

# Extinction risk from climate change: Model types

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)

#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
```

## Test for effect of model type

Here I tested if extinction predictions change based on model type.

```
#create model matrix for coefficients
#combine expert and CCVA because only 2 expert models
data.use$Model.Type =ifelse(data.use$Model.Type == "Expert", "CCVA/Expert", dat
a.use$Model.Type)
data.use$Model.Type =ifelse(data.use$Model.Type == "CCVA", "CCVA/Expert", data.
use$Model.Type)

#create model matrix for coefficients
betamat <- model.matrix(~Model.Type, data = data.use) # #create model matrix
1 - CCVA, 2 - expert, 3 - hybrid, 4 - mechanistic, 5 - SAR, 6 - SDM
```

```

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))
}

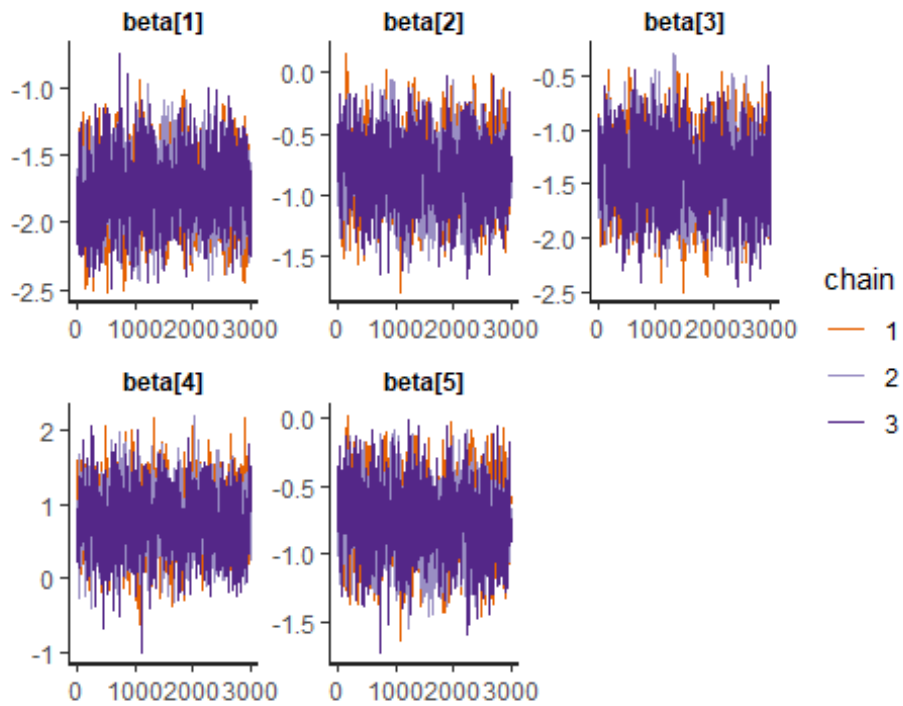
# mod=stan(file="MetaRisk2 RSTAN beta mat.stan",data=stan.data,pars=params.to
.monitor,
#          chains = 3, warmup=5000, cores=3,iter=8000,
#          init = init.fn, control=list(adapt_delta = 0.9, max_treedepth = 15
))

