Rerun of Clements et al data corrected and complete 1 for smallest

Anon

06/04/2022

————————————– Supplementary information to ————————————-

——————— META-ANALYSIS REVEALS AN EXTREME "DECLINE EFFECT" ———————
——————— IN OCEAN ACIDIFICATION IMPACTS ON FISH BEHAVIOUR ———————-

——– Jeff C. Clements, Josefin Sundin, Timothy D. Clark, Fredrik Jutfelt ———-

---

Get packages

```
library(pacman)

## Warning: package 'pacman' was built under R version 4.1.2

pacman::p_load(metafor, MCMCglmm, tidyverse, rotl, magrittr, kableExtra,
rmarkdown,gridExtra, psych, bindrcpp, pander)
library(BiocManager)

## Warning: package 'BiocManager' was built under R version 4.1.2

library(ggplot2)
library(viridis)

## Loading required package: viridisLite

library(patchwork)

## Warning: package 'patchwork' was built under R version 4.1.2
```

META-ANALYSIS - YEAR ONLINE - FULL DATASET

##attach dataset

```
decline<-read.csv(file.choose()) ##use dataset "S5 Data"
attach(decline)
```

#Note S5 corrected and complete 1 for smallest

```
head(decline)

##   obs study      authors year.online year.print if.at.pub X2017.if
if.group
## 1   1    a1 Munday et al        2009       2009     9.432    9.504
J
## 2   2    a1 Munday et al        2009       2009     9.432    9.504
J
## 3   3    a1 Munday et al        2009       2009     9.432    9.504
J
## 4   4    a1 Munday et al        2009       2009     9.432    9.504
J
## 5   5    a1 Munday et al        2009       2009     9.432    9.504
J
## 6   6    a1 Munday et al        2009       2009     9.432    9.504
J
##   avg.n            species climate cue cue.type life.stage ctrl.n
ctrl.mean
## 1    27 Amphiprion percula    Trop Yes  Habitat     Larvae     26
94.129
## 2    NA Amphiprion percula    Trop Yes  Habitat     Larvae     20
0.783
## 3    NA Amphiprion percula    Trop Yes  Habitat     Larvae     20
46.380
## 4    NA Amphiprion percula    Trop Yes  Habitat     Larvae     10
98.826
## 5    NA Amphiprion percula    Trop Yes      Kin     Larvae     30
0.912
## 6    NA Amphiprion percula    Trop Yes      Kin     Larvae     30
90.876
##     ctrl.sd oa.n oa.mean        oa.sd
## 1  2.993124   46  72.789 13.273019780
## 2  3.501682   46  83.562 10.621128750
## 3 14.002258   46  71.429 13.273019780
## 4  1.856257   16  88.258  7.044000000
## 5  2.996042   20  99.818  0.004472136
## 6  3.998375   20  99.818  0.004472136
```

##set factors

```
decline$year.online<-as.factor(decline$year.online)
decline$year.print<-as.factor(decline$year.print)
decline$obs<-as.factor(decline$obs)
decline$study<-as.factor(decline$study)
```

##view summary

```
summary(decline)

##       obs          study        authors           year.online    year.print
##  1      :  1    a3      : 48    Length:786         2018    :153   2018   :173
```

```
##  2       :  1    a87     : 40   Class :character   2015    :105   2015    : 95
##  3       :  1    a90     : 36   Mode  :character   2017    : 85   2016    : 82
##  4       :  1    a31     : 28                      2014    : 83   2013    : 78
##  5       :  1    a22     : 24                      2012    : 79   2012    : 71
##  6       :  1    a73     : 22                      2016    : 57   2017    : 71
##  (Other):780    (Other):588                       (Other):224   (Other):216
##   if.at.pub            X2017.if            if.group             avg.n
##  Length:786         Length:786         Length:786         Min.   :  4.0
##  Class :character   Class :character   Class :character   1st Qu.: 12.0
##  Mode  :character   Mode  :character   Mode  :character   Median : 18.0
##                                                           Mean   : 32.2
##                                                           3rd Qu.: 30.0
##                                                           Max.   :568.0
##                                                           NA's   :691
##    species            climate               cue              cue.type
##  Length:786         Length:786         Length:786         Length:786
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   life.stage            ctrl.n          ctrl.mean             ctrl.sd
##  Length:786         Min.   :  3.00   Min.   :   -69.78   Min.   :    0.000
##  Class :character   1st Qu.: 10.00   1st Qu.:     1.18   1st Qu.:    1.130
##  Mode  :character   Median : 18.00   Median :     9.98   Median :    5.402
##                     Mean   : 29.12   Mean   :   454.59   Mean   :  107.674
##                     3rd Qu.: 30.00   3rd Qu.:    45.29   3rd Qu.:   21.646
##                     Max.   :752.00   Max.   :154936.88   Max.   :25490.446
##
##       oa.n           oa.mean               oa.sd
##  Min.   :  2.00   Min.   :   -59.67   Min.   :    0.00
##  1st Qu.: 10.00   1st Qu.:     1.38   1st Qu.:     1.08
##  Median : 18.00   Median :    13.58   Median :     7.02
##  Mean   : 29.02   Mean   :   454.15   Mean   :   117.90
##  3rd Qu.: 34.00   3rd Qu.:    44.43   3rd Qu.:    22.00
##  Max.   :755.00   Max.   :157061.25   Max.   :36812.37
##
```

