grid(~fstimulus, margins = TRUE)
```



```
####### LMM #######
    library(lme4)
    library(lmerTest)
    library(tidyverse)

  #random effect structure: varying Intercept best represents exp. design
  #fixed effect structure
      coll.pol.f0<-lmer(log(polarization)~1 + (1|fgroup), REML=FALSE, data=drop_na(multi_mean, polar
ization))
      coll.pol.stimulus<-lmer(log(polarization)~fstimulus + (1|fgroup), REML=FALSE, data=drop_na(mul
ti_mean, polarization))
      coll.pol.trial<-lmer(log(polarization)~trial + (1|fgroup), REML=FALSE, data=drop_na(multi_mean
, polarization))
      coll.pol.inter<-lmer(log(polarization)~fstimulus*trial + (1|fgroup), REML=FALSE, data=drop_na(
multi_mean, polarization))
      coll.pol.2way <- update(coll.pol.inter, . ~ . -fstimulus:trial)

      drop1(coll.pol.2way) #test significance of fixed effects
```

```
## Single term deletions using Satterthwaite's method:
##
## Model:
## log(polarization) ~ fstimulus + trial + (1 | fgroup)
##            Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## fstimulus 0.81100 0.40550     2 134.95 11.2658 2.988e-05 ***
## trial     0.08509 0.08509     1 135.07  2.3639    0.1265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
      bbmle::AICctab(coll.pol.f0, coll.pol.stimulus, coll.pol.trial, coll.pol.inter, coll.pol.2way,
              base=TRUE, weights=TRUE,delta=TRUE,sort=TRUE)
```

```
##                   AICc  dAICc df weight
## coll.pol.2way     -47.4   0.0 6  0.492
## coll.pol.stimulus -47.3   0.2 5  0.454
## coll.pol.inter    -43.0   4.4 8  0.054
## coll.pol.f0       -31.1  16.3 3  <0.001
## coll.pol.trial    -31.0  16.5 4  <0.001
```

```
      ## model diagnostics ##
      summary(coll.pol.final <- update(coll.pol.2way, REML=TRUE))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: log(polarization) ~ fstimulus + trial + (1 | fgroup)
##    Data: drop_na(multi_mean, polarization)
##
## REML criterion at convergence: -35.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5497 -0.5086  0.0515  0.5732  2.3413
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  fgroup   (Intercept) 0.006803 0.08248
##  Residual             0.036806 0.19185
## Number of obs: 140, groups:  fgroup, 5
##
## Fixed effects:
##                    Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)       -0.701924   0.055473  13.752452 -12.654 5.81e-09 ***
## fstimulusacoustic  0.130183   0.040370 131.992739   3.225  0.00159 **
## fstimulusbimodal   0.184887   0.040127 132.016584   4.608 9.48e-06 ***
## trial              0.002871   0.001902 132.076132   1.510  0.13346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fstmlsc fstmlsb
## fstimlscstc -0.411
## fstimlsbmdl -0.381  0.529
## trial       -0.528  0.053  -0.009
```

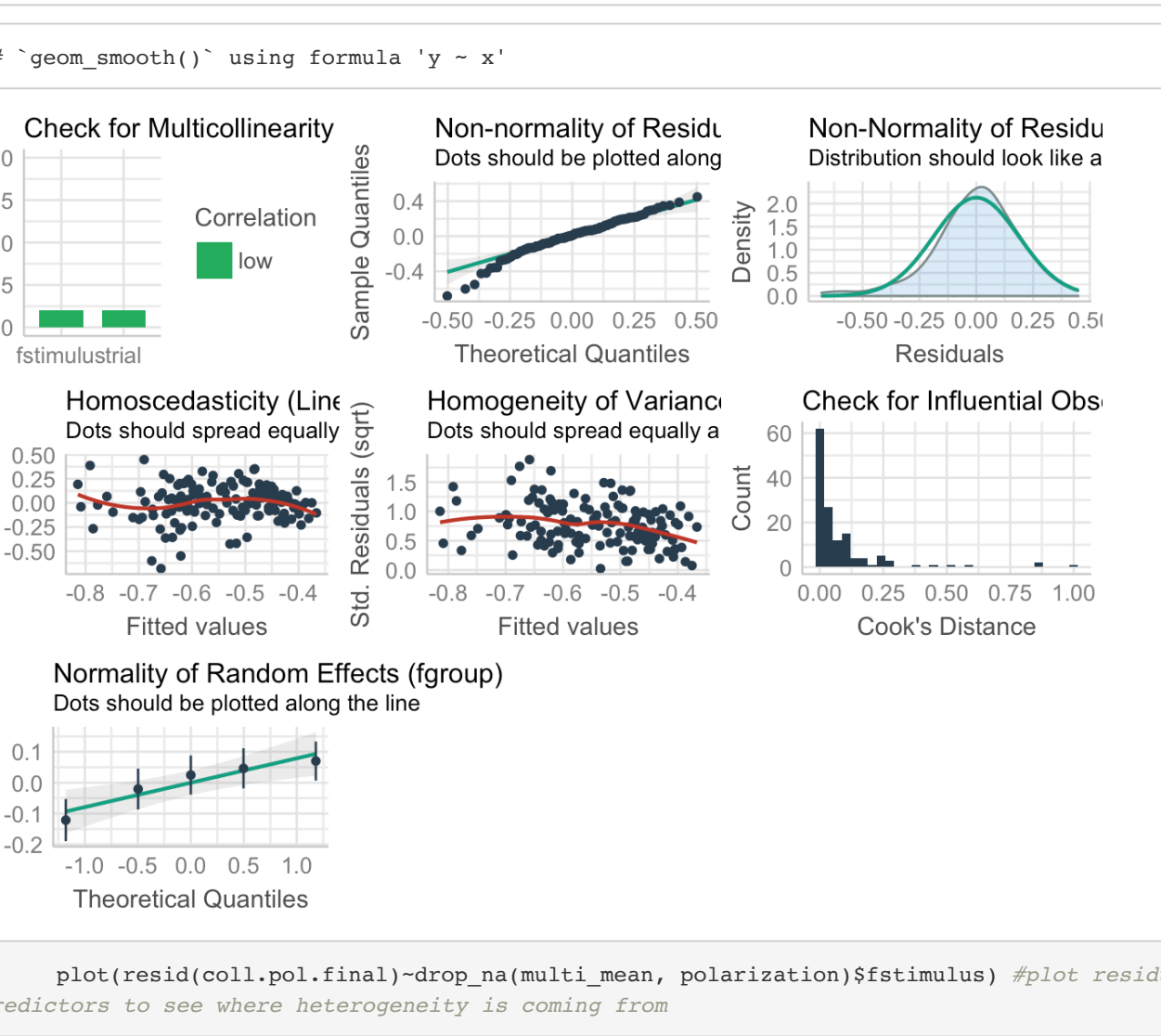
```
      performance::check_model(coll.pol.final)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

