

R-code for “Visual obstruction, but not moderate traffic noise, increases reliance on heterospecific alarm calls”

Chaminda P. Ratnayake, You Zhou, Francesca S.E. Dawson Pell, Dominique A.Potvin,
Andrew N. Radford and Robert D. Magrath

This document provides all the necessary code to reproduce the analyses presented in the paper—the data files archived in Dryad repository doi:10.5061/dryad.05qfttf2t. The structure and order of the code followed *Statistical analyses* section of the paper. Please download the files for working directory to run the code. If you encounter any problem, please contact Chaminda P. Ratnayake at `chaminda.ratnayake[at]anu.edu.au` or Robert D. Magrath at `Robert.Magrath[at]anu.edu.au`.

Required packages:

- Following packages are required to run the code in R version 4.0.3 [8]; if the packages not installed on your computer, please remove the # marks to install the packages.

```
# install.packages("tidyverse")
# install.packages("ordinal")
# install.packages("ggplot2")
# install.packages("emmeans")
# install.packages("glmmTMB")
# install.packages("lme4")
# install.packages("DHARMa")
# install.packages("bbmle")

library(tidyverse)
library(ordinal)
library(ggplot2)
library(emmeans)
library(glmmTMB)
library(lme4)
library(DHARMa)
library(bbmle)
```

Loading data

- **experiment_1.csv** this data file contains following response variables: *response_category* (1 = no response, 2 = scan in place, 3 = move while scanning, 4 = flee), *response_duration* (number of video frames); and explanatory variables: *individual_ID*, *barrier_position* (up,down), *treatment_order* (1,2). In addition, other experimentally controlled variables: *sex* (male,female), *number_of_individual_behind_the_barrier* and useful metadata : *date* (DMY) and *time*.
- **experiment_2.csv** this data file contains following response variables: *response_category* (1 = no response, 2 = scan in place, 3 = move while scanning, 4 = flee), *response_duration* (number of video frames); and explanatory variables: *individual_ID*, *barrier_position* (up,down), *traffic_noise* (present,absent), *treatment_order* (1,2,3,4). In addition, other experimentally controlled variables: *sex* (male,female), *number_of_individual_behind_the_barrier*, and useful metadata: *date* (DMY) and *time*.

```
# please change the working directory to where the downloaded DRYAD repository files  
  
setwd("/Users/chaminda/Documents/NOISE_BARRIER_PAPER")  
  
experiment_1<-read.csv("experiment_1.csv", head=T,stringsAsFactors = TRUE)  
experiment_2<-read.csv("experiment_2.csv", head=T,stringsAsFactors = TRUE)
```

Experiment 1: effect of visual restriction alone

Converting *individual_ID* and *treatment_order* as factors, and then ensure *response_category* as an ordered factor [10].

```
experiment_1$individual_ID<-as.factor(experiment_1$individual_ID)  
experiment_1$treatment_order<-as.factor(experiment_1$treatment_order)  
experiment_1$barrier_position<-as.factor(experiment_1$barrier_position)  
experiment_1$ordered_response<-factor(experiment_1$response_category,ordered = TRUE)
```

CLMM analyses for immediate responses

Fitting models using ordinal package's *clmm2* and *clmm* functions with a random factor: *individual_ID*. Please note that longer variable names may throw a warning message; however, the bug already fixed on 23 Aug 2020 in GitHub version <https://github.com/runehaubo> [4].

```
# using clmm2 function  
model0 <- ordinal::clmm2(ordered_response~ barrier_position + treatment_order,  
                           random= individual_ID,data = experiment_1, link= "probit",  
                           threshold = "equidistant", Hess = TRUE, nAGQ = 21)  
summary(model0)  
  
# using clmm function  
model1 <- ordinal::clmm(ordered_response~ barrier_position + treatment_order +  
                           (1|individual_ID),data = experiment_1, link= "probit",  
                           threshold = "equidistant", Hess = TRUE, nAGQ = 21)  
summary(model1)
```

Individual terms of the fitted model can assess using *drop1* function. The *drop1* function only works well with *clm* and *clmm* functions. For convenience, we used *clmm* function throughout the paper.

```
drop1(model1,test="Chisq",)
```

