

# Extinction risk from climate change: Dispersal modes

March 25, 2024"

## Load libraries and data

```
rm(list = ls())
root.dir = "C:/Users/mcu08001/Documents/1New Research/CC MetaRisk2/Analysis"

#Load Libraries
library(coda); library(ggplot2); library(rstan); library(bayesplot); library(
shinystan); library(loo); library(rstanarm); library(dplyr); library(ggpubr)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = FALSE)

#Load data
dataP<-read.table("MetaRisk2 aggthres 5.txt",header=T);
attach(dataP)
#note that do not need to eliminate NAs like for pre-industrial

#betareg requires no 0s or 1s
koffset = 0.001 #the k that gives the best posterior predictive check
percent2 <- adj.percent
percent2[adj.percent == 0] = koffset;
percent2[adj.percent == 1] = 1 - koffset;
dataP$percent2 <- percent2;

data.use<-dataP
N = length(data.use$percent2)
n.Study <- length(unique(data.use$Study)) #number of studies
Studyint<-as.integer(unclass(factor(data.use$Study)))
phi = data.use$Total.N
unique(Disp.Mod)

## [1] "Universal"      "None"           "Fixed"          "SpSpecific"     "Landscape"
"
## [6] "Contiguous"     "Intermediate"
```

```
load("2all_interc.rds")
mod.int = mod
loo.mod.int = loo.mod
```

