Extinction risk from climate change: Recent extinctions

March 25, 2024

Load libraries and data

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

# load libraries
library(dplyr); library(ggplot2); library(ggpubr); library(rstanarm)

## load data
# Load just IUCN extinction risk attributed to weather or climate change
CC.ext <- read.csv("CC IUCN ext risk.csv",header=T, fill = T);

# Load all extinct taxa
all.ext <- read.csv("IUCN ext data.csv",header=T, fill = T); #,colClasses="ch
aracter",colClasses="character"

# Load temp anomalies https://www.ncei.noaa.gov/access/monitoring/global-temp
erature-anomalies/anomalies
global.temps <- read.table("Temp anomalies.txt", header = T)

# when to start looking for CC
Year.threshold = 1960
```

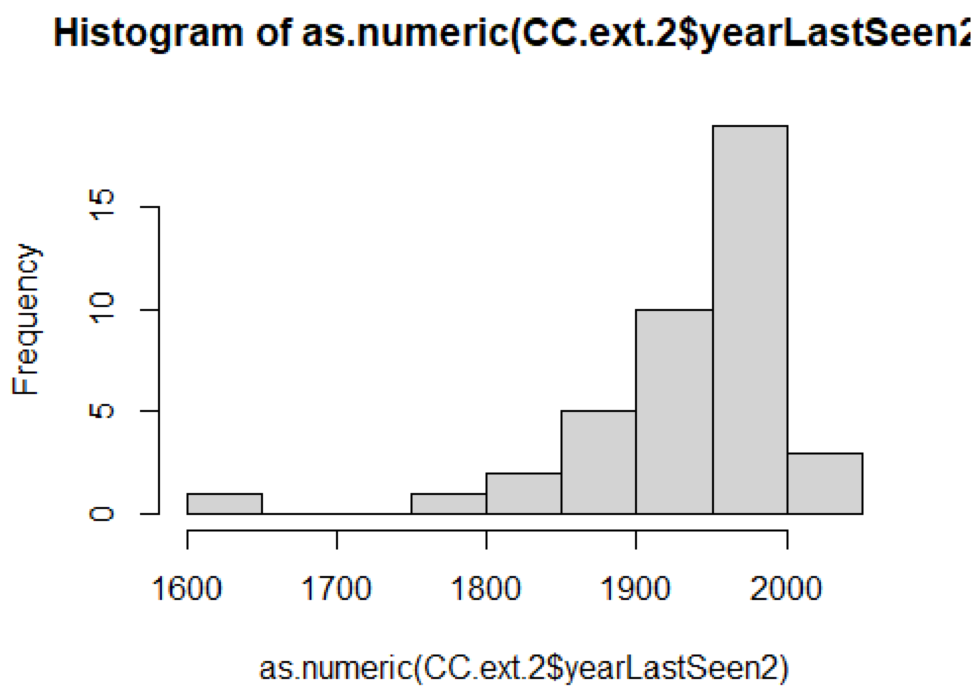
Refine temperature data

```
# Correct and refine temp dataset
global.temps$Anomaly.C <- global.temps$Anomaly - mean(global.temps$Anomaly[gl
obal.temps$Year < 1901]) #IPCC uses 1850-1900 as baseline so correct.
global.temps.sub <- global.temps[global.temps$Year > Year.threshold,]
global.temps.sub$position <- ifelse(global.temps.sub$Anomaly.C > 0, "up","dow
n")
```

Refine extinction list

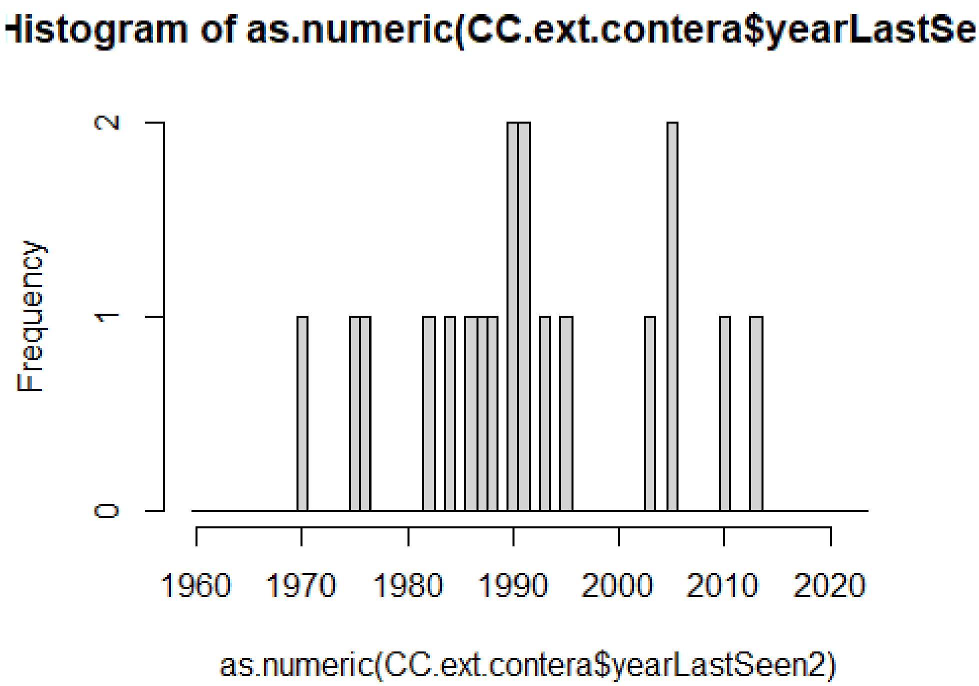
```
# Create new column in all extinction dataset for those attributed to weather
/CC and form reduced dataset CC.ext.2
cc.ext.id <- all.ext$scientificName %in% CC.ext$scientificName
```

