S2 Supplementary material: R script for repeat data analysis

```
# --- ROBUST DESIGN MARMOT PNPPSM 2019-2021 --- #

library(RMark)

# Import data

Input_file_Robust = convert.inp("select directory/S1_Robust.inp", group.df = NULL, use.comments = TRUE)

Input_file_Robust

# Set time intervals for each primary occasion (year)

#/* Sessions 1-2 = 2019 */

#/* Sessions 3-4 = 2020 */

#/* Sessions 5-6 = 2021 */

time_intervals_Robust=c(0,1,0,1,0)

marmot.proc_Robust=process.data(Input_file_Robust,model="Robust",time.intervals=time_intervals_Robust)

# Create the design data

marmot.ddl_Robust=make.design.data(marmot.proc_Robust)

marmot.ddl_Robust

# Global model

S.const=list(formula=~1)

GammaDoublePrime.random=list(formula= ~1,share=TRUE)

p.const=list(formula= ~1,share=TRUE)
```

```
model.global_Robust=mark(marmot.proc_Robust,marmot.ddl_Robust, model = "Robust",
        model.parameters=list(S=S.const,
                        GammaDoublePrime=GammaDoublePrime.random,
                        p=p.const),
        threads=2)

# Results
model.global_Robust$results$derived # best model
model.global_Robust$results$derived.vcv # best model
model.global_Robust$results$AICc

#N
#year  estimate  lcl      ucl
#2019: 19.45518 18.23343 27.07134
#2020: 15.00964 14.12410 22.21384
#2021: 23.89865 22.36090 31.98844

#SE
sqrt(2.9456414)
sqrt(2.1802240)
sqrt(3.7829843)

#CVs
sqrt(2.9456414)/19.45518*100 # 2019 = 8.8%
sqrt(2.1802240)/15.00964*100 # 2020 = 9.8%
sqrt(3.7829843)/23.89865*100 # 2021 = 8.1%
```

```
########################################

# MARK-RESIGHT MARMOT PNPPSM 2019-2021 #

########################################


# --- BOWDEN ESTIMATOR --- #

# returns N, CI, %CV, Variance and SE based on Bowden & Kufeld (1995)

# user needs to input:

# - ch = the "capture history" of individual sighting frequencies

# - unmarked.observed = the number of unmarked individuals observed

# - marked.unknown = the number of unknown marked individuals (i.e., known to be marked but not recognized)

# - alpha = the alpha-level for the desired CI (e.g., alpha = 0.05 for 95%CI)


bowden <- function(ch,unmarked.observed,marked.unknown,alpha) {

  ch.final <- c((marked.unknown/length(subset(ch, ch>0))+subset(ch, ch>0)),(subset(ch, ch==0)))

  n.marked <- length(ch.final)

  marked.observed <- sum(ch.final)

  y.mean <- (marked.observed/n.marked)

  s.squared <- (sum((ch.final-y.mean)^2))/(n.marked-1)

  N.hat <- (marked.observed+unmarked.observed)/y.mean

  N <- (N.hat + (s.squared/(y.mean^2)))/(1+(s.squared/(n.marked*(y.mean^2))))

  t.value <- qt(p=1-alpha/2, df=length(ch.final)-1)

  Var <- N^2 * (((1/n.marked - 1/N) * (s.squared/(y.mean^2))) / ((1 +
(s.squared/(n.marked*(y.mean^2))))^2))

  SE <- sqrt(Var)

  LCL <- N / exp(t.value * SE/N)

  UCL <- N * exp(t.value * SE/N)

  CV.perc <- SE/N*100

  headings <- c("N =", "LCL =", "UCL =", "%CV =", "Var =", "SE =")