## Test for effect of dispersal mode

Here I tested if extinction predictions change based on dispersal mode. I predicted that extinction risks would be highest for no dispersal, lowest for universal dispersal, and intermediate for species-specific, fixed, and contiguous dispersal.

```
#create model matrix for coefficients
betamat <- (model.matrix(~Disp.Mod,data=data.use)) #
stan.data<-list(N = N, percent = data.use$percent2, betamat = betamat, phi =
phi, S = n.Study, P = ncol(betamat), Study = Studyint)
params.to.monitor=c("beta","y_rep","stu","sigma_stu", "eta","log_lik")

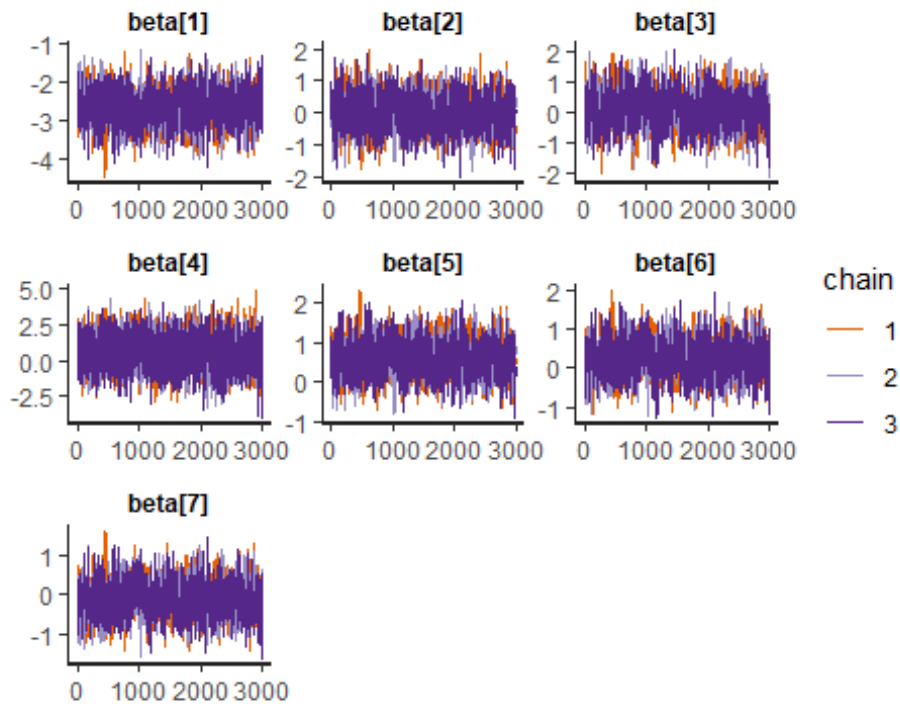
init.beta=rep(0,ncol(betamat)-1)
init.fn<- function (chain_id) {
  list(beta = c(-2.5,init.beta))
}

# mod=stan(file="MetaRisk2 RSTAN beta mat.stan",data=stan.data,pars=params.to
.monitor,
#          chains = 3, warmup=5000, cores=3,iter=8000,
#          init = init.fn, control=list(adapt_delta = 0.9, max_treedepth = 15
))

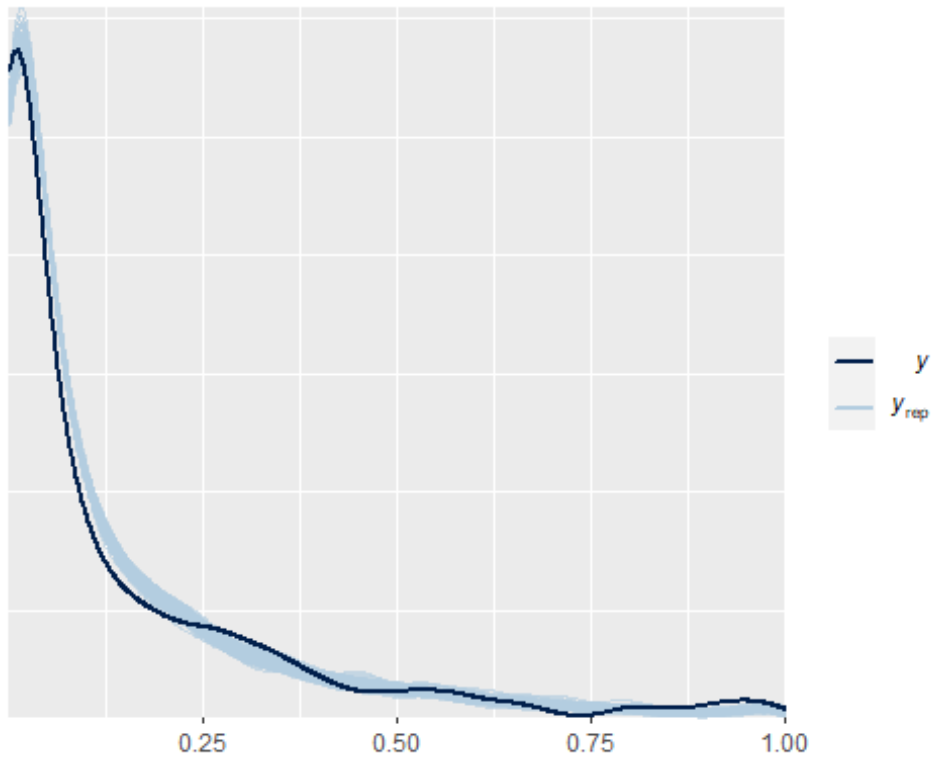
load("2dispmode.rds")
params.to.monitor2=c("beta")#
sumx = summary(mod,probs=c(.025,0.975), digits=4, pars=params.to.monitor2)
sumx$summary

##              mean      se_mean      sd      2.5%      97.5%      n_eff
## beta[1] -2.680321593 0.01365558 0.4416412 -3.5602754 -1.8257942 1045.9687
## beta[2] -0.001345932 0.01474904 0.5308035 -1.0090778 1.0440694 1295.2091
## beta[3] 0.113754770 0.02010610 0.6014701 -1.0216021 1.3269554 894.8953
## beta[4] 0.619456781 0.02155959 1.0876997 -1.5304522 2.7689707 2545.2881
## beta[5] 0.615186459 0.01410491 0.4368561 -0.2369455 1.4708353 959.2600