```
all.ext$CC.ext <- rep("N",nrow(all.ext))
all.ext$CC.ext[cc.ext.id] = "Y"
sum(all.ext$CC.ext == "Y")

## [1] 41

CC.ext.2 <- all.ext[all.ext$CC.ext == "Y",]
hist(as.numeric(CC.ext.2$yearLastSeen2))
```

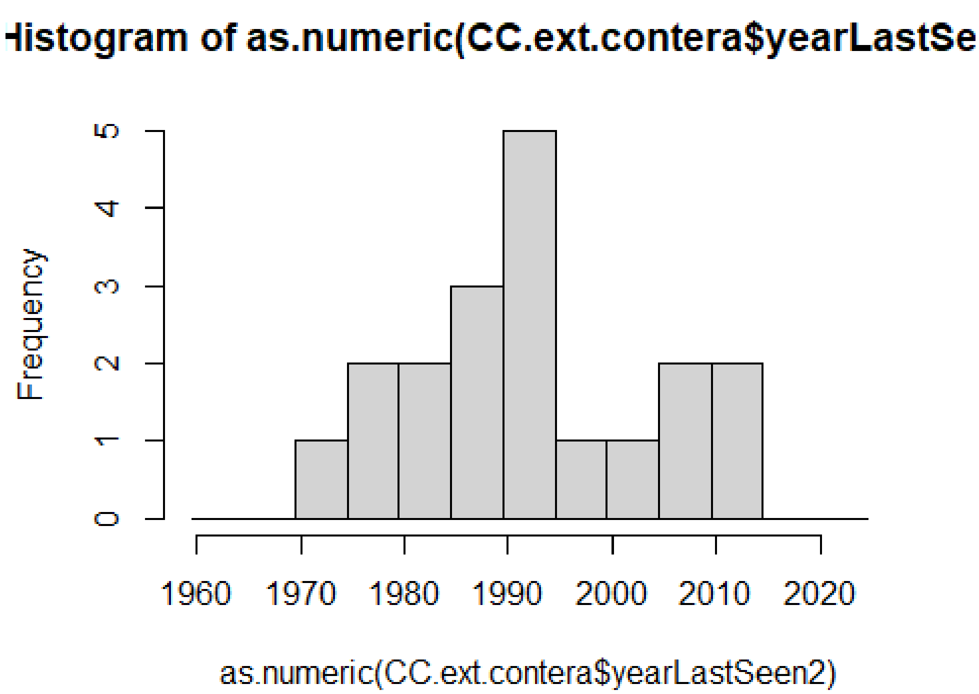



```
# Further refine the dataset to when CC was evident in threshold year
CC.ext.contera <- CC.ext.2[CC.ext.2$yearLastSeen2 > Year.threshold,]
#write.csv(CC.ext.contera, file = "CC ext.csv")
cat("starting number of extinctions =", nrow(CC.ext.contera))

