Extinction risk from climate change: through time

Feb. 27, 2024

Test for effect of time of study on results

Here I wanted to test to see if the year of study affected predictions. I split the data into original data from the 2015 analysis and new data. I also included an interaction term in case the slopes differ.

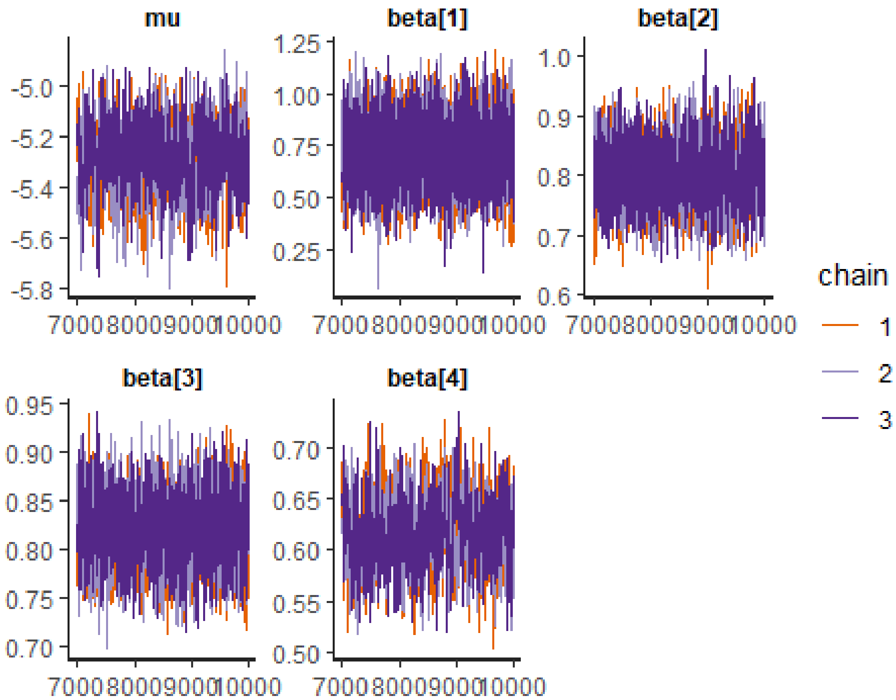
Load libraries and data

```
rm(list = ls())
 root.dir = "C:/Users/mcu08001/Documents/1New Research/CC MetaRisk2/Analysis"
# Load libraries and data
library(coda); library(ggplot2); library(rstan); library(bayesplot); library(
loo); library(rstanarm);
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = FALSE)

dataP<-read.table("Metarisk2 aggthres 5.txt",header=T); #newest data with sli
ght fix to dispersal data
dataP2<-dataP[is.finite(dataP$Pre.Ind.Rise),]; attach(dataP2) # need to elimi
nate NA s for pre-industrial rise or stat programs crash

#Bayesian stan model proportional and weighted
#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;
dataP2$percent2 <- percent2;

#create categorical factor of era ~ 90s, 00s, 10s, 20s
dataP2$era <- ifelse(Year < 2000, "1990s",
                     ifelse(Year < 2010, "2000s",
                            ifelse(Year < 2020, "2010s", "2020s")))
data.use<-dataP2

#Prepare data for models
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
```

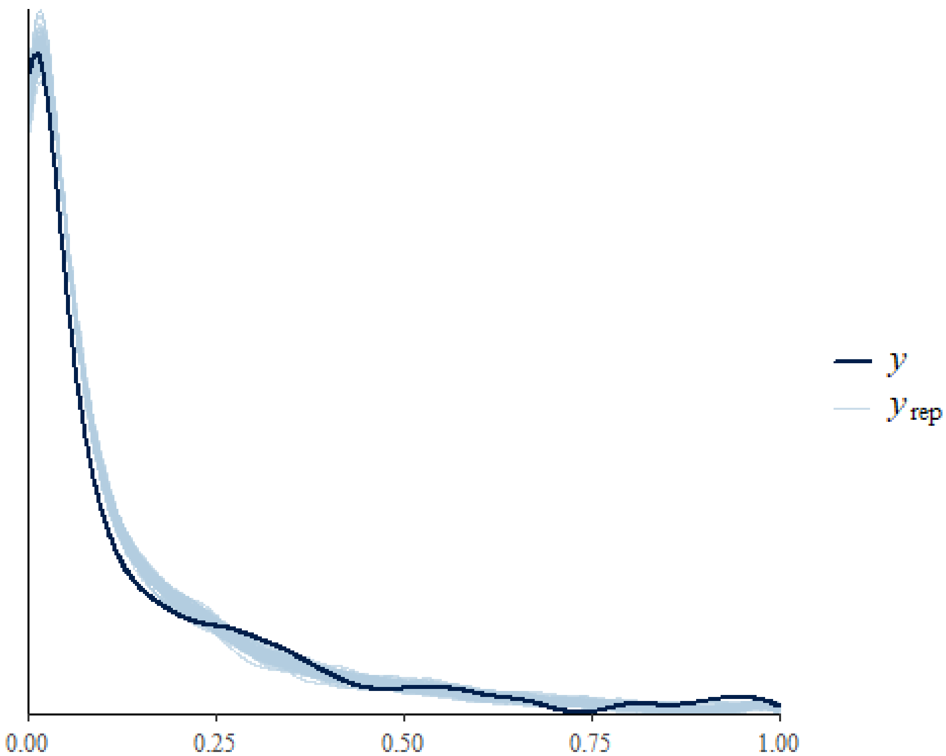
Test for effect of decade

Same-intercept model

```
#create model matrix for coefficients
betamat<-(model.matrix(~era * Pre.Ind.Rise,data=data.use))[,2:8]   #create mo
del matrix, exclude intercept which is modeled separately as mu (to allow for
informed prior)

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("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")
#
# init.fn<- function (chain_id) {
#   list(mu = -5, beta = c(0,.6,0))
# }

# mod=stan(file="MetaRisk2 RSTAN quad.stan",data=stan.data,pars=params.to.mon
itor,
#           chains = 3, warmup=5000, cores=7,iter=8000,
#          init = init.fn, control=list(adapt_delta = 0.9, max_treedepth = 15)
)

load("3timecat.rds") #minor fixes to years of publications
params.to.monitor2=c("mu","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
## mu      -5.2880314 0.0054606458 0.13127231 -5.5654559 -5.0386555  577.9074
## beta[1]  0.7260591 0.0025550184 0.14306933  0.4473888  1.0081403 3135.4870
## beta[2]  0.8011578 0.0013396310 0.04914807  0.7071176  0.8985790 1345.9924
## beta[3]  0.8199752 0.0009176027 0.03378795  0.7538011  0.8871321 1355.8583
## beta[4]  0.6161399 0.0014671797 0.03331091  0.5515311  0.6824492  515.4734
##             Rhat
## mu      1.004548
## beta[1] 1.000298
## beta[2] 1.000409
## beta[3] 1.000835
## beta[4] 1.007903

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