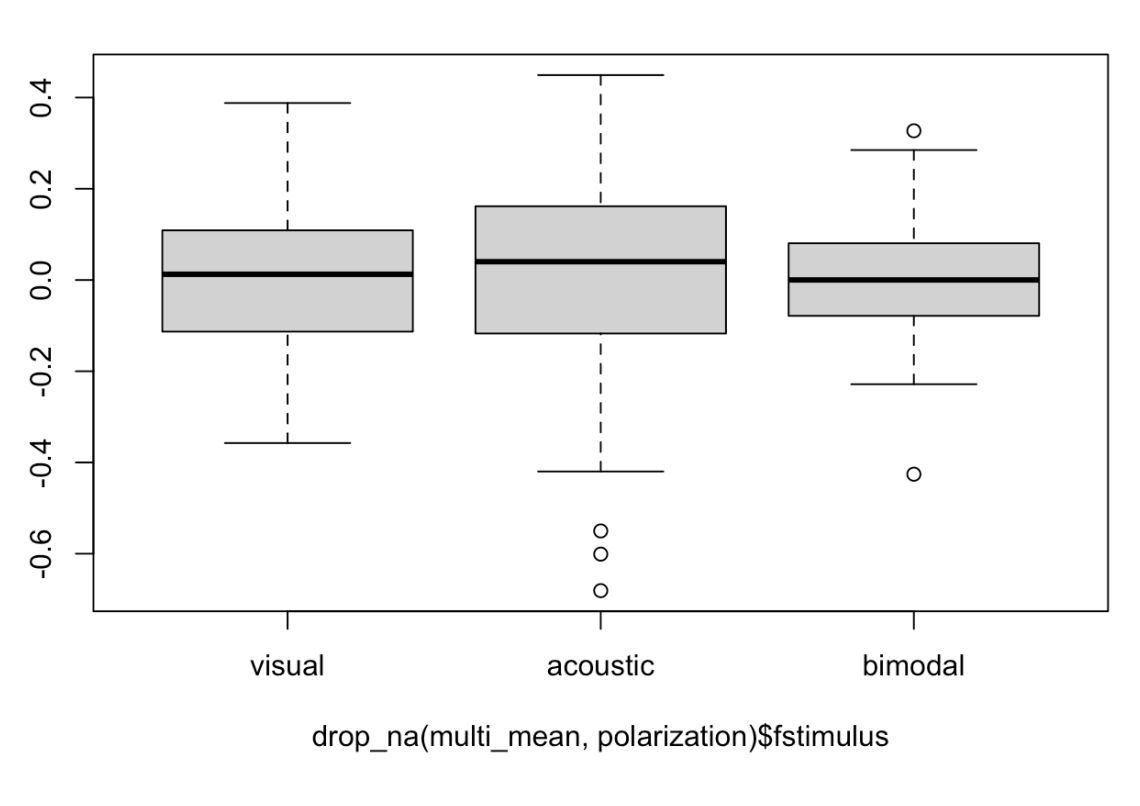
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 140 rows containing missing values (geom_text_repel).
```

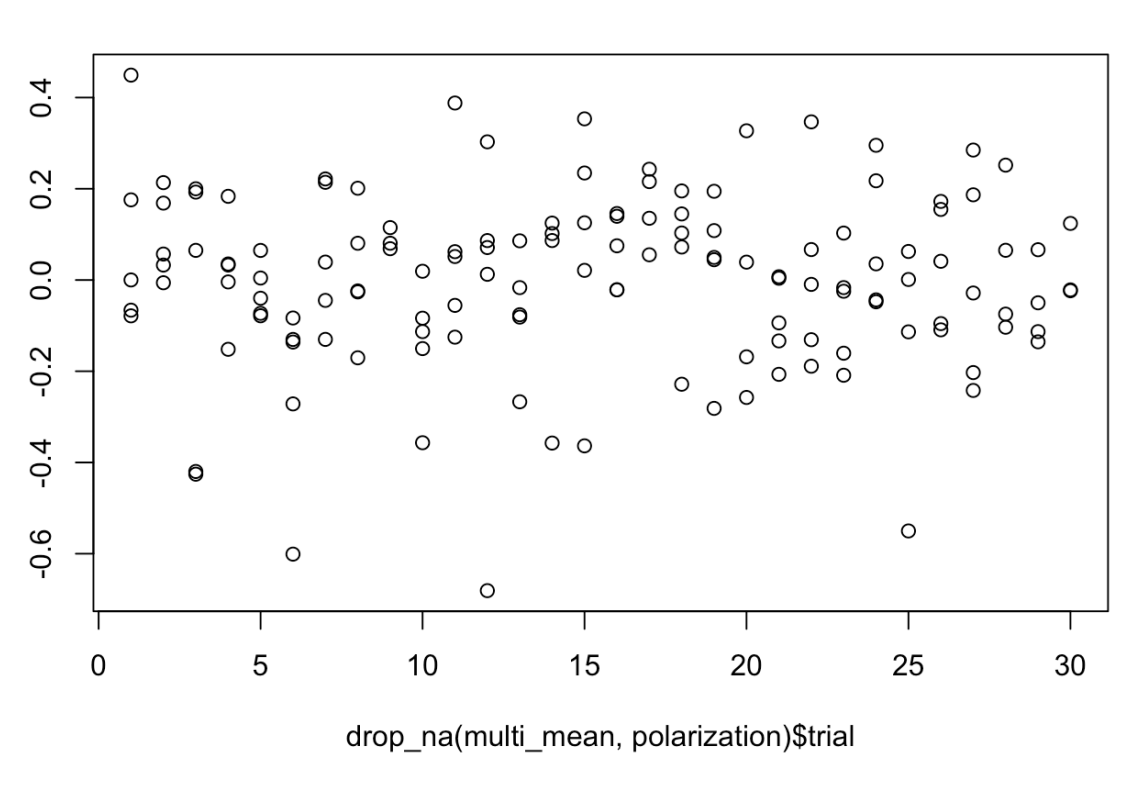
```
## `geom_smooth()` using formula 'y ~ x'
```