## starting number of extinctions = 19

# Further refine list to exclude these species based on assessment and litera
ture review
CC.ext.contera <- CC.ext.contera[CC.ext.contera$scientificName != "Telestes u
kliva",] # Invasive fish species is only threat indicated
CC.ext.contera <- CC.ext.contera[CC.ext.contera$scientificName != "Dombeya ro
driguesiana",] # over exploitation and overgrazing listed; no climate change
CC.ext.contera <- CC.ext.contera[CC.ext.contera$scientificName != "Zosterops
conspicillatus",] # invasive species and super typhoon, but typhoon in 76 so
unlikely to
CC.ext.contera <- CC.ext.contera[CC.ext.contera$scientificName != "Govenia fl
oridana",] # poaching by collectors; hurricane in 1960 noted, but likely befo
re CC effect
```

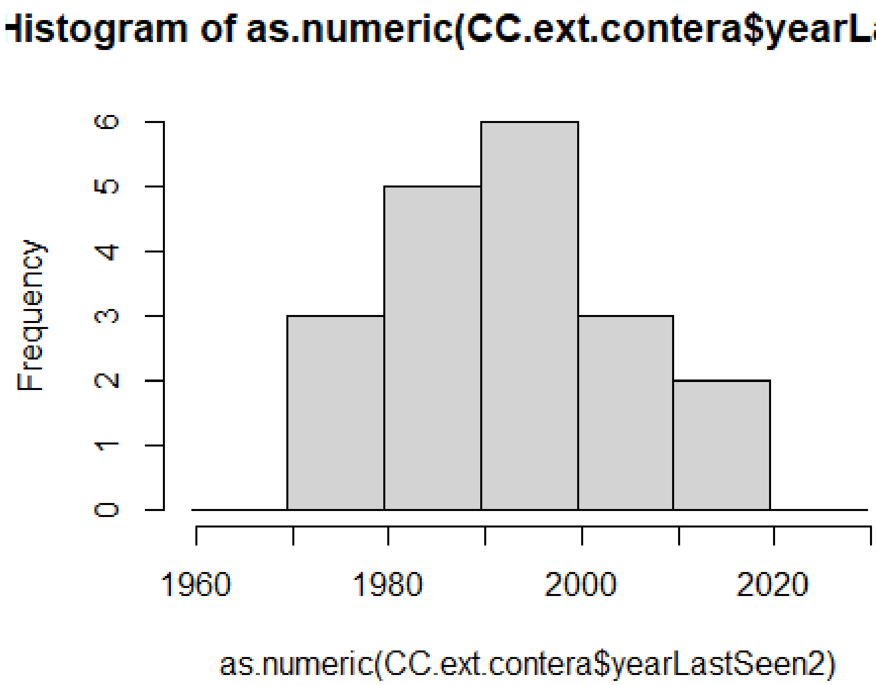
```
#Add species based on new or external literature review
CC.ext.contera[nrow(CC.ext.contera)+1,] <- all.ext[all.ext$scientificName ==
"Moho braccatus",]
CC.ext.contera[nrow(CC.ext.contera)+1,] <- all.ext[all.ext$scientificName ==
"Zosterops conspicillatus",]
CC.ext.contera[nrow(CC.ext.contera)+1,] <- all.ext[all.ext$scientificName ==
"Melamprosops phaeosoma",]
CC.ext.contera[nrow(CC.ext.contera)+1,] <- all.ext[all.ext$scientificName ==
"Corvus hawaiiensis",]

#Fix dates according to assessments
CC.ext.contera$yearLastSeen[CC.ext.contera$scientificName == "Myadestes myade
stinus"] = 1985
CC.ext.contera$yearLastSeen[CC.ext.contera$scientificName == "Cyanea dolichop
oda"] = 1990

CC.ext.contera$yearLastSeen2 = as.numeric(CC.ext.contera$yearLastSeen) +1 #ad
d one year as this is last year seen, so extinct next year?

cat("refined number of extinctions =", nrow(CC.ext.contera))

## refined number of extinctions = 19

#all extinction subset for year threshold and eliminate NAs
all.ext.sub <- all.ext[all.ext$yearLastSeen2 > Year.threshold & !is.na(all.ex
t$yearLastSeen2),]
all.ext.sub$yearLastSeen2 <- all.ext.sub$yearLastSeen2 +1

