Movement matrix goodness of fit

Benjamin Galuardi

2017-01-23

Variables used in this example (three datasets/runs)

- East Atlantic
- West Atlantic large fish
- West Atlantic small fish

```
eatl.ct = get.trans.prob(eatl.datbox, perc = F)
watl.big.ct = get.trans.prob(watl.big.datbox, perc = F)
watl.small.ct = get.trans.prob(watl.small.datbox, perc = F)
eatl = get.trans.prob(eatl.datbox, perc = T)
watl.big = get.trans.prob(watl.big.datbox, perc = T)
watl.small = get.trans.prob(watl.small.datbox, perc = T)
```

functions for multinomial variance.

The expected number of times the outcome i observed over n trials is

$$\mathrm{E}(X_i) = np_i \tag{1}$$

$$\mathrm{v}(X_i) = np_i(1 - p_i) \tag{2}$$

where n is the number of trials and p is the probability. Here, I consider each row of a movement matrix a multinomial. This is what we discussed in the proposal defense. code modified from here

Question:

I'm not sure how to represent n, but I assumed the transition between seasons was a single trial for each row.

```
# function to calculate variance of a multinomial row
get.mvar = function(x){
  p = drop(x) # use the percentages
  v = p*(1-p)
  # c = outer(p, 1-p)
  # vcv = diag(c)
  v
}

# function to sample from a posterior multinomial distribution
get.mvar.samp = function(vmu, vsd) {
  o = rtruncnorm(1, mean = vmu, sd = vsd, a = 0, 1)
  o[is.nan(o)] = 0
  o
}
```

```r
# example for seasonal transition matrices
vmat = lapply(watl.big, function(x) apply(x, 1, get.mvar))
```

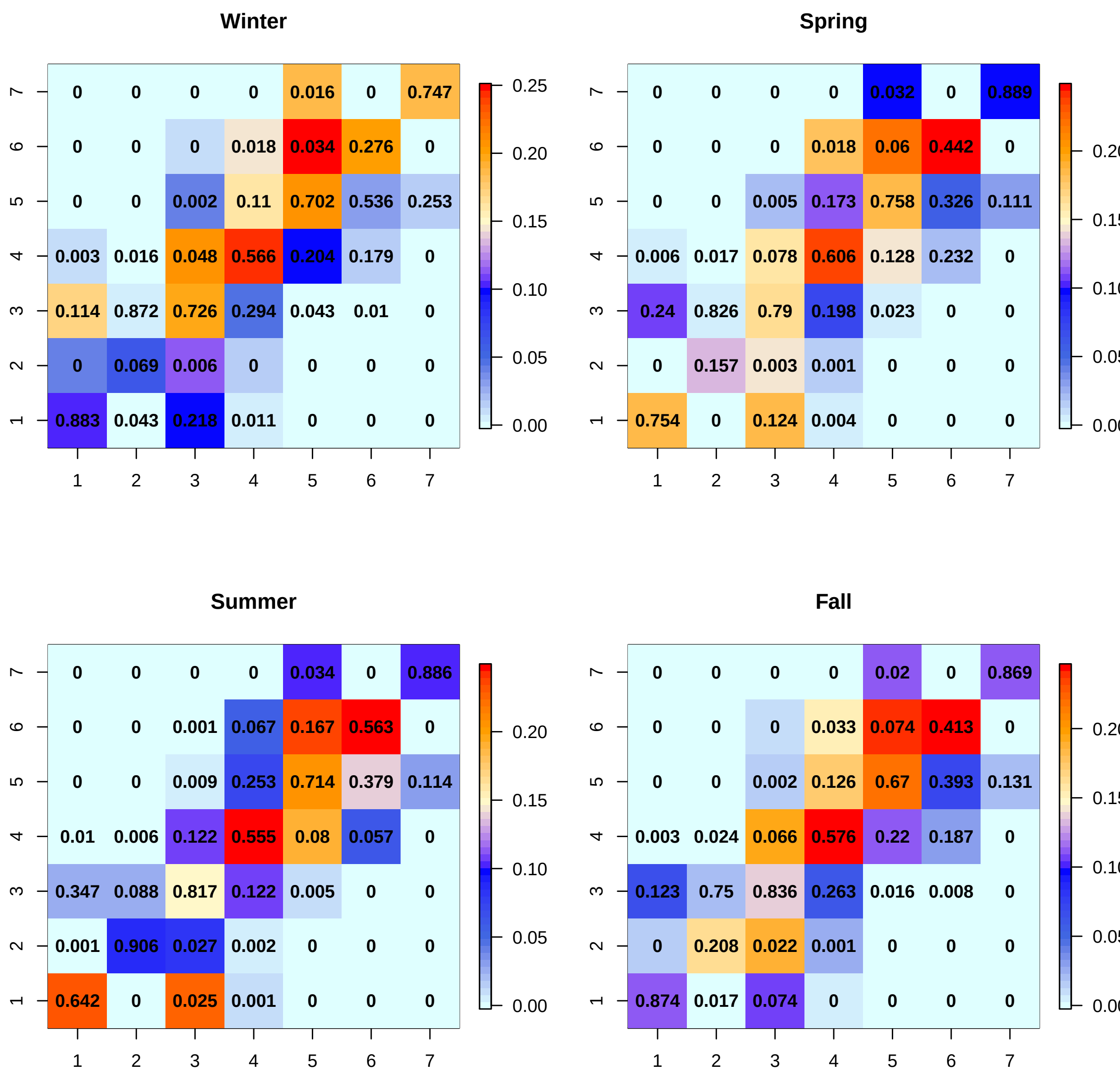
Table 1: Multinomial variance for Western bluefin >185cm, in Winter

