Supplementary material

Mark A. Stevens
Marie Auger-Méthé

Appendix S1: Mixture of Bernoulli- and Poisson-distributed demographic parameters

We model survival and fecundity as a mixture of a Bernoulli and a Poisson distribution, respectively. We model survival as a function of a random intercept and a random slope for age. We model fecundity as a function of a random intercept and a random slope for age. We model the probability of being in the Bernoulli component of the mixture as a function of a random intercept.

Contents: Section S1.1 Model specification; Section S1.2 Simulation and estimation; Section S1.3 Results

We model survival and fecundity as a function of age and individual random effects. Let y_{ia} be a binary variable for survival and z_{ia} be a count variable for fecundity of individual i at age a. We model survival and fecundity as:

$$\text{logit}(\mu_{y_{ia}}) = \beta_{0,y} + \beta_{1,y} a + b_{0,y,i} + b_{1,y,i} a$$

$$\log(\mu_{z_{ia}}) = \beta_{0,z} + \beta_{1,z} a + b_{0,z,i} + b_{1,z,i} a$$

$$\begin{bmatrix} b_{0,y,i} \\ b_{1,y,i} \\ b_{0,z,i} \\ b_{1,z,i} \end{bmatrix} \sim \text{MVN}_4 \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{b_{0,y}} & \sigma_{b_{0,y},b_{1,y}} & \sigma_{b_{0,y},b_{0,z}} & \sigma_{b_{0,y},b_{1,z}} \\ \sigma_{b_{0,y},b_{1,y}} & \sigma^2_{b_{1,y}} & \sigma_{b_{1,y},b_{0,z}} & \sigma_{b_{1,y},b_{1,z}} \\ \sigma_{b_{0,y},b_{0,z}} & \sigma_{b_{1,y},b_{0,z}} & \sigma^2_{b_{0,z}} & \sigma_{b_{0,z},b_{1,z}} \\ \sigma_{b_{0,y},b_{1,z}} & \sigma_{b_{1,y},b_{1,z}} & \sigma_{b_{0,z},b_{1,z}} & \sigma^2_{b_{1,z}} \end{bmatrix} \right)$$

where $\beta_{0,y}$ is the survival intercept, $\beta_{1,y}$ is the survival slope for age, $\beta_{0,z}$ is the fecundity intercept, $\beta_{1,z}$ is the fecundity slope for age, $\sigma^2_{b_{0,y}}$ is the variance of the survival random intercept, $\sigma^2_{b_{1,y}}$ is the variance of the survival random slope for age, $\sigma^2_{b_{0,z}}$ is the variance of the fecundity random intercept, $\sigma^2_{b_{1,z}}$ is the variance of the fecundity random slope for age, $\sigma_{b_{0,y},b_{1,y}}$ is the covariance between the survival random intercept and survival random slope for age, $\sigma_{b_{0,y},b_{0,z}}$ is the covariance between the survival random intercept and fecundity random intercept, $\sigma_{b_{0,y},b_{1,z}}$ is the covariance between the survival random intercept and fecundity random slope for age, $\sigma_{b_{1,y},b_{0,z}}$ is the covariance between the survival random slope for age and fecundity random intercept, $\sigma_{b_{1,y},b_{1,z}}$ is the covariance between the survival random slope for age and fecundity random slope for age, $\sigma_{b_{0,z},b_{1,z}}$ is the covariance between the fecundity random intercept and fecundity random slope for age, MVN_4 is a 4-dimensional multivariate normal distribution, $\text{logit}(\mu_{y_{ia}})$ is the logit-transformed probability of survival, and $\log(\mu_{z_{ia}})$ is the log-transformed expected fecundity.

