Main figure in climate change extinctions

March 25th, 2024

Load libraries and data

```
knitr::opts_chunk$set(echo = TRUE, cache.lazy = FALSE, cache = TRUE)
rm(list = ls())
 root.dir = "C:/Users/mcu08001/Documents/1New Research/CC MetaRisk2/Analysis"
library(ggplot2); library(dplyr); library(ggpubr);

dataP<-read.table("Metarisk2 aggthres 5.txt",header=T); #newest 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;

data.use<-dataP2
```

Load scenarios

```
#load estimates from MetaRisk2 facwtd preind.rmd
load("preind Weighted estimate.rds") # pred.reg.df, calculated in MetarIsk2 f
acwtd preind.RMD

#vector of temperatures to obtain values for
ssps <- c(1.2, 1.5, 2.0, 2.9, 4.3, 5.4) #2100 global temp from AR6 W1 Ch.4 Fi
g 4.2
index <- c(13, 16,21,30,44,55) # because floating point sucks
tpred <- pred.reg.df$mean_linetg[index] #return ext estimates
tpred

## [1] 0.01442579 0.01827593 0.02703627 0.05404530 0.14897341 0.29677572

#other quantities
P.Ind<-seq(from = 0, to = 5.5,by = .1) #prediction interval and frequency
P.Ind2<-seq(from = 0, to = 5.5,by = .01) #prediction interval and frequency f
or Fig. 1b

scen.table <- data.frame(Scenario = ssps,
                         "Predicted Temperature" = tpred,
```

```
                          "lower CI" = pred.reg.df$low_linetg[index],
                          "upper CI" = pred.reg.df$hi_linetg[index])

scen.table.r <- scen.table
scen.table.r <- round((scen.table.r), digits = 4)
knitr::kable(scen.table.r, caption = "Table x: Extinction risk by IPCC scenar
io, with 95% credible intervals", format = "markdown", font = 5)
```

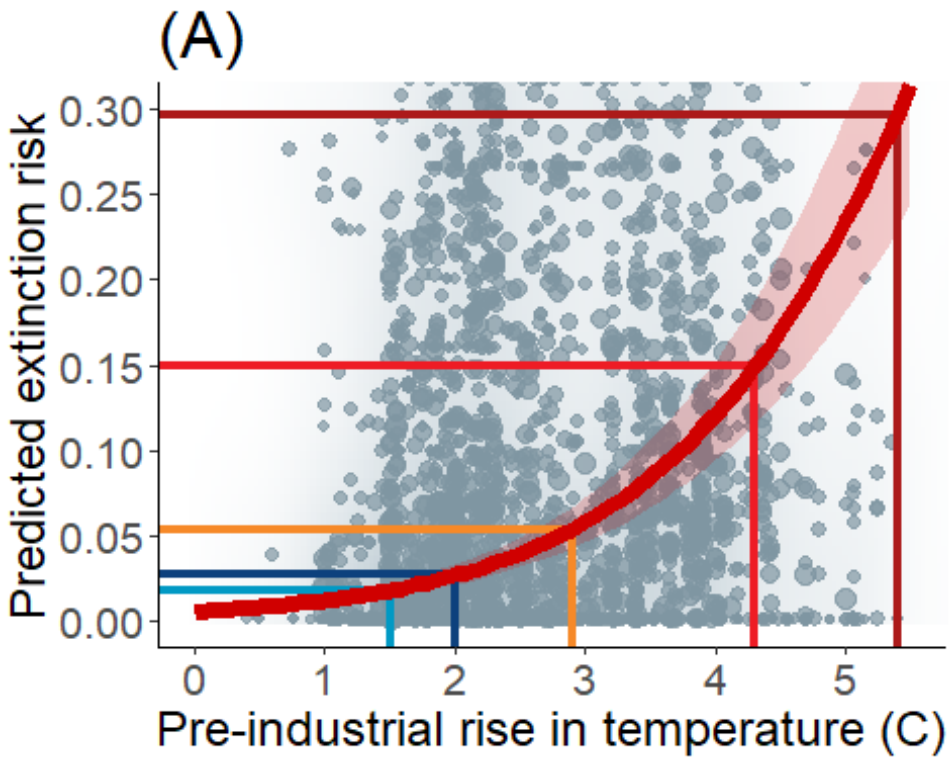
Table x: Extinction risk by IPCC scenario, with 95% credible intervals