```

```
  results <- cbind(round(N,0), round(LCL,0), round(UCL,0), round(CV.perc,1), round(Var,1), round(SE,1))
  print.table(paste(headings, results))
}

ch.2019 <- c(135,90,30,0,7,246,263,130,205,135,37,0,387,139,19,0,497,179)
unmarked.observed.2019 <- 935
marked.unknown.2019 <- 237

ch.2020 <- c(74,0,0,0,0,67,0,3,30,0,0,0,0,0,0,1,0,0,0)
unmarked.observed.2020 <- 25
marked.unknown.2020 <- 214

ch.2021 <- c(21,0,380,597,28,303,281,69,48,173,126,42,183,339,391,662,80,39,327,190,137,67)
unmarked.observed.2021 <- 104
marked.unknown.2021 <- 441

bowden.2019 <- bowden(ch=ch.2019, unmarked.observed=unmarked.observed.2019,
marked.unknown=marked.unknown.2019, alpha=0.05)
bowden.2020 <- bowden(ch=ch.2020, unmarked.observed=unmarked.observed.2020,
marked.unknown=marked.unknown.2020, alpha=0.05)
bowden.2021 <- bowden(ch=ch.2021, unmarked.observed=unmarked.observed.2021,
marked.unknown=marked.unknown.2021, alpha=0.05)

############################
# CF CMR vs. MARK-RESIGHT #
############################

# point estimates and SE for CMR
```

```
CMR.est.2019 <- 19

CMR.SE.2019 <- 1.7


CMR.est.2020 <- 15

CMR.SE.2020 <- 1.5


CMR.est.2021 <- 24

CMR.SE.2021 <- 1.9


# point estimates and SE for CTMR

CTMR.est.2019 <- 24

CTMR.SE.2019 <- 2.5


CTMR.est.2020 <- 20

CTMR.SE.2020 <- 1.7


CTMR.est.2021 <- 22

CTMR.SE.2021 <- 0.5


# t-test with p-values

t.value.2019 <- (CMR.est.2019-CTMR.est.2019)/sqrt(CMR.SE.2019^2+CTMR.SE.2019^2)

p.value.2019 <- 2*pt(q=abs(t.value.2019), df=30, lower.tail=FALSE)

p.value.2019


t.value.2020 <- (CMR.est.2020-CTMR.est.2020)/sqrt(CMR.SE.2020^2+CTMR.SE.2020^2)

p.value.2020 <- 2*pt(q=abs(t.value.2020), df=30, lower.tail=FALSE)

p.value.2020
```

```
t.value.2021 <- (CMR.est.2021-CTMR.est.2021)/sqrt(CMR.SE.2021^2+CTMR.SE.2021^2)
p.value.2021 <- 2*pt(q=abs(t.value.2021), df=30, lower.tail=FALSE)
p.value.2021

# t-test with confidence intervals
LCL.2019 <- (CMR.est.2019-CTMR.est.2019) - 1.96 * sqrt(CMR.SE.2019^2+CTMR.SE.2019^2)
UCL.2019 <- (CMR.est.2019-CTMR.est.2019) + 1.96 * sqrt(CMR.SE.2019^2+CTMR.SE.2019^2)
headings.2019 <- c("LCL =", "UCL =")
results.2019 <- cbind(round(LCL.2019,1), round(UCL.2019,1))
print.table(paste(headings.2019, results.2019))

LCL.2020 <- (CMR.est.2020-CTMR.est.2020) - 1.96 * sqrt(CMR.SE.2020^2+CTMR.SE.2020^2)
UCL.2020 <- (CMR.est.2020-CTMR.est.2020) + 1.96 * sqrt(CMR.SE.2020^2+CTMR.SE.2020^2)
headings.2020 <- c("LCL =", "UCL =")
results.2020 <- cbind(round(LCL.2020,1), round(UCL.2020,1))
print.table(paste(headings.2020, results.2020))

LCL.2021 <- (CMR.est.2021-CTMR.est.2021) - 1.96 * sqrt(CMR.SE.2021^2+CTMR.SE.2021^2)
UCL.2021 <- (CMR.est.2021-CTMR.est.2021) + 1.96 * sqrt(CMR.SE.2021^2+CTMR.SE.2021^2)
headings.2021 <- c("LCL =", "UCL =")
results.2021 <- cbind(round(LCL.2021,1), round(UCL.2021,1))
print.table(paste(headings.2021, results.2021))
# End #
```