PISCO_urchin_tuffy

Andrés Pinos-Sánchez

2025-05-15

##INPUT DATA

This data is from PISCO surveys posted by MENGE, for intertidal recruitmentacross the coast O.G. data: https://data.piscoweb.org/metacatui/view/doi%3A10.6085%2FAA%2Fpisco_recruitment.1477.1 (https://data.piscoweb.org/metacatui/view/doi%3A10.6085%2FAA%2Fpisco_recruitment.1477.1)

```
# # Input data from PISCO, data name corresponds to collection year (not deployment)
# PISCO_data_1989 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.124.1.csv")  |> mutate
(year = 1989, .before = 1)
# PISCO_data_1990 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.126.1.csv")  |> mutate
(year = 1990, .before = 1)
# PISCO_data_1991 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.128.1.csv")  |> mutate
(year = 1991, .before = 1)
# PISCO_data_1992 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.130.1.data") |> mutate
(year = 1992, .before = 1)
# PISCO_data_1993 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.132.1.data") |> mutate
(year = 1993, .before = 1)
# PISCO_data_1994 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.134.1.csv")  |> mutate
(year = 1994, .before = 1)
# PISCO_data_1995 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.136.1.data") |> mutate
(year = 1995, .before = 1)
# PISCO_data_1996 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.138.1.data") |> mutate
(year = 1996, .before = 1)
# # 1997 skipped
# PISCO_data_1998 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.140.1.data") |> mutate
(year = 1998, .before = 1)
# PISCO_data_1999 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.142.1.data") |> mutate
(year = 1999, .before = 1)
# PISCO_data_2000 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.144.1.csv")  |> mutate
(year = 2000, .before = 1)
# PISCO_data_2001 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.146.1.data") |> mutate
(year = 2001, .before = 1)
# PISCO_data_2002 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.148.1.csv")  |> mutate
(year = 2002, .before = 1)
# PISCO_data_2003 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.150.1.data") |> mutate
(year = 2003, .before = 1)
# PISCO_data_2004 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.152.1.csv")  |> mutate
(year = 2004, .before = 1)
# PISCO_data_2005 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.154.1.csv")  |> mutate
(year = 2005, .before = 1)
# PISCO_data_2006 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.156.1.data") |> mutate
(year = 2006, .before = 1)
# PISCO_data_2007 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.1364.1.data")|> mutate
(year = 2007, .before = 1)
# PISCO_data_2008 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.1366.1.csv") |> mutate
(year = 2008, .before = 1)
# PISCO_data_2009 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.1438.1.csv") |> mutate
(year = 2009, .before = 1)
# PISCO_data_2010 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.1446.1.csv") |> mutate
(year = 2010, .before = 1)
# PISCO_data_2011 <- read_csv("Datasets/doi_10.6085_AA_pisco_recruitment.1464.1.txt") |> mutate
(year = 2011, .before = 1)
#
# # Combine all datasets
# PISCO_all_years <- bind_rows(PISCO_data_1989, PISCO_data_1990, PISCO_data_1991,
#                              PISCO_data_1992, PISCO_data_1993, PISCO_data_1994,
#                              PISCO_data_1995, PISCO_data_1996, PISCO_data_1998,
#                              PISCO_data_1999, PISCO_data_2000, PISCO_data_2001,
```

```
#                              PISCO_data_2002, PISCO_data_2003, PISCO_data_2004,
#                              PISCO_data_2005, PISCO_data_2006, PISCO_data_2007,
#                              PISCO_data_2008, PISCO_data_2009, PISCO_data_2010,
#                              PISCO_data_2011)

# # Print and explore
# print()
# summary()

# Save table
# write.csv(PISCO_all_years, "PISCO_all_years.csv", row.names = FALSE)
```

MODEL URCHINRECRUITMENT

Objective: 1. estimate the normalized year to year variance (SD) of incoming recruits using a log normal distribution of recruitment counts

I'd be building a hierarchical structure where my incoming settlers data informs a log-normal distributed annual means, and those annual means inform a normally distributed year to year variance (SD)

```
# Load nimble
source("attach.nimble.R")

# Load PISCO data
pisco <- read_csv("PISCO_all_years.csv") #PISCO_all_years, #read_csv("PISCO_all_years.csv"), # s
ource("PISCO_data_prep.R")
```

```
## Rows: 77630 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): sample_month, site_code, exposure, zone, collector_type, sampler, ...