## beta[6] 0.273851530 0.01451614 0.4486174 -0.5982427 1.1465713 955.1024
## beta[7] -0.093121916 0.01440194 0.4396526 -0.9457437 0.7721975 931.9173
##              Rhat
## beta[1] 1.003791
## beta[2] 1.003251
## beta[3] 1.004081
## beta[4] 1.000794
## beta[5] 1.004323
## beta[6] 1.002750
## beta[7] 1.004252

#checks
traceplot(mod,pars=params.to.monitor2,inc_warmup=FALSE)
```



```
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
  fun = 'dens_overlay'
)
```



```
#calculate loo
# log_lik_1 <- extract_log_lik(mod, merge_chains = FALSE)
# r_eff <- relative_eff(exp(log_lik_1), cores = 6)
# loo.mod <- loo(log_lik_1, r_eff = r_eff, cores = 6)
loo.mod #

##
## Computed from 9000 by 3235 log-likelihood matrix
##
##           Estimate      SE
## elpd_loo    7622.5   96.9
## p_loo       1975.4   25.6
## looic       -15245.0 193.7
## -----
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##               Count Pct.    Min. n_eff
## (-Inf, 0.5] (good)    645  19.9%    368
## (0.5, 0.7] (ok)      780  24.1%    118
## (0.7, 1] (bad)     1545  47.8%     14
## (1, Inf) (very bad)  265   8.2%      3
## See help('pareto-k-diagnostic') for details.

#create data frame of looics from two models
load("2dispmode.rds")
```

```
table.data<-data.frame(
  Model = c("Intercept-only model","Model including model type"),
  LOOic = c(loo.mod.int$estimates[3],loo.mod$estimates[3]),
  SE = c(loo.mod.int$estimates[6],loo.mod$estimates[6])
)
knitr::kable(table.data, caption = "Table 1: Comparisons of LOOic between baseline and model with type", format = "markdown")
```