Scenario	Predicted.Temperature	lower.CI	upper.CI
1.2	0.0144	0.0115	0.0179
1.5	0.0183	0.0146	0.0226
2.0	0.0270	0.0217	0.0334
2.9	0.0540	0.0429	0.0669
4.3	0.1490	0.1157	0.1879
5.4	0.2968	0.2302	0.3714

Create figures

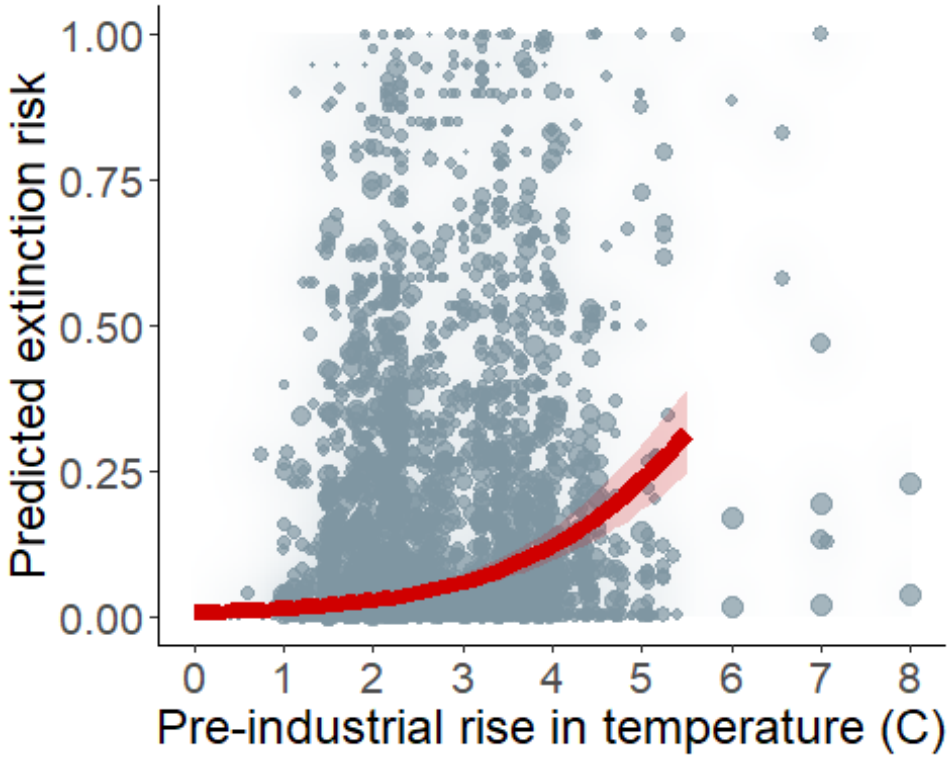
```
Fig1<-ggplot(data = pred.reg.df)+
  stat_density2d(data = data.use, aes(x=Pre.Ind.Rise, y=percent2, fill = ..de
nsity..^.5),
                 geom = "tile", contour = FALSE, n = 200, show.legend = FALSE
) +
  scale_fill_continuous(low = "white", high = "#B1C2CB") +
  scale_x_continuous(breaks = seq(0,5,1)) +
  geom_point(data = data.use, aes(x=Pre.Ind.Rise, y=percent2, size = log(Tota
l.N)), alpha = 0.7, shape = 20, color = "#7F96A2") +
  geom_ribbon(data = pred.reg.df, aes(x=P.Ind,ymin=low_linetg,ymax=hi_linetg)
,alpha=.2,fill="#D00000")+

  geom_segment(x = ssps[2], xend = ssps[2], y = -Inf, yend = tpred[2], color
= "#0199C4", linewidth = 1.5) +
  geom_segment(x = ssps[3], xend = ssps[3], y = -Inf, yend = tpred[3], color
= "#0D417D", linewidth = 1.5) +
  geom_segment(x = ssps[4], xend = ssps[4], y = -Inf, yend = tpred[4], color
= "#F68928", linewidth = 1.5) +
  geom_segment(x = ssps[5], xend = ssps[5], y = -Inf, yend = tpred[5], color
= "#EE1C25", linewidth = 1.5) +
  geom_segment(x = ssps[6], xend = ssps[6], y = -Inf, yend = tpred[6], color
= "#AC1A1A", linewidth = 1.5) +

  geom_segment(y = tpred[2], xend = ssps[2], x = -Inf, yend = tpred[2], color
= "#0199C4", linewidth = 1.5) +
  geom_segment(y = tpred[3], xend = ssps[3], x = -Inf, yend = tpred[3], color
```

```
= "#0D417D", linewidth = 1.5) +
  geom_segment(y = tpred[4], xend = ssps[4], x = -Inf, yend = tpred[4], color
= "#F68928", linewidth = 1.5) +
  geom_segment(y = tpred[5], xend = ssps[5], x = -Inf, yend = tpred[5], color
= "#EE1C25", linewidth = 1.5) +
  geom_segment(y = tpred[6], xend = ssps[6], x = -Inf, yend = tpred[6], color
= "#AC1A1A", linewidth = 1.5)+

  geom_line(data = pred.reg.df, aes(x=P.Ind,y=mean_linetg),size=3,color="#D00
000")+

  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  ggtitle("(A)") +
  theme_classic()+ coord_cartesian(xlim = c(0,5.5), ylim = c(0,.30)) + scale_
y_continuous(breaks = seq(0,.3,.05)) +
  theme(axis.title=element_text(size=18),title=element_text(size=20),axis.tex
t = element_text(size=16))+
  guides(size=F)
Fig1
```




```
#ggsave("Fig1 overall.png",width=6,height=4.8,unit="in",dpi="print")

max(data.use$Pre.Ind.Rise)

## [1] 8
```