Preparing Figure 3a [9].

```
Figure_3a<-ggplot2::ggplot(experiment_1,aes(x = barrier_position,
                                              fill = ordered_response)) +
  geom_bar(position = "fill",width=0.5, colour="black") +
  labs(x ="", y = "Proportion of responses",face = "bold",size = 14) +
  scale_fill_manual(values =c("#FFFFFF","#E1E0DF","#999999","#000000"),
                     name="Immediate\nresponse",
                     labels = c("No response","Scan in place","Move while scanning","Flee")) +
  scale_x_discrete(labels=c("down" = "\nBarrier down", "up" = "\nBarrier up")) +
  theme_bw()+
  theme(panel.spacing.x=unit(c(0), "lines"))+
  theme(plot.title = element_blank(),axis.title.x = element_blank()) +
  theme(legend.text = element_text(colour="black", size = 10,
                                    margin = margin(r = 2, unit = 'cm')))+
  theme(legend.key = element_rect(size = 4),
        legend.key.height = unit(0.5, "cm"),
        legend.key.width = unit(0.5, "cm"),
        legend.key.size = unit(0.75, 'lines'))+
  theme(axis.title.y = element_text(size=12,colour ="#000000")) +
  theme(panel.grid.major = element_line(colour = "transparent")) +
  theme(panel.grid.minor = element_line(colour = "transparent")) +
  theme(axis.text.y = element_text(size=11,colour ="#000000")) +
  theme(axis.text.x = element_text(size=12,colour ="#000000")) +
  theme(strip.background = element_blank(),
        strip.text.y = element_blank(),strip.placement = "outside",
        strip.text = element_text(size=12,face = "bold"))
```

Figure_3a

GLMM analyses for response duration

Fitting models with count data (number of frames) as response variable with generalized linear mixed-effect models with *poisson*, *nbinom1* and *nbinom2* as link functions [7] [2].

```
model_poisson <-glmmTMB::glmmTMB(response_duration~ barrier_position + treatment_order +
                                      (1|individual_ID),data= experiment_1,family = poisson)
model_nbinom1 <-glmmTMB::glmmTMB(response_duration~ barrier_position + treatment_order +
                                      (1|individual_ID),data= experiment_1,family = nbinom1)
model_nbinom2 <-glmmTMB::glmmTMB(response_duration~ barrier_position + treatment_order +
                                      (1|individual_ID),data= experiment_1,family = nbinom2)
# model selection
bbmle::AICtab(model_poisson,model_nbinom1,model_nbinom2)
```

Model validation using *sumulateResidualfunction* and dispersion test. Please note that test values slightly different each time when you run the simulation.

```
# simulation of residuals of the fitted model and diagnostic plots
res <- DHARMA::simulateResiduals(fittedModel = model_nbino2)
plot(res)
```

Summary of the final model *model_nbino2*.

```
summary(model_nbino2)
```

Individual term assessment.

```
drop1(model_nbino2,test="Chisq")
```

Estimated mean and pair-wise difference of barrier position [6].

```
es1<-emmeans::emmeans(model_nbino2,list(pairwise~barrier_position))
summary(es1)
```

Intervals *back-transformed* from the *logscale* to create new data frame.

```
# model estimates
es1.emm <- emmeans::emmeans(model_nbino2,"barrier_position")
glmmTMB::VarCorr(model_nbino2)
total.SD = sqrt(0.6636 ^2)

# creating data frame
df1<-summary(es1.emm, type="response", bias.adjust = TRUE, sigma= total.SD)
df1
```

Preparing Figure 3b.

```
# plotting graph
Figure_3b<-ggplot2::ggplot(df1, aes(x=barrier_position, y=response,
                                         fill=barrier_position)) +
  geom_point(position=position_dodge(.9), colour="black", stat="identity",size=4) +
  geom_errorbar(position=position_dodge(.9),size= 0.6, width=.25,
                 aes(ymin=lower.CL, ymax=upper.CL)) +
  labs(x = "", y = "Predicted response duration (frames)",face = "bold",size = 12) +
  scale_x_discrete(labels=c("down"= "\nBarrier down", "up" = "\nBarrier up")) +
  coord_cartesian(ylim=c(0,1700)) +
  theme_bw() +
  theme(axis.title.y=element_text(size=12,colour ="#000000")) +
  theme(axis.text.y = element_text(size=11,colour ="#000000")) +
  theme(axis.text.x = element_text(size=12,colour ="#000000")) +
  theme(panel.grid.major = element_line(colour = "transparent")) +
  theme(panel.grid.minor = element_line(colour = "transparent")) +
  theme(legend.position = "none") +theme(strip.background = element_blank(),
    strip.text.y = element_blank(),strip.placement = "outside",
    strip.text = element_text(size=12,face = "bold"))
```