```
      plot(resid(coll.pol.final)~drop_na(multi_mean, polarization)$fstimulus) #plot residuals versus
predictors to see where heterogeneity is coming from
```



```
      plot(resid(coll.pol.final)~drop_na(multi_mean, polarization)$trial)
```



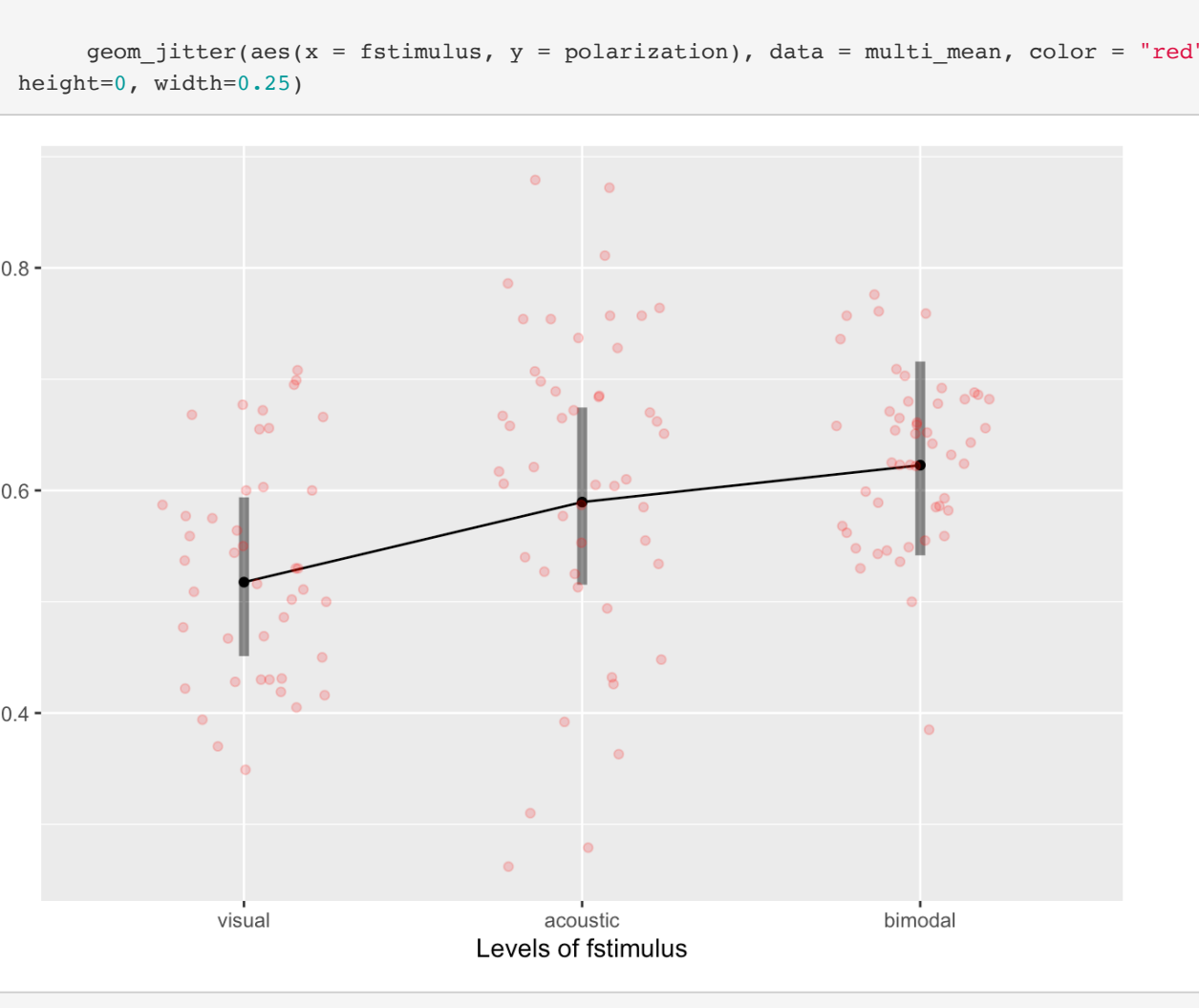
```
      ## model predictions ##
      library(emmeans)
      joint_tests(coll.pol.final)
```

```
##  model term df1    df2 F.ratio p.value
##  fstimulus    2 132.07  11.039 <.0001
##  trial        1 132.15   2.278 0.1336
```

```
      (coll.pol_EMMs <- emmip(coll.pol.final, ~fstimulus, type = "response", adjust = "mvt", CIs = T