hist(as.numeric(CC.ext.contera$yearLastSeen2), breaks = seq(Year.threshold-.5
,2023.5,1))
ext.counts <- hist(as.numeric(CC.ext.contera$yearLastSeen2), breaks = seq(Yea
r.threshold-.5,2023.5,1))['counts']
```



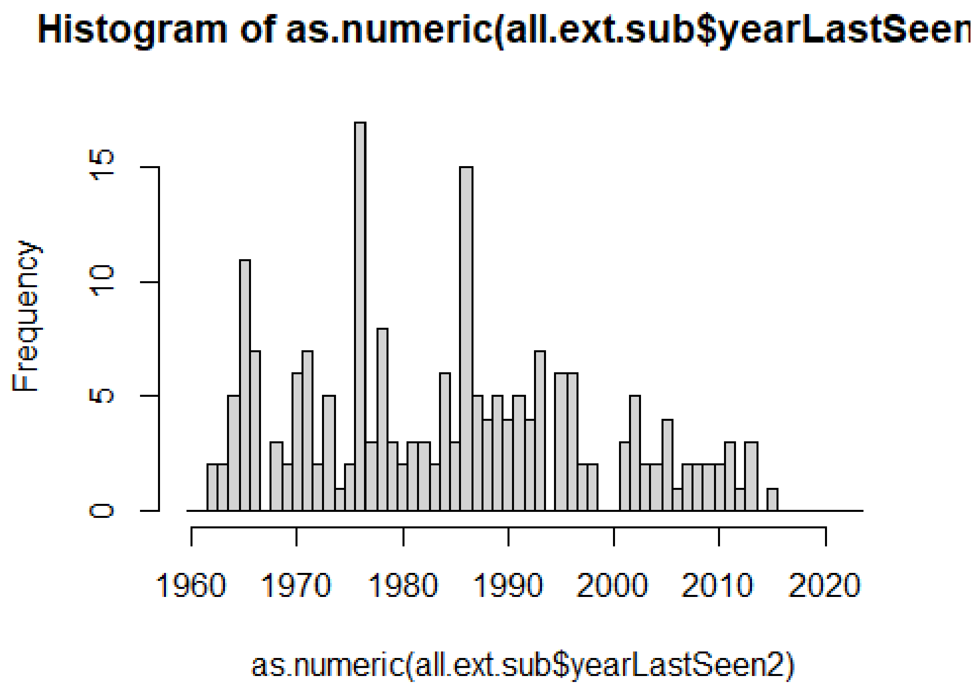

```
ext.counts5 <- hist(as.numeric(CC.ext.contera$yearLastSeen2), breaks = seq(Ye
ar.threshold-.5,2025.5,5))['counts']
```



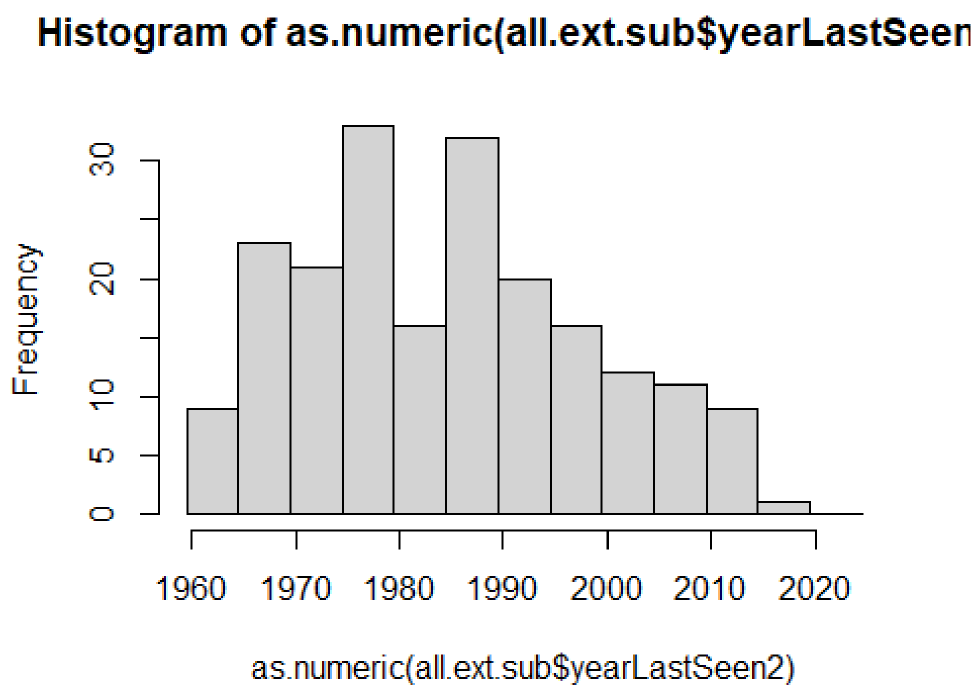
```
ext.counts10 <- hist(as.numeric(CC.ext.contera$yearLastSeen2), breaks = seq(Y
ear.threshold-.5,2030.5,10))['counts']
```



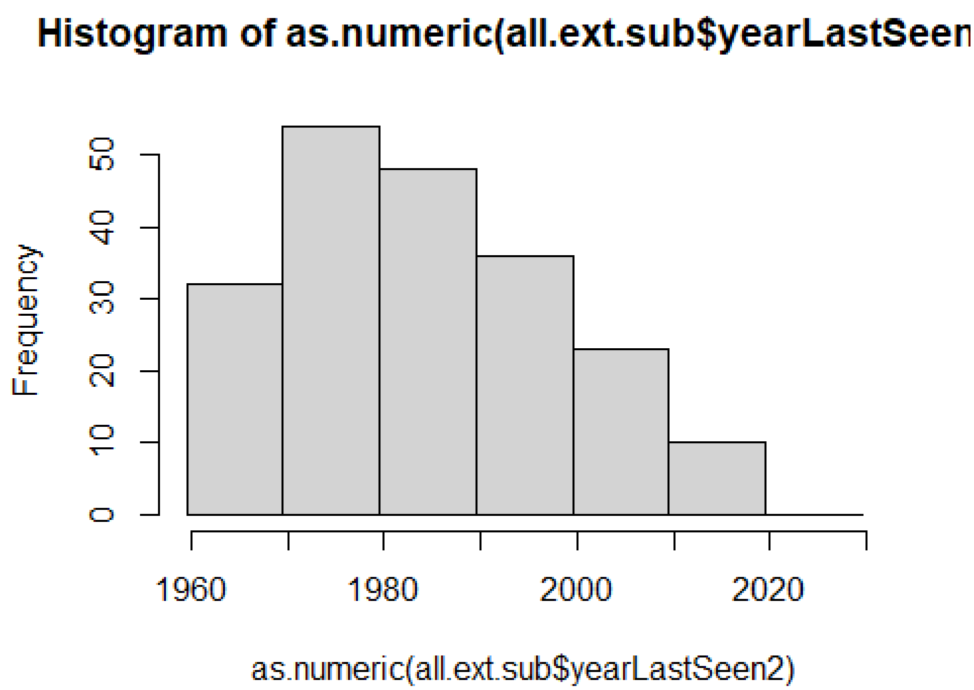