##subset by year

```
y2009 <- filter(decline, year.online == "2009")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2009$obs <- 1:nrow(y2009)
y2010 <- filter(decline, year.online == "2010")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2010$obs <- 1:nrow(y2010)
y2011 <- filter(decline, year.online == "2011")[,-
```

```
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2011$obs <- 1:nrow(y2011)
y2012 <- filter(decline, year.online == "2012")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2012$obs <- 1:nrow(y2012)
y2013 <- filter(decline, year.online == "2013")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2013$obs <- 1:nrow(y2013)
y2014 <- filter(decline, year.online == "2014")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2014$obs <- 1:nrow(y2014)
y2015 <- filter(decline, year.online == "2015")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2015$obs <- 1:nrow(y2015)
y2016 <- filter(decline, year.online == "2016")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2016$obs <- 1:nrow(y2016)
y2017 <- filter(decline, year.online == "2017")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2017$obs <- 1:nrow(y2017)
y2018 <- filter(decline, year.online == "2018")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2018$obs <- 1:nrow(y2018)
y2019 <- filter(decline, year.online == "2019")[,-
match(c("avg.n","cue.type","if.at.pub","X2017.if","if.group"), colnames
(decline))]
y2019$obs <- 1:nrow(y2019)
```

##comupte effect sizes for each year

```
lnRR2009 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
009,append=TRUE)
lnRR2010 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
010,append=TRUE)
lnRR2011 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
011,append=TRUE)
lnRR2012 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
012,append=TRUE)
```

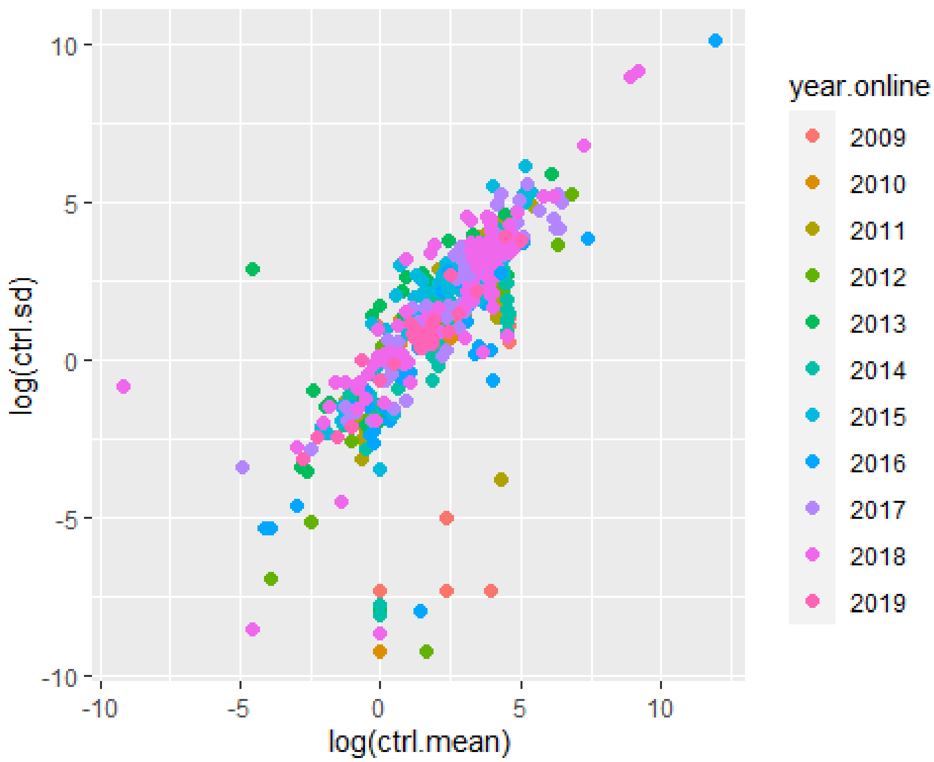
```
lnRR2013 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
013,append=TRUE)
lnRR2014 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
014,append=TRUE)
lnRR2015 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
015,append=TRUE)
lnRR2016 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
016,append=TRUE)
lnRR2017 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
017,append=TRUE)
lnRR2018 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
018,append=TRUE)
lnRR2019 <- escalc(measure= "ROM",
m1i=oa.mean,sd1i=oa.sd,n1i=oa.n,m2i=ctrl.mean,sd2i=ctrl.sd,n2i=ctrl.n,data=y2
019,append=TRUE)

#Note log(m1i/m21i) produced NAs for 2011, 2012, 2013, 2015, 2017
```

##remove NAs

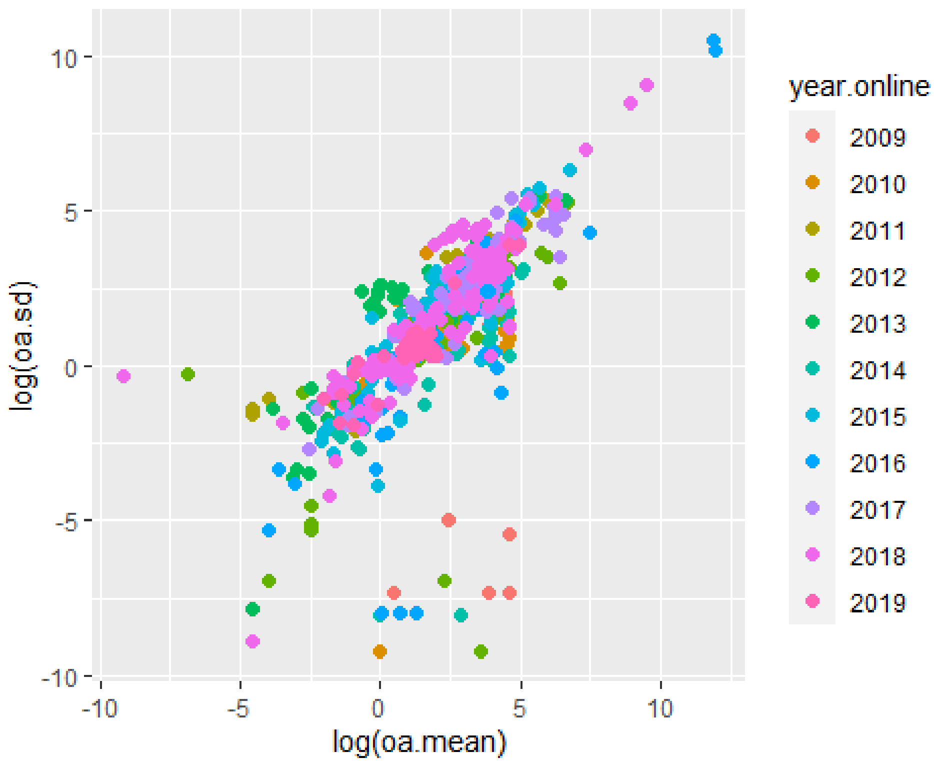
```
lnRR2009clean<-na.omit(lnRR2009)
lnRR2010clean<-na.omit(lnRR2010)
lnRR2011clean<-na.omit(lnRR2011)
lnRR2012clean<-na.omit(lnRR2012)
lnRR2013clean<-na.omit(lnRR2013)
lnRR2014clean<-na.omit(lnRR2014)
lnRR2015clean<-na.omit(lnRR2015)
lnRR2016clean<-na.omit(lnRR2016)
lnRR2017clean<-na.omit(lnRR2017)
lnRR2018clean<-na.omit(lnRR2018)
lnRR2019clean<-na.omit(lnRR2019)
```