## dbl  (6): year, replicate, proportion_sampled, count_classcode, count, metho...
## date (2): deploy_date, collect_date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Filter for taxa 66,67,70 and drop unwanted sites
urchins_raw <- pisco %>%
  filter(count_classcode %in% c("66","67","70"), # data 2001 - 2011
         !site_code %in% c("CMEN00","CMES00","IFBRXX","KHLX00",
                          "PSGX00","SHCX00","TRHX00","IPTAXX")) # 14 sites

# Group (transects) by year, site_code, zone, replicate,
# then sum within transects, and compute annual mean and sd
urchins <- urchins_raw %>%
  group_by(year, site_code, zone, replicate) %>%
  summarise(count = sum(count))
```

```
## `summarise()` has grouped output by 'year', 'site_code', 'zone'. You can
## override using the `.groups` argument.
```

```
# Log-transform counts (add 1 to avoid log(0))
urchins <- urchins %>%
  mutate(log_count = log(count + 1))



# Nimble model - Fun begins
model_code <- nimbleCode({

  # Hyperpriors
  mu         ~ dnorm(0, 1)           # Overall mean
  sigma_year ~ T(dt(0, 1, 1), 0, )   # Half-Cauchy for year SD
  sigma_obs  ~ T(dt(0, 1, 1), 0, )   # Half-Cauchy for residual SD

  # Random year effects
  for(j in 1:n_years){
    year_effect[j] ~ dnorm(mu, sd = sigma_year)
  }#j

  # Observation model
  for(i in 1:n_obs){

    log_count[i] ~ dnorm(mean = year_effect[year_index[i]], sd = sigma_obs)

  }#i

})#model_code



# Prep for MCMC

# Constants & data
parameters <- c("mu", "sigma_year", "sigma_obs", "year_effect")

nimble_constants <- list(n_obs = nrow(urchins),
                         n_years = length(sort(unique(urchins$year))),
                         year_index = match(urchins$year, sort(unique(urchins$year))))

nimble_data <- list(log_count = urchins$log_count)

# MCMC settings
ni <- 50000
nb <- 10000
nt <- 40
nc <- 3

# Run MCMC
mcmc_output <- nimbleMCMC(code      = model_code,
                          data      = nimble_data,
                          constants = nimble_constants,
                          monitors  = parameters,
                          niter     = ni,
```

```
                          nburnin   = nb,
                          nchains   = nc,
                          thin      = nt,
                          summary   = TRUE,
                          samplesAsCodaMCMC = TRUE)
```

```
## Defining model
## Building model
## Setting data and initial values
## Running calculate on model
##   [Note] Any error reports that follow may simply reflect missing values in model variables.
## Checking model sizes and dimensions
##   [Note] This model is not fully initialized. This is not an error.
##          To see which variables are not initialized, use model$initializeInfo().
##          For more information on model initialization, see help(modelInitialization).
## Checking model calculations
## [Note] NAs were detected in model variables: mu, logProb_mu, sigma_year, sigma_obs, year_effe
ct, logProb_year_effect, logProb_log_count.
## [Note] Infinite values were detected in model variables: logProb_sigma_year, logProb_sigma_ob
s.