```
all.ext.counts <- hist(as.numeric(all.ext.sub$yearLastSeen2), breaks = seq(Ye
ar.threshold-.5,2023.5,1))['counts']
```




```
all.ext.counts5 <- hist(as.numeric(all.ext.sub$yearLastSeen2), breaks = seq(Y
ear.threshold-.5,2025.5,5))['counts']
```



```
all.ext.counts10 <- hist(as.numeric(all.ext.sub$yearLastSeen2), breaks = seq(
Year.threshold-.5,2030.5,10))['counts']
```




```
prop.CC.exts5 <- ifelse(all.ext.counts5$counts == 0,0,ext.counts5$counts/all.
ext.counts5$counts)
prop.CC.exts10 <- ifelse(all.ext.counts10$counts == 0,0,ext.counts10$counts/a
ll.ext.counts10$counts)

CC.ext.years <- data.frame(CC.ext.years = as.numeric(CC.ext.contera$yearLastS
een2),
                           id = rank(as.numeric(CC.ext.contera$yearLastSeen2)
, ties.method = 'first'))

all.ext.years <- data.frame(all.ext.years = as.numeric(all.ext.sub$yearLastSe
en2))
nonCC.ext.years <- data.frame(nonCC.ext.years = CC.ext.contera$yearLastSeen2
%in% all.ext.sub$yearLastSeen2)

prop.CC.ext10.df <- data.frame(Year = seq(Year.threshold + 5,2015,10),
  prop.CC.exts10 = prop.CC.exts10[1:6],
  CC.ext = ext.counts10$counts[1:6],
  all.ext = all.ext.counts10$counts[1:6],
  Ind = factor(seq(1,6,)))

cat("average proportion extinction risk from climate change = ",mean(prop.CC.
ext10.df[,2]))
```

```
## average proportion extinction risk from climate change =  0.1094706
```

```r
mod <- stan_glmer(cbind(CC.ext,all.ext) ~ scale(Year) + (1|Ind), family = "bi
nomial", data = prop.CC.ext10.df, chains = 3, iter = 10000)
```

```
##
## SAMPLING FOR MODEL 'binomial' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 4.5e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would tak
e 0.45 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.61 seconds (Warm-up)
## Chain 1:                0.785 seconds (Sampling)
## Chain 1:                1.395 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'binomial' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.7e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would tak
e 0.17 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
```

```
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2: 
## Chain 2:  Elapsed Time: 0.65 seconds (Warm-up)
## Chain 2:                0.536 seconds (Sampling)
## Chain 2:                1.186 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'binomial' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 1.5e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would tak
e 0.15 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3: 
## Chain 3: 
## Chain 3: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 3: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 3: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 3: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 3: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 3: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 3: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 3: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 3: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 3: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 3: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 3: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 3: 
## Chain 3:  Elapsed Time: 0.626 seconds (Warm-up)
## Chain 3:                0.827 seconds (Sampling)
## Chain 3:                1.453 seconds (Total)
## Chain 3: 

