

Code for producing results and figures of “Estimating sperm whale cue rates to inform passive acoustic density estimation via cue counting”

Marques, T. A. and Marques, C. S. and Gkikopoulou, K. C.

08 June 2023

Contents

Introduction	1
About the data	1
Reading the data	2
Initial data exploration and pre-processing	2
Estimating cue rates	7
Model fitting	8
GLMs	8
GLMMs	11
Predictions for new year-location combinations	12
Notes	16
Aknowledgements	17

Introduction

This document presents the code required to reproduce the figures and results in the manuscript “Estimating sperm whale cue rates to inform passive acoustic density estimation via cue counting”, submitted to The Journal of the Acoustical Society of America, by **add all names here**.

About the data

The dataset lies within `ddata1`, the single object in `data_4_article_clickrates_deep_dive.rda` that gets frontloaded below. This data was created via an internal ACCURATE document from the data that corresponds to the times of detections for each echolocation click from the focal animal detected in each tag, and those times were obtained from the DTAG raw sound files as described in the methods section of the paper. This is combined with the information for the duration of the tag records to obtain cue production rates, number of sounds per animal per unit time. For future reference, the processing of the objects considered here was done in an RMarkdown dynamic report entitled `Cue_Rates_For_Sperm_Whales.Rmd`. This document is shared as part of a separate data paper, where the times of echolocation clicks in this unique DTAG dataset, along with the DTAGs depth profiles, are shared to be used by others.

The data consists of summaries of numbers of regular echolocation clicks per deep dive cycle, for each of the sperm whale tags considered on the manuscript. Note that despite having been recorded at the deep dive cycle level, to be used in the manuscript “A sperm whale cautionary tale about estimating acoustic cue rates for deep divers”, submitted to The Journal of the Acoustical Society of America, by Marques, T. A., Marques, C. S. & and Gkikopoulou, K. C., the first step in the data pre-processing here is to pool data for each tag record, as the tag is the fundamental (and more importantly independent) sampling unit.

Reading the data

We begin by reading the deep dive cycle data in:

```
# file created in Cue_Rates_For_Sperm_Whales.Rmd
# Reading the data that contain the information per deep dive cycle - object ddata1
load("../data_4_article_clickrates_deep_dive.rda")
```

Initial data exploration and pre-processing

We actually have data from whales which have been subjected to SONAR exposure under controlled exposure experiments. We discard the data from those tags here, since evaluating the effect of SONAR is the focus of other research programs. Understanding the effects of sonar exposure on whale behaviour, and in particular cue rate production, is a separate research thread which requires information to be analysed at a much finer resolution, and for which context would have to be considered. For the majority of the tags the context information is unavailable to us within the ACCURATE project.

```
#removing the tags for animals we know were exposed to sonar
DDCs<-ddata1[ddata1$sonar!="sonar",]
```

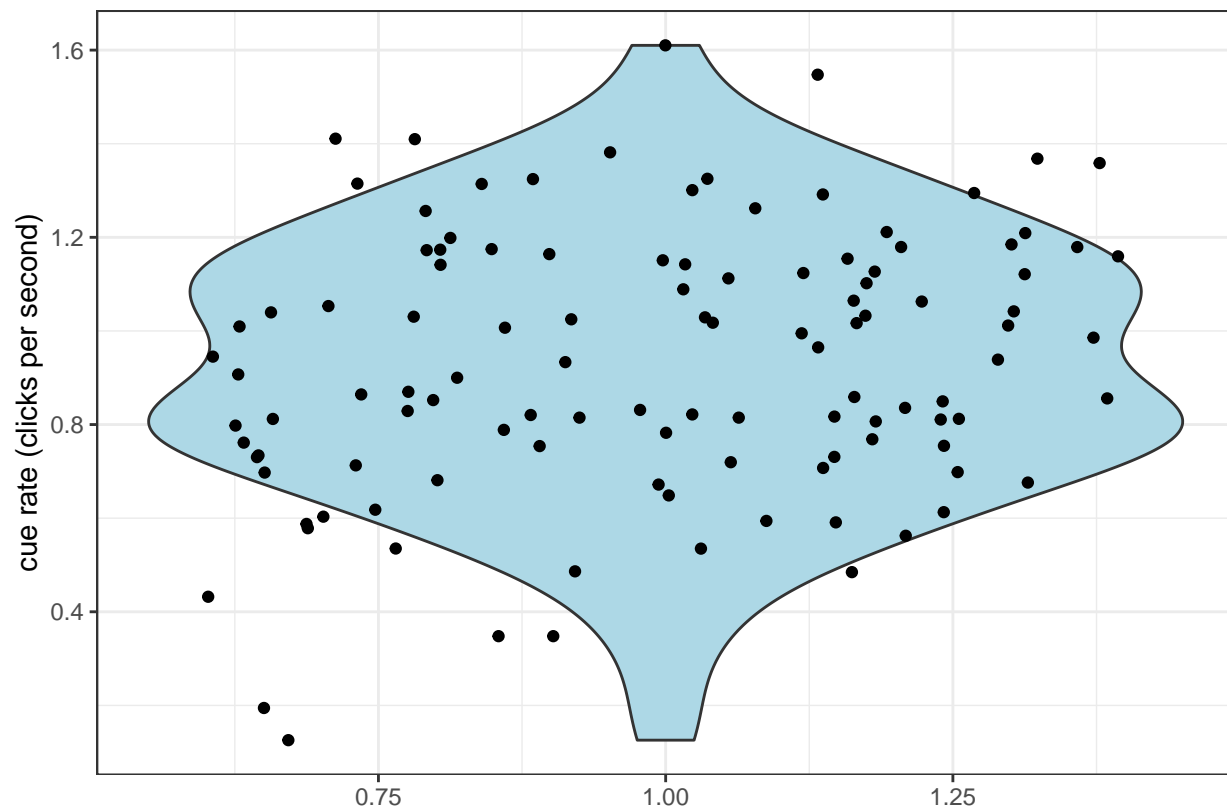
Since we will treat tags, not deep dive cycles, as the independent sampling units, we aggregate the deep dive cycle data for each tag into a single tag record

```
# Creating the data per tag
tags<-DDCs%>%
  group_by(tag)%>%
  summarise(location=unique(location), year=unique(year), sex=unique(sex),
    duration= sum(durations,na.rm=T),nclicks=sum(nclick,na.rm=T),
    crate=sum(nclick,na.rm=T)/sum(durations,na.rm=T),ddc=max(absdives+1,na.rm = T))
```

We have a total of 113 whales for which whales were not, knowingly, exposed to sonar.