```r
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 3220 log-likelihood matrix.
##
##           Estimate    SE
## elpd_loo    7696.8  95.7
## p_loo       1858.4  24.9
## looic     -15393.5 191.4
## ------
## MCSE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. ESS
## (-Inf, 0.7]   (good)     1569  48.7%   114
##    (0.7, 1]   (bad)      1436  44.6%   <NA>
##    (1, Inf)   (very bad)  215   6.7%   <NA>
## See help('pareto-k-diagnostic') for details.

load("3timecat.rds")
loo.mod2=loo.mod # rename loo.mod so can load base model

load("2pre_lowb.rds") #load base model
```

```
loo.mod1=loo.mod

#create data frame of looics from two models
table.data<-data.frame(
  Model = c("Base model","Model with decade"),
  LOOic = c(loo.mod1$estimates[3],loo.mod2$estimates[3]),
  SE = c(loo.mod1$estimates[6],loo.mod2$estimates[6])
)
knitr::kable(table.data, caption = "Comparisons of LOOic between baseline and
Model with decade", format = "markdown")
```

Comparisons of LOOic between baseline and Model with decade

Model	LOOic	SE
Base model	-15440.81	190.9764
Model with decade	-15393.54	191.3967

```
Looic.diff = loo.mod2$estimates[3] - loo.mod1$estimates[3]
table.data

##               Model    LOOic       SE
## 1        Base model -15440.81 190.9764
## 2 Model with decade -15393.54 191.3967

cat("Delta LoOic = ", Looic.diff)

## Delta LoOic =  47.27183

#Prediction range
P.Ind<-seq(from = 0.4, to = 5.5,by = .1) #don't extrapolate rise  min(Pre.Ind
.Rise)= 0.4

load("3timecat.rds")
modx = mod
load("2pre_lowb.rds")

#Calculate estimates; note original is 1 in matrix
posterior=as.data.frame(modx);
a.all <- posterior[["mu"]]
b.90 <- posterior[["beta[1]"]]
b.00 <- posterior[["beta[2]"]]
b.10 <- posterior[["beta[3]"]]
b.20 <- posterior[["beta[4]"]]

#For each decade
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {a.all[x] + b.90[
x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
```

```
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_90 = pred.reg.quant[2,],
                          low_line_90 = pred.reg.quant[1,],
                          hi_line_90= pred.reg.quant[3,])

pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {a.all[x] + b.00[
x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_00 = pred.reg.quant[2,]
pred.reg.df$low_line_00 = pred.reg.quant[1,]
pred.reg.df$hi_line_00= pred.reg.quant[3,]

pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {a.all[x] + b.10[
x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_10 = pred.reg.quant[2,]
pred.reg.df$low_line_10 = pred.reg.quant[1,]
pred.reg.df$hi_line_10= pred.reg.quant[3,]

pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {a.all[x] + b.20[
x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_20 = pred.reg.quant[2,]
pred.reg.df$low_line_20 = pred.reg.quant[1,]
pred.reg.df$hi_line_20= pred.reg.quant[3,]

Fig1<-ggplot(data = pred.reg.df)+
      geom_ribbon(aes(x=P.Ind,ymin=low_line_90,ymax=hi_line_90),alpha=.7,fill
="#416788")+
  geom_line(aes(x=P.Ind,y=mean_line_90),size=1.5,color="#416788")+
      geom_ribbon(aes(x=P.Ind,ymin=low_line_00,ymax=hi_line_00),alpha=.7,fill
="#5b507b")+
  geom_line(aes(x=P.Ind,y=mean_line_00),size=1.5,color="#5b507b")+
      geom_ribbon(aes(x=P.Ind,ymin=low_line_10,ymax=hi_line_10),alpha=.7,fill
="#5b307b")+
  geom_line(aes(x=P.Ind,y=mean_line_10),size=1.5,color="#5b307b")+
  geom_ribbon(aes(x=P.Ind,ymin=low_line_20,ymax=hi_line_20),alpha=.5,fill="#5
b000b")+
  geom_line(aes(x=P.Ind,y=mean_line_20),size=1.5,color="#5b000b")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+
  theme(axis.title=element_text(size=18),title=element_text(size=20),axis.tex
t = element_text(size=16))+
```