RUE, plotit=FALSE))
```

```
##   fstimulus      yvar         SE       df       LCL       UCL tvar     xvar
## 1    visual 0.5175901 0.02439801 7.436439 0.4511569 0.5938057    1   visual
## 2  acoustic 0.5895542 0.02721238 6.854768 0.5153634 0.6744254    1 acoustic
## 3   bimodal 0.6227034 0.02862377 6.744572 0.5415555 0.7160108    1  bimodal
```

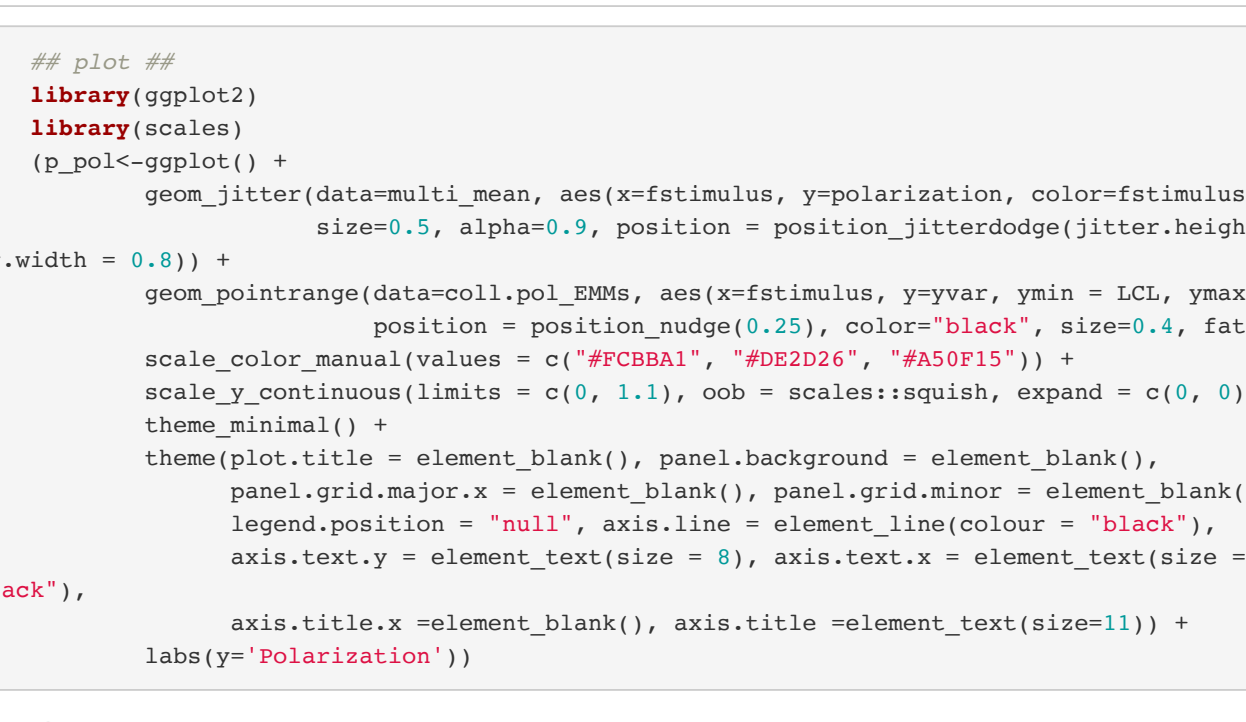
```
      emmip(coll.pol.final, ~fstimulus, type = "response", adjust = "mvt", CIs = TRUE, plotit=TRUE)
+
        geom_jitter(aes(x = fstimulus, y = polarization), data = multi_mean, color = "red", alpha=0.
2, height=0, width=0.25)
```