Table 1: Comparisons of LOOic between baseline and model with type

Model	LOOic	SE
Intercept-only model	-15213.77	193.3243
Model including model type	-15245.01	193.7378

```
Looic.diff = loo.mod$estimates[3] - loo.mod.int$estimates[3]
cat("Difference in LOOic =", Looic.diff)

## Difference in LOOic = -31.24051
```

## Results

Delta LOOIC of -31.2 suggests that model type matters.

```
load("2dispmod.rds")
modx = mod
#cats <- c("Contiguous", "Fixed", "Intermediate", "Landscape", "None", "Taxa-specific", "Universal")
cats <- c("Contiguous", "Fixed", "None", "Taxa-specific", "Universal")
#Calculate estimates; note original is 1 in matrix
posterior=as.data.frame(modx);

n.total <- nrow(data.use) #total N

#calculate risks from beta matrix
beta.1<-posterior[["beta[1]"]]
beta.2<-posterior[["beta[1]"]]+posterior[["beta[2]"]]
beta.3<-posterior[["beta[1]"]]+posterior[["beta[3]"]]
beta.4<-posterior[["beta[1]"]]+posterior[["beta[4]"]]
beta.5<-posterior[["beta[1]"]]+posterior[["beta[5]"]]
beta.6<-posterior[["beta[1]"]]+posterior[["beta[6]"]]
beta.7<-posterior[["beta[1]"]]+posterior[["beta[7]"]]

beta.cat<-cbind(beta.1,beta.2,beta.5,beta.6,beta.7)# too small sample size: beta.3,beta.4,

#use common definition for global median for graph, or else it varies a bit
```

```

posterior2=as.data.frame(mod.int)
grand.mean = posterior2[["mu"]]
grand.mean.pred <- invlogit(quantile(grand.mean, probs = c(0.025, 0.5, 0.975)
))
grand.mean.pred

##          2.5%          50%          97.5%
## 0.06568248 0.07596676 0.08662260

#calculate median and credible intervals
pred.cat = invlogit(apply(beta.cat, 2, quantile, probs = c(0.025, 0.5, 0.975)
,na.rm=TRUE))
pred.cat.df <- data.frame(x = cats,
                          mean = pred.cat[2,],
                          low = pred.cat[1,],
                          hi = pred.cat[3,])

#arrange in increasing order#
pred.cat.df$cats = with (pred.cat.df, reorder(cats, mean))

#calculate risks relative to global median
rbeta.cat = invlogit(beta.cat)-invlogit(grand.mean)
rpred.cat = (apply(rbeta.cat, 2, quantile, probs = c(0.025, 0.5, 0.975),na.rm
=TRUE))
rpred.cat.df <- data.frame(x = cats,
                          mean = rpred.cat[2,],
                          low = rpred.cat[1,],
                          hi = rpred.cat[3,])

#Differences from zero for fig
glob.mean.over <- rep("n",nrow(rpred.cat.df))
glob.mean.over[rpred.cat.df$mean > 0 & rpred.cat.df$low > 0] = "y"
glob.mean.over[rpred.cat.df$mean < 0 & rpred.cat.df$hi < 0] = "y"
rpred.cat.df$glob.mean.over <- glob.mean.over

#arrange in increasing order#
rpred.cat.df$cats = with (rpred.cat.df, reorder(cats, mean))

#show values
pred.cat.r <- pred.cat.df[,1:4]
pred.cat.r[,2:4] <- round((pred.cat.r[,2:4]), digits = 4)
opts <- options(knitr.kable.NA = "")
knitr::kable(pred.cat.r, caption = "Table 2: Extinction risk by dispersal mod
e, with 95% credible intervals", format = "markdown", font = 7, longtable = T
)