Tag recording duration ranged from 0.144 to 25.781 hours. The observed cue rates per tag varied between 0.126 and 1.61, with a mean value of 0.93 and median value of 0.933. The individual cue rates per tag record, pooled across years and locations, are shown in the following figure:

```
ggplot(tags,aes(x=1,y=crate),fill="lightblue")+
  theme_bw()+geom_violin(fill="lightblue")+
  geom_jitter()+ylab("cue rate (clicks per second)")+xlab("")
```



Below we present a table with the locations and years covered by these tags:

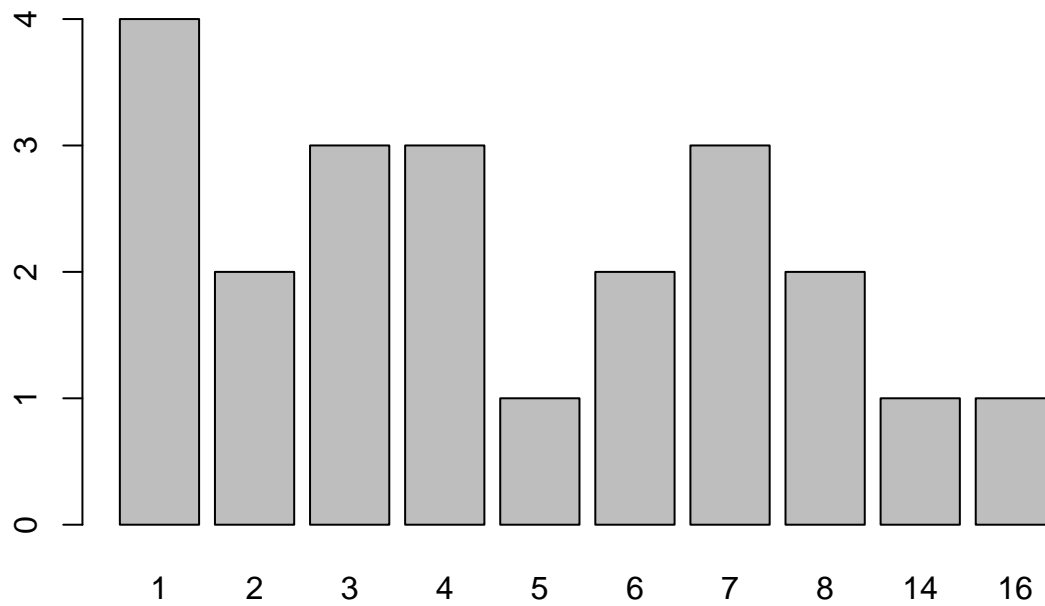
```
kable(table(tags$location, tags$year))
```

	2001	2002	2003	2005	2009	2010	2013	2014	2015	2016	2017	2018	2019	2020	2021
Azores	0	0	0	0	0	8	0	0	0	0	0	3	5	6	3
DOMINICA	0	0	0	0	0	0	0	7	16	4	1	4	0	0	0
Gulf of Mexico	4	14	8	0	0	0	0	0	0	0	0	0	0	0	0
Kaikoura	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0
Mediterranean	1	0	7	0	0	0	0	0	0	0	0	0	0	0	0
North Atlantic	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0
Delaware															
Norway	0	0	0	0	1	3	0	0	0	2	0	0	2	0	0
Norway	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Andenes															

The number of tags per year-location combination varies considerably. Out of a total of 8 different locations and 15 different years, leading therefore to 120 possible year-location combinations, tags are not available for the majority of these possible combinations (98 combinations), with only 22 combinations having any tags associated with.

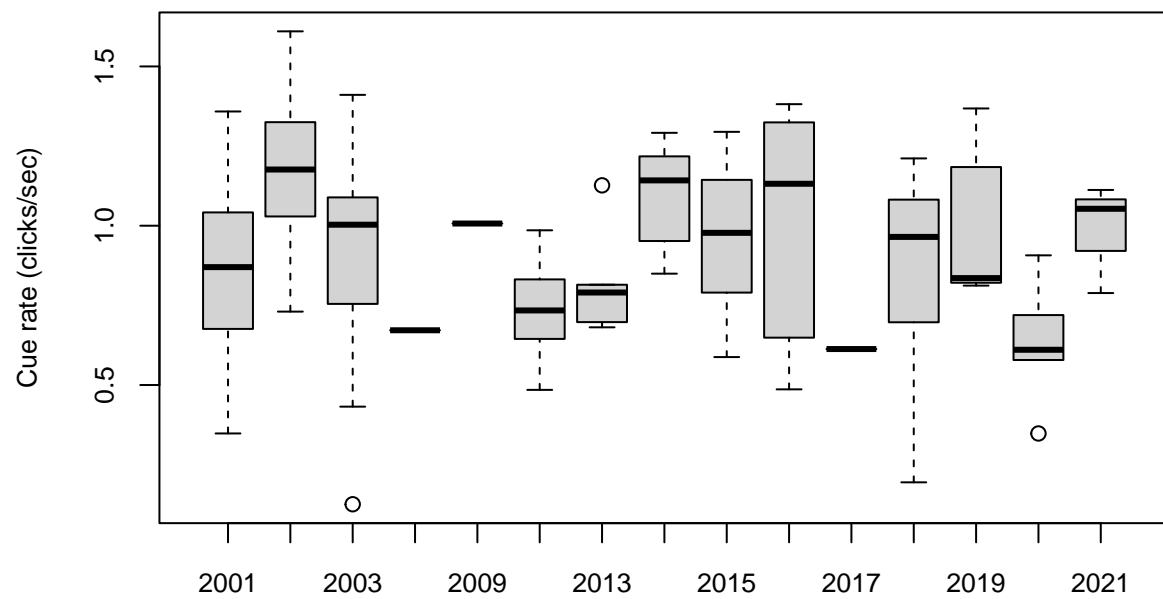
What might be the number of tag records required to obtain a reliable year-location cue rate estimate remains hard to evaluate, but several year-location combinations are certainly below that minimum, namely for those with less than a handful of tags. The distribution of the number of tags per year-location combination for which we have tags is represented in the image below:

```
#note the need to remove the first count  
#corresponding to un-interesting year-location combinations  
#with 0 tags  
barplot(table(table(tags$location,tags$year))[-1])
```