We model the probability of being in the Bernoulli component of the mixture as a function of a random intercept:

$$\text{logit}(\pi_{i}) = \beta_{0,\pi} + b_{0,\pi,i}$$

$$b_{0,\pi,i} \sim \text{N}(0, \sigma^2_{b_{0,\pi}})$$

where π_{i} is the probability of being in the Bernoulli component of the mixture for individual i, $\beta_{0,\pi}$ is the intercept, $b_{0,\pi,i}$ is the random intercept for individual i, $\sigma^2_{b_{0,\pi}}$ is the variance of the random intercept, N is a univariate normal distribution, and $\text{logit}(\pi_{i})$ is the logit-transformed probability of being in the Bernoulli component of the mixture.

Appendix S2: R and JAGS code to assess model performance in estimating population-level parameters, effects of annual survival and reproductive success probabilities

We provide the R and JAGS code to assess model performance in estimating population-level parameters, effects of annual survival and reproductive success probabilities.

Contents: Section S2.1 Simulation and estimation; Section S2.2 Results

```
# R code for Stevens et al. "A mixture model for demographic parameters"
# Simulation and estimation
# Section S2.1
# Section S2.2

# Clear the workspace
rm(list = ls())

# Load the JAGS library
library(R2jags)

# Number of repeated simulations
n.sim <- 100

# Number of individuals
n.ind <- 1000

# Number of age classes
n.age <- 5

# Number of years
n.year <- 50

# Mean survival probability (logit scale)
mean.survival <- 0.5

# Mean fecundity (log scale)
mean.fecundity <- 0.5

# Age effect on survival
age.eff.survival <- 0.1

# Age effect on fecundity
age.eff.fecundity <- 0.1

# Random intercept SD for survival
sd.int.survival <- 0.1

# Random slope SD for survival
sd.slope.survival <- 0.1

# Random intercept SD for fecundity
sd.int.fecundity <- 0.1

# Random slope SD for fecundity
sd.slope.fecundity <- 0.1

# Random intercept SD for Bernoulli component
sd.int.bernoulli <- 0.1

# Covariance between survival random intercept and random slope
cov.int.slope.survival <- 0.0

# Covariance between survival random intercept and fecundity random intercept
cov.int.survival.fecundity <- 0.0

# Covariance between survival random intercept and fecundity random slope
cov.int.survival.slope.fecundity <- 0.0

# Covariance between survival random slope and fecundity random intercept
cov.slope.survival.fecundity <- 0.0

# Covariance between survival random slope and fecundity random slope
cov.slope.survival.slope.fecundity <- 0.0

# Covariance between fecundity random intercept and fecundity random slope
cov.int.slope.fecundity <- 0.0

# Probability of being in the Bernoulli component of the mixture (logit scale)
mean.bernoulli <- 0.0

# Number of MCMC chains
n.chain <- 3

# Number of MCMC iterations
n.iter <- 5000

# Number of MCMC burn-in iterations
n.burnin <- 2500

# Number of MCMC thinning rate
n.thin <- 1

# Number of MCMC samples per chain after burn-in and thinning
n.post <- (n.iter - n.burnin) / n.thin

# Number of MCMC samples across all chains
n.samples <- n.post * n.chain

# Simulation and estimation loop
for (i in 1:n.sim){
  cat("\n\n\n\n", "Simulation", i, "of", n.sim, "\n\n\n\n")

  # Simulation
  # Age-specific survival probability (logit scale)
  age.survival <- mean.survival + (age.eff.survival * (1:(n.age) - 1) / (n.age - 1))

  # Age-specific fecundity (log scale)
  age.fecundity <- mean.fecundity + (age.eff.fecundity * (1:(n.age) - 1) / (n.age - 1))

  # Random intercept for survival
  ri.survival <- rnorm(n.ind, 0, sd.int.survival)

  # Random slope for survival
  rs.survival <- rnorm(n.ind, 0, sd.slope.survival)

  # Random intercept for fecundity
  ri.fecundity <- rnorm(n.ind, 0, sd.int.fecundity)

  # Random slope for fecundity
  rs.fecundity <- rnorm(n.ind, 0, sd.slope.fecundity)

  # Random intercept for Bernoulli component
  ri.bernoulli <- rnorm(n.ind, 0, sd.int.bernoulli)

  # Covariance matrix
  cov.matrix <- matrix(c(sd.int.survival^2, cov.int.slope.survival, cov.int.survival.fecundity, cov.int.survival.slope.fecundity,