## Warning: There were 3 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
msum <- data.frame(summary(mod, digits = 4, prob=c(.025, .5, .975)))
msum
```

```
##                                               mean        mcse        sd
## (Intercept)                            -2.38682157 0.003792843 0.3304822
## scale(Year)                             0.79646932 0.004752118 0.4042722
## b[(Intercept) Ind:1]                   -0.17367097 0.005592586 0.5168484
## b[(Intercept) Ind:2]                    0.02382196 0.004266507 0.3824263
## b[(Intercept) Ind:3]                    0.09876254 0.003878536 0.3484525
## b[(Intercept) Ind:4]                    0.11215698 0.003930648 0.3526428
## b[(Intercept) Ind:5]                   -0.09491635 0.004647563 0.3979306
```

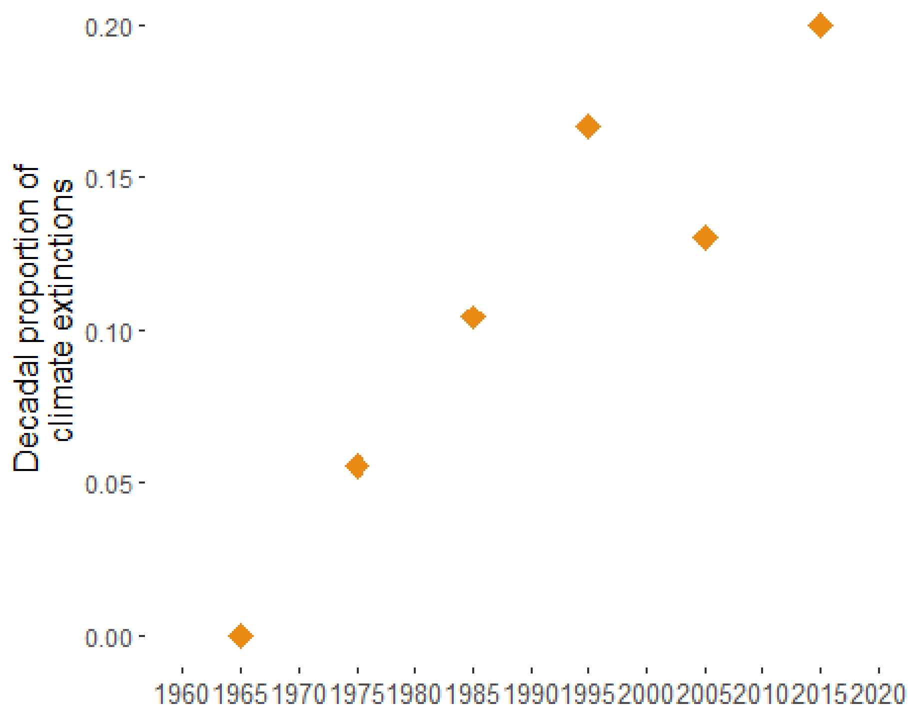
```
## b[(Intercept) Ind:6]                 -0.08408673 0.005994914 0.4539394
## Sigma[Ind:(Intercept),(Intercept)]   0.26268304 0.009789956 0.6260864
## mean_PPD                             3.23790000 0.007390000 0.9797406
## log-posterior                      -25.17182850 0.033615788 2.4317255
##                                            X2.5.          X50.      X97.5.
## (Intercept)                        -3.081532e+00  -2.373437260  -1.7873053
## scale(Year)                         4.775070e-02   0.782657428   1.6271890
## b[(Intercept) Ind:1]               -1.533708e+00  -0.044605428   0.5535329
## b[(Intercept) Ind:2]               -7.647867e-01   0.003046955   0.9195519
## b[(Intercept) Ind:3]               -5.367132e-01   0.031529992   0.9624995
## b[(Intercept) Ind:4]               -5.142488e-01   0.041975809   0.9666400
## b[(Intercept) Ind:5]               -1.072043e+00  -0.026488227   0.6156097
## b[(Intercept) Ind:6]               -1.162576e+00  -0.016576910   0.7201410
## Sigma[Ind:(Intercept),(Intercept)]  1.115812e-04   0.072061349   1.6672997
## mean_PPD                            1.500000e+00   3.166666667   5.3333333
## log-posterior                      -3.080115e+01 -24.863667664 -21.3231588
##                                    n_eff      Rhat
## (Intercept)                         7592 1.0001181
## scale(Year)                         7237 1.0002041
## b[(Intercept) Ind:1]                8541 1.0004731
## b[(Intercept) Ind:2]                8034 1.0000090
## b[(Intercept) Ind:3]                8071 1.0000796
## b[(Intercept) Ind:4]                8049 0.9999883
## b[(Intercept) Ind:5]                7331 1.0001025
## b[(Intercept) Ind:6]                5734 1.0002609
## Sigma[Ind:(Intercept),(Intercept)]  4090 1.0003417
## mean_PPD                           17577 1.0000466
## log-posterior                       5233 1.0002379
```