## Compiling
##   [Note] This may take a minute.
##   [Note] Use 'showCompilerOutput = TRUE' to see C++ compilation details.
## running chain 1...
```

```
## |-------------|-------------|-------------|-------------|
## |-------------------------------------------------------|
```

```
## running chain 2...
```

```
## |-------------|-------------|-------------|-------------|
## |-------------------------------------------------------|
```

```
## running chain 3...
```

```
## |-------------|-------------|-------------|-------------|
## |-------------------------------------------------------|
```

```
# MCMC outputs
attach.nimble(mcmc_output$samples)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following object is masked from 'package:purrr':
##
##     transpose
##
## The following object is masked from 'package:nimble':
##
##     cube
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
summary(mcmc_output)
```

```
##         Length Class     Mode
## samples 3      mcmc.list list
## summary 4      -none-    list
```

```
print(mcmc_output$summary)
```

```
## $chain1
##                       Mean      Median     St.Dev.   95%CI_low  95%CI_upp
## mu              0.363951399 0.358933644 0.128445821  0.09950289 0.61698984
## sigma_obs       0.327340941 0.326884410 0.009960114  0.30986819 0.34788555
## sigma_year      0.406819367 0.388548000 0.107345060  0.25427547 0.67361390
## year_effect[1]  0.659861477 0.653310148 0.121164758  0.43744167 0.90340321
## year_effect[2]  0.759963581 0.758605415 0.108033106  0.55817737 0.97196324
## year_effect[3]  0.705002483 0.706531804 0.092193558  0.52385178 0.87742709
## year_effect[4]  0.950977986 0.950078160 0.092961412  0.76352347 1.14265517
## year_effect[5]  0.529370748 0.529881054 0.041809691  0.44674525 0.61221918
## year_effect[6]  0.012483803 0.012786609 0.039999041 -0.06679308 0.09026055
## year_effect[7]  0.022818453 0.022296595 0.039418970 -0.05765294 0.09758570
## year_effect[8]  0.002892884 0.003505966 0.036034109 -0.07034788 0.07016620
## year_effect[9]  0.109612700 0.109852240 0.037197076  0.03730520 0.18062810
## year_effect[10] 0.181532803 0.183439738 0.038359879  0.10314144 0.25231625
## year_effect[11] 0.089729383 0.087707145 0.036455869  0.02099091 0.16043563
##
## $chain2
##                       Mean      Median    St.Dev.   95%CI_low  95%CI_upp
## mu              0.355507671 0.356029666 0.13148004  0.11193481 0.61696537
## sigma_obs       0.327788781 0.327659030 0.01009170  0.30864551 0.34769476
## sigma_year      0.411192701 0.392471172 0.11229693  0.25007396 0.65878539
## year_effect[1]  0.656943759 0.660214462 0.13035309  0.39641229 0.90199643
## year_effect[2]  0.751657991 0.751461054 0.11085158  0.53944449 0.97485746
## year_effect[3]  0.703944803 0.700846297 0.08962037  0.53122551 0.87699155
## year_effect[4]  0.951152520 0.954524135 0.09151262  0.77163540 1.12842947
## year_effect[5]  0.530597500 0.531912300 0.04267033  0.44500059 0.61354503
## year_effect[6]  0.014891073 0.015423683 0.04019509 -0.05944972 0.09218436
## year_effect[7]  0.019447361 0.019785420 0.04101811 -0.06331519 0.09752413
## year_effect[8]  0.002481337 0.001654778 0.03553346 -0.06520400 0.07128823
## year_effect[9]  0.111308204 0.109859100 0.03579143  0.04493703 0.18581392
## year_effect[10] 0.179843180 0.180068799 0.03656496  0.10571310 0.25435979
## year_effect[11] 0.087628324 0.087236470 0.03680059  0.01721638 0.15972341
##
## $chain3
##                       Mean      Median     St.Dev.   95%CI_low  95%CI_upp