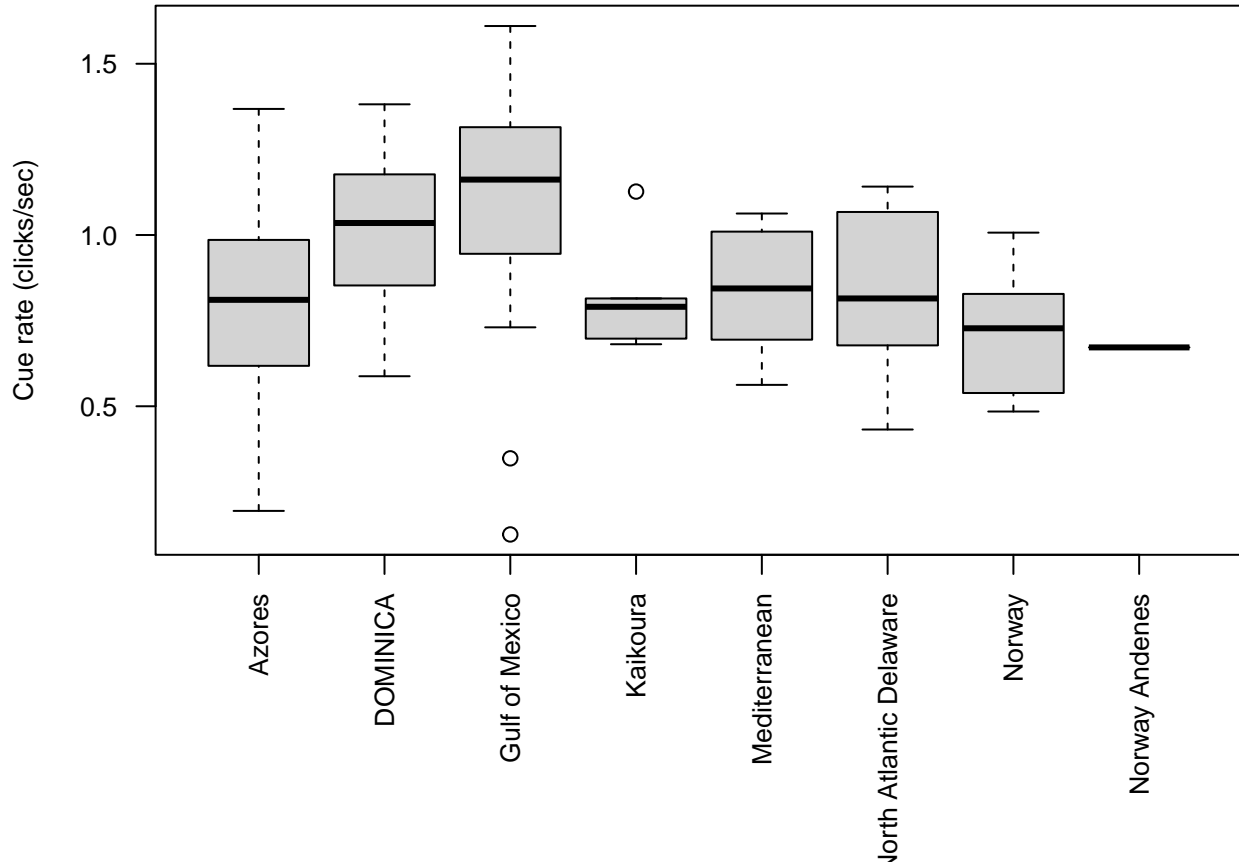
We can take a look at the cue rates (pooled across locations) per year

```
with(tags,boxplot(crate~year,ylab="Cue rate (clicks/sec)",xlab="",cex.lab=0.8,cex.axis=0.8))
```



and those (pooled across years) per location

```
par(mar=c(8,4,0.2,0.2))
with(tags,boxplot(crate~location,las=2,ylab="Cue rate (clicks/sec)",xlab="",cex.lab=0.8,cex.axis=0.8))
```



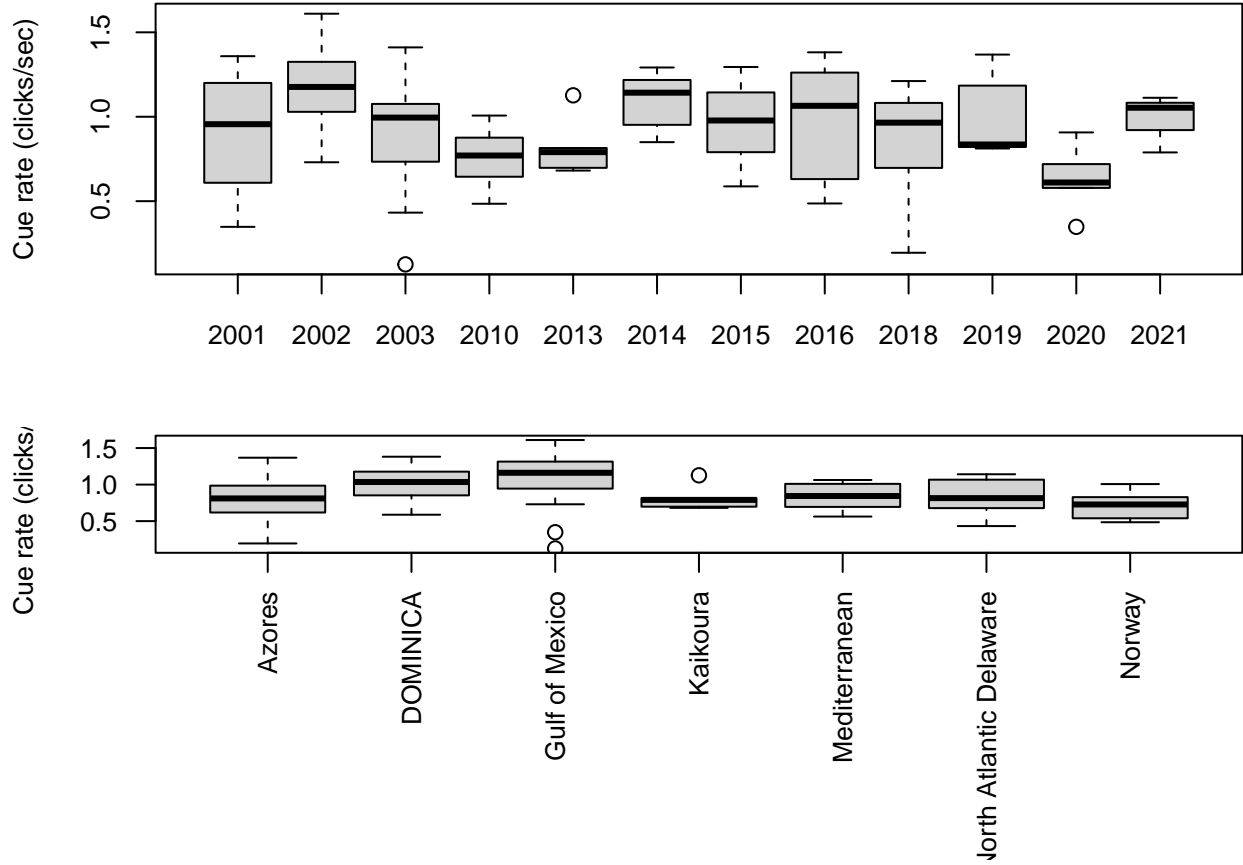
We are hoping to explain variability in cue rates as a function of year and location. To do so, year-location combinations with a small number of tags are difficult to deal with. In particular, for those with a single tag, a year-location effect would be hard to estimate: with a model with an interaction, the effect would be strictly unidentifiable. We therefore removed years and locations for which only a single tag existed, effectively removing the single tag from Norway Andenes, also the only tag from 2005. Additionally, we grouped tags from single-tag year-location combinations into an adjacent year for the same same location. Hence, we pooled