```
invlogit(as.numeric(msum[2,]))
```

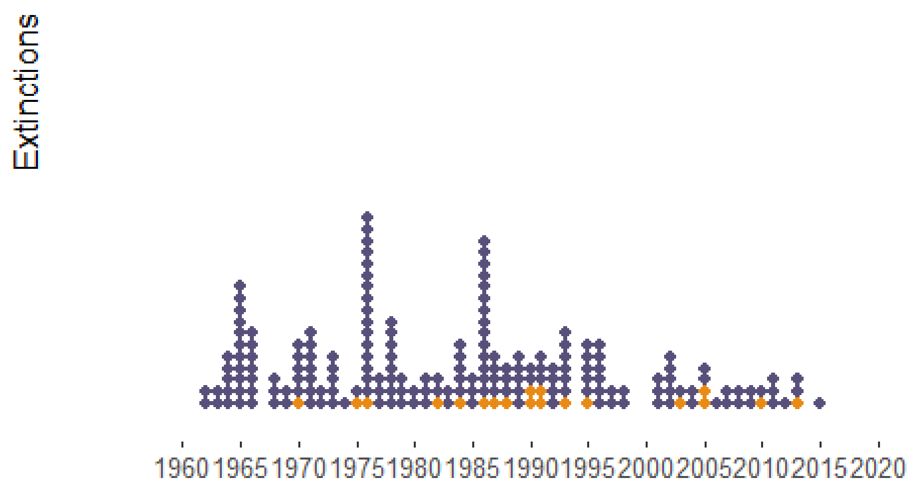
```
## [1] 0.6892187 0.5011880 0.5997137 0.5119354 0.6862526 0.8357842 1.0000000
## [8] 0.7310987
```

Make plots

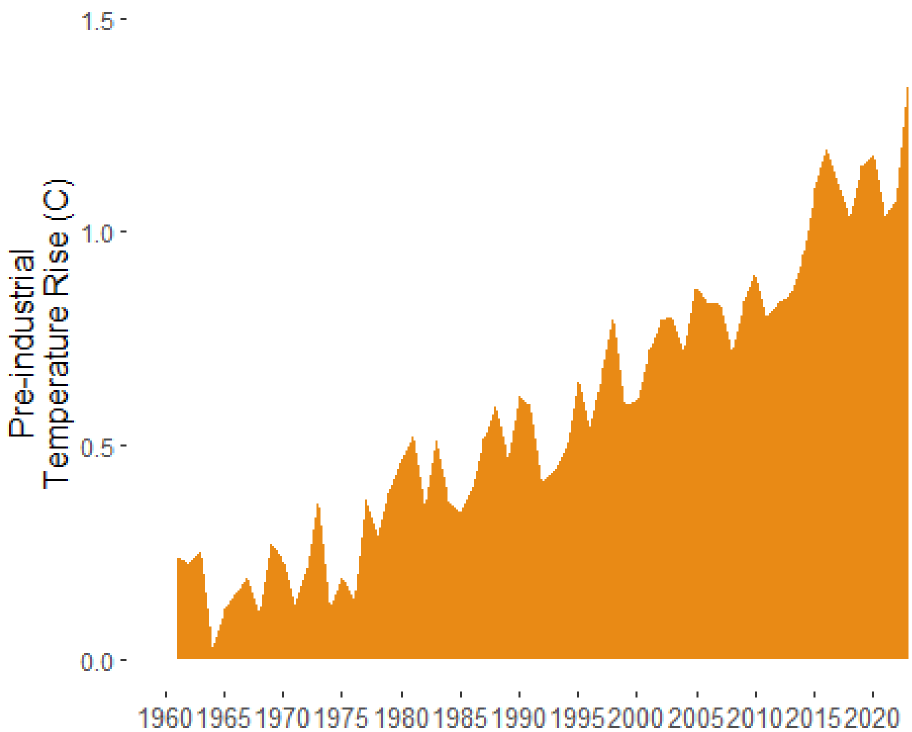
```
Fig4a <- ggplot(data = prop.CC.ext10.df) +
  geom_point(aes(x = Year, y = prop.CC.exts10),color = "#e98a15", fill = "#e9
8a15", shape = 23, size = 3) +
   theme(axis.title=element_text(size=12), axis.text.x = element_text(size =
10),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(
),panel.background = element_rect(fill = "white")) +
    scale_x_continuous(NULL, limits = c(Year.threshold, 2023), breaks = seq(Y
ear.threshold,2023,5)) +
  scale_y_continuous(name = "Decadal proportion of \n climate extinctions", l
imits = c(0,.21),breaks = seq(0,.21,.05))
Fig4a
```



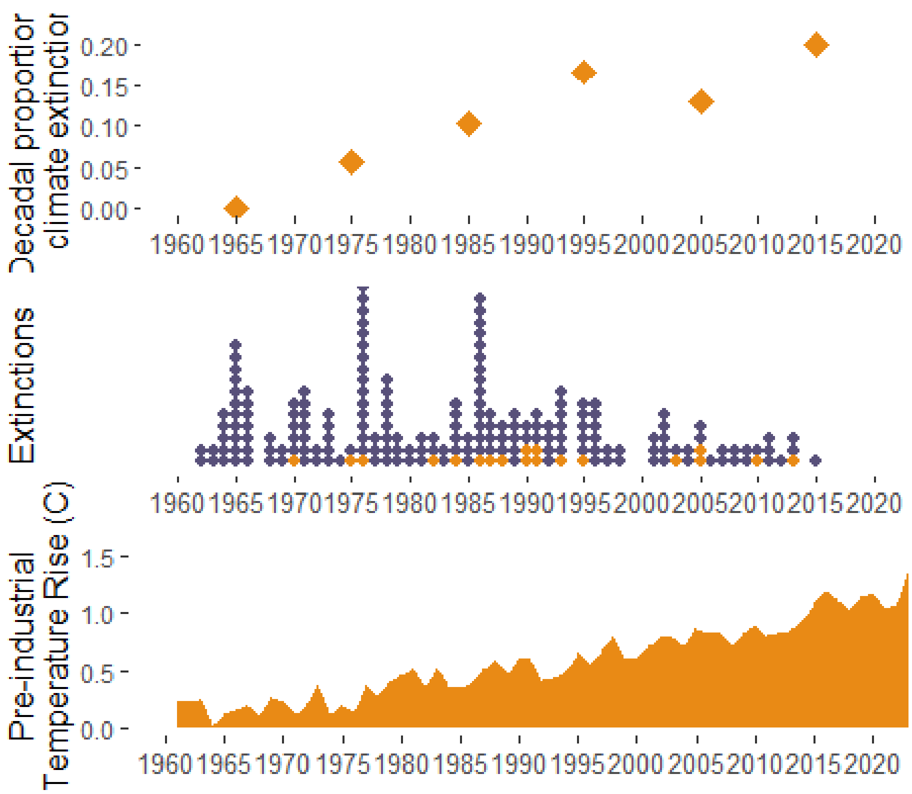