Figure_3b

Experiment 2: combined effects of visual and acoustic restrictions

Converting *individual_ID* and *treatment_order* as factors and then ensure *response_category* as an ordered factor.

```
experiment_2$individual_ID<-as.factor(experiment_2$individual_ID)
experiment_2$treatment_order<-as.factor(experiment_2$treatment_order)
experiment_2$barrier_position<-as.factor(experiment_2$barrier_position)
experiment_2$traffic_noise<-as.factor(experiment_2$traffic_noise)
experiment_2$ordered_response<-factor(experiment_2$response_category,ordered = TRUE)

# creating new data frame without two rows where video recordings unavailable
experiment_2_new<- experiment_2[which(!is.na(experiment_2$response_category)), ]
```

CLMM analyses for immediate responses

Interaction and additive models using *clmm* function.

```
model2<- ordinal::clmm(ordered_response~ traffic_noise * barrier_position +
                         treatment_order + (1|individual_ID), data = experiment_2_new,
                         threshold ="equidistant", link="probit", Hess = TRUE,nAGQ = 20)

model3<- ordinal::clmm(ordered_response~ traffic_noise + barrier_position +
                         treatment_order + (1|individual_ID), data = experiment_2_new,
                         threshold ="equidistant", link="probit", Hess = TRUE,nAGQ = 20)
```

Testing interaction between *traffic_noise* and *barrier_position*.

```
anova(model2,model3)
```

Final model summary.

```
summary(model3)
```

Assessing individual terms.

```
drop1(model3,test="Chisq")
```

Preparing Figure 4a.

```
# adding labels to panels
labels <- c("absent" ="No traffic noise\n", "present" ="Traffic noise\n")

# plotting graph
Figure_4a<-ggplot2::ggplot(experiment_2_new,aes(x = barrier_position,
                                                fill = ordered_response)) +
  geom_bar(position = "fill",width=0.5, colour="black") +
  facet_wrap(~traffic_noise,strip.position = "top", nrow = 1,
             labeller = labeller(traffic_noise=labels)) +
  labs(x = "", y = "Proportion of responses",face = "bold",size = 12) +
  scale_fill_manual(values =c("#FFFFFF","#E1E0DF","#999999","#000000"),
                    name="Immediate\nresponse",
                    labels = c("No response","Scan in place","Move while scanning","Flee")) +
  scale_x_discrete(labels=c("up" = "Barrier up", "down" = "Barrier down")) +
  theme_bw () +
  theme(panel.spacing.x=unit(c(0), "lines"))+
  theme(plot.title = element_blank(),axis.title.x = element_blank()) +
  theme(legend.text = element_text(colour="black", size = 10,
                                    margin = margin(r = 2, unit = 'cm')) +
  theme(legend.key = element_rect(size = 4),
        legend.key.height = unit(0.5, "cm"),
        legend.key.width = unit(0.5, "cm"),
        legend.key.size = unit(0.75, 'lines'))+
  theme(axis.title.y =element_text(size=12)) +
  theme(panel.grid.major = element_line(colour = "transparent")) +
  theme(panel.grid.minor = element_line(colour = "transparent")) +
  theme(axis.text.y = element_text(size=12,colour ="#000000")) +
  theme(axis.text.x = element_text(size=12,colour ="#000000")) +
  theme(strip.background = element_blank(),
        strip.text.y = element_blank(),strip.placement = "outside",
        strip.text = element_text(size=12,face = "bold"))
```

Figure_4a

GLMM analyses for response duration

Subset data, convert counts as response variable (1,0) for Binomial GLMM.

```
# only positive responses
respdu_pos <- subset(experiment_2_new,response_duration > 0)

# creating dichotomous variable (resp_bino, 1 = response , 0 = no response)
experiment_2_new$resp_bino <- as.numeric(experiment_2_new$response_duration > 0)

# convert response variable as factor
experiment_2_new$resp_bino <- as.factor(experiment_2_new$resp_bino)
```

Initial model formulation using response duration - video frame counts [1].

```
model6<- lme4::glmer(response_duration~ traffic_noise * barrier_position +
                        treatment_order + (1|individual_ID),data=experiment_2_new,family = poisson)
summary(model6)
```

