

Shifts in the size of fish from a culturally important recreational fishery

Mark Scheuerell

07 August 2020

Data

The original data were provided to me by Tom Quinn on 16 June 2020 in the form of an MS Excel file titled `Tengu_derby_leaders through 2019 derby.xls`. I exported the one worksheet of interest (data in kg) as `~/data/tengu_derby_data.csv`.

```
## set data dir
datadir <- here::here("data")
## import raw data
raw_data <- readr::read_csv(file.path(datadir, "tengu_derby_data.csv"))

## Parsed with column specification:
## cols(
##   derby = col_double(),
##   year = col_double(),
##   month = col_character(),
##   days = col_double(),
##   members = col_double(),
##   total_catch = col_double(),
##   n_over_10 = col_double(),
##   n_over_5 = col_double(),
##   size_1 = col_double(),
##   size_2 = col_double(),
##   size_3 = col_double(),
##   size_4 = col_double(),
##   size_5 = col_double()
## )
```

Fishing effort

This dataset lacks the necessary detail to calculate a proper index of the catch per unit effort (CPUE) because effort is ill-defined. That is, although we know the total number of days per year the derby was open and the total number of anglers that participated per year, we don't know how many days *each* angler fished. Therefore, we assumed that each angler fished every day the derby

was open, which is almost certainly an overestimate of the true effort, but we have no reason to believe there would be any systematic change over time.

```
## CPUE
cpue <- raw_data$total_catch / (raw_data$days * raw_data$members)

## time series plot of CPUE
par(mai = c(0.9, 0.9, 0.1, 0.1))
plot.ts(ts(cpue, start = min(raw_data$year)), lwd = 2, col = "blue",
        las = 1, log = "y",
        ylab = "CPUE", xlab = "Year", yaxt = "n")
axis(2, at = c(0.01, 0.02, 0.05, 0.1), las = 1)
```