##view mean-variance relationship

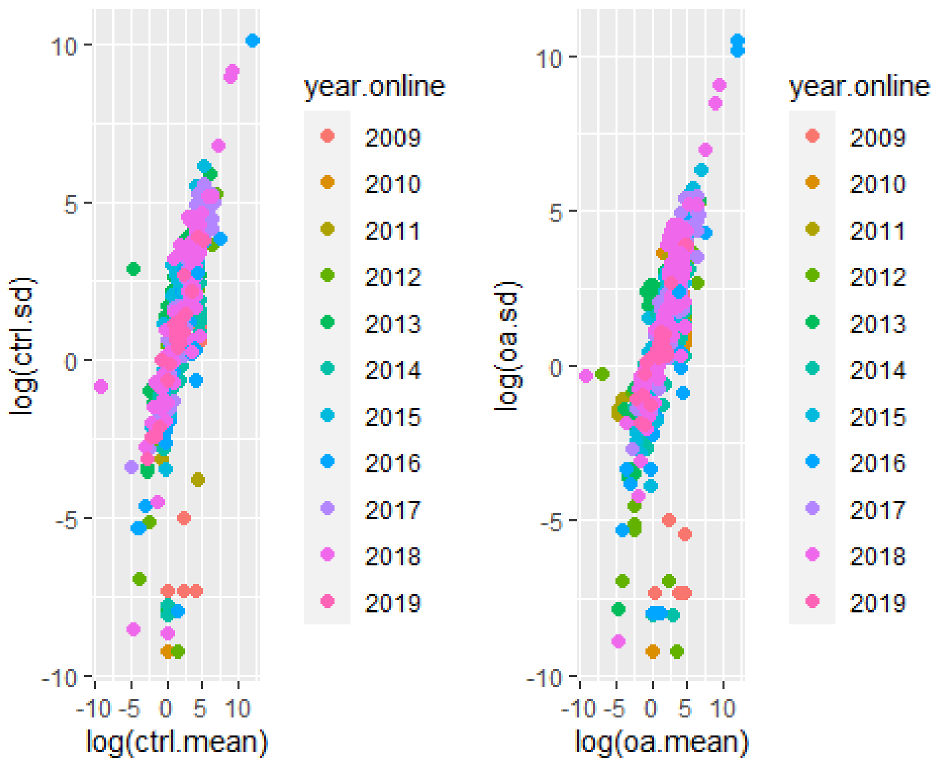
```
pp1<-ggplot(decline,aes(x=log(ctrl.mean),y=log(ctrl.sd),col=year.online))+
geom_point(size=2,na.rm = TRUE)
pp2<-ggplot(decline,aes(x=log(oa.mean),y=log(oa.sd),col=year.online))+
geom_point(size=2,na.rm = TRUE)
pp1 #Note added

## Warning in log(ctrl.mean): NaNs produced

## Warning in log(ctrl.mean): NaNs produced
```




```
pp2 #Note added

## Warning in log(oa.mean): NaNs produced

## Warning in log(oa.mean): NaNs produced
```




```
grid.arrange(pp1,pp2, nrow =1)

## Warning in log(ctrl.mean): NaNs produced

## Warning in log(ctrl.mean): NaNs produced

## Warning in log(oa.mean): NaNs produced

## Warning in log(oa.mean): NaNs produced
```




```
#Note there are a number of NAs created due the the log of a negative being
not computible
```