```
      contrast(emmeans(coll.pol.final, ~fstimulus, type = "response", adjust = "mvt"),
                                                list(`visual-bimodal` = c(-1,0,1), `acoustic-
bimodal` = c(0,-1,1)))
```

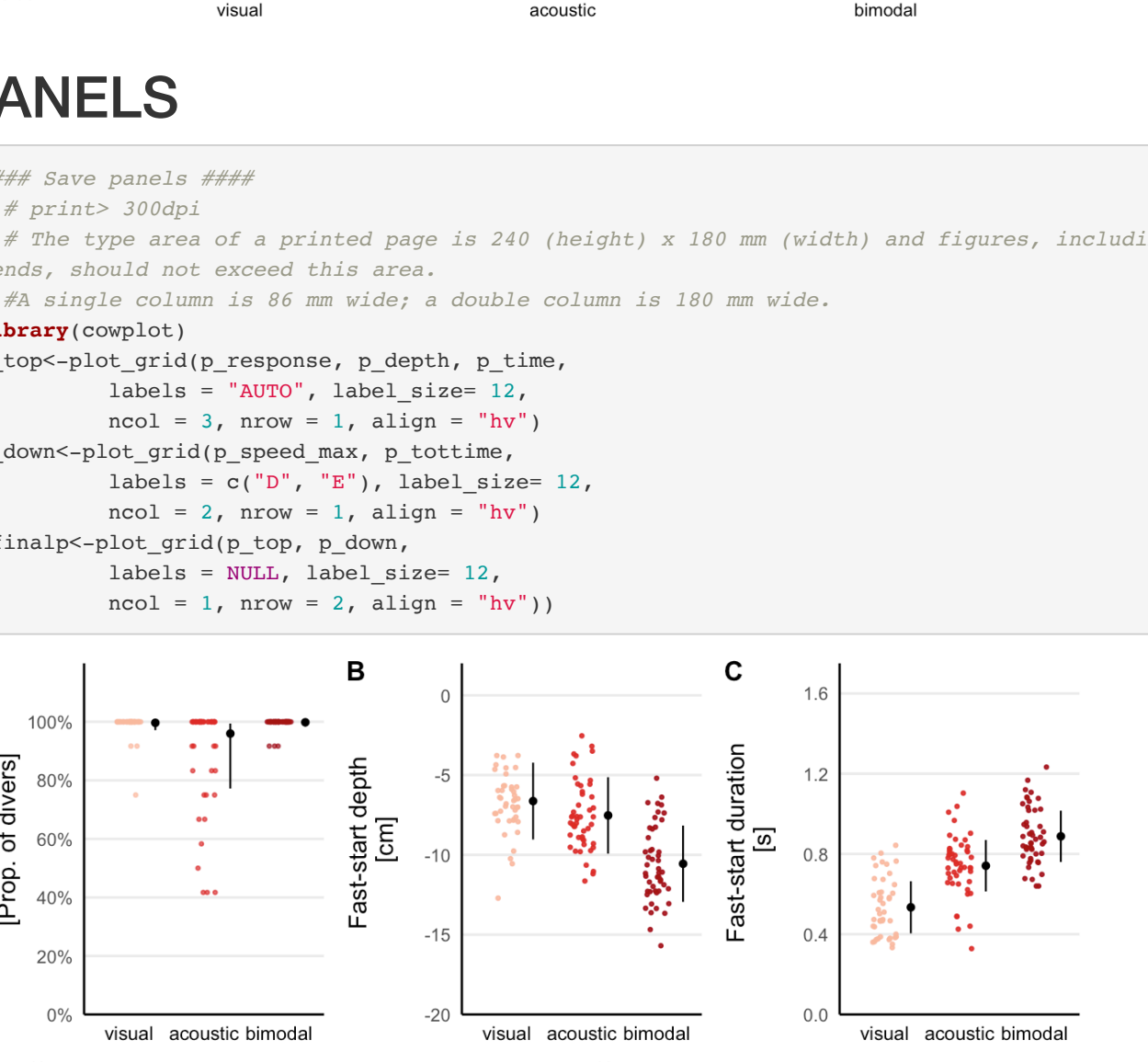
```
##  contrast         ratio     SE  df t.ratio p.value
##  visual-bimodal    1.20 0.0483 132 4.607   <.0001
##  acoustic-bimodal  1.06 0.0412 132 1.401   0.1636
##
## Degrees-of-freedom method: kenward-roger
## Tests are performed on the log scale
```