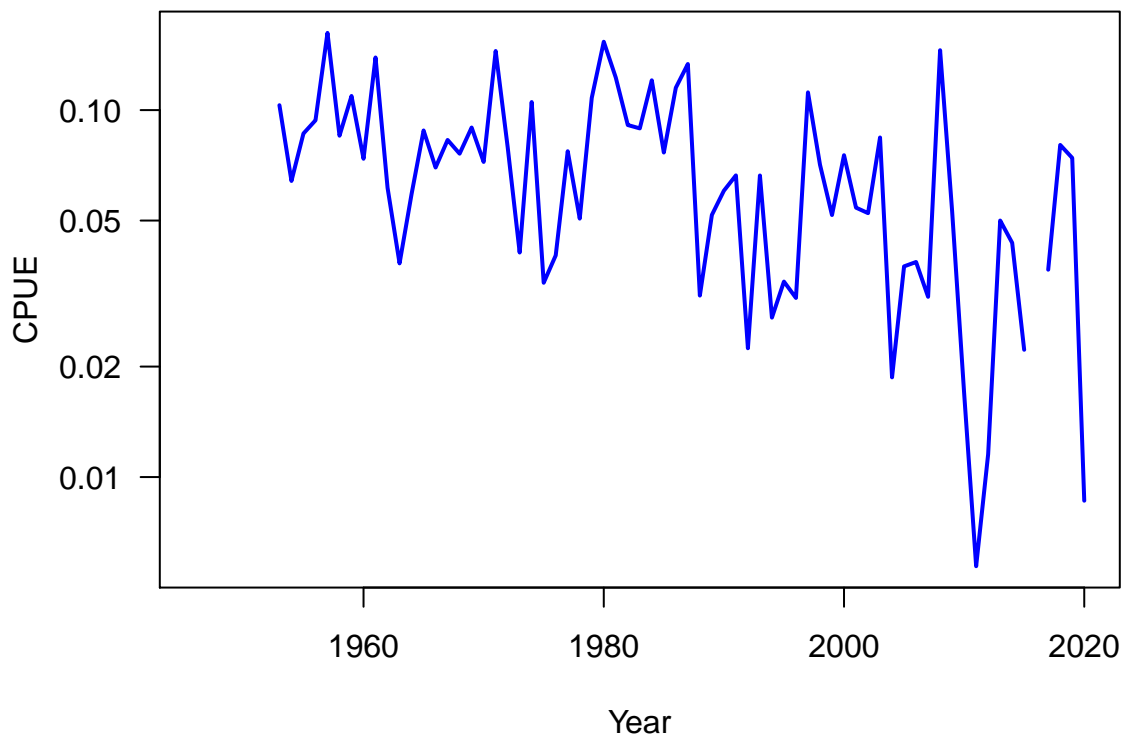


Figure 1. Time series of catch per unit effort of blackmouth from the Tengu Derby. Note that the y-axis is on a natural log scale.

Changes in catch over time

The CPUE appears to generally decline over time, so I fit two random walk models (with and without a bias term) to estimate the data support for a decline. Because CPUE is necessarily bounded by zero at a minimum, I fit models to the log-transformed CPUE. Specifically, the models are

$$c_t = c_{t-1} + u + e_t \quad (1)$$

and

$$c_t = c_{t-1} + e_t \quad (2)$$

where c_t is the \log_e -transformed CPUE, u is a bias term, and $e_t \sim N(0, \sigma)$.

```
## model setup
mod_list <- list(
  B = matrix(1),
  U = matrix("u"),
  Q = matrix("q"),
  Z = matrix(1),
  A = matrix(0),
  R = matrix(0)
)
## response
l_cpue <- matrix(log(cpue), nrow = 1)
```

Fit biased random walk

```
## fit model with bias (Eqn 1)
cpue_brw <- MARSS(l_cpue, model = mod_list)

## Success! abstol and log-log tests passed at 17 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 17 iterations.
## Log-likelihood: -74.14775
## AIC: 154.2955   AICc: 154.6705
##
##      Estimate
## U.u    -0.0406
## Q.q     0.4947
## x0.x0  -1.7089
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
## 95% CI for bias
cpue_brw <- MARSSparamCIs(cpue_brw)
```

Fit unbiased random walk

```
## fit model without bias (Eqn 2)
mod_list$U <- matrix(0)
cpue_rw <- MARSS(l_cpue, model = mod_list)

## Success! abstol and log-log tests passed at 17 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 17 iterations.
## Log-likelihood: -74.25323
## AIC: 152.5065   AICc: 152.6911
##
##      Estimate
## Q.q      0.496
## x0.x0    -1.953
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

The 95% confidence interval for the bias term (u) is (-0.21, 0.13), and the model with a bias term has an AIC value ~1.8 units greater than the model without a bias term, suggesting there is no data support for a systematic downward trend in the log-transformed CPUE over time.

Variance in catch over time

The time series of CPUE (Figure 1) appears to exhibit increasing variance over time, so here is a plot of the residuals from the better fitting random walk model (Eqn 2).

```
## Model residuals
rw_resids <- t(residuals(cpue_rw)$state.residuals)