## mu              0.363679703 0.363483874 0.124964382  0.11699038 0.60264230
## sigma_obs       0.327704133 0.327393582 0.009816192  0.30920849 0.34777014
## sigma_year      0.407260895 0.394955758 0.101375970  0.24991195 0.65417049
## year_effect[1]  0.651925018 0.654027641 0.127738719  0.40002008 0.88612227
## year_effect[2]  0.756749600 0.753522818 0.110413979  0.54647055 0.98408947
## year_effect[3]  0.704600054 0.705545437 0.091114517  0.51639154 0.88670884
## year_effect[4]  0.950596096 0.950280166 0.089112453  0.77915953 1.12255147
## year_effect[5]  0.531120285 0.531071631 0.042668349  0.44747400 0.61359708
## year_effect[6]  0.013110444 0.011679555 0.039148571 -0.06108112 0.09503036
## year_effect[7]  0.020977361 0.021906577 0.039307415 -0.05447264 0.09774076
## year_effect[8]  0.003241656 0.004739071 0.036415096 -0.06816611 0.06920552
## year_effect[9]  0.112186152 0.112866528 0.037701341  0.03727322 0.18502883
## year_effect[10] 0.181369852 0.181845042 0.036893467  0.10631247 0.24893388
## year_effect[11] 0.089346752 0.088560040 0.037424260  0.01569415 0.16629420
##
## $all.chains
```

```
##                        Mean      Median     St.Dev.   95%CI_low  95%CI_upp
## mu              0.361046258 0.359092861 0.128341402  0.10918417 0.61437050
## sigma_obs       0.327611285 0.327242582 0.009955214  0.30895265 0.34777014
## sigma_year      0.408424321 0.391520722 0.107081432  0.25054763 0.66823177
## year_effect[1]  0.656243418 0.654684770 0.126478252  0.40388163 0.90097413
## year_effect[2]  0.756123724 0.754645259 0.109789880  0.54657004 0.97838439
## year_effect[3]  0.704515780 0.704242509 0.090952967  0.51962382 0.87904555
## year_effect[4]  0.950908868 0.951619853 0.091179183  0.77196042 1.13092689
## year_effect[5]  0.530362844 0.531028515 0.042376940  0.44673595 0.61350050
## year_effect[6]  0.013495107 0.013234799 0.039783301 -0.06304511 0.09275562
## year_effect[7]  0.021081058 0.021497592 0.039932964 -0.06013742 0.09771928
## year_effect[8]  0.002871959 0.003410743 0.035985369 -0.06813994 0.06995826
## year_effect[9]  0.111035685 0.110883445 0.036908621  0.04054315 0.18419838
## year_effect[10] 0.180915278 0.181436168 0.037276275  0.10554757 0.25186777
## year_effect[11] 0.088901486 0.087809421 0.036894766  0.01726138 0.16234900
```

```
mcmcplot(mcmc_output$samples)
```

```
##                                                                   Preparing plo
ts for mu.  7% complete.
```

```
##                                                                   Preparing plo
ts for sigma.  14% complete.
```

```
##                                                                   Preparing plo
ts for sigma.  21% complete.
```

```
##                                                                   Preparing plo
ts for year.  29% complete.
```

```
##                                                                   Preparing plo
ts for year.  36% complete.
```

```
##                                                                   Preparing plo
ts for year.  43% complete.
```

```
##                                                                   Preparing plo
ts for year.  50% complete.
```

```
##                                                                   Preparing plo
ts for year.  57% complete.
```

```
##                                                                   Preparing plo
ts for year.  64% complete.
```

```
##                                                                                Preparing plo
ts for year.  71% complete.