```
      ## plot ##
      library(ggplot2)
      library(scales)
      (p_pol<-ggplot() +
            geom_jitter(data=multi_mean, aes(x=fstimulus, y=polarization, color=fstimulus),
                        size=0.5, alpha=0.9, position = position_jitterdodge(jitter.height = 0, jitt
er.width = 0.8)) +
            geom_pointrange(data=coll.pol_EMMs, aes(x=fstimulus, y=yvar, ymin = LCL, ymax = UCL),
                        position = position_nudge(0.25), color="black", size=0.4, fatten=1.2) +
            scale_color_manual(values = c("#FCBBA1", "#DE2D26", "#A50F15")) +
            scale_y_continuous(limits = c(0, 1.1), oob = scales::squish, expand = c(0, 0)) +
            theme_minimal() +
            theme(plot.title = element_blank(), panel.background = element_blank(),
                  panel.grid.major.x = element_blank(), panel.grid.minor = element_blank(),
                  legend.position = "null", axis.line = element_line(colour = "black"),
                  axis.text.y = element_text(size = 8), axis.text.x = element_text(size = 8, color="
black"),
                  axis.title.x =element_blank(), axis.title =element_text(size=11)) +
            labs(y='Polarization'))
```



PANELS

```
#### Save panels ####
  # print> 300dpi
  # The type area of a printed page is 240 (height) x 180 mm (width) and figures, including their le
gends, should not exceed this area.
  #A single column is 86 mm wide; a double column is 180 mm wide.
library(cowplot)
p_top<-plot_grid(p_response, p_depth, p_time,
        labels = "AUTO", label_size= 12,
        ncol = 3, nrow = 1, align = "hv")
p_down<-plot_grid(p_speed_max, p_tottime,
        labels = c("D", "E"), label_size= 12,
        ncol = 2, nrow = 1, align = "hv")
(finalp<-plot_grid(p_top, p_down,
        labels = NULL, label_size= 12,
        ncol = 1, nrow = 2, align = "hv"))
```



```
#
# ggsave("figure4_CuesInd.png", plot = finalp, type = "cairo", path = "../../figs/",
#        scale = 1, width = 180, height = 133, units = "mm",
#        dpi = "print")
#
# p_birds
# ggsave("figure3_BirdCues.png", plot = p_birds, type = "cairo", path = "../../figs/",
#        scale = 1, width = 86, height = 60, units = "mm",
#        dpi = "print")
#
# (p_field<-plot_grid(p_natDiving, labels = c("B"), label_size = 12, ncol=1, nrow=1, align="hv"))
# ggsave("figure1_NatDiving.png", plot = p_field, type = "cairo", path = "../../figs/",
#        scale = 1, width = 86, height = 60, units = "mm",
#        dpi = "print")
#
# (finalSI<-plot_grid(p_coh, p_pol, labels = "AUTO", label_size = 12, ncol=2, nrow=1, align="hv"))
# ggsave("figureSI_CuesColl.png", plot = finalSI, type = "cairo", path = "../../figs/",
#        scale = 1, width = 180, height = 60, units = "mm",
#        dpi = "print")
```

TABLES

```
library(sjPlot)
```

```
## Learn more about sjPlot with 'browseVignettes("sjPlot")'.
```

```
##
## Attaching package: 'sjPlot'
```

```
## The following objects are masked from 'package:cowplot':
##
##     plot_grid, save_plot
```