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

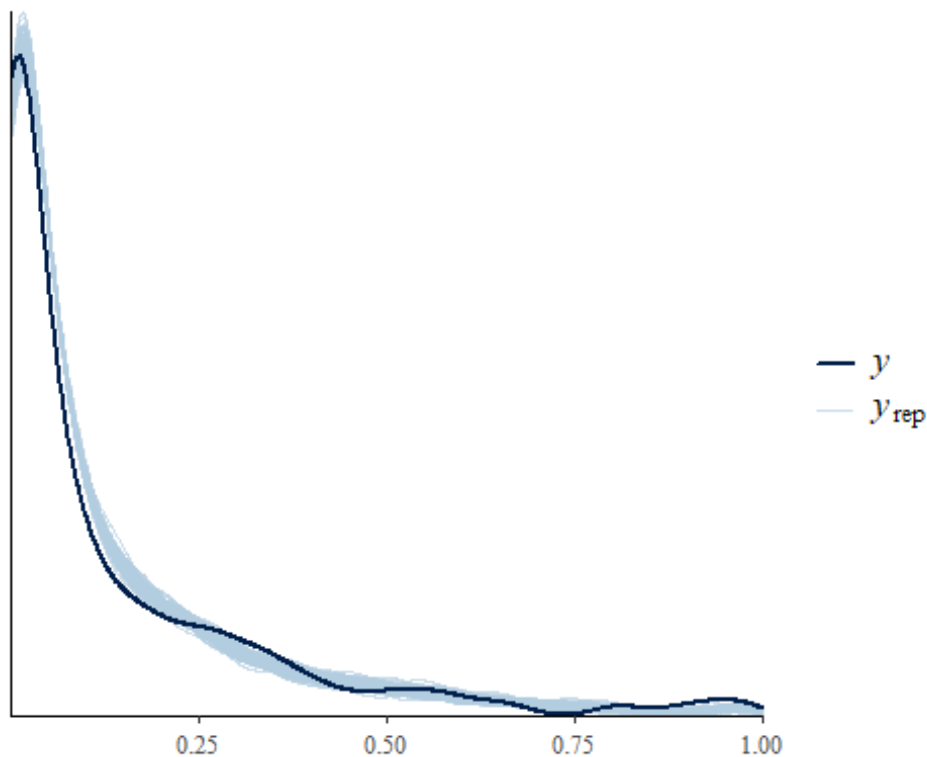
##              mean      se_mean      sd      2.5%      97.5%      n_eff
## beta[1] -1.7661192 0.007196151 0.2386452 -2.22147782 -1.2858741 1099.7775
## beta[2] -0.7999735 0.008271636 0.2584639 -1.33331071 -0.3067969  976.3758
## beta[3] -1.3898398 0.008151574 0.3002158 -1.98310721 -0.8110390 1356.3886
## beta[4]  0.7951301 0.012716526 0.3954273  0.02775297  1.5522717  966.9327
## beta[5] -0.7467022 0.007071374 0.2381908 -1.23081117 -0.2885844 1134.5990
##              Rhat
## beta[1] 1.007166
## beta[2] 1.007067
## beta[3] 1.006671
## beta[4] 1.006511
## beta[5] 1.007426

#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    7615.3   96.8
## p_loo       1982.5   24.8
## looic      -15230.6 193.5
## -----
## MCSE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##               Count Pct.    Min. ESS
## (-Inf, 0.7] (good)   1409  43.6%   127
## (0.7, 1]   (bad)    1560  48.2%   <NA>
## (1, Inf)   (very bad) 266   8.2%   <NA>
## See help('pareto-k-diagnostic') for details.
```

```
#create data frame of looics from two models
load("2modelc.rds")
loo.mod2=loo.mod # rename loo.mod so can load n
mod2 = mod
```

```
load("2all_interc.rds")

table.data<-data.frame(
  Model = c("Intercept-only model","Model including model type"),
  LOOic = c(loo.mod$estimates[3],loo.mod2$estimates[3]),
  SE = c(loo.mod$estimates[6],loo.mod2$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	-15230.61	193.5366

```
Loaic.diff = loo.mod2$estimates[3] - loo.mod$estimates[3]
cat("Difference in LOOic =", Loaic.diff)

## Difference in LOOic = -16.8415
```