                         cov.int.slope.survival, sd.slope.survival^2, cov.slope.survival.fecundity, cov.slope.survival.slope.fecundity,
                         cov.int.survival.fecundity, cov.slope.survival.fecundity, sd.int.fecundity^2, cov.int.slope.fecundity,
                         cov.int.survival.slope.fecundity, cov.slope.survival.slope.fecundity, cov.int.slope.fecundity, sd.slope.fecundity^2),
                       nrow = 4, byrow = TRUE)

  # Cholesky decomposition
  chol.matrix <- chol(cov.matrix)

  # Simulating random effects
  random.effects <- matrix(NA, nrow = n.ind, ncol = 4)
  for (j in 1:n.ind){
    random.effects[j, ] <- t(chol.matrix) %*% rnorm(4, 0, 1)
  }

  # Probability of being in the Bernoulli component of the mixture (logit scale)
  prob.bernoulli <- mean.bernoulli + ri.bernoulli

  # Simulating Bernoulli component of the mixture
  bernoulli <- matrix(NA, nrow = n.ind, ncol = n.year)
  for (j in 1:n.ind){
    bernoulli[j, 1] <- rbinom(1, 1, prob.bernoulli[j])
    for (t in 2:n.year){
      bernoulli[j, t] <- bernoulli[j, 1]
    }
  }

  # Probability of survival (logit scale)
  prob.survival <- matrix(NA, nrow = n.ind, ncol = n.year - 1)
  for (j in 1:n.ind){
    for (t in 1:(n.year - 1)){
      prob.survival[j, t] <- age.survival[1] + age.survival[2] * (t - 1) / (n.year - 2) + random.effects[j, 1] + random.effects[j, 2] * (t - 1) / (n.year - 2)
    }
  }

  # Survival
  survival <- matrix(NA, nrow = n.ind, ncol = n.year - 1)
  for (j in 1:n.ind){
    for (t in 1:(n.year - 1)){
      survival[j, t] <- rbinom(1, 1, plogis(prob.survival[j, t]))
    }
  }

  # Expected fecundity (log scale)
  exp.fecundity <- matrix(NA, nrow = n.ind, ncol = n.year - 1)
  for (j in 1:n.ind){
    for (t in 1:(n.year - 1)){
      exp.fecundity[j, t] <- age.fecundity[1] + age.fecundity[2] * (t - 1) / (n.year - 2) + random.effects[j, 3] + random.effects[j, 4] * (t - 1) / (n.year - 2)
    }
  }

  # Fecundity
  fecundity <- matrix(NA, nrow = n.ind, ncol = n.year - 1)
  for (j in 1:n.ind){
    for (t in 1:(n.year - 1)){
      fecundity[j, t] <- rbinom(1, 1, exp(exp.fecundity[j, t]))
      if (bernoulli[j, t] == 1){
        fecundity[j, t] <- 0
      }
    }
  }

  # Data
  data <- list(survival = survival,
               fecundity = fecundity,
               n.ind = n.ind,
               n.year = n.year)

  # Parameters
  parameters <- c("age.survival",
                  "age.fecundity",
                  "sd.int.survival",
                  "sd.slope.survival",
                  "sd.int.fecundity",
                  "sd.slope.fecundity",
                  "sd.int.bernoulli",
                  "cov.int.slope.survival",
                  "cov.int.survival.fecundity",
                  "cov.int.survival.slope.fecundity",
                  "cov.slope.survival.fecundity",
                  "cov.slope.survival.slope.fecundity",
                  "cov.int.slope.fecundity",
                  "mean.bernoulli",
                  "ri.survival",
                  "rs.survival",
                  "ri.fecundity",
                  "rs.fecundity",
                  "ri.bernoulli")

  # Initial values
  inits <- function(){
    list(age.survival = rnorm(2, 0, 1),
         age.fecundity = rnorm(2, 0, 1),
         sd.int.survival = runif(1, 0, 1),
         sd.slope.survival = runif(1, 0, 1),
         sd.int.fecundity = runif(1, 0, 1),
         sd.slope.fecundity = runif(1, 0, 1),
         sd.int.bernoulli = runif(1, 0, 1),
         cov.int.slope.survival = runif(1, -1, 1),
         cov.int.survival.fecundity = runif(1, -1, 1),
         cov.int.survival.slope.fecundity = runif(1, -1, 1),
         cov.slope.survival.fecundity = runif(1, -1, 1),
         cov.slope.survival.slope.fecundity = runif(1, -1, 1),
         cov.int.slope.fecundity = runif(1, -1, 1),
         mean.bernoulli = rnorm(1, 0, 1),
         ri.survival = rnorm(n.ind, 0, 1),
         rs.survival = rnorm(n.ind, 0, 1),
         ri.fecundity = rnorm(n.ind, 0, 1),
         rs.fecundity = rnorm(n.ind, 0, 1),
         ri.bernoulli = rnorm(n.ind, 0, 1))