```
  tab_model(diving_gm_final, mean_depth.final, mean_time.final, mean_speed_max.final, mean_tottime.f
inal,
        pred.labels = c("Intercept", "stimulus: acoustic", "stimulus: bimodal", "exposure (trial)"
),
        dv.labels = c("Responsiveness [Prop. of divers]", "Fast-start depth [cm]", "Fast-start dur
ation [s]",
                     "Fast-start maximum speed [cm/s]", "Total dive duration [log(s)]"),
        file="../../figs/tab_GroupMeans_diving.html")
```

	Responsiveness [Prop. of divers]			Fast-start depth [cm]			Fast-start duration [s]			Fast-start maximum speed [cm/s]			Total dive duration [log(s)]		
Predictors	*Odds Ratios*	*CI*	*p*	*Estimates*	*CI*	*p*	*Estimates*	*CI*	*p*	*Estimates*	*CI*	*p*	*Estimates*	*CI*	*p*
Intercept	292.38	41.08 – 2081.16	**<0.001**	-6.45	-8.00 – -4.91	**<0.001**	0.55	0.46 – 0.64	**<0.001**	19.31	15.50 – 23.12	**<0.001**	1.58	1.32 – 1.85	**<0.001**
stimulus: acoustic	0.08	0.03 – 0.20	**<0.001**	-0.90	-1.58 – -0.21	**0.010**	0.21	0.15 – 0.26	**<0.001**	-5.52	-7.09 – -3.95	**<0.001**	-0.20	-0.33 – -0.07	**0.003**
stimulus: bimodal	2.14	0.51 – 8.93	0.299	-3.93	-4.61 – -3.24	**<0.001**	0.35	0.30 – 0.41	**<0.001**	2.02	0.45 – 3.58	**0.011**	0.13	0.00 – 0.26	**0.043**
exposure (trial)	1.00	0.97 – 1.04	0.874	-0.01	-0.04 – 0.02	0.479	-0.00	-0.00 – 0.00	0.419	0.08	0.01 – 0.16	**0.026**	-0.01	-0.02 – -0.00	**0.001**
Random Effects															
σ^2	3.29			2.77			0.02			14.53			0.10		
τ_{00}	3.30 $_{fgroup}$			2.46 $_{fgroup}$			0.01 $_{fgroup}$			15.52 $_{fgroup}$			0.07 $_{fgroup}$		
ICC	0.50			0.47			0.30			0.52			0.41		
N	5 $_{fgroup}$			5 $_{fgroup}$			5 $_{fgroup}$			5 $_{fgroup}$			5 $_{fgroup}$		
Observations	140			140			140			140			140		
Marginal R^2 / Conditional R^2	0.239 / 0.620			0.356 / 0.659			0.475 / 0.634			0.275 / 0.649			0.131 / 0.491		

```
  tab_model(birds_gm,
        pred.labels = c("Intercept", "type: attack", "sampling: midday", "sampling: afternoon",
                "type [attack] * sampling [midday]", "type [attack] * sampling [afternoon]
"),
        dv.labels = "Disturbance cues",
        file="../../figs/tab_PredatorCues.html")
```

	Disturbance cues		
Predictors	*Incidence Rate Ratios*	*CI*	*p*
Intercept	11.00	6.89 – 17.57	**<0.001**
type: attack	0.91	0.47 – 1.77	0.780
sampling: midday	1.73	0.92 – 3.26	0.091
sampling: afternoon	0.82	0.42 – 1.61	0.561
type [attack] * sampling [midday]	3.49	1.44 – 8.42	**0.005**
type [attack] * sampling [afternoon]	6.20	2.49 – 15.48	**<0.001**
Observations	24		
R^2 Nagelkerke	0.976		

```
  tab_model(DiveTime.final,
        pred.labels = c("Intercept", "exposure [trial]"),
        dv.labels = "Total dive duration [s]",
        file="../../figs/tab_NatDiving.html")
```

	Total dive duration [s]		
Predictors	*Estimates*	*CI*	*p*
Intercept	4.32	3.81 – 4.82	**<0.001**
exposure [trial]	-0.05	-0.11 – 0.00	0.064
Random Effects			
σ^2	0.52		
$\tau_{00\ fsite}$	0.17		
ICC	0.25		
N_{fsite}	6		
Observations	63		
Marginal R^2 / Conditional R^2	0.041 / 0.279		