## Results

Delta LOOIC of -16.6 suggests that model type matters.

```
load("2modelc.rds")
modx = mod
cats <- c("CCVA/Expert", "Hybrid", "Mechanistic", "SAR", "SDM")
#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.cat<-cbind(beta.1,beta.2,beta.3,beta.4,beta.5)

#use common definition for global median for graph, or else it varies a bit
load("2all_interc.rds")
```

```

posterior2=as.data.frame(mod)
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.df

##          x          mean          low          hi          cats
## beta.1 CCVA/Expert 0.14554629 0.09783829 0.21655198 CCVA/Expert
## beta.2 Hybrid 0.07129339 0.05686129 0.08960667 Hybrid
## beta.3 Mechanistic 0.04092374 0.02829563 0.05886825 Mechanistic
## beta.4 SAR 0.27442480 0.16717005 0.41650011 SAR
## beta.5 SDM 0.07500361 0.06537937 0.08596596 SDM

rpred.cat.df

```

	x	mean	low	hi	glob.mean.over
## beta.1	CCVA/Expert	0.069600666	0.01974798	0.14076642	y
## beta.2	Hybrid	-0.004653215	-0.02301703	0.01630619	n
## beta.3	Mechanistic	-0.035025882	-0.05158843	-0.01461889	y
## beta.4	SAR	0.198616215	0.09105620	0.34132388	y
## beta.5	SDM	-0.001054976	-0.01518890	0.01439341	n

## cats

## beta.1 CCVA/Expert

## beta.2 Hybrid

## beta.3 Mechanistic

## beta.4 SAR

## beta.5 SDM

### #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", s
ize = 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

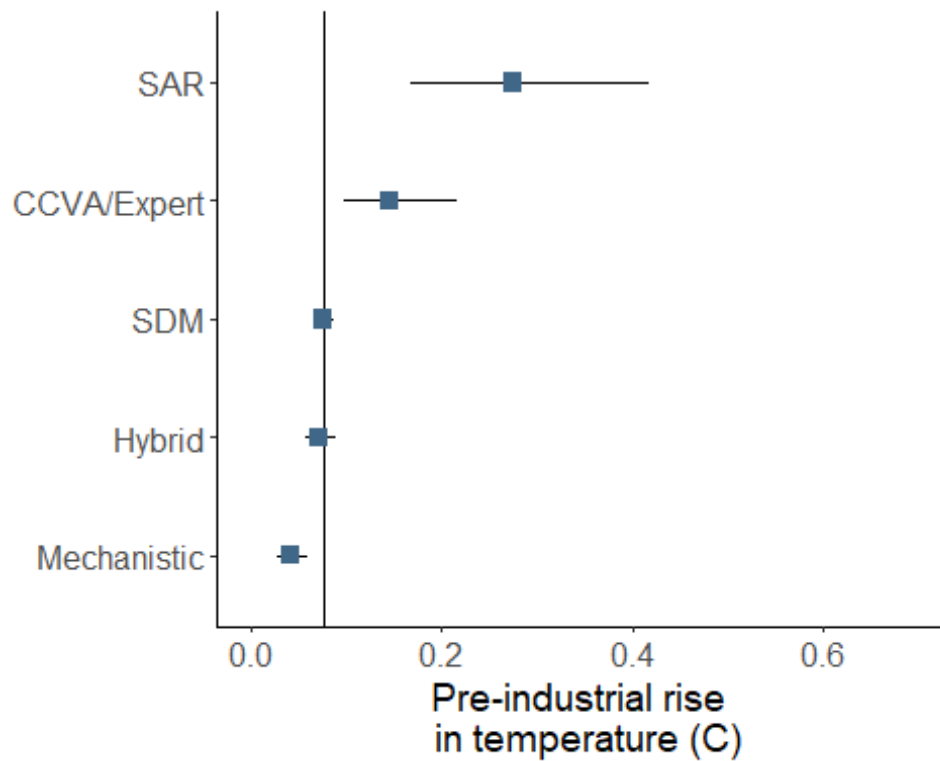
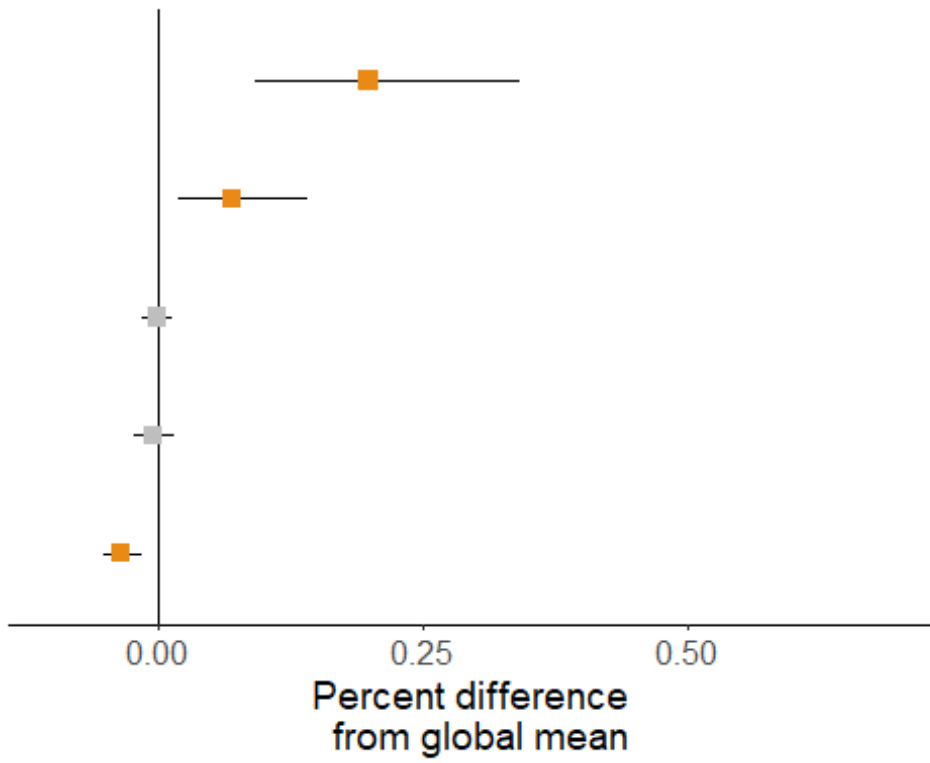