##look at lnRR by year

```
MLMA_2009_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2009)

## Warning: Ratio of largest to smallest sampling variance extremely large.
May not
## be able to obtain stable results.

summary(MLMA_2009_lnRR)

##
## Multivariate Meta-Analysis Model (k = 19; method: REML)
##
##    logLik  Deviance       AIC       BIC      AICc
## -39.8746   79.7493   83.7493   85.5300   84.5493
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed  factor
## sigma^2    4.7709  2.1842     19     no     obs
##
## Test for Heterogeneity:
## Q(df = 18) = 4124865078.0293, p-val < .0001
##
## Model Results:
```

```
##
## estimate      se    zval    pval   ci.lb   ci.ub
##   1.2647  0.5046  2.5062  0.0122  0.2757  2.2538  *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2010_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2010)
```

```
## Warning: Ratio of largest to smallest sampling variance extremely large.
May not
## be able to obtain stable results.
```

```
summary(MLMA_2010_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 48; method: REML)
##
##   logLik  Deviance       AIC       BIC      AICc
## -91.7106  183.4213  187.4213  191.1216  187.6940
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed  factor
## sigma^2    2.8052  1.6749     48     no     obs
##
## Test for Heterogeneity:
## Q(df = 47) = 1407483.9857, p-val < .0001
##
## Model Results:
##
## estimate      se    zval    pval   ci.lb   ci.ub
##   1.6112  0.2473  6.5150  <.0001  1.1265  2.0959  ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2011_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2011)
summary(MLMA_2011_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 51; method: REML)
##
##   logLik  Deviance       AIC       BIC      AICc
## -45.9064   91.8128   95.8128   99.6368   96.0681
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed  factor
## sigma^2    0.1494  0.3865     51     no     obs
##
```

```
## Test for Heterogeneity:
## Q(df = 50) = 230.2425, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb   ci.ub
##  -0.0421  0.0715  -0.5886  0.5561  -0.1823  0.0981
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2012_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2012)
```

```
## Warning: Ratio of largest to smallest sampling variance extremely large.
May not
## be able to obtain stable results.
```

```
summary(MLMA_2012_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 79; method: REML)
##
##    logLik   Deviance        AIC        BIC       AICc
## -140.8044   281.6089   285.6089   290.3223   285.7689
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed  factor
## sigma^2    1.2981  1.1393     79     no     obs
##
## Test for Heterogeneity:
## Q(df = 78) = 29286.6213, p-val < .0001
##
## Model Results:
##
## estimate      se    zval    pval    ci.lb   ci.ub
##   0.1285  0.1401  0.9170  0.3591  -0.1461  0.4031
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2013_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2013)
```

```
## Warning: Ratio of largest to smallest sampling variance extremely large.
May not
## be able to obtain stable results.
```

```
summary(MLMA_2013_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 54; method: REML)
##
```

```
##   logLik  Deviance       AIC       BIC      AICc
## -94.3731  188.7463  192.7463  196.6869  192.9863
##
## Variance Components:
##
##               estim    sqrt  nlvls  fixed  factor
## sigma^2    1.0529  1.0261     54     no     obs
##
## Test for Heterogeneity:
## Q(df = 53) = 488.1779, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb    ci.ub
##  -0.4809  0.1672  -2.8769  0.0040  -0.8086  -0.1533  **
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2014_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2014)
summary(MLMA_2014_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 83; method: REML)
##
##    logLik   Deviance        AIC        BIC       AICc
## -160.6395   321.2790   325.2790   330.0925   325.4309
##
## Variance Components:
##
##               estim    sqrt  nlvls  fixed  factor
## sigma^2    2.7935  1.6714     83     no     obs
##
## Test for Heterogeneity:
## Q(df = 82) = 631228.7966, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb   ci.ub
##  -0.2355  0.1869  -1.2601  0.2076  -0.6018  0.1308
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2015_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2015)
summary(MLMA_2015_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 105; method: REML)
##
##   logLik  Deviance       AIC       BIC      AICc
```

```
## -77.7151  155.4302  159.4302  164.7190  159.5490
##
## Variance Components:
##
##               estim    sqrt  nlvls  fixed  factor
## sigma^2    0.0887  0.2979    105     no     obs
##
## Test for Heterogeneity:
## Q(df = 104) = 368.7129, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb    ci.ub
##  -0.0936  0.0410  -2.2806  0.0226  -0.1740  -0.0132  *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2016_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2016)
```

```
## Warning: Ratio of largest to smallest sampling variance extremely large.
May not
## be able to obtain stable results.
```

```
summary(MLMA_2016_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 57; method: REML)
##
##   logLik  Deviance       AIC       BIC      AICc
## -43.5933   87.1866   91.1866   95.2373   91.4130
##
## Variance Components:
##
##               estim    sqrt  nlvls  fixed  factor
## sigma^2    0.2241  0.4734     57     no     obs
##
## Test for Heterogeneity:
## Q(df = 56) = 70900.5484, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb   ci.ub
##  -0.0350  0.0672  -0.5211  0.6023  -0.1666  0.0966
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2017_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2017)
summary(MLMA_2017_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 85; method: REML)
##
##   logLik  Deviance       AIC       BIC      AICc
## -61.0486  122.0972  126.0972  130.9589  126.2454
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed  factor
## sigma^2    0.1465  0.3828     85     no     obs
##
## Test for Heterogeneity:
## Q(df = 84) = 2256.3271, p-val < .0001
##
## Model Results:
##
## estimate      se    zval    pval    ci.lb   ci.ub
##   0.0186  0.0493  0.3777  0.7057  -0.0780  0.1152
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2018_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2018)
```

```
## Warning: Ratio of largest to smallest sampling variance extremely large.
May not
## be able to obtain stable results.
```

```
summary(MLMA_2018_lnRR)
```

```
##
## Multivariate Meta-Analysis Model (k = 153; method: REML)
##
##    logLik   Deviance        AIC        BIC       AICc
## -239.3455   478.6911   482.6911   488.7388   482.7716
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed  factor
## sigma^2    0.7201  0.8486    153     no     obs
##
## Test for Heterogeneity:
## Q(df = 152) = 1572.6246, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb   ci.ub
##  -0.0315  0.0755  -0.4172  0.6765  -0.1794  0.1164
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MLMA_2019_lnRR <- rma.mv(yi ~ 1, V = vi, random=~1|obs, data=lnRR2019)
summary(MLMA_2019_lnRR)

##
## Multivariate Meta-Analysis Model (k = 52; method: REML)
##
##   logLik  Deviance       AIC       BIC      AICc
## -47.0733   94.1465   98.1465  102.0102   98.3965
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed  factor
## sigma^2    0.1813  0.4258     52     no     obs
##
## Test for Heterogeneity:
## Q(df = 51) = 302.1940, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb   ci.ub
##  -0.0709  0.0646  -1.0963  0.2730  -0.1976  0.0558
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

##set prior

```
prior <- list(R=list(V = 1, nu =0.002), G = list(G = list(V=1, nu = 0.002)))
```

##run bayesian MLMA models