- the single tag from Norway in 2009 with the 3 tags from the year 2010;
- the single tag from Dominica in 2017 with the remaining 4 tags from 2016;
- the single tag from the Mediterranean in 2001 with the 7 tags from the year 2003.

```
#removing the location with a single tag
tags<-tags[tags$location!="Norway Andenes",]
#grouping single tag per year into adjacent years
tags$year[tags$location=="Norway" & tags$year==2009]<- 2010
tags$year[tags$location=="DOMINICA" & tags$year==2017]<- 2016
tags$year[tags$location=="Mediterranean" & tags$year==2001]<- 2003
```

Given this data pooling, the above plots become:

```
par(mfrow=c(2,1),mar=c(4,4,0.2,0.2))
with(tags,boxplot(crate~year,ylab="Cue rate (clicks/sec)",xlab="",cex.lab=0.8,cex.axis=0.8))
par(mar=c(8,4,0.2,0.2))
with(tags,boxplot(crate~location,las=2,ylab="Cue rate (clicks/sec)",xlab="",cex.lab=0.8,cex.axis=0.8))
```



Estimating cue rates

Our objective is to obtain a cue rate estimate, and its desired precision to include in a cue counting density estimator. We assume cue rate will depend on a number of covariates, here in particular location and time.

Ignoring predictions to begin with, from first principles, location might be a sensible fixed effects covariate, since different locations will present different depths and prey distributions, and hence foraging at different depths might occur, and consequently different cue rates per location (across years). On the other hand, it seems like the variability from year might be not driven by year itself, but as random fluctuations over time, perhaps more sensibly accounted for as factor (or a random effect). If one believes this to be the case, to predict cue rates for:

1. a sampled location and a new year, one could consider to use a model with location as a fixed effect, propagating the variability of year as a random effect;
2. for a new location and year, one could would use a model with both location and year as random effects.

In other words, intuitively year seems more sensibly modeled as a random effect, while location might be more sensibly modeled as a fixed effect.

We can distinguish different levels of difficulty in terms of cue rate estimation with regards to the available information to do so. From easiest to hardest, we might want to estimate a cue rate for:

- a year-location combination we have data for;
- a location we have data for at a year we do not have data for;
- a year we have data for at a location we do not have data for;
- a completely new year-location combination.

From a conceptual point of view, if we had enough data across years and locations, one might want to:

1. treat as fixed effects those covariates for which we observed the level for which predictions are desired, but
2. treat as random effects those covariates for which the level at which we would like to predict were not observed

When considering a model with random effect(s) care must be had to propagate into predictions the variability associated with predicting for a new, previously unobserved, level of the random effect. As will be described below, this is not necessarily straightforward, and different alternatives are available to do so.

Model fitting

We will assume that cue rates, a strictly positive quantity, follow a Gamma distribution, and a log-link function will be considered within a generalized linear model (GLM) or generalized linear mixed model (GLMM) to ensure positive predictions.

GLMs

We begin by looking at GLM models where both location and year are treated as fixed effects, first with year as a numerical covariate, including or not interaction terms between year and location and then with year as a factor. The latter seems more sensible a priori, as noted above.

```
#run models
CRglm1<-glm(crate~location+year,data=tags,family=Gamma(link="log"))
#summary(CRglm1)
CRglm2<-glm(crate~location+year+year:location,data=tags,family=Gamma(link="log"))
#summary(CRglm2)
#making a new variable, year as factor
tags$fyyear<-as.factor(tags$year)
CRglm3<-glm(crate~location+fyyear,data=tags,family=Gamma(link="log"))
#summary(CRglm3)
CRglm4<-glm(crate~location+fyyear+fyyear:location,data=tags,family=Gamma(link="log"))
#summary(CRglm4)

kable(AIC(CRglm1,CRglm2,CRglm3,CRglm4))
```

	df	AIC
CRglm1	9	51.91597
CRglm2	12	57.84711
CRglm3	17	51.20435
CRglm4	19	54.76608

According to AIC the best model considers both location and year as factors, while the interaction between these factors is not deemed relevant. Given the nature of year and expectations as noted above for temporal effects, using year as a factor seems indeed more sensible any way.

```
#define indexes of year-location combinations we are most interested in
iGoM2002<-tags$location=="Gulf of Mexico" & tags$year==2002
iGoM2001<-tags$location=="Gulf of Mexico" & tags$year==2001
iDom2015<-tags$location=="DOMINICA" & tags$year==2015
iDom2018<-tags$location=="DOMINICA" & tags$year==2018
```

We illustrate how to estimate the cue rate, and its precision, for all year location combinations for which we have more than 3 tags. The choice of 3 is arbitrary, but we considered that 3 or less tags would be unreliable (can we perhaps look into it? given var across locations vs within locations, one might be better off MRSE-wise to estimate from other locations). We do note

- a location and year for which we have available data. We consider here the cases for which we have the most tags, namely
 - the Gulf of Mexico in 2002 (14 tags), and
 - Dominica in 2015 (16 tags), and
- for these same locations, two different years when we have less data, namely
 - 2001 for the Gulf of Mexico (4 tags) and
 - 2018 for Dominica (4 tags).

```
# create all possible unique location-year combinations
all.comb<-expand.grid(location=sort(unique(tags$location)),year=sort(unique(tags$year)))
# define minimum number of tags required to produce an estimate
nmin<-3
# select only those
index.min<-which(table(tags$location,tags$year)>3)
# create an object to hold results by location-year
byly <- all.comb[index.min,]
```

```
#now, for each combination
for(i in 1:nrow(byly)){
  index <- tags$location==byly$location[i] & tags$year==byly$year[i]
  #get the actual number of tags
  byly$ntags[i] <- sum(index)
  #calculate empirical cue rate
  byly$ecr[i] <- mean(tags$crate[index])
  #calculate empirical standard deviation
  byly$ecrsd[i] <- sd(tags$crate[index])
  #and the margin for a confidence interval
  byly$margin.ecr[i] <- qt(0.975,byly$ntags[i]-1)*byly$ecrsd[i]/sqrt(byly$ntags[i])
}
# get the lower and upper 95% CI
byly$lcl.ecr <- with(byly,ecr - margin.ecr)
byly$ucl.ecr <- with(byly,ecr + margin.ecr)
```

```
#this might be safely deleted as it is all above
ntags.GoM2002 <- sum(iGoM2002)
ntags.GoM2001 <- sum(iGoM2001)
ntags.Dom2015 <- sum(iDom2015)
ntags.Dom2018 <- sum(iDom2018)
mean.cr.GoM2002 <- mean(tags$crate[iGoM2002])
mean.cr.GoM2001 <- mean(tags$crate[iGoM2001])
mean.cr.Dom2015 <- mean(tags$crate[iDom2015])
mean.cr.Dom2018 <- mean(tags$crate[iDom2018])
sd.cr.GoM2002 <- sd(tags$crate[iGoM2002])
sd.cr.GoM2001 <- sd(tags$crate[iGoM2001])
sd.cr.Dom2015 <- sd(tags$crate[iDom2015])
sd.cr.Dom2018 <- sd(tags$crate[iDom2018])
margin.cr.GoM2002 <- qt(0.975,ntags.GoM2002-1)*sd(tags$crate[iGoM2002])/sqrt(ntags.GoM2002)
margin.cr.GoM2001 <- qt(0.975,ntags.GoM2001-1)*sd(tags$crate[iGoM2001])/sqrt(ntags.GoM2001)
margin.cr.Dom2015 <- qt(0.975,ntags.Dom2015-1)*sd(tags$crate[iDom2015])/sqrt(ntags.Dom2015)
margin.cr.Dom2018 <- qt(0.975,ntags.Dom2018-1)*sd(tags$crate[iDom2018])/sqrt(ntags.Dom2018)
```