	1	2	3	4	5	6	7
1	0.1034	0.0409	0.1704	0.0109	0.0000	0.0000	0.0000
2	0.0000	0.0643	0.0060	0.0000	0.0000	0.0000	0.0000
3	0.1010	0.1117	0.1990	0.2077	0.0412	0.0101	0.0000
4	0.0032	0.0160	0.0461	0.2456	0.1626	0.1467	0.0000
5	0.0000	0.0000	0.0017	0.0983	0.2092	0.2487	0.1889
6	0.0000	0.0000	0.0001	0.0176	0.0333	0.1996	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0158	0.0000	0.1889

Plot the variance for each cell, each season (colorbar)

Overlay with original movement matrix values

Example using Western bluefin >185cm



sample from posterior

```
samp = data.frame(sapply(1:7, function(x) get.mvar.samp(watl.big[[1]][x,], vmat[[1]][x,])))
names(samp) = 1:7
kable(samp, row.names = 1:7, caption = 'Posterior sample: Western Atlantic bluefin > 185cm, Winter seas
```

Table 2: Posterior sample: Western Atlantic bluefin > 185cm, Winter season

	1	2	3	4	5	6	7
1	0.8952	0.0427	0.1925	0.0092	0.0000	0.0000	0.0000
2	0.0766	0.1335	0.2093	0.0060	0.0000	0.0000	0.0000
3	0.3652	0.8779	0.6493	0.2873	0.0440	0.0101	0.0000
4	0.0006	0.0163	0.0988	0.6622	0.1492	0.1808	0.0000
5	0.0000	0.0000	0.0607	0.0861	0.8687	0.5023	0.2628
6	0.0000	0.0000	0.0055	0.1078	0.1534	0.5465	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.2236	0.0000	0.6696