Initial model validation using Pearson residual test and simulate function. The following code adopted from *Zuur and Ieno (2016)* [11] and modified for this purpose. The initial model is highly over-dispersed and zero-inflated; the observed zeros are true zeros (responses to the treatments); therefore, we use hurdle models for further examination.

```
E0<-resid(model6,type="pearson")
N <-nrow(experiment_2_new)
p <-length(fixef(model6)) + 1
sum(E0^2)/(N-p)

par(mfrow = c(1,1), mar = c(5,5,2,2), cex.lab = 1.5)
F1 <- fitted(model6)
plot(x = F1,y = E0, xlab = "Fitted values", ylab = "Pearson residuals")
abline(h = 0, lty = 2)

Ysim<- simulate(model6,10000)
dim(Ysim)

zeros <- vector(length=10000)
N <- nrow(experiment_2_new)

for (i in 1:10000) {
  zeros[i]<- sum(Ysim[,i] == 0) / N
}

ZerosWithBreaks<- cut(zeros,breaks=seq(from=0,to=0.6,by=0.05))
par(mar = c(5,5,2,2))
plot(x = seq(0, 0.6, by = 0.05)[1:12],y = as.numeric(table(ZerosWithBreaks)),
      xlim = c(0, 0.6), type = "h", axes = FALSE,
      xlab = "Percentage of zeros", ylab = "Frequency",cex.lab = 1.5)
axis(2)
axis(1, at = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6),
      labels = c("%", "10%", "20%", "30%", "40%", "50%", "60%"))
points(x = sum(experiment_2_new$response_duration ==0) / N, y = 0,
       pch = 16, cex = 1.5, col = 2)
box()
```

Hurdle models with *truncated_poisson* and *truncated_nbino2* [3].

```
model_poisson_a <- glmmTMB::glmmTMB(response_duration ~ traffic_noise * barrier_position +
                                         treatment_order + (1|individual_ID), experiment_2_new,
                                         zi=~.,family = truncated_poisson)
model_poisson_b <- glmmTMB::glmmTMB(response_duration ~ traffic_noise + barrier_position +
                                         treatment_order + (1|individual_ID), experiment_2_new,
                                         zi=~.,family = truncated_poisson)
model_nbino2_a <- glmmTMB::glmmTMB(response_duration ~ traffic_noise * barrier_position +
                                         treatment_order + (1|individual_ID), experiment_2_new,
                                         zi=~.,family = truncated_nbino2)
model_nbino2_b <- glmmTMB::glmmTMB(response_duration ~ traffic_noise + barrier_position +
                                         treatment_order + (1|individual_ID), experiment_2_new,
                                         zi=~.,family = truncated_nbino2)

# model selection
bbmle::AICtab (model_poisson_a,model_poisson_b, model_nbino2_a,model_nbino2_b)
```

Model validation using *simulateResidualfunction* and dispersion test. Please note that test values can slightly vary due to simulated data [5].

```
# simulation of residuals of the fitted model and diagnostic plots
res1 <- simulateResiduals(fittedModel = model_nbinom2_b)
plot(res1)
```

Summary of the final hurdle model *model_nbinom2_b*.

```
summary(model_nbinom2_b)
```

Individual term assessment.

```
drop1(model_nbinom2_b, test="Chisq")
```

Testing interaction between *traffic_noise* and *barrier_position*.

```
anova(model_nbinom2_b, model_nbinom2_a)
```

Estimated mean and pair-wise difference between *traffic_noise* and *barrier_position*.

```
es2<-emmeans(model_nbinom2_b, list(pairwise~ barrier_position + traffic_noise))
summary(es2)
```

Intervals *back-transformed* from the *logscale* to create data frame.

```
# model estimates
es2.emm <- emmeans::emmeans(model_nbinom2_b, "barrier_position", "traffic_noise")
glmmTMB::VarCorr(model_nbinom2_b)
total.SD = sqrt(0.65344 ^2 + 1.8817 ^2)

# creating data frame
df2<-summary(es2.emm, type="response", bias.adjust = TRUE, sigma= total.SD)
df2
```

Preparing Figure 4b.

```
# adding labels to panel
labels1 <- c("absent" ="No traffic noise\n", "present" ="Traffic noise\n")

#plotting graph
Figure_4b<-ggplot2::ggplot(df2, aes(x=barrier_position, y=response,
                                         fill=barrier_position)) +
  geom_point(position=position_dodge(.9), colour="black", stat="identity",size=4) +
  geom_errorbar(position=position_dodge(.9),size= 0.6, width=.25,
                aes(ymin=lower.CL, ymax=upper.CL)) +
  labs(x = "", y = "Predicted response duration (frames)",face = "bold",size = 12) +
  scale_x_discrete(labels=c("down" = "\nBarrier down", "up" = "\nBarrier up")) +
  facet_wrap(~ traffic_noise,strip.position = "top", nrow = 1,
             labeller = labeller(traffic_noise=labels1)) +
  coord_cartesian(ylim=c(0,1600)) +
  scale_fill_manual(values=c("#FFFFFF", "#CCCCCC")) +
  theme_bw() +
  theme(axis.title.y=element_text(size=12,colour ="#000000")) +
  theme(panel.grid.major = element_line(colour = "transparent")) +
  theme(panel.grid.minor = element_line(colour = "transparent")) +
  theme(axis.text.y = element_text(size=12,colour ="#000000")) +
  theme(axis.text.x = element_text(size=12,colour ="#000000")) +
  theme(panel.spacing.x=unit(c(0), "lines")) +
  theme(legend.position = "none") +
  theme(strip.background = element_blank(),
        strip.text.y = element_blank(),
        strip.placement = "outside",
        strip.text = element_text(size=12,face = "bold")) +
  geom_text(data=data.frame(x= 1, y= 1550,noi=c("0","1"),
                            label= c("", "")),aes(x,y,label=label),
            inherit.aes=FALSE)
```