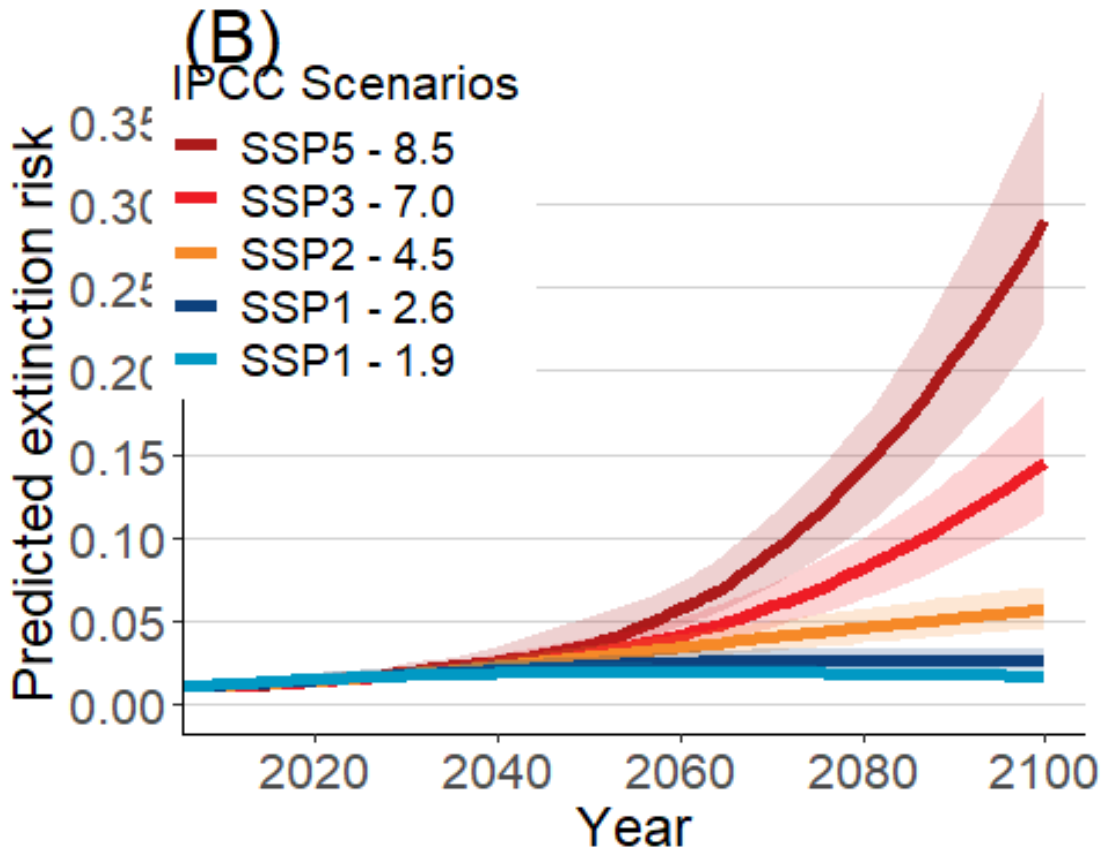
```
FigS1<-ggplot(data = pred.reg.df)+
  stat_density2d(data = data.use, aes(x=Pre.Ind.Rise, y=percent2, fill = ..density..^.5),
                 geom = "tile", contour = FALSE, n = 200, show.legend = FALSE) +
  scale_fill_continuous(low = "white", high = "#B1C2CB") +
  scale_x_continuous(breaks = seq(0,8,1)) +
  geom_point(data = data.use, aes(x=Pre.Ind.Rise, y=percent2, size = log(Total.N)), alpha = 0.7, shape = 20, color = "#7F96A2") +
  geom_ribbon(data = pred.reg.df, aes(x=P.Ind,ymin=low_linetg,ymax=hi_linetg),alpha=.2,fill="#D00000")+
  geom_line(data = pred.reg.df, aes(x=P.Ind,y=mean_linetg),size=3,color="#D00000")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction risk")+
  theme_classic()+ coord_cartesian(xlim = c(0,8), ylim = c(0,1)) +
  theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text = element_text(size=16))+
  guides(size=F)
FigS1
```




```
#ggsave("FigS1 overall.png",width=8,height=6,unit="in",dpi="print")

scenes <- read.table("ipcc scenario temps.txt",header=T); attach(scenes) # import IPCC scenarios by 5 year increment
```