```
model_magnitude_bayes_2009 <- MCMCglmm(yi ~ 1, mev = lnRR2009clean$vi, random
= ~obs, data = lnRR2009clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2010 <- MCMCglmm(yi ~ 1, mev = lnRR2010clean$vi, random
= ~obs, data = lnRR2010clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2011 <- MCMCglmm(yi ~ 1, mev = lnRR2011clean$vi, random
= ~obs, data = lnRR2011clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2012 <- MCMCglmm(yi ~ 1, mev = lnRR2012clean$vi, random
= ~obs, data = lnRR2012clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2013 <- MCMCglmm(yi ~ 1, mev = lnRR2013clean$vi, random
= ~obs, data = lnRR2013clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2014 <- MCMCglmm(yi ~ 1, mev = lnRR2014clean$vi, random
= ~obs, data = lnRR2014clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2015 <- MCMCglmm(yi ~ 1, mev = lnRR2015clean$vi, random
= ~obs, data = lnRR2015clean, prior = prior, burnin = 10000, nitt = 1000000,
```

```
thin = 100, verbose = FALSE)
model_magnitude_bayes_2016 <- MCMCglmm(yi ~ 1, mev = lnRR2016clean$vi, random
= ~obs, data = lnRR2016clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2017 <- MCMCglmm(yi ~ 1, mev = lnRR2017clean$vi, random
= ~obs, data = lnRR2017clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2018 <- MCMCglmm(yi ~ 1, mev = lnRR2018clean$vi, random
= ~obs, data = lnRR2018clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
model_magnitude_bayes_2019 <- MCMCglmm(yi ~ 1, mev = lnRR2019clean$vi, random
= ~obs, data = lnRR2019clean, prior = prior, burnin = 10000, nitt = 1000000,
thin = 100, verbose = FALSE)
```

##get model summaries

```
summary(model_magnitude_bayes_2009)

##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: 25.50312
##
##  G-structure:  ~obs
##
##     post.mean l-95% CI u-95% CI eff.samp
## obs     2.775 0.000254    7.682     3382
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units     2.641 0.0001652    7.454     3402
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)    1.2576   0.1623   2.2779     9900 0.0214 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(model_magnitude_bayes_2010)

##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: 47.96303
##
```

```
##  G-structure:  ~obs
##
##     post.mean l-95% CI u-95% CI eff.samp
## obs     1.503 0.000221    3.592     1465
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units      1.44 0.0001554    3.581     1481
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)     1.608    1.092    2.094     9900 <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_magnitude_bayes_2011)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: -22.41812
##
##  G-structure:  ~obs
##
##     post.mean l-95% CI u-95% CI eff.samp
## obs   0.08059 0.000156   0.2066     4074
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units   0.07997 0.0002795   0.2052     4046
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)  -0.04295 -0.18941  0.10260     9900 0.557
```

```
summary(model_magnitude_bayes_2012)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: 59.08081
##
##  G-structure:  ~obs
```

```
##
##     post.mean  l-95% CI u-95% CI eff.samp
## obs    0.6783 0.0002013    1.587     1221
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units    0.6667 0.0001916    1.577     1145
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)    0.1284  -0.1513   0.4142     9900 0.367
```

```
summary(model_magnitude_bayes_2013)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: 43.18556
##
##  G-structure:  ~obs
##
##     post.mean  l-95% CI u-95% CI eff.samp
## obs    0.5386 0.0001451      1.4     1666
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units    0.5793 0.0001794    1.418     1650
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp   pMCMC
## (Intercept)   -0.4834  -0.8329  -0.1558     9900 0.00485 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_magnitude_bayes_2014)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: 111.3108
##
##  G-structure:  ~obs
##
```

```
##     post.mean  l-95% CI u-95% CI eff.samp
## obs      1.36 0.0001924    3.311      794
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units     1.513 0.0002124    3.382    871.8
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)   -0.2372  -0.6166   0.1277     9900 0.215
```

```
summary(model_magnitude_bayes_2015)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: -75.42929
##
##  G-structure:  ~obs
##
##     post.mean  l-95% CI u-95% CI eff.samp
## obs   0.04687 0.0002211   0.1109     2555
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units   0.04625 0.0002058   0.1104     2502
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)  -0.09389 -0.17484 -0.01138     9900 0.0271 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_magnitude_bayes_2016)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: -13.02221
##
##  G-structure:  ~obs
##
##     post.mean l-95% CI u-95% CI eff.samp
```

```
## obs    0.1165 0.000213   0.2809     2837
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units    0.1199 0.0002043   0.2845     2914
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)  -0.03432 -0.16933  0.09890     9900 0.618
```

```
summary(model_magnitude_bayes_2017)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: -38.78002
##
##  G-structure:  ~obs
##
##     post.mean  l-95% CI u-95% CI eff.samp
## obs   0.07534 0.0002619   0.1786     2250
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units    0.0773 0.0002296   0.1798     2316
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)   0.01866 -0.07752  0.11824     9900 0.714
```

```
summary(model_magnitude_bayes_2018)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: 46.44447
##
##  G-structure:  ~obs
##
##     post.mean  l-95% CI u-95% CI eff.samp
## obs    0.3761 0.0002349   0.8502    667.7
##
##  R-structure:  ~units
```

```
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     0.358 0.000263   0.8427    668.2
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)   -0.0306  -0.1856   0.1112     9900 0.677
```

```
summary(model_magnitude_bayes_2019)
```

```
##
##  Iterations = 10001:999901
##  Thinning interval  = 100
##  Sample size  = 9900
##
##  DIC: -22.40152
##
##  G-structure:  ~obs
##
##     post.mean  l-95% CI u-95% CI eff.samp
## obs   0.09641 0.0002422   0.2465     3493
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI u-95% CI eff.samp
## units   0.09703 0.0001994   0.2452     3606
##
##  Location effects: yi ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)  -0.07104 -0.20086  0.06430     9900 0.282
```

##extract posteriors