The point estimates and respective 95% confidence intervals using a standard average are:

- Gulf of Mexico in 2002 (14 tags): 1.18 (1.04-1.33)
- Gulf of Mexico in 2001 (4 tags): 0.9 (0.23-1.58)
- Dominica in 2015 (16 tags): 0.97 (0.85-1.08)

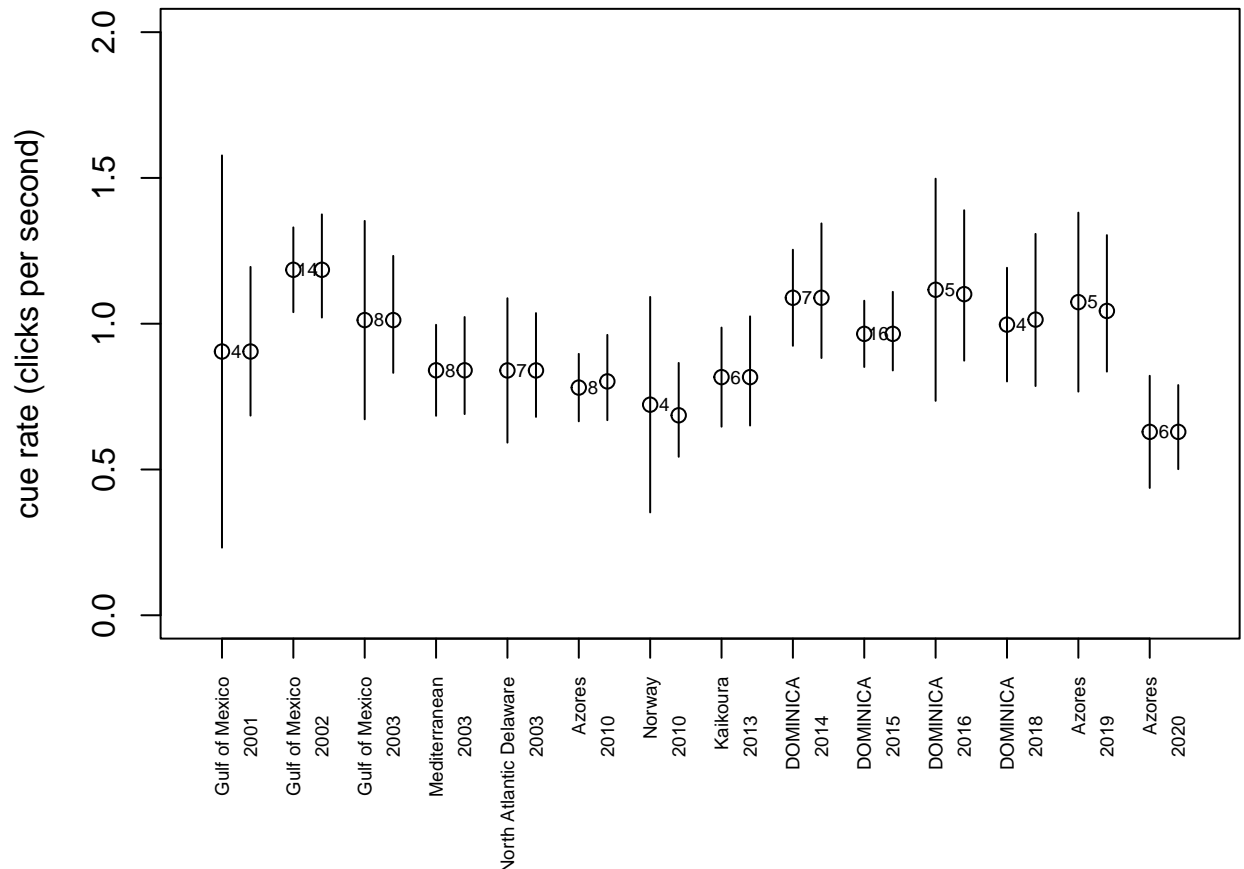
- Dominica in 2018 (4 tags): 1 (0.8-1.19)

If we consider a GLM model with both year and location as fixed effects, the point estimates and respective 95% confidence intervals are easiest to obtain by recoding the year and location factors to have the desired levels of each factor as the intercept, and then exponentiating back to the linear scale (for confidence intervals, we create these on the link scale and then transform back to the response scale the interval limits).

```
#now, for each combination
for(i in 1:nrow(byly)){
  index <- tags$location==byly$location[i] & tags$year==byly$year[i]
  # recode to get the current baseline
  # define the current year as baseline
  iyear <- which(byly$year==byly$year[i])
  #new levels
  levelsy<-c(byly$year[i],unique(byly$year[byly$year!=byly$year[i]]))
  #recode levels for year
  tags$fyear<-factor(tags$fyear,levels=levelsy)
  # define the current location as baseline
  ilocation <- which(byly$location==byly$location[i])
  #new levels
  levelsl<-c(byly$location[i],unique(byly$location[byly$location!=byly$location[i]]))
  tags$location<-factor(tags$location,levels=levelsl)
  #fit the model for the current year and location as baseline
  CRglm4yl<-glm(crate~location+fyear,data=tags,family=Gamma(link="log"))
  # get the cue rate
  byly$glm.cr[i] <- exp(summary(CRglm4yl)$coefficients[1,1])
  byly$glm.lci.cr[i] <- exp(summary(CRglm4yl)$coefficients[1,1]-qt(0.975,summary(CRglm4yl)$df.residual))
  byly$glm.uci.cr[i] <- exp(summary(CRglm4yl)$coefficients[1,1]+qt(0.975,summary(CRglm4yl)$df.residual)*s
}
```