```

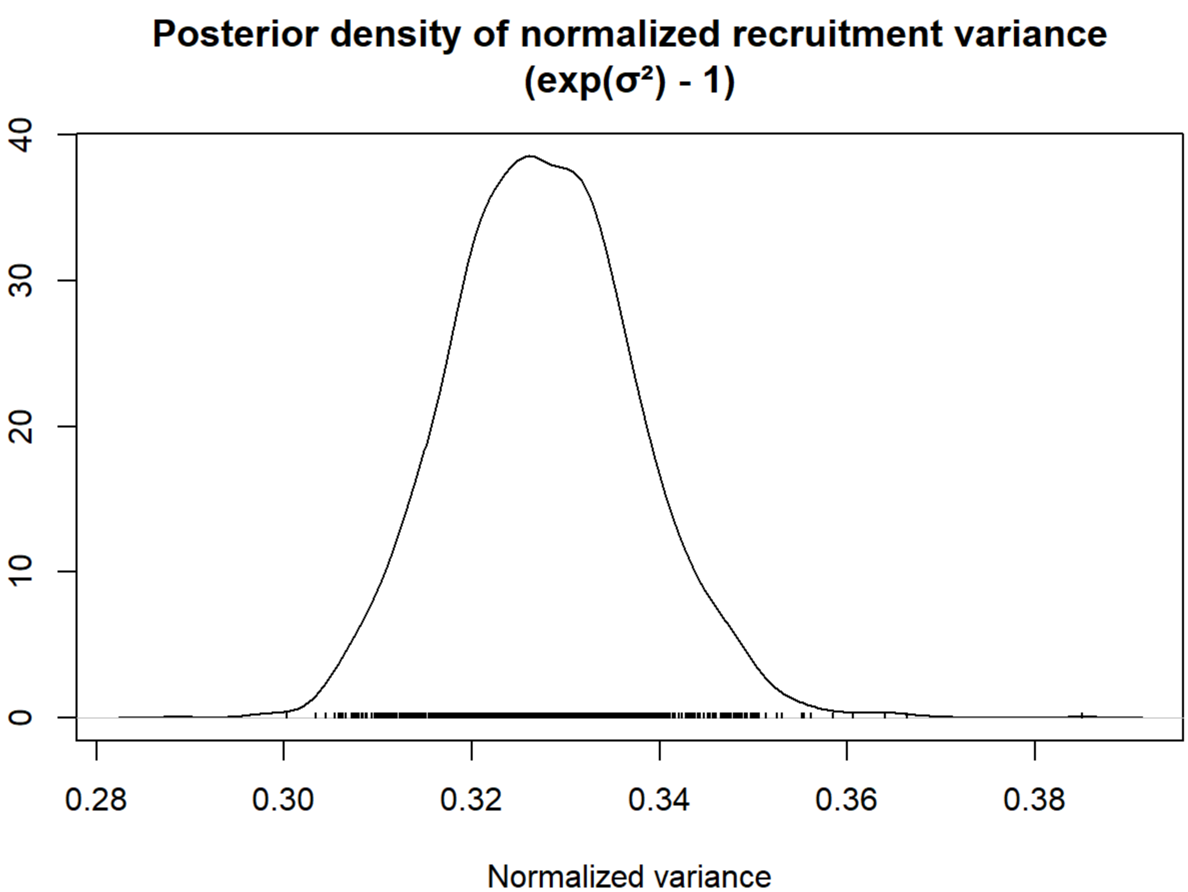
```
##                                                                                Preparing plo
ts for year.  79% complete.
```

```
##                                                                                Preparing plo
ts for year.  86% complete.
```

```
##                                                                                Preparing plo
ts for year.  93% complete.