```
Fig4b <- ggplot() +
  geom_dotplot(data = all.ext.years, aes(x=all.ext.years),binwidth = 1, fill
= "#58507A", color = "#58507A", method = "histodot") +
  geom_dotplot(data = CC.ext.years, aes(x=CC.ext.years),binwidth = 1, color =
"#e98a15", fill = "#e98a15", method = "histodot") +
  theme(axis.title=element_text(size=12), axis.text.x = element_text(size = 1
0),axis.text.y = element_text(color = "white"),
        axis.ticks.y = element_line(color = "white"), panel.grid.major = elem
ent_blank(), panel.grid.minor = element_blank(),panel.background = element_re
ct(fill = "white")) +
  scale_y_continuous(name = "Extinctions \n ") +
  scale_x_continuous(NULL, limits = c(Year.threshold, 2023),breaks = seq(Year
.threshold,2023,5)) +
  guides(size=F) #+ theme_classic()

## 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.

Fig4b
```




```
Fig4c <- ggplot() +
  geom_area(data = global.temps.sub, aes(x = Year, y = Anomaly.C), fill = "#e
98a15") +
  ylab("Pre-industrial \n Temperature Rise (C)") + ylim(0,1.5) +
  scale_x_continuous(NULL, limits = c(Year.threshold, 2023),breaks = seq(Year
.threshold,2023,5)) +
  theme(axis.title.y=element_text(size=12), axis.title.x=element_blank(), axi
s.text.x = element_text(size=10),
        legend.position = "none",panel.grid.major = element_blank(), panel.gr
id.minor = element_blank(),panel.background = element_rect(fill = "white"))#+
Fig4c
```




```r
ggarrange(Fig4a, NULL, Fig4b, NULL, Fig4c, nrow=5, heights = c(4, 0.05, 4, 0.05, 4))
```



```
#ggsave("Metarisk2 extinction now.png",width=6,height=7,unit="in",dpi=2400)
```