```
  guides(size="none")
Fig1
```

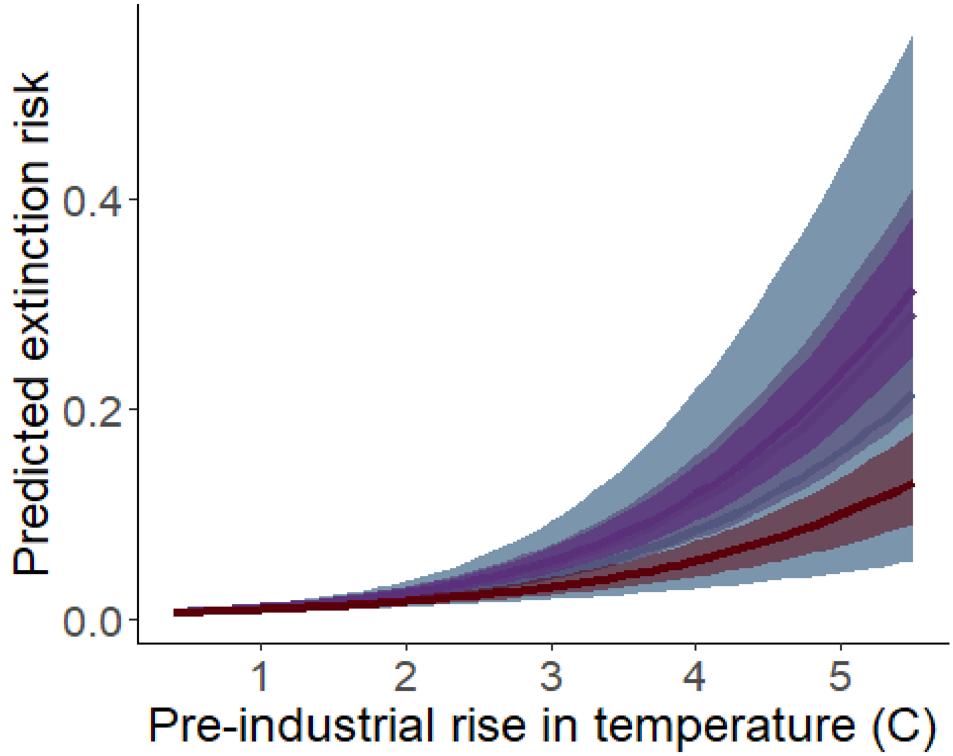
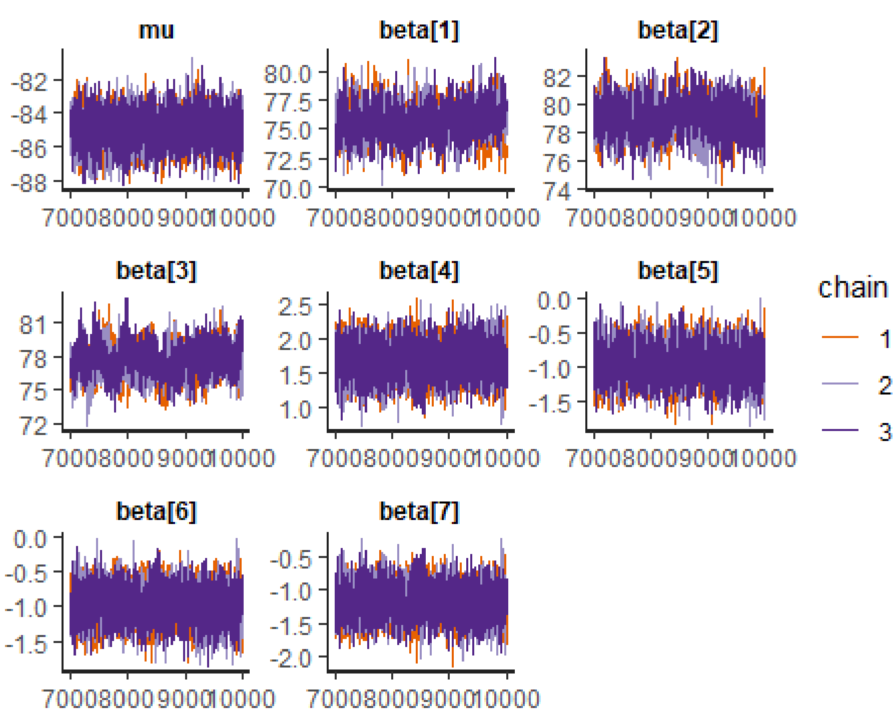


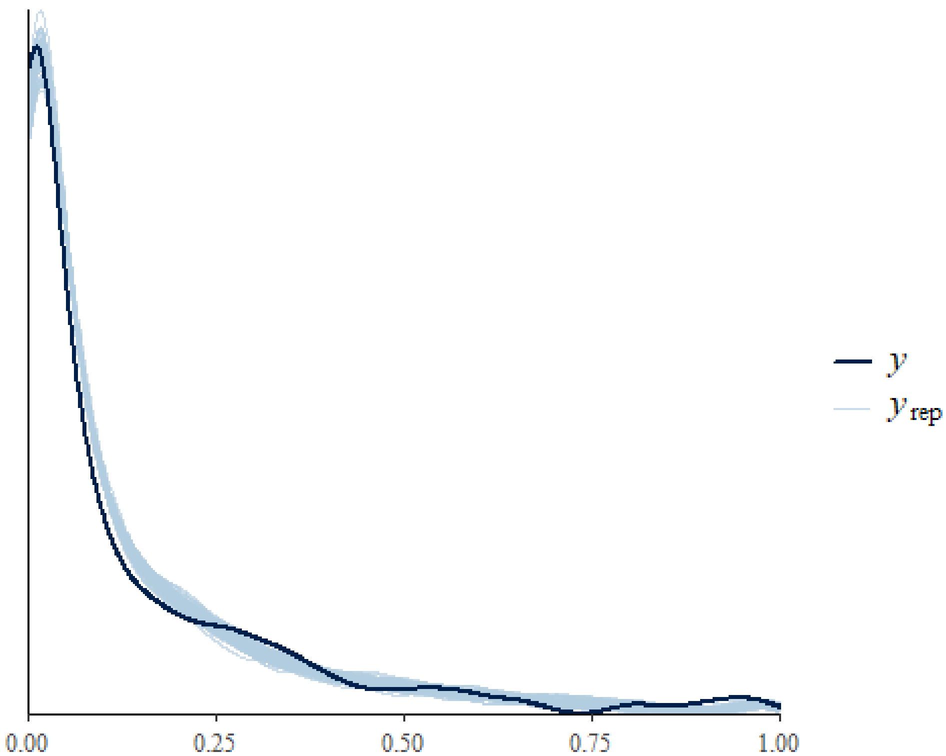
Fig. Predicted extinction risk versus pre-industrial temperature rise across decades.

```
#ggsave("FigSx pre by time2.png",width=8,height=6,unit="in",dpi="print")

#create model matrix for coefficients
betamat<-(model.matrix(~era * Pre.Ind.Rise,data=data.use))[,2:8]   #create mo
del matrix, exclude intercept which is modeled separately as mu (to allow for
informed prior)

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("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")

init.fn<- function (chain_id) {
  list(mu = -5, beta = c(0,.6,0))
}

# mod=stan(file="MetaRisk2 RSTAN quad.stan",data=stan.data,pars=params.to.mon
itor,
#          chains = 3, warmup=5000, cores=7,iter=8000,
#         init = init.fn, control=list(adapt_delta = 0.9, max_treedepth = 15)
)
```

```r
load("2timecat.rds")
params.to.monitor2=c("mu","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
## mu       -84.9961345 0.022471307 0.9980861 -86.951943 -83.0749515 1972.7834
## beta[1]   75.6887125 0.063825635 1.4795180  72.873159  78.5681504  537.3413
## beta[2]   78.9427324 0.069128527 1.1989960  76.592042  81.3684189  300.8298
## beta[3]   77.6286423 0.089559967 1.3870857  74.921892  80.4000343  239.8716
## beta[4]    1.6669894 0.005723864 0.2471274   1.198919   2.1645435 1864.0744
## beta[5]   -0.9289795 0.005845043 0.2530085  -1.432143  -0.4502896 1873.6781
## beta[6]   -0.9689136 0.005810456 0.2508352  -1.474347  -0.4929148 1863.6157
## beta[7]   -1.1604418 0.005946263 0.2500400  -1.660770  -0.6844859 1768.1981
##              Rhat
## mu       1.002339
## beta[1] 1.007874
## beta[2] 1.005096
## beta[3] 1.009684
## beta[4] 1.003703
## beta[5] 1.003548
## beta[6] 1.003760
## beta[7] 1.004610