```

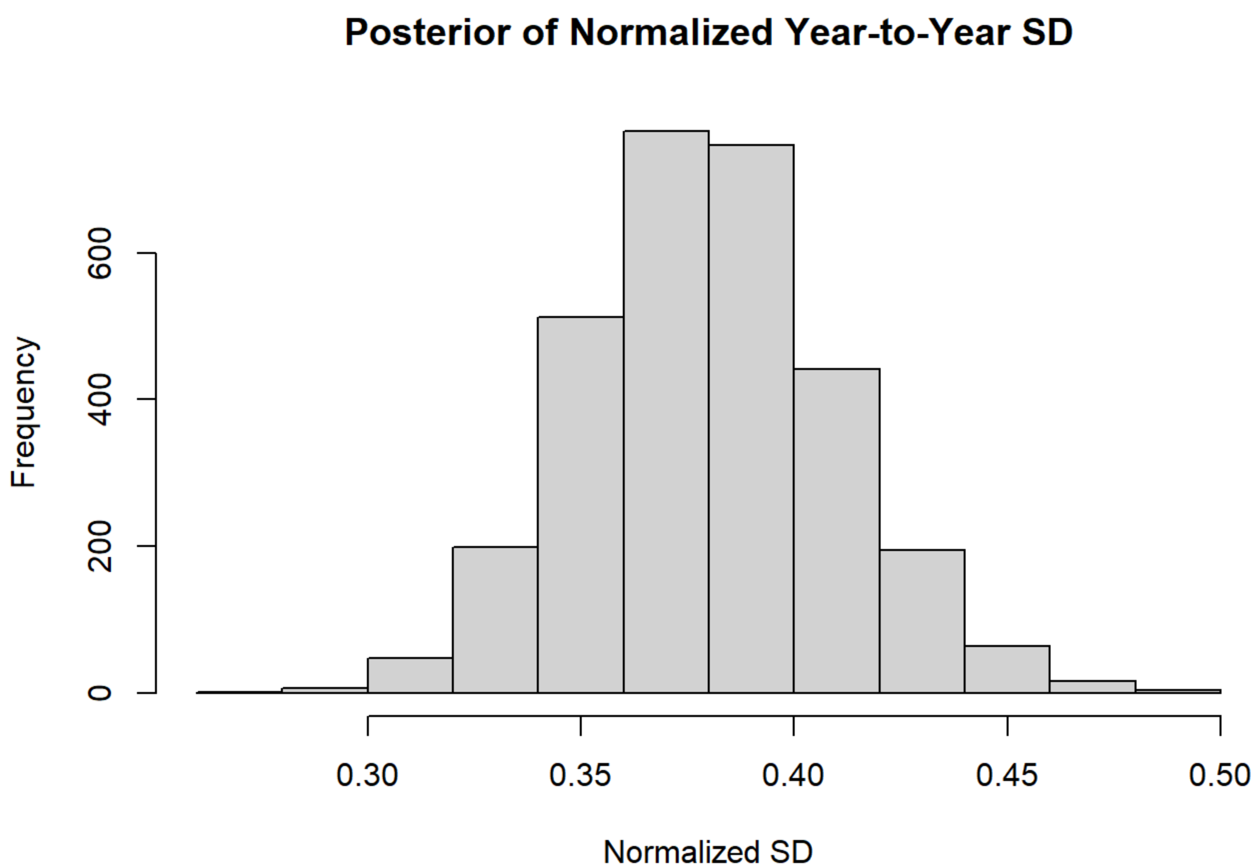
```
##                                                                                Preparing plo
ts for year.  100% complete.
```

```
densplot(mcmc_output$samples[, "sigma_obs"],
         main = "Posterior density of normalized recruitment variance\n(exp(σ²) - 1)",
         xlab = "Normalized variance")
```



```
# Normalize the year effects (back-transform from log-scale)
year_mean_recruit <- exp(year_effect)  # back to count scale
year_mean_norm <- sweep(year_mean_recruit, 1, rowMeans(year_mean_recruit), "/")
norm_sd <- apply(year_mean_norm, 1, sd)

# Summary of normalized SD
hist(norm_sd, main = "Posterior of Normalized Year-to-Year SD",
     xlab = "Normalized SD")
```




```
mean(norm_sd)
```

```
## [1] 0.3798434
```

```
quantile(norm_sd, probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.3226070 0.3793081 0.4426450
```

```
#Plot posterior density of CV
norm_sd_mcmc <- as.mcmc(norm_sd)
plot(density(norm_sd_mcmc),
     main = "Posterior density of recruitment norm_sd",
     xlab = "Norm_sd")
```