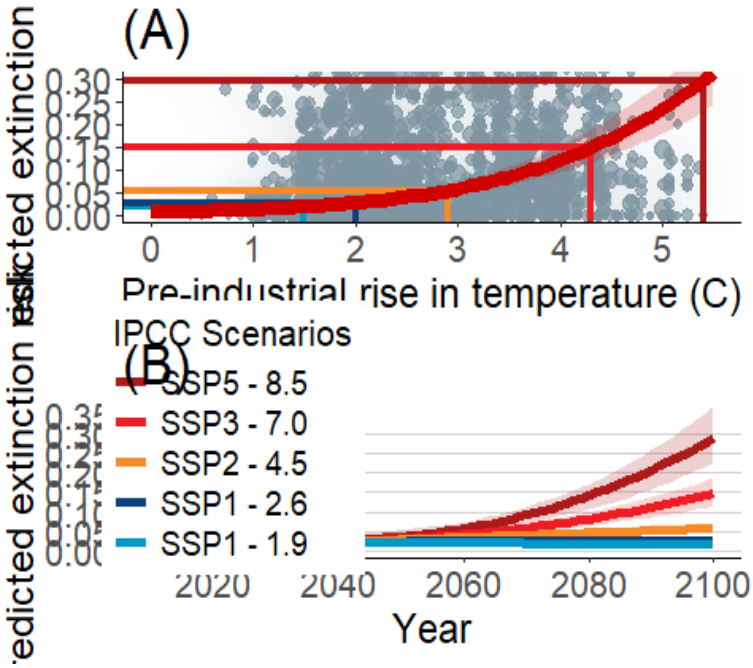
```
## The following object is masked from dataP2:
##
##     Year

fine.time <- seq(2000,2100,1)
#create finer interpolated predictions
sm.1.1.9 <- loess(SSP1.1.9 ~ Year, span = 0.75)
sm.1.2.6 <- loess(SSP1.2.6 ~ Year, span = 0.75)
sm.2.4.5 <- loess(SSP2.4.5 ~ Year, span = 0.75)
sm.3.7.0 <- loess(SSP3.7.0 ~ Year, span = 0.75)
sm.5.8.5 <- loess(SSP5.8.5 ~ Year, span = 0.75)

pSSP1.1.9 <- predict(sm.1.1.9, newdata = fine.time)
pSSP1.2.6 <- predict(sm.1.2.6, newdata = fine.time)
pSSP2.4.5 <- predict(sm.2.4.5, newdata = fine.time)
pSSP3.7.0 <- predict(sm.3.7.0, newdata = fine.time)
pSSP5.8.5 <- predict(sm.5.8.5, newdata = fine.time)

fine.scenes <- data.frame(Year = fine.time,SSP1.1.9 = pSSP1.1.9, SSP1.2.6 = p
SSP1.2.6, SSP2.4.5 = pSSP2.4.5, SSP3.7.0 = pSSP3.7.0, SSP5.8.5 = pSSP5.8.5)
fine.scenes[,2:6] <- round(fine.scenes[,2:6],2)
pred.reg.hi = data.frame(x = pred.reg.df2[,1],pred.reg.df2[11:13])

scen.preds<-merge(fine.scenes,pred.reg.hi,by.x = "SSP1.1.9",by.y = "x")
names(scen.preds)[names(scen.preds) %in% c("mean_linetg","low_linetg","hi_lin
etg")]<- c("mean_line.1.9","low_line.1.9","hi_line.1.9")

scen.preds<-merge(scen.preds,pred.reg.hi,by.x = "SSP1.2.6",by.y = "x")
names(scen.preds)[names(scen.preds) %in% c("mean_linetg","low_linetg","hi_lin
etg")]<- c("mean_line.2.6","low_line.2.6","hi_line.2.6")

scen.preds<-merge(scen.preds,pred.reg.hi,by.x = "SSP2.4.5",by.y = "x")
names(scen.preds)[names(scen.preds) %in% c("mean_linetg","low_linetg","hi_lin
etg")]<- c("mean_line.4.5","low_line.4.5","hi_line.4.5")

scen.preds<-merge(scen.preds,pred.reg.hi,by.x = "SSP3.7.0",by.y = "x")
names(scen.preds)[names(scen.preds) %in% c("mean_linetg","low_linetg","hi_lin
etg")]<- c("mean_line.7.0","low_line.7.0","hi_line.7.0")

scen.preds<-merge(scen.preds,pred.reg.hi,by.x = "SSP5.8.5",by.y = "x")
names(scen.preds)[names(scen.preds) %in% c("mean_linetg","low_linetg","hi_lin
etg")]<- c("mean_line.8.5","low_line.8.5","hi_line.8.5")
#
Fig1b<-ggplot(data = scen.preds)+