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



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




```r
#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 3220 log-likelihood matrix.
##
##          Estimate    SE
## elpd_loo   7621.1  96.4
## p_loo      1932.1  30.2
## looic    -15242.3 192.9
## ------
## MCSE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. ESS
## (-Inf, 0.7]   (good)     1589  49.3%   113
##    (0.7, 1]   (bad)      1379  42.8%   <NA>
```

```
##    (1, Inf)   (very bad)  252   7.8%   <NA>
## See help('pareto-k-diagnostic') for details.
```

```
load("2timecat.rds")
loo.mod2=loo.mod # rename loo.mod so can load base model
load("2pre_lowb.rds") #load base model

#create data frame of looics from two models
table.data<-data.frame(
  Model = c("Base model","Model with decade (full)"),
  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 = "Comparisons of LOOic between baseline and
Full Model with decade", format = "markdown")
```

Comparisons of LOOic between baseline and Full Model with decade

Model	LOOic	SE
Base model	-15440.81	190.9764
Model with decade (full)	-15242.28	192.8742

```
Looic.diff = loo.mod2$estimates[3] - loo.mod$estimates[3]
print(Looic.diff)

## [1] 198.5329
```

Previous study versus current study

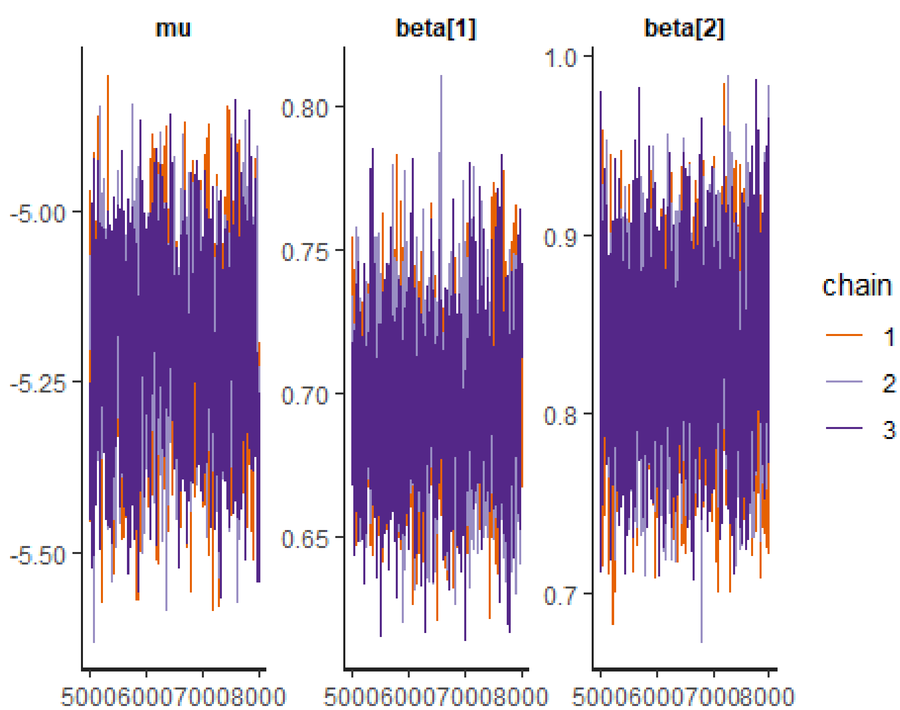
Interactive time model LOOic is larger than additive time model, delta LOOic = +6.0. Also, the interaction term overlaps with zero. No support for interactive model.

```
#create model matrix for coefficients
betamat<-(model.matrix(~Time:Pre.Ind.Rise,data=data.use))[,2:3]  #create mode
l matrix, exclude intercept which is modeled separately as mu (to allow for i
nformed prior)

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("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")

init.fn<- function (chain_id) {
  list(mu = -5, beta = c(0,.6,0))
}

# mod=stan(file="MetaRisk2 RSTAN quad.stan",data=stan.data,pars=params.to.mon
itor,
#           chains = 3, warmup=5000, cores=7,iter=8000,
#          init = init.fn, control=list(adapt_delta = 0.9, max_treedepth = 15)
```

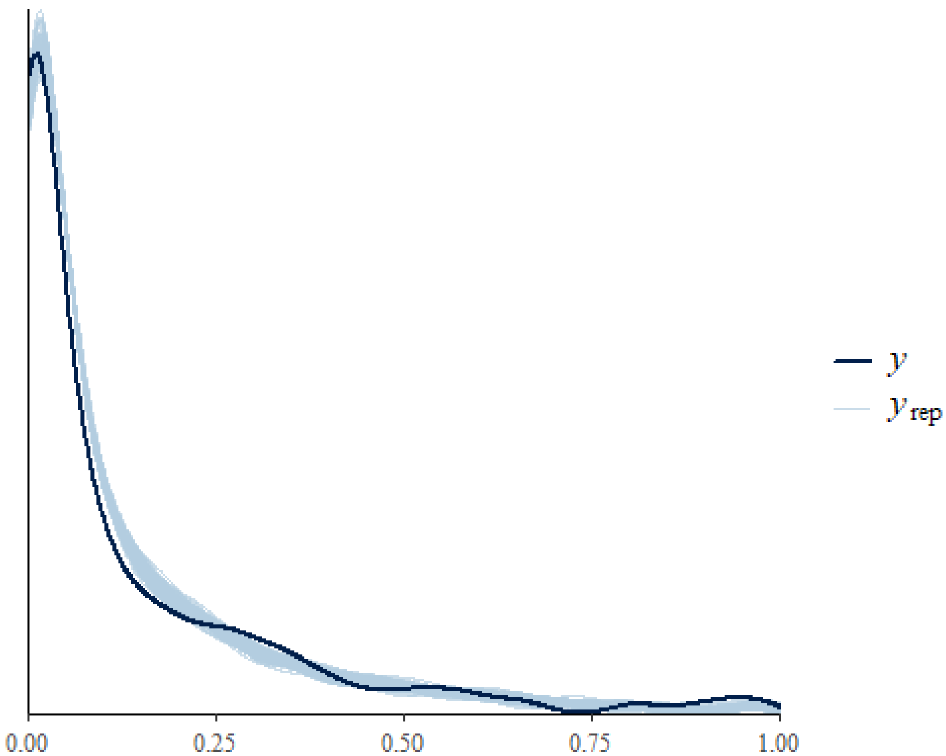
```
)

load("2pre_time2.rds")
params.to.monitor2=c("mu","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
## mu       -5.1964200 0.0048349612 0.12286201 -5.4520887 -4.9665072  645.7275
## beta[1]   0.6974673 0.0009971936 0.02574175  0.6477136  0.7483416  666.3726
## beta[2]   0.8344729 0.0013639486 0.04387019  0.7487497  0.9218539 1034.5298
##              Rhat
## mu       1.002062
## beta[1]  1.004699
## beta[2]  1.001381
```

```
#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'
)
```




```r
#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 3220 log-likelihood matrix.
##
##           Estimate    SE
## elpd_loo    7707.0  95.8
## p_loo       1854.5  24.7
## looic     -15414.0 191.6
## ------
## MCSE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. ESS
## (-Inf, 0.7]   (good)     1610  50.0%   142
##    (0.7, 1]   (bad)      1368  42.5%   <NA>
##    (1, Inf)   (very bad)  242   7.5%   <NA>
## See help('pareto-k-diagnostic') for details.

load("2pre_time2.rds")
loo.mod2=loo.mod # rename loo.mod so can load base model
mod2 = mod
load("2pre_lowb.rds") #load base model
```

```
#create data frame of looics from two models
table.data<-data.frame(
  Model = c("Base model","Model including study period"),
  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 = "Comparisons of LOOic between baseline and
Model with study period", format = "markdown")
```