```
sol2009 <- model_magnitude_bayes_2009$Sol
VCV2009 <- model_magnitude_bayes_2009$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2009$VCV))]
sol2010 <- model_magnitude_bayes_2010$Sol
VCV2010 <- model_magnitude_bayes_2010$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2010$VCV))]
sol2011 <- model_magnitude_bayes_2011$Sol
VCV2011 <- model_magnitude_bayes_2011$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2011$VCV))]
sol2012 <- model_magnitude_bayes_2012$Sol
VCV2012 <- model_magnitude_bayes_2012$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2012$VCV))]
sol2013 <- model_magnitude_bayes_2013$Sol
VCV2013 <- model_magnitude_bayes_2013$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2013$VCV))]
sol2014 <- model_magnitude_bayes_2014$Sol
```

```
VCV2014 <- model_magnitude_bayes_2014$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2014$VCV))]
sol2015 <- model_magnitude_bayes_2015$Sol
VCV2015 <- model_magnitude_bayes_2015$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2015$VCV))]
sol2016 <- model_magnitude_bayes_2016$Sol
VCV2016 <- model_magnitude_bayes_2016$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2016$VCV))]
sol2017 <- model_magnitude_bayes_2017$Sol
VCV2017 <- model_magnitude_bayes_2017$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2017$VCV))]
sol2018 <- model_magnitude_bayes_2018$Sol
VCV2018 <- model_magnitude_bayes_2018$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2018$VCV))]
sol2019 <- model_magnitude_bayes_2019$Sol
VCV2019 <- model_magnitude_bayes_2019$VCV[,-match("sqrt(mev):sqrt(mev).meta",
colnames(model_magnitude_bayes_2019$VCV))]
```

##get folded normal function

```
mu.fnorm <- function(mu, sigma){dnorm(mu, 0, sigma)*2*sigma^2 +
mu*(2*pnorm(mu, 0, sigma) -1)}
```

##get magnitude means + variance

```
magnitude_mean_2009 <- mu.fnorm(sol2009[,1], sqrt(rowSums(VCV2009)))
magnitude_2009_mean <- data.frame(mean_mag = mean(magnitude_mean_2009), L_CI
= HPDinterval(magnitude_mean_2009)[1], U_CI =
HPDinterval(magnitude_mean_2009)[2])
magnitude_mean_2010 <- mu.fnorm(sol2010[,1], sqrt(rowSums(VCV2010)))
magnitude_2010_mean <- data.frame(mean_mag = mean(magnitude_mean_2010), L_CI
= HPDinterval(magnitude_mean_2010)[1], U_CI =
HPDinterval(magnitude_mean_2010)[2])
magnitude_mean_2011 <- mu.fnorm(sol2011[,1], sqrt(rowSums(VCV2011)))
magnitude_2011_mean <- data.frame(mean_mag = mean(magnitude_mean_2011), L_CI
= HPDinterval(magnitude_mean_2011)[1], U_CI =
HPDinterval(magnitude_mean_2011)[2])
magnitude_mean_2012 <- mu.fnorm(sol2012[,1], sqrt(rowSums(VCV2012)))
magnitude_2012_mean <- data.frame(mean_mag = mean(magnitude_mean_2012), L_CI
= HPDinterval(magnitude_mean_2012)[1], U_CI =
HPDinterval(magnitude_mean_2012)[2])
magnitude_mean_2013 <- mu.fnorm(sol2013[,1], sqrt(rowSums(VCV2013)))
magnitude_2013_mean <- data.frame(mean_mag = mean(magnitude_mean_2013), L_CI
= HPDinterval(magnitude_mean_2013)[1], U_CI =
HPDinterval(magnitude_mean_2013)[2])
magnitude_mean_2014 <- mu.fnorm(sol2014[,1], sqrt(rowSums(VCV2014)))
magnitude_2014_mean <- data.frame(mean_mag = mean(magnitude_mean_2014), L_CI
= HPDinterval(magnitude_mean_2014)[1], U_CI =
HPDinterval(magnitude_mean_2014)[2])
magnitude_mean_2015 <- mu.fnorm(sol2015[,1], sqrt(rowSums(VCV2015)))
magnitude_2015_mean <- data.frame(mean_mag = mean(magnitude_mean_2015), L_CI
```

```
= HPDinterval(magnitude_mean_2015)[1], U_CI =
HPDinterval(magnitude_mean_2015)[2])
magnitude_mean_2016 <- mu.fnorm(sol2016[,1], sqrt(rowSums(VCV2016)))
magnitude_2016_mean <- data.frame(mean_mag = mean(magnitude_mean_2016), L_CI
= HPDinterval(magnitude_mean_2016)[1], U_CI =
HPDinterval(magnitude_mean_2016)[2])
magnitude_mean_2017 <- mu.fnorm(sol2017[,1], sqrt(rowSums(VCV2017)))
magnitude_2017_mean <- data.frame(mean_mag = mean(magnitude_mean_2017), L_CI
= HPDinterval(magnitude_mean_2017)[1], U_CI =
HPDinterval(magnitude_mean_2017)[2])
magnitude_mean_2018 <- mu.fnorm(sol2018[,1], sqrt(rowSums(VCV2018)))
magnitude_2018_mean <- data.frame(mean_mag = mean(magnitude_mean_2018), L_CI
= HPDinterval(magnitude_mean_2018)[1], U_CI =
HPDinterval(magnitude_mean_2018)[2])
magnitude_mean_2019 <- mu.fnorm(sol2019[,1], sqrt(rowSums(VCV2019)))
magnitude_2019_mean <- data.frame(mean_mag = mean(magnitude_mean_2019), L_CI
= HPDinterval(magnitude_mean_2019)[1], U_CI =
HPDinterval(magnitude_mean_2019)[2])
```

##view ES magnitudes and uncertainty