```

Table 2: Extinction risk by dispersal mode, with 95% credible intervals

	x	mean	low	hi
beta.1	Contiguous	0.0644	0.0276	0.1387
beta.2	Fixed	0.0641	0.0354	0.1120
beta.5	None	0.1125	0.0967	0.1305
beta.6	Taxa-specific	0.0827	0.0662	0.1020
beta.7	Universal	0.0588	0.0505	0.0682

```
rpred.cat.r <- rpred.cat.df[,1:5]
rpred.cat.r[,2:4] <- round((rpred.cat.r[,2:4]), digits = 4)
opts <- options(knitr.kable.NA = "")
knitr::kable(rpred.cat.r, caption = "Table 3: Relative extinction risk by dispersal mode, with 95% credible intervals", format = "markdown", font = 7, longtable = T)
```

Table 3: Relative extinction risk by dispersal mode, with 95% credible intervals

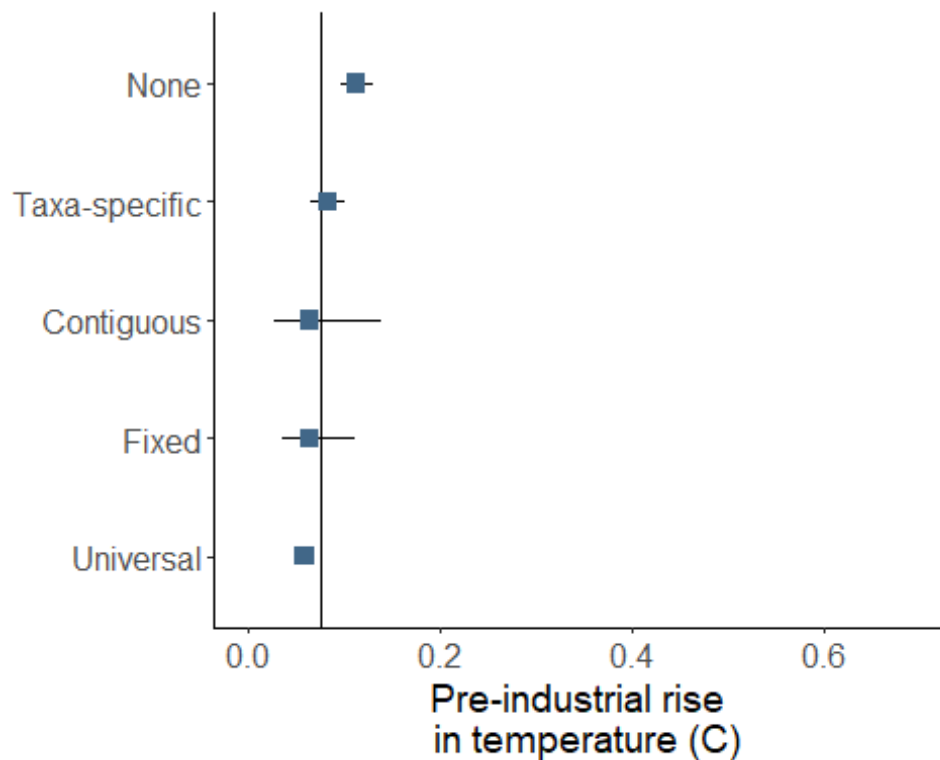
	x	mean	low	hi	glob.mean.over
beta.1	Contiguous	-0.0111	-0.0507	0.0626	n
beta.2	Fixed	-0.0117	-0.0431	0.0371	n
beta.5	None	0.0367	0.0175	0.0570	y
beta.6	Taxa-specific	0.0069	-0.0130	0.0281	n
beta.7	Universal	-0.0171	-0.0313	-0.0030	y