Comparisons of LOOic between baseline and Model with study period

Model	LOOic	SE
Base model	-15440.81	190.9764
Model including study period	-15414.03	191.5567

```
Looic.diff = loo.mod2$estimates[3] - loo.mod$estimates[3]
print(Looic.diff)

## [1] 26.78257
```

Results - Analysis of original and new studies

I found no model support for an additive time factor because the change in LOOic is positive = 26.8, meaning that the model with time is worse than the model without time.

```
P.Ind<-seq(from = 0.4, to = 5.5,by = .1) #don't extrapolate rise  min(Pre.Ind
.Rise)= 0.4

load("2pre_time2.rds")
modx = mod
load("2pre_lowb.rds")

#Calculate estimates; note original is 1 in matrix
posterior=as.data.frame(modx);
beta.new<-posterior[["beta[1]"]]
beta.old<-posterior[["beta[2]"]]

#compare new vs old slopes
beta.diff<-beta.new-beta.old;
beta.diff.comp <- (quantile(beta.diff, probs = c(0.025, 0.5, 0.975)))
beta.diff.comp
```

```
##        2.5%         50%       97.5%
## -0.22577003 -0.13723216 -0.04974919

#For new data
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ beta.new[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_new = pred.reg.quant[2,],
                          low_line_new = pred.reg.quant[1,],
                          hi_line_new= pred.reg.quant[3,])

#for original data
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ beta.old[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_orig = pred.reg.quant[2,]
pred.reg.df$low_line_orig = pred.reg.quant[1,]
pred.reg.df$hi_line_orig= pred.reg.quant[3,]

Fig1<-ggplot(data = pred.reg.df)+
      geom_ribbon(aes(x=P.Ind,ymin=low_line_orig,ymax=hi_line_orig),alpha=.7,
fill="#5b507b")+
  geom_line(aes(x=P.Ind,y=mean_line_orig),size=1.5,color="#5b507b")+
  geom_ribbon(aes(x=P.Ind,ymin=low_line_new,ymax=hi_line_new),alpha=.5,fill="
#416788")+
  geom_line(aes(x=P.Ind,y=mean_line_new),size=1.5,color="#416788")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+
  theme(axis.title=element_text(size=18),title=element_text(size=20),axis.tex
t = element_text(size=16))+
  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.

Fig1
```

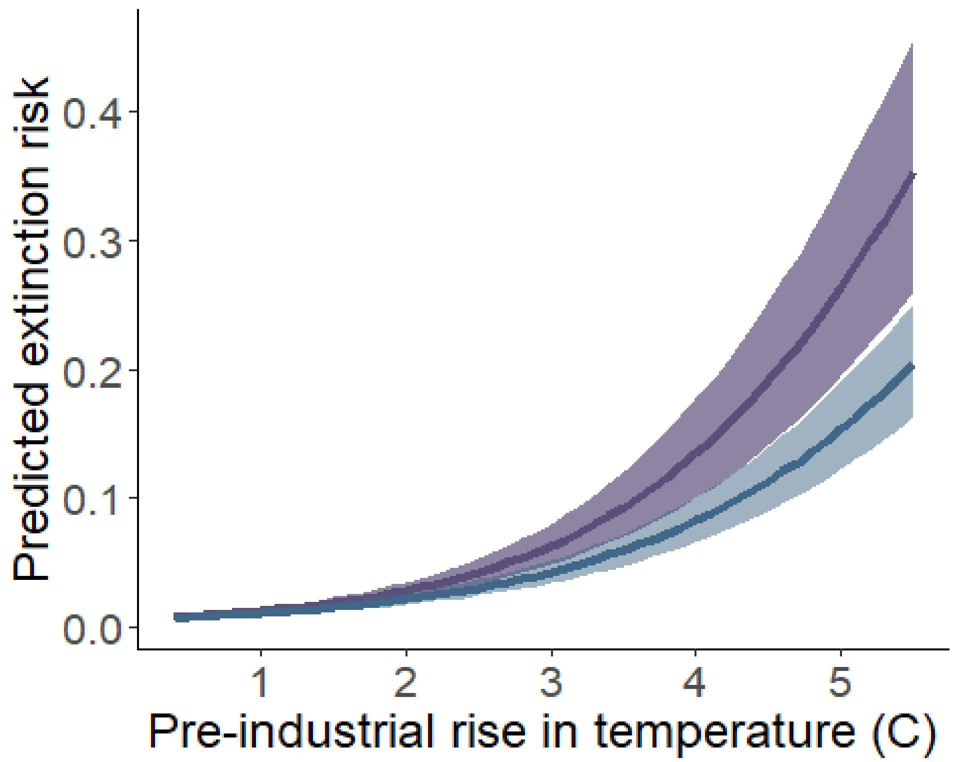
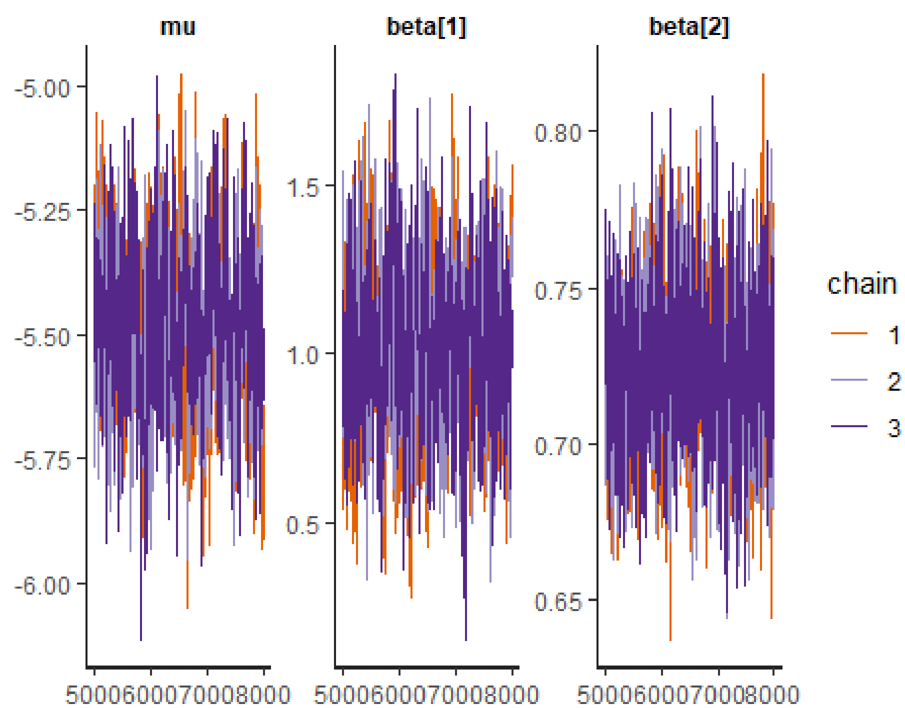