```
magnitude_2009_mean

##   mean_mag     L_CI     U_CI
## 1 2.142856 1.500926 2.919172

magnitude_2010_mean

##   mean_mag     L_CI     U_CI
## 1  1.93814 1.580976 2.319668

magnitude_2011_mean

##    mean_mag      L_CI     U_CI
## 1 0.3230558 0.2282608 0.430983

magnitude_2012_mean

##    mean_mag      L_CI     U_CI
## 1 0.9341205 0.7736961 1.106872

magnitude_2013_mean

##    mean_mag      L_CI     U_CI
## 1 0.9342499 0.7086135 1.181569

magnitude_2014_mean

##   mean_mag     L_CI     U_CI
## 1 1.369658 1.148612 1.596379

magnitude_2015_mean
```

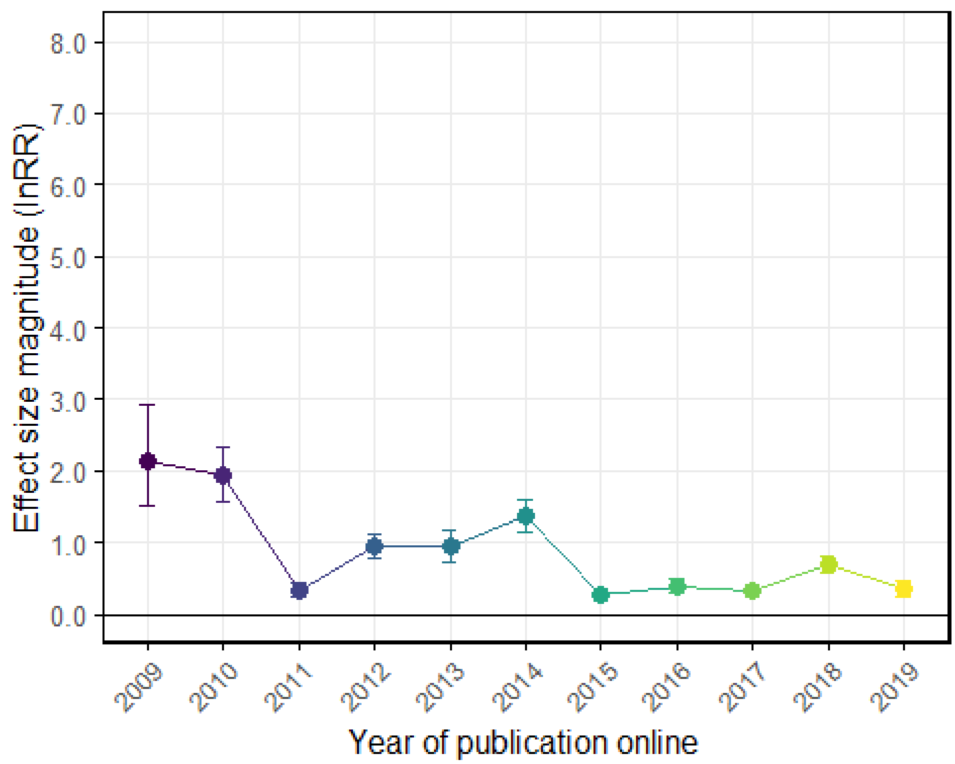
```
##    mean_mag      L_CI      U_CI
## 1 0.2552839 0.1933206 0.3177689

magnitude_2016_mean

##   mean_mag      L_CI      U_CI
## 1 0.390167 0.3060117 0.4823499

magnitude_2017_mean

##    mean_mag      L_CI      U_CI
## 1 0.3127445 0.2462618 0.3801782

magnitude_2018_mean

##   mean_mag      L_CI      U_CI
## 1 0.684231 0.5706654 0.7982046

magnitude_2019_mean

##    mean_mag      L_CI      U_CI
## 1 0.3553184 0.2490392 0.4625114
```

#Note Code below not in original. Made to allow a data frame to be constructed and plotted from.

```
magnitudedata = rbind(magnitude_2009_mean, magnitude_2010_mean,
magnitude_2011_mean, magnitude_2012_mean, magnitude_2013_mean,
magnitude_2014_mean, magnitude_2015_mean, magnitude_2016_mean,
magnitude_2017_mean, magnitude_2018_mean, magnitude_2019_mean)

yearlabel = c("2009", "2010", "2011", "2012", "2013", "2014", "2015", "2016",
"2017", "2018", "2019")

magnitudedata = cbind(yearlabel, magnitudedata )
magnitudedata

##    yearlabel  mean_mag      L_CI      U_CI
## 1       2009 2.1428562 1.5009257 2.9191723
## 2       2010 1.9381398 1.5809756 2.3196675
## 3       2011 0.3230558 0.2282608 0.4309830
## 4       2012 0.9341205 0.7736961 1.1068723
## 5       2013 0.9342499 0.7086135 1.1815686
## 6       2014 1.3696578 1.1486118 1.5963790
## 7       2015 0.2552839 0.1933206 0.3177689
## 8       2016 0.3901670 0.3060117 0.4823499
## 9       2017 0.3127445 0.2462618 0.3801782
## 10      2018 0.6842310 0.5706654 0.7982046
## 11      2019 0.3553184 0.2490392 0.4625114
```

#Note Plot from above model was not included in the original code. This has been built from scratch. Should be the same as Fig 1B

```
Decline_magnitude<-ggplot(magnitudedata,aes(x=yearlabel, y=mean_mag, colour=yearlabel))  + geom_line(aes(group=1)) + scale_color_viridis(discrete=TRUE)+ geom_point(size=3) +  geom_errorbar(aes (ymin = L_CI, ymax = U_CI), width=0.2) +  scale_y_continuous(breaks = seq(0, 15, by = 1), minor_breaks = NULL, limits=c(0,8),labels = scales::number_format(accuracy = 0.1)) +  theme(legend.position = "none") + xlab("Year of publication online") + ylab("Effect size magnitude (lnRR)")  + theme_minimal(12) + theme(panel.border = element_rect(colour = "black", fill=NA, size=1)) + theme(legend.position = "none") + theme(axis.ticks=element_line()) + theme(axis.text.x = element_text(angle = 45, hjust=1))+ geom_hline(yintercept = 0)