## Create figures

```
#Figures
Fig1a<-ggplot(data = pred.cat.df)+
  geom_vline(xintercept=grand.mean.pred[2]) +
  geom_errorbar(aes(y = cats, xmin = low, xmax = hi), width = 0) +
  geom_point(stat = "identity", aes(y = cats, x = mean), color = "#416788", size = 3, shape = 15) +
  xlab("Pre-industrial rise \n in temperature (C)") + xlim(c(0,.7)) +
  theme_classic()+
  theme(axis.title.y = element_blank(),axis.title=element_text(size=14),axis.text = element_text(size=12))+
  guides(size=F)
```

```
## Warning: The `scale` argument of `guides()` cannot be `FALSE`. Use "none"
## instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

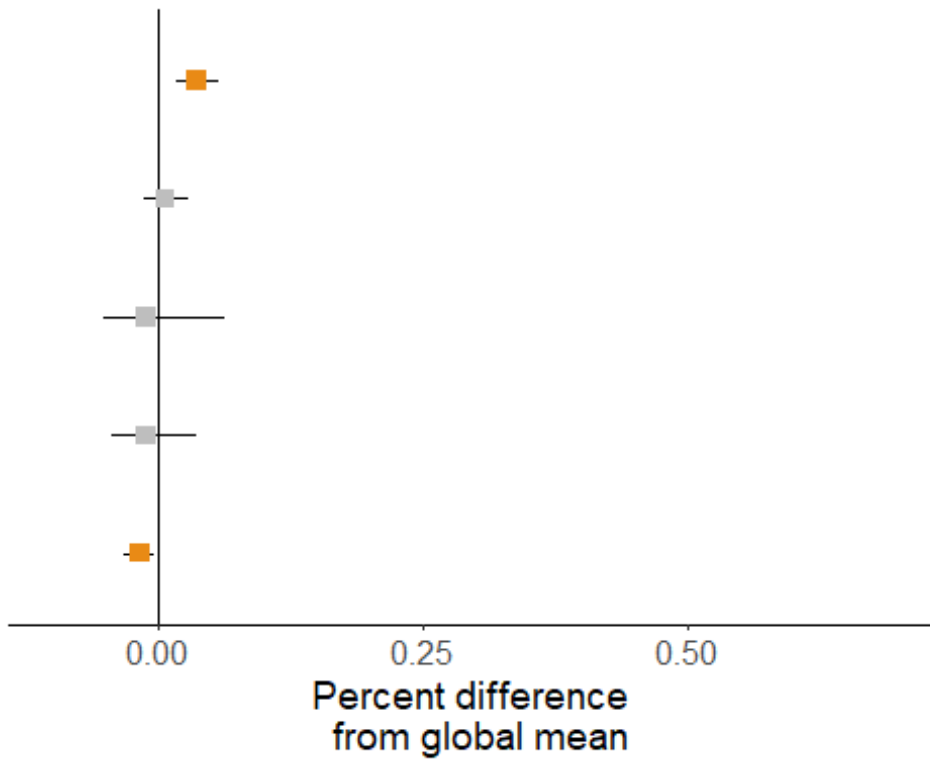
Fig1a



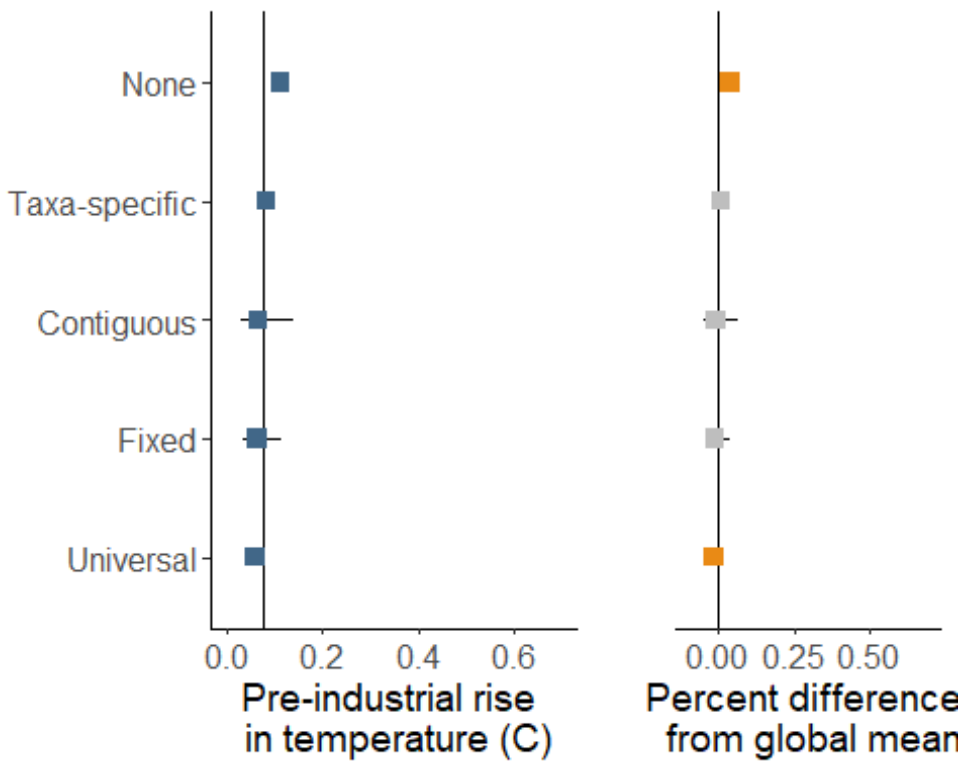
```
Fig1b<-ggplot(data = rpred.cat.df)+
  geom_vline(xintercept=0) +
  geom_errorbar(aes(y = cats, xmin = low, xmax = hi), width = 0) +
  geom_point(stat = "identity", aes(y = cats, x = mean, color = glob.mean.ove
r), size = 3, shape = 15) +
  scale_color_manual(values=c('grey', '#E98a15'))+
  xlab("Percent difference \n from global mean") + xlim(c(-.1,.7)) +
  theme_classic()+
  theme(axis.title.y = element_blank(),axis.ticks.y = element_blank(),axis.li
ne.y = element_blank(),axis.text.y = element_blank(),
        axis.title=element_text(size=14),axis.text = element_text(size=12),le
gend.position = "none")+
  guides(size=F)
```

Fig1b