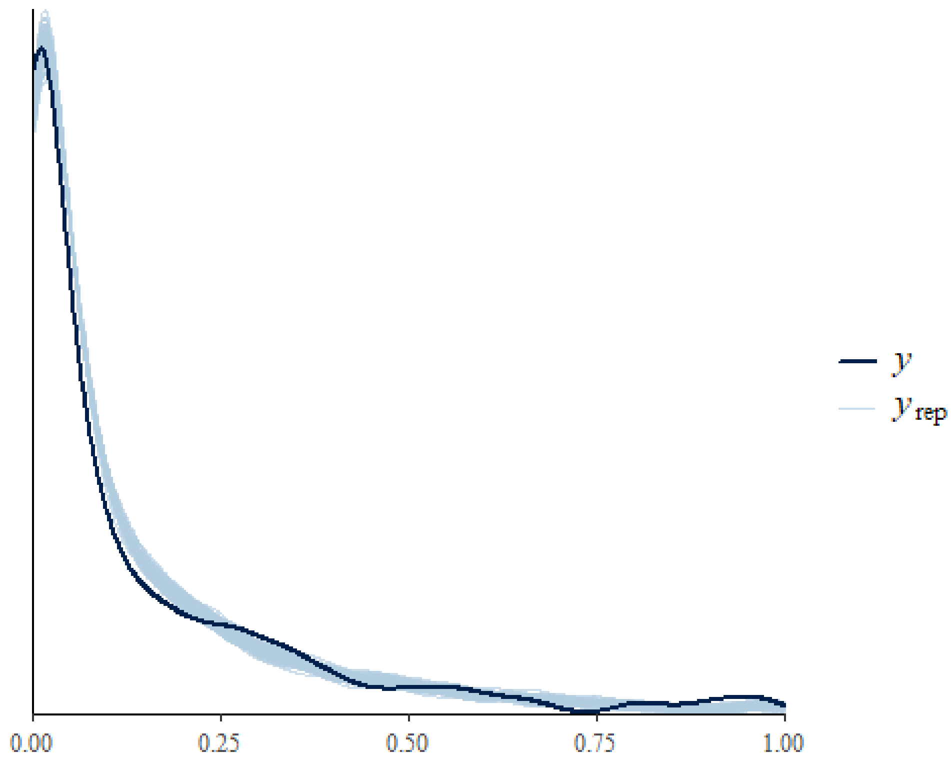
Fig. x. Predicted extinction risk versus pre-industrial temperature rise for original (red) and new (blue) estimates.

```
#ggsave("FigSx pre by time.png",width=8,height=6,unit="in",dpi="print")

#create model matrix for coefficients
betamat<-(model.matrix(~Time+Pre.Ind.Rise,data=data.use))[,2:3] #create model
matrix, exclude intercept which is modeled separately as mu (to allow for inf
ormed prior)

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("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")

init.fn<- function (chain_id) {
  list(mu = -5, beta = c(0,.6,0))
}

# mod=stan(file="MetaRisk2 RSTAN quad.stan",data=stan.data,pars=params.to.mon
itor,
#           chains = 3, warmup=5000, cores=7,iter=8000,
#          init = init.fn, control=list(adapt_delta = 0.9, max_treedepth = 15)
)

load("2pre_time3.rds")
params.to.monitor2=c("mu","beta")#
```

```r
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
## mu       -5.4787482 0.0088089946 0.15404755 -5.7941425 -5.1939370 305.8137
## beta[1]   1.0088886 0.0107254538 0.22718633  0.5820817  1.4671272 448.6762
## beta[2]   0.7244421 0.0009296898 0.02348723  0.6786863  0.7706699 638.2451
##              Rhat
## mu       1.014057
## beta[1]  1.008983
## beta[2]  1.001456

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




```r
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 3220 log-likelihood matrix.
##
##          Estimate    SE
## elpd_loo   7696.0  95.9
## p_loo      1864.7  25.2
## looic    -15391.9 191.9
## ------
## MCSE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. ESS
## (-Inf, 0.7]   (good)     1588  49.3%   121
##    (0.7, 1]   (bad)      1379  42.8%   <NA>
##    (1, Inf)   (very bad)  253   7.9%   <NA>
## See help('pareto-k-diagnostic') for details.

load("2pre_time3.rds")
loo.mod2=loo.mod # rename loo.mod so can load base model
mod2 = mod
load("2pre_lowb.rds") #load base model
```

```r
#create data frame of looics from two models
table.data<-data.frame(
  Model = c("Base model","Model including study period"),
  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 = "Comparisons of LOOic between baseline and
Model with study period", format = "markdown")
```

Comparisons of LOOic between baseline and Model with study period

Model	LOOic	SE
Base model	-15440.81	190.9764
Model including study period	-15391.91	191.8692

```r
Looic.diff = loo.mod2$estimates[3] - loo.mod$estimates[3]
print(Looic.diff)

## [1] 48.90551
```

Results - Same slope model of original and new studies

I found no model support for an additive time factor because the change in LOOic is positive = 48.9 and greater than for the same-intercept model, meaning that this model with time is worse than the model without time and with the same intercept.