Decline_magnitude
```




```
ggsave(Decline_magnitude, filename = 'Decline magnitude corrected and complete model 1 for smallest lnRR.png', device=png,  width = 4.2, height = 4.6, units = "in", res = 800)
```

CREATE SCATTERPLOT FIGURE TO VISUALIZE MEAN EFFECT SIZE MAGNITUDE FOR EACH OBSERVATION OVER TIME (FIG 1A)

##attach dataset

```
decline_allobs<-read.csv(file.choose()) ##use dataset "S10 Data"
attach(decline_allobs)
```

```
## The following objects are masked from decline:
##
##     ctrl.n, obs, study, year.online, year.print
```

```
summary(decline_allobs)
```

```
##       obs           study            year.online     year.print
##  Min.   :  1.0   Length:839         Min.   :2009   Min.   :2009
##  1st Qu.:210.5   Class :character   1st Qu.:2012   1st Qu.:2013
##  Median :420.0   Mode  :character   Median :2015   Median :2015
##  Mean   :420.0                      Mean   :2015   Mean   :2015
##  3rd Qu.:629.5                      3rd Qu.:2017   3rd Qu.:2018
##  Max.   :839.0                      Max.   :2019   Max.   :2019
##
##      ctrl.n          lnrr.mag
##  Min.   :  3.00   Min.   :0.0000
##  1st Qu.: 10.00   1st Qu.:0.1020
##  Median : 18.00   Median :0.2806
##  Mean   : 28.77   Mean   :0.7628
##  3rd Qu.: 30.00   3rd Qu.:0.8243
##  Max.   :752.00   Max.   :8.4360
##                   NA's   :53
```

##Create plot

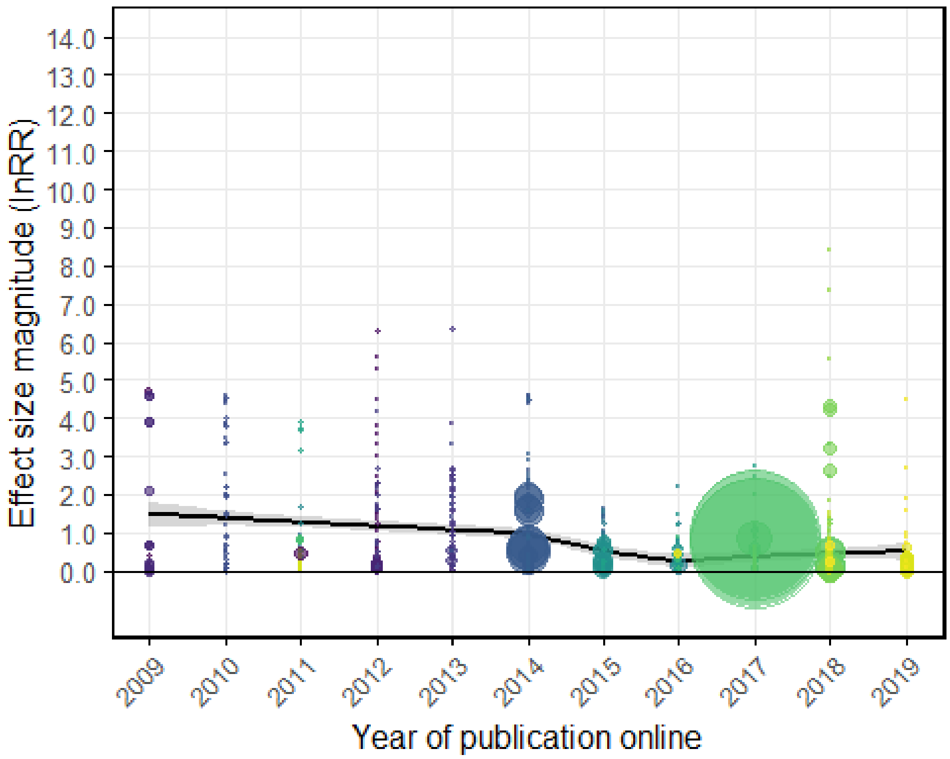
#Note I had to fix the specifications of the scale continuous both x and y Original below not working: scale_x_continuous(breaks = round(seq(min(study$Year$), max(study$Year), by = 1),1)) + scale_y_continuous(breaks = round(seq(min(study$lnrr), 15, by = 1),1)) #Note Code given did not produce a graph that looked exactly the same as in the figures of the paper. Changes to aesthetics were add to make it look the same as the published version.

```
Decline_studies_loess<-ggplot(decline_allobs,aes(x=year.online, y=lnrr.mag,
color=study)) + geom_smooth(method="loess", se=TRUE, fullrange=TRUE,
level=0.95,color="black") + geom_point(size=ctrl.n*0.03,alpha=0.6) +
scale_size(range = c(1, 2), name="Sample size")+
scale_color_viridis(discrete=TRUE)+ xlab("Year of publication
online")+ylab("Effect size magnitude (lnRR)") + scale_x_continuous(breaks =
round(seq(min(2009), max(2019), by = 1),1)) + scale_y_continuous(breaks =
seq(0, 14, by = 1), minor_breaks = NULL, limits=c(-1,14),labels =
scales::number_format(accuracy = 0.1)) + theme_minimal(12) +
theme(legend.position = "none") + theme(panel.grid.minor = element_blank())
+ theme(panel.border = element_rect(colour = "black", fill=NA, size=1)) +
theme(legend.position = "none")+ theme(axis.ticks=element_line()) +
theme(axis.text.x = element_text(angle = 45, hjust=1))+ geom_hline(yintercept
= 0)

Decline_studies_loess
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 53 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 53 rows containing missing values (geom_point).
```




```
ggsave(Decline_studies_loess, filename = 'Decline magnitude corrected and
complete 1 for smallest lnRR.png', device=png,  width = 4.2, height = 4.6,
units = "in", res = 800)

## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 53 rows containing non-finite values (stat_smooth).

## Warning: Removed 53 rows containing missing values (geom_point).
```