  geom_hline(yintercept = c(0,.05,.1,.15,.2,.25,.3), color = "lightgrey") +
  geom_ribbon(aes(x=Year,ymin=low_line.8.5,ymax=hi_line.8.5),alpha=.2,fill="#
AC1A1A")+
  geom_smooth(aes(x=Year,y=mean_line.8.5,color="#AC1A1A"),linewidth=2, se=F)+
  geom_ribbon(aes(x=Year,ymin=low_line.7.0,ymax=hi_line.7.0),alpha=.2,fill="#
```

```
EE1C25")+
  geom_smooth(aes(x=Year,y=mean_line.7.0,color="#EE1C25"),linewidth=2, se=F)+
  geom_ribbon(aes(x=Year,ymin=low_line.4.5,ymax=hi_line.4.5),alpha=.2,fill="#
F68928")+
  geom_smooth(aes(x=Year,y=mean_line.4.5,color="#F68928"),linewidth=2, se=F)+
  geom_ribbon(aes(x=Year,ymin=low_line.2.6,ymax=hi_line.2.6),alpha=.2,fill="#
0D417D")+
  geom_smooth(aes(x=Year,y=mean_line.2.6,color="#0D417D"),linewidth=2, se=F)+
  geom_ribbon(aes(x=Year,ymin=low_line.1.9,ymax=hi_line.1.9),alpha=.2,fill="#
0199C4")+
  geom_smooth(aes(x=Year,y=mean_line.1.9,color="#0199C4"),linewidth=2, se=F)+
  ggtitle("(B)") +
  xlab("Year") + ylab("Predicted extinction risk")+ scale_y_continuous(breaks
= seq(0,.35,.05)) +
  scale_x_continuous(breaks = seq(2000,2100,20)) + coord_cartesian(xlim = c(2
010,2100), ylim = c(0,.35)) +
  theme_classic() + theme(axis.title=element_text(size=18),title=element_text
(size=20),axis.text = element_text(size=16),legend.title = element_text(size=
16),
                          legend.text = element_text(size=14),legend.position
= c(.18, .8)) +
  scale_color_identity("IPCC Scenarios",guide = "legend", breaks = c("#AC1A1A
","#EE1C25","#F68928","#0D417D","#0199C4"),
                       labels =c("SSP5 - 8.5","SSP3 - 7.0","SSP2 - 4.5","SSP1
- 2.6","SSP1 - 1.9")) #+ guides(color=guide_legend(title="New Legend Title"))

## 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.

Fig1b

## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
#ggsave("Fig1b scenes.png",width=6,height=4.8,unit="in",dpi="print")

ggarrange(Fig1, Fig1b, nrow = 2)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
#ggsave("Fig1 combined.png",width=6,height=10,unit="in",dpi="print")
```