```r
P.Ind<-seq(from = 0.4, to = 5.5,by = .1) #don't extrapolate rise  min(Pre.Ind
.Rise)= 0.4

load("2pre_time3.rds")
modx = mod
load("2pre_lowb.rds")

#Calculate estimates; note original is 1 in matrix
posterior=as.data.frame(modx);
alpha.old<-posterior[["beta[1]"]]
beta<-posterior[["beta[2]"]]

#For new data
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
```

```r
                                  mean_line_new = pred.reg.quant[2,],
                                  low_line_new = pred.reg.quant[1,],
                                  hi_line_new= pred.reg.quant[3,])

#for original data
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ alpha.old[x] + beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_orig = pred.reg.quant[2,]
pred.reg.df$low_line_orig = pred.reg.quant[1,]
pred.reg.df$hi_line_orig= pred.reg.quant[3,]

Fig1<-ggplot(data = pred.reg.df)+
      geom_ribbon(aes(x=P.Ind,ymin=low_line_orig,ymax=hi_line_orig),alpha=.7,
fill="#5b507b")+
  geom_line(aes(x=P.Ind,y=mean_line_orig),size=1.5,color="#5b507b")+
  geom_ribbon(aes(x=P.Ind,ymin=low_line_new,ymax=hi_line_new),alpha=.5,fill="
#416788")+
  geom_line(aes(x=P.Ind,y=mean_line_new),size=1.5,color="#416788")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+
  theme(axis.title=element_text(size=18),title=element_text(size=20),axis.tex
t = element_text(size=16))+
  guides(size=F)
Fig1
```

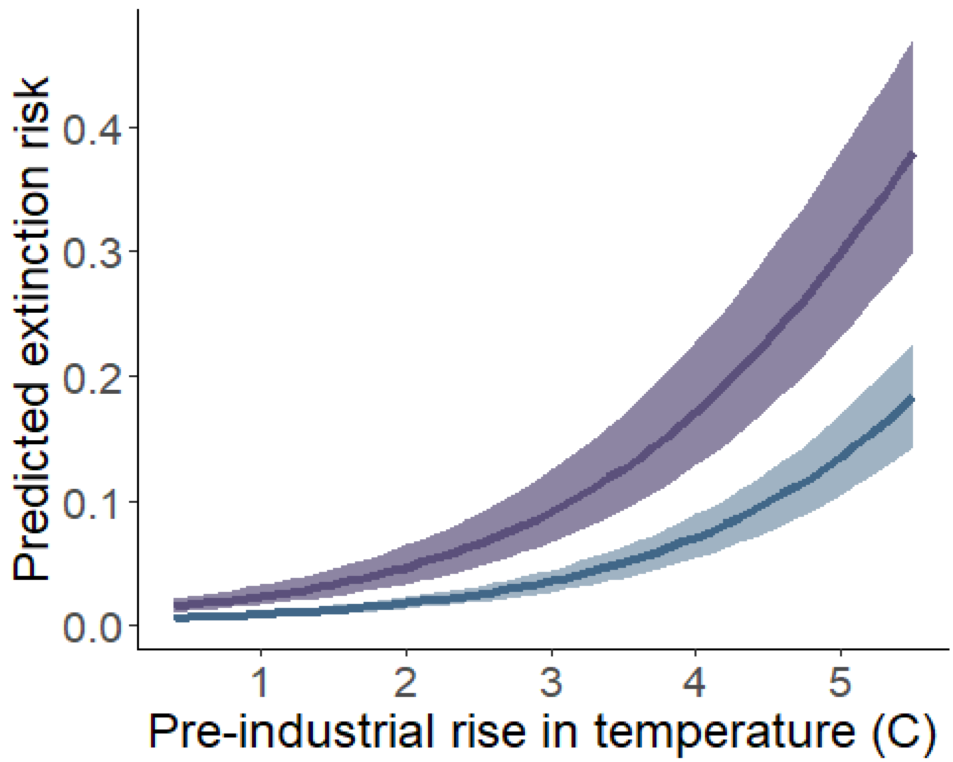


Fig. x. Predicted extinction risk versus pre-industrial temperature rise for original (red) and new (blue) estimates.

```
ggsave("FigSx time same slope.png",width=8,height=6,unit="in",dpi="print")
```

Conclusion

The model with study period as a factor and its interaction with temperature does not improve upon the base model. The LOOic is larger, indicating poorer fit, however the 95% credible intervals for the slopes do not overlap zero.

Comparing old studies and all studies

```
P.Ind<-seq(from = 0.4, to = 5.5,by = .1) #don't extrapolate rise  min(Pre.Ind
.Rise)= 0.4

load("2pre_time2.rds")
modx = mod
posterior=as.data.frame(modx);

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

```
posterior.all = as.data.frame(mod)
#Calculate estimates; note original is 1 in matrix

beta.all<-posterior.all[["beta"]]
beta.old<-posterior[["beta[2]"]]

#compare new vs old slopes
beta.diff<-beta.all-beta.old;
beta.diff.comp <- (quantile(beta.diff, probs = c(0.025, 0.5, 0.975)))
beta.diff.comp

##        2.5%         50%       97.5%
## -0.28827098 -0.18896031 -0.08986062

#For all data
pred.reg = sapply(1:length(posterior.all$mu), FUN = function(x) {posterior.al
l$mu[x] + beta.all[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_all = pred.reg.quant[2,],
                          low_line_all = pred.reg.quant[1,],
                          hi_line_all = pred.reg.quant[3,],
                          range_line_all = pred.reg.quant[3,] - pred.reg.quan
t[1,])

#for original data
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ beta.old[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_orig = pred.reg.quant[2,]
pred.reg.df$low_line_orig = pred.reg.quant[1,]
pred.reg.df$hi_line_orig= pred.reg.quant[3,]
pred.reg.df$range_line_orig= pred.reg.quant[3,] - pred.reg.quant[1,]

pred.reg.df$prop.lower = (pred.reg.df$range_line_all)-(pred.reg.df$range_line
_orig)

Fig1<-ggplot(data = pred.reg.df)+
      geom_ribbon(aes(x=P.Ind,ymin=low_line_orig,ymax=hi_line_orig),alpha=.7,
fill="#5b507b")+
  geom_line(aes(x=P.Ind,y=mean_line_orig),size=1.5,color="#5b507b")+
  geom_ribbon(aes(x=P.Ind,ymin=low_line_all,ymax=hi_line_all),alpha=.5,fill="
#416788")+
  geom_line(aes(x=P.Ind,y=mean_line_all),size=1.5,color="#416788")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+
  theme(axis.title=element_text(size=18),title=element_text(size=20),axis.tex
```

```
t = element_text(size=16))+
  guides(size=F)
Fig1
```