Goodness of Fit

function: compare pairs of rows of two matrices, using loglinear model

returns the pchisq value

χ^2 multinomial goodness of fit test from here

$$\chi^2 = \sum_i \frac{(f_i - e_i)^2}{e_i} \tag{3}$$

```
# first arguement is counts, second argument is probability. In this way, row-wise comparisons of two m
# p-values of the log-linear (chisq test) are returned

get.loglin <- function(mat1, mat2){
  fm = NULL
  for(i in 1:nrow(mat1)){
    fm[[i]] =  loglin(rbind(mat1[i,], mat2[i,]+1e-30),c(1), print = F, fit = F)
  }
  unlist(lapply(fm, function(x) 1-pchisq(x$lrt, x$df)))
}

get.mgof <- function(mat1, mat2){
  fm = NULL
  for(i in 1:nrow(mat1)){
    fm[i] =  chisq.test(mat1[i,], p = mat2[i,]+1e-30)$p.value
  }
 fm
}
```

I tried several versions of log-linear and χ^2 tests. Log linear models, and their GLM versions, are essentially wrappers on χ^2 tests for ease of use and flexibility in using contingency tables in various formats.

Since each row of the movments matrices is dependent on it's associated region i nthe previous time step, the matrix as a whole cannot be considered a contingency table. Each row, however, may be considered it's own 1-dimensional contingency table.

To compare any two rows, within a matrix or between matrices, the best method appeared to be the χ^2 test. To set this up, counts within a row are compared to the probablity in another row. The probability from the second matirx row is simply $\frac{x_i}{\sum x_i}$, and is the markovian result we have used to date.

To do goodness of fit comparisons, I used the `get.mgof` function. If any probabilities are 0, the χ^2 has issues. I added $1e-30$ to alleviate this problem. Perhaps there is a better way?

Compare between seasons of same run/stock (e.g. Western Atlantic > 185cm)

```r
bw.bs = get.mgof(watl.big.ct[[1]], watl.big[[2]])
print(bw.bs)
```

```
FALSE [1] 2.370314e-100  0.000000e+00 1.512262e-285  9.761739e-37  4.226017e-13
FALSE [6]  0.000000e+00  7.020061e-03
```

Compare between runs, same season

```r
# same row, same season, large and small WABFT
# (bg.sm = get.loglin(watl.big.ct[[1]], watl.small[[1]]))

bg.sm = data.frame(sapply(1:4, function(x) get.mgof(watl.big.ct[[x]], watl.small[[x]])))
names(bg.sm) =  names(watl.big)
kable(bg.sm, row.names = 1:7, caption = "Western Atlantic bluefin > 185 cm vs < 185 cm", digits = 4)
```

Table 3: Western Atlantic bluefin > 185 cm vs < 185 cm

	Winter	Spring	Summer	Fall
1	0.0000	0.0000	0.0000	0.0000
2	0.0209	0.8376	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000
5	0.0144	0.6251	0.0000	0.0000
6	0.0000	0.0078	0.0096	0.0000
7	1.0000	0.2848	0.0000	0.1587

Compare stocks and season

```
e.w.s = data.frame(sapply(1:4, function(x) get.mgof(watl.big.ct[[x]], eatl[[x]])))
names(e.w.s) =  names(watl.big)

# library(xtable)

# print(xtable(e.w.s, include.rownames = T), type = 'html')
kable(e.w.s, row.names = 1:7, caption = "Western Atlantic bluefin > 185 cm vs Eastern Atlantic bluefin"
```

Table 4: Western Atlantic bluefin > 185 cm vs Eastern Atlantic bluefin

	Winter	Spring	Summer	Fall
1	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.6999	0.0000
7	0.9228	0.9756	0.7166	0.8458

Compare between stocks, non-seasonal

this doesn't work... need a different test if this comparison is neccesary

```
# get overall transition rate; multiply the seasonal matrices
b1 = watl.big
b2 = eatl

ball.w = b1[[1]]%*%b1[[2]]%*%b1[[3]]%*%b1[[4]]
ball.e = b2[[1]]%*%b2[[2]]%*%b2[[3]]%*%b2[[4]]

(e.w = get.loglin(ball.e, ball.w))
```

Tradeoff between degrees of freedom and likelihood ratio within the χ^2 test

this helped answer a question I had earlier