Fig. 1. Predicted extinction risk based on model type

```
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
```





*Fig. 1. Predicted extinction risk based on model type*

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

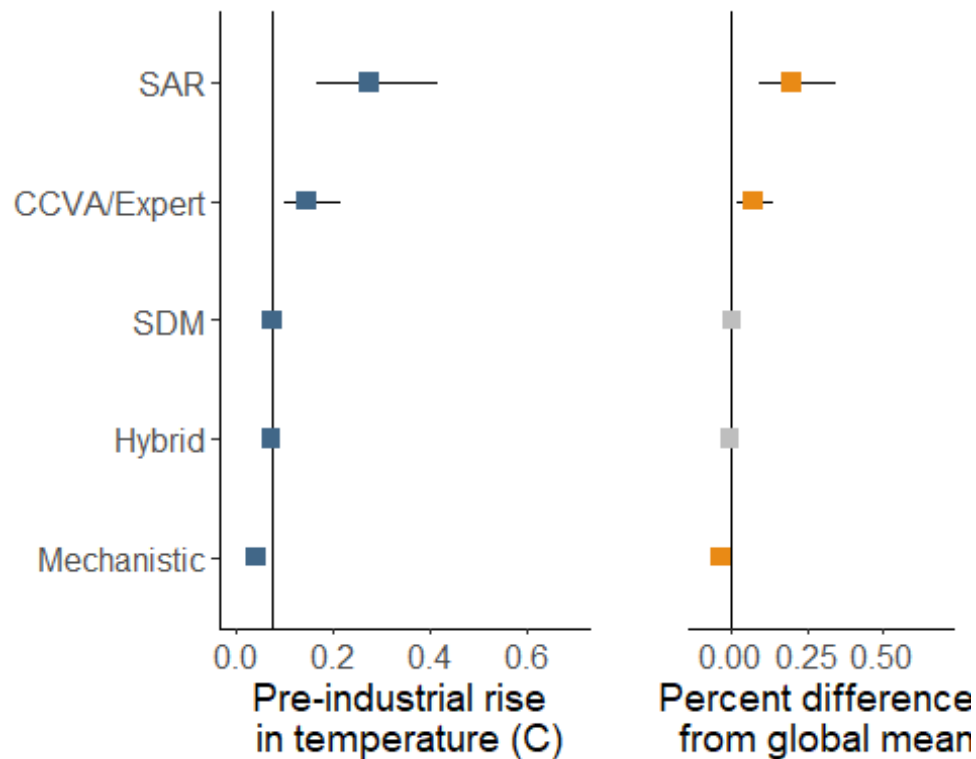


Fig. 2. Predicted extinction risk based on model type

```
#ggsave("Metarisk2_taxa.png",width=4,height=3,unit="in",dpi="print")
```

## Conclusion

Model types matter for extinction risks. Mechanistic models tend to estimate lower risks while CCVA and SAR models estimate higher risks.

```
#catalog sample sizes
N.1.st<-length(unique(dataP$Study[dataP$Model.Type == "CCVA" | dataP$Model.Type == "Expert"]))
N.1.mod<-length((dataP$Study[dataP$Model.Type == "CCVA" | dataP$Model.Type == "Expert"]))
#
# N.2.st<-length(unique(dataP$Study[dataP$Model.Type == "Expert"]))
# N.2.mod<-length((dataP$Study[dataP$Model.Type == "Expert"]))

N.3.st<-length(unique(dataP$Study[dataP$Model.Type == "Hybrid"]))
N.3.mod<-length((dataP$Study[dataP$Model.Type == "Hybrid"]))

N.4.st<-length(unique(dataP$Study[dataP$Model.Type == "Mechanistic"]))
N.4.mod<-length((dataP$Study[dataP$Model.Type == "Mechanistic"]))
```

```

N.5.st<-length(unique(dataP$Study[dataP$Model.Type == "SAR"]))
N.5.mod<-length((dataP$Study[dataP$Model.Type == "SAR"]))

N.6.st<-length(unique(dataP$Study[dataP$Model.Type == "SDM"]))
N.6.mod<-length((dataP$Study[dataP$Model.Type == "SDM"]))

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

```

*Table 2: Number of studies and models for each factor*

Factor	Studies	Models
CCVA/Expert	23	155
Hybrid	61	345
Mechanistic	46	266
SAR	13	105
SDM	391	2364

```

#combine expert and CCVA
dataP$Model.Type2 <- ifelse(dataP$Model.Type == "CCVA", "Expert", Model.Type)

nspecies <- dataP %>%
  group_by(Model.Type2) %>%
  summarize(median = median(Total.N), mean = mean(Total.N))

knitr::kable(nspecies, caption = "Table 3: Mean and median number of species
for each approach", format = "markdown")

```

*Table 3: Mean and median number of species for each approach*

Model.Type2	median	mean
Expert	28	52909.1161
Hybrid	30	965.8870
Mechanistic	6	664.2105
SAR	192	85785.0667
SDM	30	1580.2470

## Variation explained

```
#After Gelman 2019 R2 for Bayesian
#
#Load model and beta matrix - check if mu is modeled separately
load("2modelc.rds")
posterior=as.data.frame(mod);
data.use$Model.Type =ifelse(data.use$Model.Type == "Expert", "CCVA/Expert", data.use$Model.Type)
data.use$Model.Type =ifelse(data.use$Model.Type == "CCVA", "CCVA/Expert", data.use$Model.Type)

#create model matrix for coefficients
betamat <- model.matrix(~Model.Type, data = data.use)

#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])
}

#Calculate 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 = 0.08309257 0.1156264 0.1585276
```

```

#Total model With random effects
y.pred.c <-(as.matrix(posterior[, (K+1):(3235+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)
cat("Overall model R2 = ", quantile(R2.v.c, probs = c(0.025, 0.5, 0.975), na.rm
= T))

## Overall model R2 = 0.7662082 0.7867021 0.8068025

```