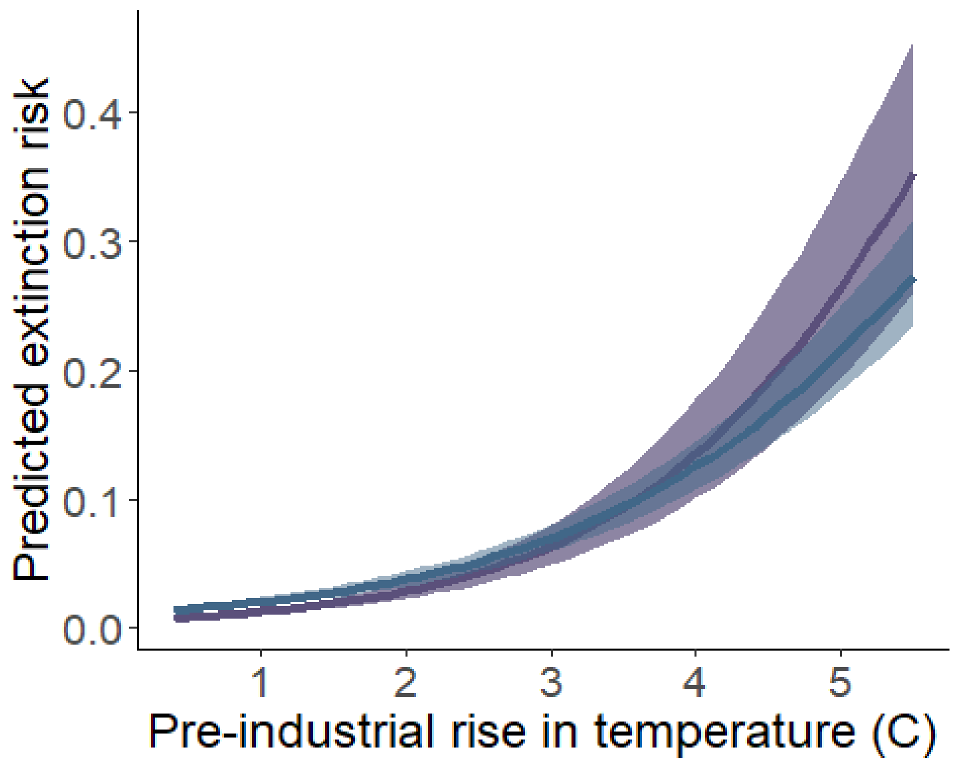


Fig. 1. Predicted extinction risk versus pre-industrial temperature rise for original (red) and all (blue) estimates.

```
manual_labels <- c("range_line" = "Label A",
                   "Group B" = "Label B",
                   "Group C" = "Label C")

Fig2<-ggplot(data = pred.reg.df)+
  geom_hline(yintercept = 0, color = "darkgrey") +
  geom_line(aes(x=P.Ind,y=range_line_orig,color="#5b507b"),size=1.5)+
  geom_line(aes(x=P.Ind,y=range_line_all,color="#416788"),size=1.5)+
  geom_line(aes(x=P.Ind,y=prop.lower/1,color="#F68928"),size=1.5)+
  scale_y_continuous(sec.axis = sec_axis(~.*1, name = "Difference in range"))
+

  xlab("Pre-industrial rise in temperature (C)") + ylab("Credible interval ra
nge")+
  theme_classic()+
  scale_color_identity("Credible interval range",guide = "legend", breaks = c
("#5b507b","#416788","#F68928"),
                        labels =c("Original","New","Difference")) +
  theme(axis.title=element_text(size=18),title=element_text(size=20),axis.tex
```

```
t = element_text(size=16),legend.position = c(.25, .83), legend.title = eleme
nt_text(size = 18), legend.text = element_text(size = 16))

## Warning: A numeric `legend.position` argument in `theme()` was deprecated
in ggplot2
## 3.5.0.
## ℹ Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

  #guides(size=F)
Fig2
```

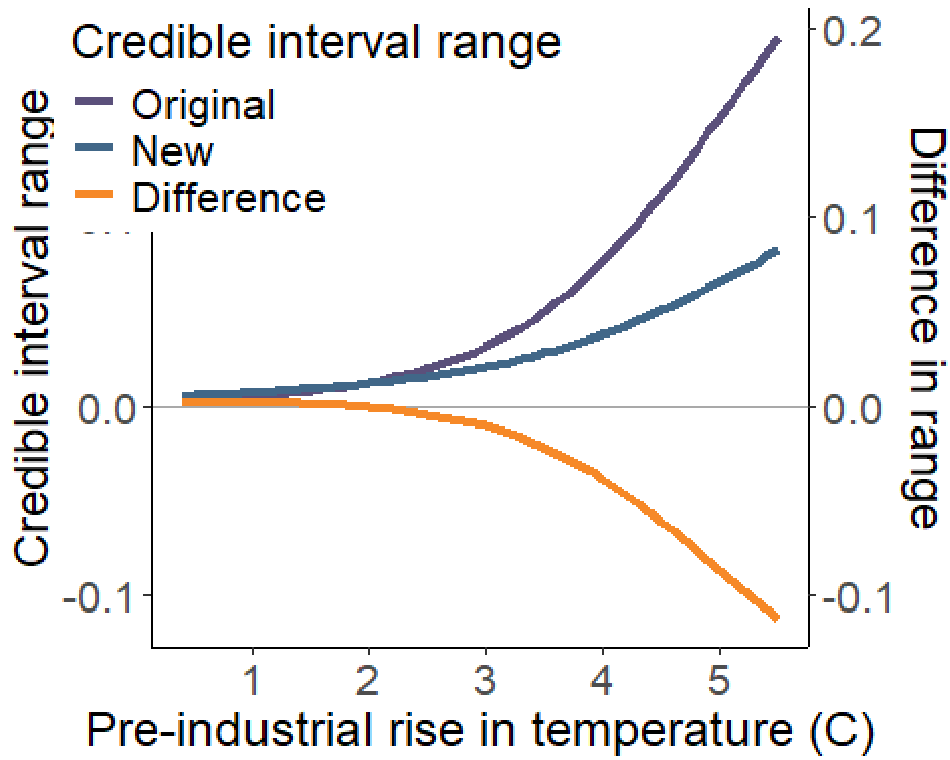


Fig. 2. Predicted extinction risk versus pre-industrial temperature rise for original (red) and all (blue) estimates.

```
#ggsave("FigS8 pre all vs old.png",width=8,height=6,unit="in",dpi="print")
```