```
ggarrange(Fig1a, NULL, Fig1b, ncol=3, widths = c(4,.4, 2))
```



```
#ggsave("Metarisk2 dispersal.png",width=4,height=3,unit="in",dpi="print")
```

## Conclusion

Dispersal affects extinction risks. As predicted, extinction risks were reduced when universal dispersal was assumed and increased when no dispersal was assumed.

```
#catalog sample sizes
cats2 <- c("Contiguous", "Fixed", "Intermediate", "Landscape", "None", "Taxa-
specific", "Universal")
N.1.st<-length(unique(dataP$Study[dataP$Disp.Mod == "Contiguous"]))
N.1.mod<-length((dataP$Study[dataP$Disp.Mod == "Contiguous"]))
#
N.2.st<-length(unique(dataP$Study[dataP$Disp.Mod == "Fixed"]))
N.2.mod<-length((dataP$Study[dataP$Disp.Mod == "Fixed"]))

N.3.st<-length(unique(dataP$Study[dataP$Disp.Mod == "Intermediate"]))
N.3.mod<-length((dataP$Study[dataP$Disp.Mod == "Intermediate"]))

N.4.st<-length(unique(dataP$Study[dataP$Disp.Mod == "Landscape"]))
N.4.mod<-length((dataP$Study[dataP$Disp.Mod == "Landscape"]))

N.5.st<-length(unique(dataP$Study[dataP$Disp.Mod == "None"]))
N.5.mod<-length((dataP$Study[dataP$Disp.Mod == "None"]))

N.6.st<-length(unique(dataP$Study[dataP$Disp.Mod == "SpSpecific"]))
N.6.mod<-length((dataP$Study[dataP$Disp.Mod == "SpSpecific"]))

N.7.st<-length(unique(dataP$Study[dataP$Disp.Mod == "Universal"]))
N.7.mod<-length((dataP$Study[dataP$Disp.Mod == "Universal"]))

table.data<-data.frame(
  Factor = cats2,
  Studies = c(N.1.st,N.2.st,N.3.st,N.4.st,N.5.st,N.6.st, N.7.st),
  Models = c(N.1.mod,N.2.mod,N.3.mod,N.4.mod,N.5.mod,N.6.mod, N.7.mod)
)
knitr::kable(table.data, caption = "Table 4: Number of studies and models for
each factor", format = "markdown")
```

Table 4: Number of studies and models for each factor

Factor	Studies	Models
Contiguous	8	15
Fixed	11	319
Intermediate	2	13
Landscape	3	6
None	226	1091
Taxa-specific	69	436
Universal	350	1355

## Variation explained

```
#After Gelman 2019 R2 for Bayesian
#
#Load model and beta matrix - check if mu is modeled separately
load("2dispmod.rds")
posterior=as.data.frame(mod);
betamat <- (model.matrix(~Disp.Mod,data=data.use)) #

#posterior=as.data.frame(mod.int);
#betamat = as.matrix(rep(1,9000))

#Variables and matrices
S = 9000; #samples
K = ncol(betamat); #factors
p.mat <- as.matrix(posterior[,1:K])
y = dataP$percent2
y.mat = t(matrix(rep(y,S), nrow = N, ncol = S))
y.mean <- mean(y)

#Calculate y.pred for fixed effects only
y.pred <- matrix(rep(NA, N*S), nrow = S, ncol = N)
theta <- y.pred
for (i in 1:N) {
  theta[,i] = invlogit(p.mat %*% betamat[i,])#rows = samples, cols = i
  y.pred[,i] = (theta[,i] * data.use$Total.N[i])/(theta[,i] * data.use$Total.N[i] + (1-theta[,i]) * data.use$Total.N[i])
}

#Calcluate residual variance
res.f = y.mat - y.pred
RSS.f = rowSums((res.f)^2)
res.v.f = 1/(N -1) * RSS.f

#Calculate fit variance
pred.v.f = 1/(N-1) * rowSums((y.pred)^2)

#Calculate R2
R2.v.f = pred.v.f/(pred.v.f + res.v.f)
cat("fixed effects R2 = ", quantile(R2.v.f,probs = c(0.025, 0.5, 0.975),na.rm = T))

## fixed effects R2 = 0.07747646 0.102867 0.1354669

#Total model With random effects
y.pred.c <- (as.matrix(posterior[, (K+1):(N+K)])) #calculated in STAN, with all RE and weightings
```

```

#Calculate residual variance
res.c = y.mat - y.pred.c
RSS.c = rowSums((res.c)^2)
res.v.c = 1/(N-1) * RSS.c

#Calculate fit variance

pred.v.c = 1/(N-1) * rowSums(y.pred.c^2)

#Calculate full model R2
R2.v.c = pred.v.c/(pred.v.c + res.v.c)
print(" ")
## [1] " "

cat("Overall model R2 = ", quantile(R2.v.c, probs = c(0.025, 0.5, 0.975), na.rm
= T))

## Overall model R2 = 0.7686331 0.7893581 0.8095618

```