We compare below standard averages against values obtained from the glm (between the value for the estimates for each year-location combination corresponds we show the number of tags available for each year-location combination)

```
#standard average
par(mfrow=c(1,1),mar=c(6,4,0.1,0.4))
plot(x=(1:nrow(byly))-0.2,y=byly$ecr,xaxt="n",ylim=c(0,2),xlim=c(0.5,nrow(byly)+0.5),xlab="",ylab="cue :
with(byly,segments(x0=(1:nrow(byly))-0.2,x1=(1:nrow(byly))-0.2,y0=lcl.ecr,y1=ucl.ecr))
#GLM
points(x=(1:nrow(byly))+0.2,y=byly$glm.cr)
with(byly,segments(x0=(1:nrow(byly))+0.2,x1=(1:nrow(byly))+0.2,y0=glm.lci.cr,y1=glm.uci.cr))
#draw axis and annotations
axis(1, at=(1:nrow(byly))-0.2,byly$location,cex.axis=0.6,las=2)
axis(1, at=(1:nrow(byly))+0.2,byly$year,tick=FALSE,cex.axis=0.6,las=2,line=1)
text(x=(1:nrow(byly)),y=byly$ecr,labels=byly$ntags,cex=0.6)
```



What these estimates illustrate is that point estimates are not that different from each other, with almost complete overlap in confidence intervals, except for the GoM in 2002, with slightly higher estimated cue rate. It also illustrates how a cue rate estimate drawn from a reduced number of tags could result in inadmissible estimates (the naive 95% CI for the GoM in 2001 approaches 0, while negative values for the cue rate are not possible). From that perspective, estimates from the fitted model might be better, as these will avoid negative values by construction, induced by the log link. Interestingly, estimates from the GLM model are slightly less precise for all but the most variable year-location combination, the GoM in 2001, where the strength borrowed from all the tags analysis means the model based estimate is considerably more precise.

GLMMs

Attempting a model with location as a fixed effect, but with year as a random effect, to predict cue rate for a location we have data from, in a year we do not have data at that location. Therefore we consider a Gamma log-link GLMM for cue rate. Let us consider we are predicting the cue rate for Dominica and the GoM in the year 2012. Note the corresponding estimate would be the same in any non-observed year, it's a generic estimate for any non-observed level of the (year) random effect.

```
#define DOMINICA as baseline
tags$location<-factor(tags$location,levels=c("DOMINICA","Gulf of Mexico","Azores","Kaikoura","Mediterranean"))
#run model
crglmerDom<-glmer(crate~location+(1|fyear),data=tags,family=Gamma(link="log"))
#define GoM as baseline
tags$location<-factor(tags$location,levels=c("Gulf of Mexico","DOMINICA","Azores","Kaikoura","Mediterranean"))
# run model
crglmerGoM<-glmer(crate~location+(1|fyear),data=tags,family=Gamma(link="log"))
```

The empirical mean cue rate across all years for the GoM is 1.09 clicks per second and for Dominica is 1.02

clicks per second. On the other hand, considering the GLMM, those values are 1.05 clicks per second and 1.06 clicks per second for the GoM and Dominica, respectively. So while the observed cue rate was larger in the GoM, it is estimated by the GLMM as being (even if ever just) higher in Dominica.

This implies that most of the variability that the model with two fixed effects (location and year) was attributing to changes per location are accounted for by random fluctuations in variability per year in the GLMM, and the random effect just happened to be higher for years in which the GoM was sampled (in particular 2002).

Note that we do not present here precision measures for the estimated cue rate in 2012. Obtaining the correct precision on the estimates for unobserved levels of the (year) random effect is not straightforward. We do so for a model with both year and location as random effects below.

Predictions for new year-location combinations

The above results seem to suggest that it might be quite difficult to separate location and year effects.

A precautionary approach, when predicting a cue rate for new year and location combinations, might be to use both year and location as random effects in a random effects model and then propagate the uncertainty associated with the overall mean with respect to both random effects, year and location, that would be unobserved. This is implemented here:

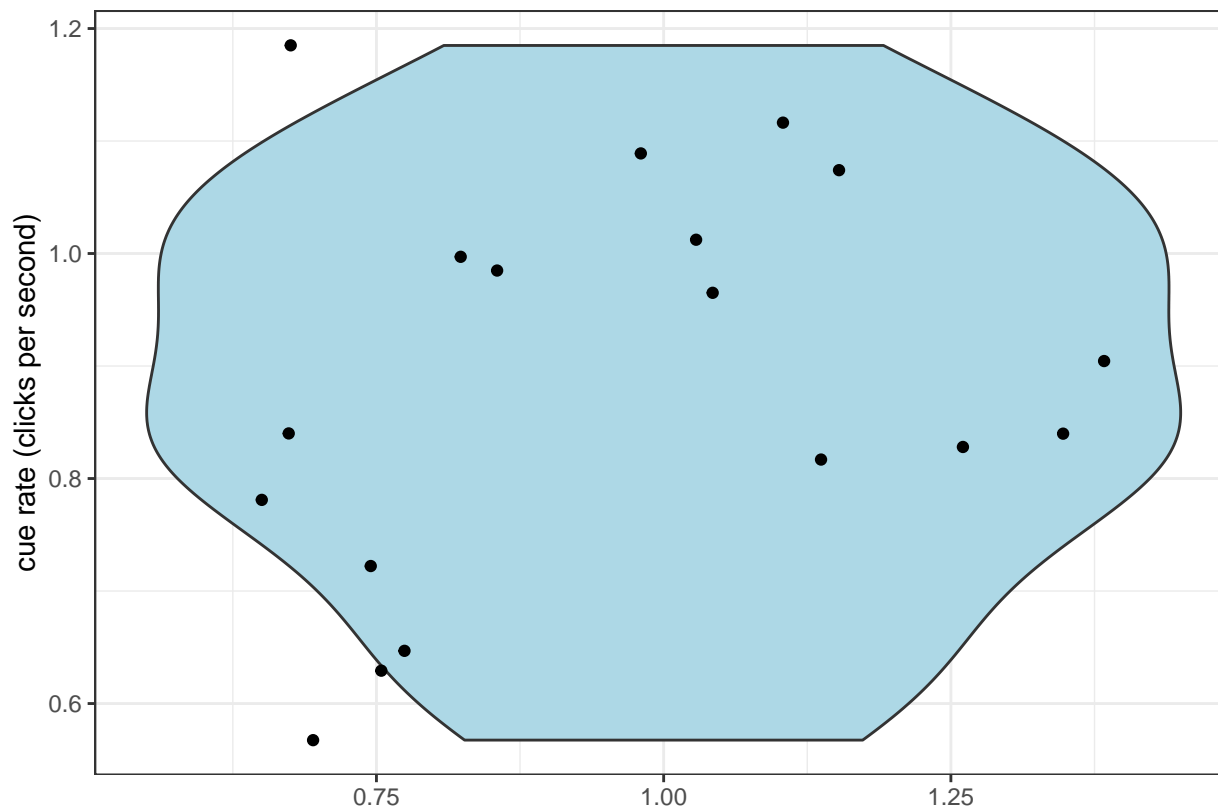
```
crglmer<-glmer(crate~(1|location)+(1|fyear),data=tags,family=Gamma(link="log"))
#crglmer<-glmer(crate~(1/as.factor(paste0(location+fyear))),data=tags,family=Gamma(link="log"))
```

Under this representation, the cue rate would be estimated, for any new location-year combination as 0.84 clicks per second.