  }

  # Model
  model <- "model{
    # Priors
    age.survival[1] ~ dnorm(0.0, 1.0E-6)
    age.survival[2] ~ dnorm(0.0, 1.0E-6)
    age.fecundity[1] ~ dnorm(0.0, 1.0E-6)
    age.fecundity[2] ~ dnorm(0.0, 1.0E-6)
    sd.int.survival ~ dunif(0.0, 1.0E3)
    sd.slope.survival ~ dunif(0.0, 1.0E3)
    sd.int.fecundity ~ dunif(0.0, 1.0E3)
    sd.slope.fecundity ~ dunif(0.0, 1.0E3)
    sd.int.bernoulli ~ dunif(0.0, 1.0E3)
    cov.int.slope.survival ~ dunif(-1.0, 1.0)
    cov.int.survival.fecundity ~ dunif(-1.0, 1.0)
    cov.int.survival.slope.fecundity ~ dunif(-1.0, 1.0)
    cov.slope.survival.fecundity ~ dunif(-1.0, 1.0)
    cov.slope.survival.slope.fecundity ~ dunif(-1.0, 1.0)
    cov.int.slope.fecundity ~ dunif(-1.0, 1.0)
    mean.bernoulli ~ dnorm(0.0, 1.0E-6)
    cov.matrix[1, 1] <- sd.int.survival^2
    cov.matrix[2, 2] <- sd.slope.survival^2
    cov.matrix[3, 3] <- sd.int.fecundity^2
    cov.matrix[4, 4] <- sd.slope.fecundity^2
    cov.matrix[1, 2] <- cov.int.slope.survival
    cov.matrix[1, 3] <- cov.int.survival.fecundity
    cov.matrix[1, 4] <- cov.int.survival.slope.fecundity
    cov.matrix[2, 1] <- cov.int.slope.survival
    cov.matrix[2, 3] <- cov.slope.survival.fecundity
    cov.matrix[2, 4] <- cov.slope.survival.slope.fecundity
    cov.matrix[3, 1] <- cov.int.survival.fecundity
    cov.matrix[3, 2] <- cov.slope.survival.fecundity
    cov.matrix[3, 4] <- cov.int.slope.fecundity
    cov.matrix[4, 1] <- cov.int.survival.slope.fecundity
    cov.matrix[4, 2] <- cov.slope.survival.slope.fecundity
    cov.matrix[4, 3] <- cov.int.slope.fecundity
    chol.matrix <- chol(cov.matrix)
    for (j in 1:n.ind){
      random.effects[j, 1:4] ~ dmnorm(0.0, chol.matrix)
    }
    prob.bernoulli <- mean.bernoulli + ri.bernoulli
    for (j in 1:n.ind){
      bernoulli[j, 1] ~ dbern(prob.bernoulli[j])
      for (t in 2:n.year){
        bernoulli[j, t] <- bernoulli[j, 1]
      }
    }
    for (j in 1:n.ind){
      for (t in 1:(n.year - 1)){
        prob.survival[j, t] <- age.survival[1] + age.survival[2] * (t - 1) / (n.year - 2) + random.effects[j, 1] + random.effects[j, 2] * (t - 1) / (n.year - 2)
        survival[j, t] ~ dbern(plogis(prob.survival[j, t]))
        exp.fecundity[j, t] <- age.fecundity[1] + age.fecundity[2] * (t - 1) / (n.year - 2) + random.effects[j, 3] + random.effects[j, 4] * (t - 1) / (n.year - 2)
        fecundity[j, t] ~ dbern(exp(exp.fecundity[j, t]))
        fecundity[j, t] <- fecundity[j, t] * (1 - bernoulli[j, t])
      }
    }
    age.survival.diff <- age.survival[2] - age.survival[1]
    age.fecundity.diff <- age.fecundity[2] - age.fecundity[1]
    sd.int.survival.sq <- sd.int.survival^2
    sd.slope.survival.sq <- sd.slope.survival^2
    sd.int.fecundity.sq <- sd.int.fecundity^2
    sd.slope.fecundity.sq <- sd.slope.fecundity^2
    sd.int.bernoulli.sq <- sd.int.bernoulli^2
    mean.bernoulli.prob <- plogis(mean.bernoulli)
    ri.bernoulli.prob <- plogis(ri.bernoulli)
    ri.bernoulli.diff <- ri.bernoulli.prob - mean.bernoulli.prob
    ri.bernoulli.diff.abs <- abs(ri.bernoulli.diff)
    ri.bernoulli.diff.sq <- ri.bernoulli.diff^2
    ri.bernoulli.diff.sq.sum <- sum(ri.bernoulli.diff.sq)
    ri.bernoulli.diff.abs.sum <- sum(ri.bernoulli.diff.abs)
    ri.bernoulli.sq <- ri.bernoulli^2
    ri.bernoulli.sq.sum <- sum(ri.bernoulli.sq)
    ri.bernoulli.var <- var(ri.bernoulli)
    ri.bernoulli.var.prop <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq)
    ri.bernoulli.var.prop.1 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (exp(mean.bernoulli) * (1.0 - exp(mean.bernoulli)))))
    ri.bernoulli.var.prop.2 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (mean.bernoulli.prob * (1.0 - mean.bernoulli.prob))))
    ri.bernoulli.var.prop.3 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.4 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.5 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.6 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.7 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.8 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.9 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.10 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.11 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.12 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.13 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.14 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.15 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.16 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.17 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.18 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.19 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.20 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.21 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.22 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.23 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.24 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.25 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.26 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.27 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.28 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.29 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.30 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.31 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.32 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.33 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.34 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.35 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.36 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.37 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.38 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.39 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.40 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.41 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.42 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.43 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.44 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.45 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.46 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.47 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.48 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.49 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.50 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.51 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.52 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.53 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.54 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.55 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.56 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.57 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.58 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.59 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.60 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.61 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.62 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.63 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.64 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.65 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.66 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.67 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.68 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.69 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.70 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.71 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.72 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - ri.bernoulli.prob)
    ri.bernoulli.var.prop.73 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob))))
    ri.bernoulli.var.prop.74 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli.prob)))) * (1.0 - mean.bernoulli.prob)
    ri.bernoulli.var.prop.75 <- ri.bernoulli.var / (ri.bernoulli.var + sd.int.bernoulli.sq + (1.0 / (ri.bernoulli.prob * (1.0 - ri.bernoulli
```