Figure_4b

-
- Examining two separate models: Binomial GLMM comparable with Zero-inflated model part of the Hurdle model; GLMM with link function *nbinom2* using the subset of 51 responses, comparable with the conditional model(part) of the Hurdle model.

Binomial GLMM

```
# binomial model with interaction
bino <- glmmTMB::glmmTMB(resp_bino~ traffic_noise* barrier_position +
                           treatment_order + (1|individual_ID) ,data=experiment_2_new,
                           family=binomial(link = "logit"))

# binomial model without interaction
bino1 <- glmmTMB::glmmTMB(resp_bino~ traffic_noise + barrier_position +
                           treatment_order + (1|individual_ID) ,data=experiment_2_new,
                           family=binomial(link = "logit"))
```

Testing interaction between *traffic_noise* and *barrier_position*.

```
anova(bino1,bino)
```

Summary of the model: *bino1*.

```
summary (bino1)
```

Negative binomial GLMM

```
# negative binomial model with interaction
nb <- glmmTMB::glmmTMB(response_duration ~ traffic_noise * barrier_position +
    treatment_order + (1|individual_ID), respdu_pos, family=nbinom2(link="log"))

# negative binomial model without interaction
nb1 <- glmmTMB::glmmTMB(response_duration ~ traffic_noise + barrier_position +
    treatment_order + (1|individual_ID), respdu_pos, family=nbinom2(link="log"))
```

Testing interaction between *traffic_noise* and *barrier_position*.

```
anova(nb,nb1)
```

Summary of the model: *nb1*.

```
summary (nb1)
```

References

- [1] Douglas Bates et al. “Fitting Linear Mixed-Effects Models Using lme4”. In: *Journal of Statistical Software* 67.1 (2015), pp. 1–48. doi: 10.18637/jss.v067.i01.
- [2] Ben Bolker and R Development Core Team. *bbmle: Tools for General Maximum Likelihood Estimation*. R package version 1.0.23.1. 2020. URL: <https://github.com/bbolker/bbmle>.
- [3] Mollie E. Brooks et al. “glmmTMB Balances Speed and Flexibility Among Packages for Zero-inflated Generalized Linear Mixed Modeling”. In: *The R Journal* 9.2 (2017), pp. 378–400. URL: <https://journal.r-project.org/archive/2017/RJ-2017-066/index.html>.
- [4] Rune Haubo Bojesen Christensen. *ordinal: Regression Models for Ordinal Data*. R package version 2019.12-10. 2019. URL: <https://github.com/runehaubo/ordinal>.
- [5] Florian Hartig. *DHARMA: Residual Diagnostics for Hierarchical (Multi-Level / Mixed) Regression Models*. R package version 0.3.3.0. 2020. URL: <http://florianhartig.github.io/DHARMA/>.
- [6] Russell Lenth. *emmeans: Estimated Marginal Means, aka Least-Squares Means*. R package version 1.5.2-1. 2020. URL: <https://github.com/rvlenth/emmeans>.
- [7] Arni Magnusson et al. *glmmTMB: Generalized Linear Mixed Models using Template Model Builder*. R package version 1.0.2.1. 2020. URL: <https://github.com/glmmTMB/glmmTMB>.
- [8] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2020. URL: <https://www.R-project.org/>.

- [9] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: <https://ggplot2.tidyverse.org>.
- [10] Hadley Wickham et al. “Welcome to the tidyverse”. In: *Journal of Open Source Software* 4.43 (2019), p. 1686. doi: 10.21105/joss.01686.
- [11] Alain F. Zurr and Elena N. Ieno. *Beginner’s Guide to Zero-Inflated Models with R*. Highland Statistics Ltd, 2016.