We note explicitly this is quite lower than the overall mean of all tags cue rates 0.93 clicks per second, a consequence of the tags being considerably unbalanced across years and locations, with more tags in year-location combinations which happened to have higher cue rates. This is nonetheless closer to, as would be expected, the mean of the average cue rates per year-location combination (0.89 clicks per second).

Note that the variability on the observed cue rates per year-location combination is not that large:

```
crates.psycomb<-data.frame(cr=tapply(X=tags$crate,INDEX = paste0(tags$location,tags$year),FUN=mean),sy=
ggplot(crates.psycomb,aes(x=1,y=cr),fill="lightblue")+
  theme_bw()+geom_violin(fill="lightblue")+
  geom_jitter()+ylab("cue rate (clicks per second)")+xlab("")
```



```
#boxplot(tapply(X=tags$crate,INDEX = paste0(tags$location,tags$year),FUN=mean))
```

To estimate the precision on this mean estimate for cue rate we would need to propagate the variability associated with the random effects into the point estimate of the intercept of the model.

How can that be done? Bolker et al. note here under Predictions and/or confidence (or prediction) intervals on predictions “none of the following approaches takes the uncertainty of the random effects parameters into account...” and suggest that “if you want to take RE parameter uncertainty into account, a Bayesian approach is probably the easiest way to do it.”

Ben Bolker (BB) further provided several possible approaches as a reply to a question TAM posted on stack exchange:

<https://stats.stackexchange.com/questions/616697/how-to-estimate-precision-on-a-prediction-for-a-glmm-for-an-unobserved-level-of/>

BB describes a possible approach via an (intuitive) parametric bootstrapp. The suggested idea would be to resample values from the distribution of the overall mean, and then add sampled values from the variability associated with each of the random effects, and compute the variability of the resulting estimates. These have been referred to in the literature as “population prediction intervals”. We note in passing that BB notes that this bootstrap procedure might be hard to justify from a theoretical point of view:

<https://stats.stackexchange.com/questions/590595/justification-for-population-prediction-intervals>

The options would be:

1. a procedure involving the analytic estimates of the variances of the parameters
2. a quick (parametric) bootstrapp procedure
3. a true (parametric) bootstrapp procedure (see below I actually implemented a possibly quite weird mix of a non-parametric+parametric bootstrap)

4. a full Bayesian implementation

Here I attempt to implement options 2 and 3.

In this case, the quick parametric bootstrapp would be

```
set.seed(123)
B<-9999
means<-rnorm(B,mean=summary(crglmer)$coefficients[1,1],sd=summary(crglmer)$coefficients[1,2])
desv.year<-rnorm(B,mean=0,sd=sqrt(as.numeric(summary(crglmer)$varcor[1])))
desv.loc<-rnorm(B,mean=0,sd=sqrt(as.numeric(summary(crglmer)$varcor[2])))
#estimates at a new year location combination, on the link scale
est.lmean.crs<-means+desv.year+desv.loc
#estimates on the scale of the response
est.mean.crs<-exp(est.lmean.crs)
```

and now to get a confidence interval we can simply use the percentile method, leading to a mean estimate of 0.84 and 95% confidence intervals of 0.62-1.14.

Note that I did not implement the methods strictly as described by BB, instead I did “almost” that, following what I had intuitively thought about originally. BB suggested the following procedure:

- draw MVN samples (using `MASS::mvrnorm` or one of the other alternatives in the R ecosystem) from the distribution with the combined FE sampling variance + RE variance
- exponentiate them, and
- draw Gamma samples based on those mean values

but it is unclear to me, **and would love some additional insight from BB**, why:

- one would want to implement the third step, draw from a Gamma, since what we are interested is in the variation on the mean of a new year-location combination, not the variance in observations from such a location;
- (this is potentially a mute point/detail, but nonetheless I would like to know) what said Gamma would be, given via `glmer` we only get the mean value of the Gamma. Where in the output of `glmer` lies the estimate for the dispersion parameter of the corresponding Gamma (as happens to be reported by say a `glm` call with the argument `family=Gamma`)?

As per the answer by BB (see also section 5.3 in Bolker 2008 <https://math.mcmaster.ca/~bolker/emdbook/c hap7A.pdf>), while the above procedure is sensible, this approach will ignore the variability associated with the estimation of the random effects, and a “full” parametric bootstrap approach would first resample then re-fit the model.

I implement here below this second bootstrap, which would account for the variability in estimating the random effects, by sampling year-location combinations.

```
#define location-year combination as a variable
tags$ly<-with(tags,paste0(location,year))
#unique location-year combinations
lys<-unique(tags$ly)
#number of combinations
nlys<-length(lys)
B<-9999
res.boot<-numeric(B)
#for each bootstrapp resample
for(i in 1:B){
  #select a reasmple of location-year combinations
  ly1<-sample(lys,1)
  boot.tags<-tags[tags$ly==ly1,]
  for(j in 2:nlys){
```

```

    boot.tags<-rbind(boot.tags,tags[tags$ly==sample(lys,1),])
  }
  #fit the model
  #NOTE: while I silence the messages, we get lots of "boundary (singular) fit: see help('isSingular')"
  crglmer.boot<-glmer(crate~(1|location)+(1|fyear),data=boot.tags,family=Gamma(link="log"))
  means.b<-rnorm(1,mean=summary(crglmer.boot)$coefficients[1,1],sd=summary(crglmer.boot)$coefficients[1,2])
  desv.year.b<-rnorm(1,mean=0,sd=sqrt(as.numeric(summary(crglmer.boot)$varcor[1])))
  desv.loc.b<-rnorm(1,mean=0,sd=sqrt(as.numeric(summary(crglmer.boot)$varcor[2])))
  #estimates at a new year location combination, on the link scale
  est.lmean.crs.b<-means.b+desv.year.b+desv.loc.b
  #estimates on the scale of the response
  res.boot[i]<-exp(est.lmean.crs.b)
}