```
  tab_model(coll.cov.final, coll.pol.final,
        pred.labels = c("Intercept", "stimulus: acoustic", "stimulus: bimodal", "exposure (trial)"
),
        dv.labels = c("Cohesion (log)", "Polarization"),
        file="../../figs/tab_GroupMeansColl_diving.html")
```

	Cohesion (log)			Polarization		
Predictors	*Estimates*	*CI*	*p*	*Estimates*	*CI*	*p*
Intercept	-0.99	-1.25 – -0.72	**<0.001**	-0.70	-0.81 – -0.59	**<0.001**
stimulus: acoustic	0.19	0.04 – 0.34	**0.015**	0.13	0.05 – 0.21	**0.001**
stimulus: bimodal	-0.10	-0.25 – 0.05	0.182	0.18	0.11 – 0.26	**<0.001**
exposure (trial)	-0.00	-0.01 – 0.01	0.564	0.00	-0.00 – 0.01	0.131
Random Effects						
σ^2	0.14			0.04		
τ_{00}	0.06 $_{fgroup}$			0.01 $_{fgroup}$		
ICC	0.30			0.16		
N	5 $_{fgroup}$			5 $_{fgroup}$		
Observations	140			140		
Marginal R^2 / Conditional R^2	0.075 / 0.356			0.129 / 0.265		

SESSION INFO

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS  10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] sjPlot_2.8.4     cowplot_1.1.0    plyr_1.8.6       scales_1.1.1
##  [5] lmerTest_3.1-2  lme4_1.1-23     Matrix_1.2-18   emmeans_1.5.0  
##  [9] DHARMa_0.3.3.0  MASS_7.3-53     forcats_0.5.0   stringr_1.4.0  
## [13] dplyr_1.0.2     purrr_0.3.4     tidyr_1.1.2     tibble_3.0.3   
## [17] ggplot2_3.3.2   tidyverse_1.3.0 readr_1.3.1    
## 
## loaded via a namespace (and not attached):
##   [1] readxl_1.3.1        backports_1.1.10    Hmisc_4.4-1        
##   [4] splines_4.0.2       TH.data_1.0-10      digest_0.6.25      
##   [7] qqplotr_0.0.4       foreach_1.5.0       htmltools_0.5.0    
##  [10] fansi_0.4.1         magrittr_1.5        checkmate_2.0.0    
##  [13] cluster_2.1.0       doParallel_1.0.15   see_0.6.2          
##  [16] modelr_0.1.8        sandwich_2.5-1      bdsmatrix_1.3-4    
##  [19] jpeg_0.1-8.1        colorspace_1.4-1    blob_1.2.1         
##  [22] rvest_0.3.6         ggrepel_0.8.2       haven_2.3.1        
##  [25] xfun_0.17           crayon_1.3.4        jsonlite_1.7.1     
##  [28] survival_3.2-3      zoo_1.8-8           iterators_1.0.12   
##  [31] glue_1.4.2          gtable_0.3.0        sjstats_0.18.0     
##  [34] sjmisc_2.8.5        DEoptimR_1.0-8      mvtnorm_1.1-1      
##  [37] DBI_1.1.0           ggeffects_1.0.1     Rcpp_1.0.5         
##  [40] xtable_1.8-4        performance_0.5.0   htmlTable_2.0.1    
##  [43] foreign_0.8-80      Formula_1.2-3       stats4_4.0.2       
##  [46] htmlwidgets_1.5.1   httr_1.4.2          RColorBrewer_1.1-2 
##  [49] ellipsis_0.3.1      pkgconfig_2.0.3     farver_2.0.3       
##  [52] nnet_7.3-14         qgam_1.3.2          dbplyr_1.4.4       
##  [55] utf8_1.1.4          tidyselect_1.1.0    labeling_0.3       
##  [58] rlang_0.4.7         later_1.1.0.1       effectsize_0.3.2   
##  [61] munsell_0.5.0       cellranger_1.1.0    tools_4.0.2        
##  [64] cli_2.0.2           generics_0.0.2      sjlabelled_1.1.6   
##  [67] broom_0.7.0         ggridges_0.5.2      evaluate_0.14      
##  [70] fastmap_1.0.1       yaml_2.2.1          knitr_1.29         
##  [73] fs_1.5.0            robustbase_0.93-7   nlme_3.1-149       
##  [76] mime_0.9            xml2_1.3.2          doBy_4.6.7         
##  [79] gap_1.2.2           compiler_4.0.2      pbkrtest_0.4-8.6   
##  [82] rstudioapi_0.11     png_0.1-7           reprex_0.3.0       
##  [85] statmod_1.4.34      stringi_1.5.3       parameters_0.8.5   
##  [88] lattice_0.20-41     nloptr_1.2.2.2      vctrs_0.3.4        
##  [91] pillar_1.4.6        lifecycle_0.2.0     estimability_1.3   
##  [94] data.table_1.13.0   insight_0.13.0      httpuv_1.5.4       
##  [97] R6_2.4.1            latticeExtra_0.6-29 promises_1.1.1     
## [100] gridExtra_2.3       codetools_0.2-16    boot_1.3-25        
## [103] assertthat_0.2.1    withr_2.2.0         Deriv_4.0.1        
## [106] multcomp_1.4-13     mgcv_1.8-33         bayestestR_0.7.2   
## [109] parallel_4.0.2      hms_0.5.3           grid_4.0.2         
## [112] rpart_4.1-15        coda_0.19-3         minqa_1.2.4        
## [115] rmarkdown_2.3       bbmle_1.0.23.1      numDeriv_2016.8-1.1
## [118] shiny_1.5.0         lubridate_1.7.9     base64enc_0.1-3
```