```

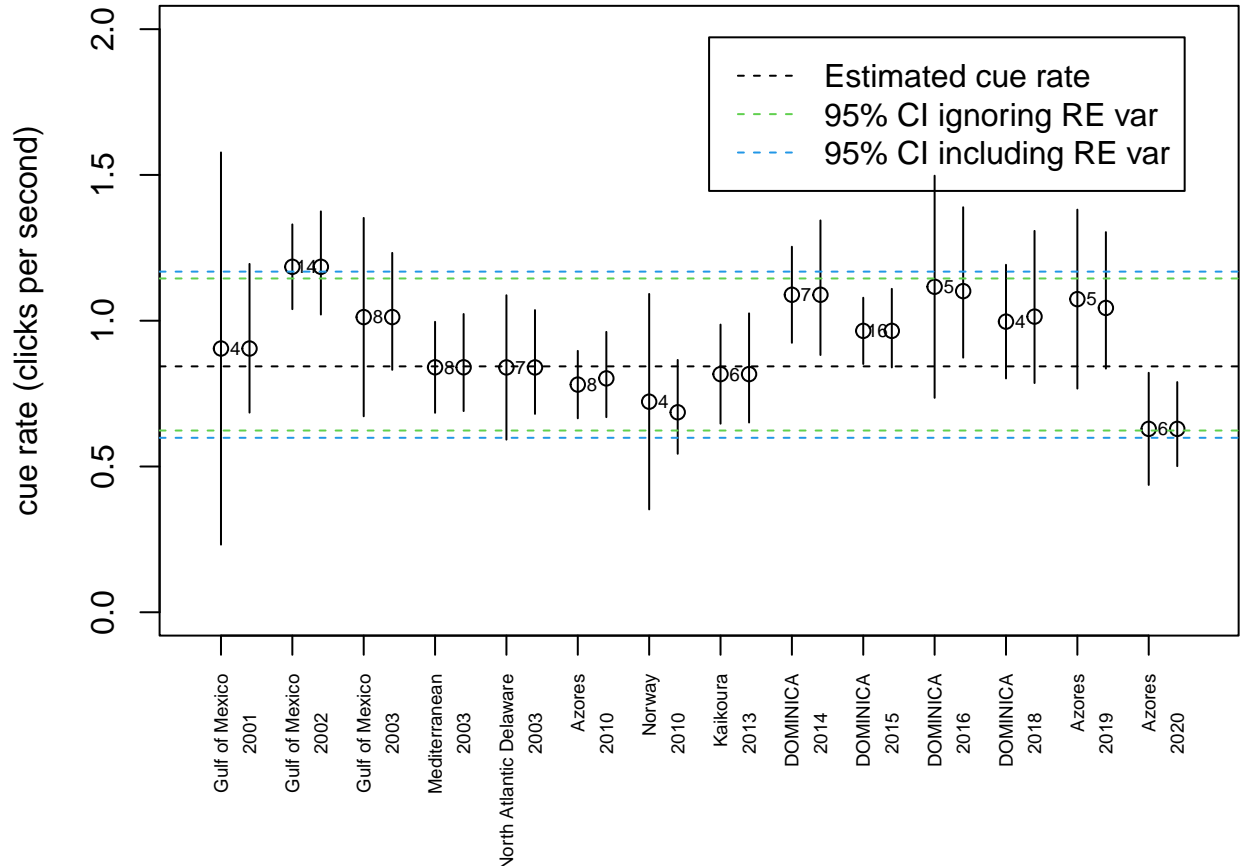
and now to get a confidence interval we can again use the percentile method, leading to a 95% confidence interval of 0.6-1.17. As expected, the 95% CI are, even if just ever so slightly, wider than when ignoring the component of variation due to the random effects estimation. We can see that overlaid with the point and interval estimates presented before:

```

#standard average
par(mfrow=c(1,1),mar=c(6,4,0.1,0.4))
plot(x=(1:nrow(byly))-0.2,y=byly$ecr,xaxt="n",ylim=c(0,2),xlim=c(0.5,nrow(byly)+0.5),xlab="",ylab="cue rate",
with(byly,segments(x0=(1:nrow(byly))-0.2,x1=(1:nrow(byly))+0.2,y0=lcl.ecr,y1=ucl.ecr))
#GLM
points(x=(1:nrow(byly))+0.2,y=byly$glm.cr)
with(byly,segments(x0=(1:nrow(byly))+0.2,x1=(1:nrow(byly))+0.2,y0=glm.lci.cr,y1=glm.uci.cr))
#draw axis and annotations
axis(1, at=(1:nrow(byly))-0.2,byly$location,cex.axis=0.6,las=2)
axis(1, at=(1:nrow(byly))+0.2,byly$year,tick=FALSE,cex.axis=0.6,las=2,line=1)
text(x=(1:nrow(byly)),y=byly$ecr,labels=byly$ntags,cex=0.6)

#the estimated mean cue rate from GLMER
abline(h=exp(summary(crglmer)$coefficients[1,1]),lty=2)
#adding a new year location combination - 95% CI that ignore RE estimation variability
abline(h=quantile(est.mean.crs,probs=c(0.025,0.975)),col=3,lty=2)
#adding a new year location combination - 95% CI accounting for said variability
abline(h=quantile(res.boot,probs=c(0.025,0.975)),col=4,lty=2)
legend("topright",legend=c("Estimated cue rate","95% CI ignoring RE var","95% CI including RE var"),lty=

```



This result is quite interesting. As expected, if predicting for a new location-year not yet observed, we include within the confidence interval all but one of the cue rate point estimates obtained empirically for observed year-location combinations. The fact that we do not include the Gulf of Mexico in the year 2002, one of the year locations with the largest number of tags (14 tags) suggests that this year-location combination the cue rates were exceptionally high. The cause for that remains unresolved (sex info available? must check).

Notes

Some notes added on 8th June 2023, for BB benefit and for future reference. I've noticed that I might have mis-implemented your suggestions in (at least) two ways (above and beyond the "almost" bit I had already noted)

1. I now realize I got sidetracked and instead of what you refer to as full parametric bootstrap, I actually created a mix of non-parametric with parametric bootstrap: first randomize independent sampling units (year-location combinations? individual tags? I lean towards the former, as that encompasses the second level of variability) observations, re-fit the model at each iteration, sampling from the variances associated with each random effect. **Note:** While I can get an estimate of the mean at each bootstrap iteration, how do I get the variability on the random effects? At each iteration, there is none. Do I simply generate a single value for a possible mean of a year-location combination at each bootstrap resample? I guess so! (though this seems, intuitively, quite computationally ineffective).
2. For option #3, I might have ignored your comment "including the RE variance but not the FE sampling variance" since I sampled from the intercept distribution as well as for the random effects at each bootstrap resample (to be consistent with what I had done in option #2)

For option #3, regarding the non-parametric bootstrap, I actually resampled data from year-location combinations, not really year and location independently. The latter would mean one would end up with non-existing year-location combinations. However, said resampling scheme would be more consistent with a

model where year-location combo was a single random effect. The downside of doing it that way from the start is that we lose the ability to evaluate what is most important in explaining the variability, year or location.

The main question remains. What is the full parametric procedure that BB suggested? How do I resample before re-fitting? What do I resample? Does that actually mean, at each iteration

1. simulate data from the fitted model
2. re-estimate the fitted model based on simulated data
3. get the intercept value

The variance on these intercepts is the variance I want, which accounts for variability in random effect estimation.

This seems circular to me, but maybe that is what one is supposed to do?

Aknowledgements

We thank Ben Bolker for helpful advice via the answer to our question on “Stack Exchange” at <https://stats.stackexchange.com/questions/616697/how-to-estimate-precision-on-a-prediction-for-a-glmm-for-an-unobserved-level-of/>