## time series plot of CPUE
par(mai = c(0.9, 0.9, 0.1, 0.1))
plot.ts(ts(rw_resids, start = min(raw_data$year)), lwd = 2, col = "blue",
        las = 1, ylab = "Model residuals", xlab = "Year")
```

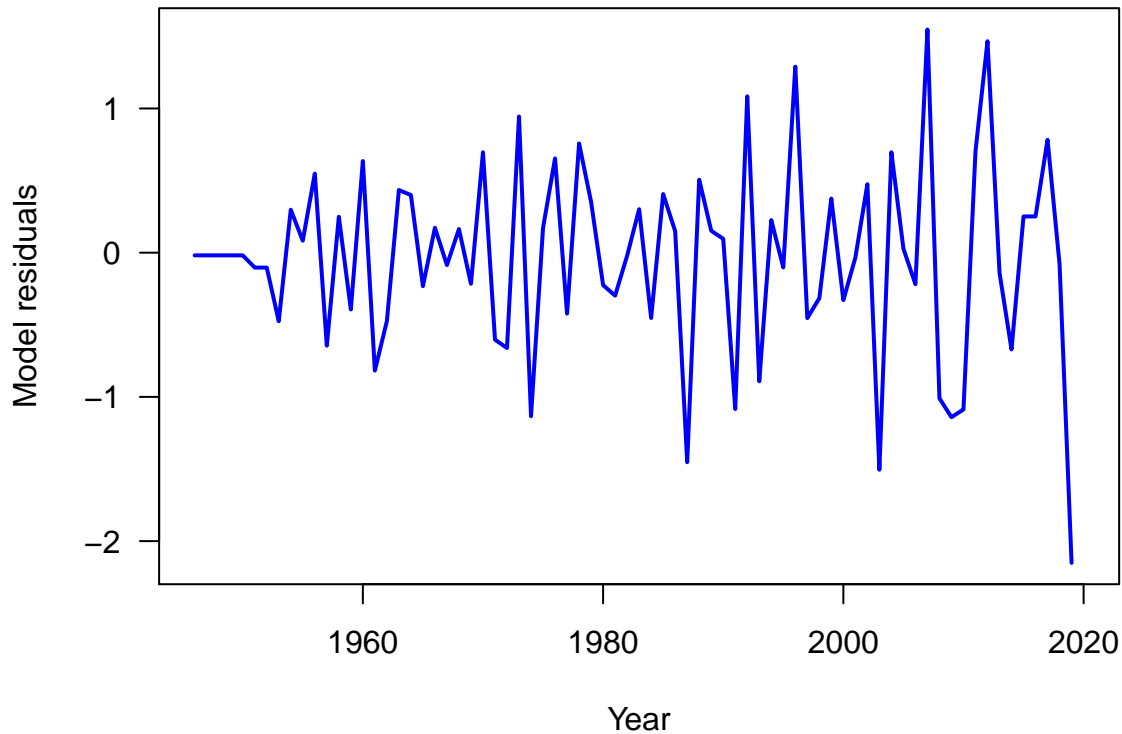


Figure 2. Time series of the residuals from the random walk model (Eqn 2).

The residuals do indeed appear heteroscedastic with increasing variance over time. The model was fit to log-transformed CPUE, so it's not clear another transformation would help.

Changes in fish size over time

The data set contains three different indicators of fish size over time:

- 1) the total number of fish over 10 pounds (~4.55 kg);
- 2) the total number of fish over 5 pounds (~2.27 kg); and
- 3) the masses (kg) of the 5 largest fish.

Clearly (1) and (2) will be correlated, as (1) is a subset of (2). To see how they relate to (3), I summed the sizes of the 5 largest fish and compare that metric to (1) and (2). Here are the correlations among the three.

```
## total mass of 5 largest fish
sum_of_top_5 <- apply(raw_data[,c(9:13)], 1, sum, na.rm = TRUE)
sum_of_top_5[sum_of_top_5 == 0] <- NA

## 3 size metrics
fish_sizes <- cbind("N over 10 lbs" = raw_data$n_over_10,
                    "N over 5 lbs" = raw_data$n_over_5,
                    "Sum of top 5 (kg)" = sum_of_top_5) %>%
```

```

ts(start = min(raw_data$year))

## correlation among the 3 metrics
cor(fish_sizes, use = "pairwise.complete.obs") %>%
  round(2)

##           N over 10 lbs N over 5 lbs Sum of top 5 (kg)
## N over 10 lbs           1.00         0.68           0.82
## N over 5 lbs            0.68         1.00           0.62
## Sum of top 5 (kg)       0.82         0.62           1.00

```

As suspected, all three metrics are correlated, with the total number of fish over 10 pounds and the total mass (kg) of the five largest fish being particularly so.

Here are plots of the three size metrics over time.

```

## time series plot of fish size metrics
par(mai = c(0.9, 0.9, 0.1, 0.1))
plot.ts(fish_sizes, lwd = 2, col = "blue",
        main = "", yax.flip = TRUE,
        ylab = c("", "", ""), xlab = "Year")

```

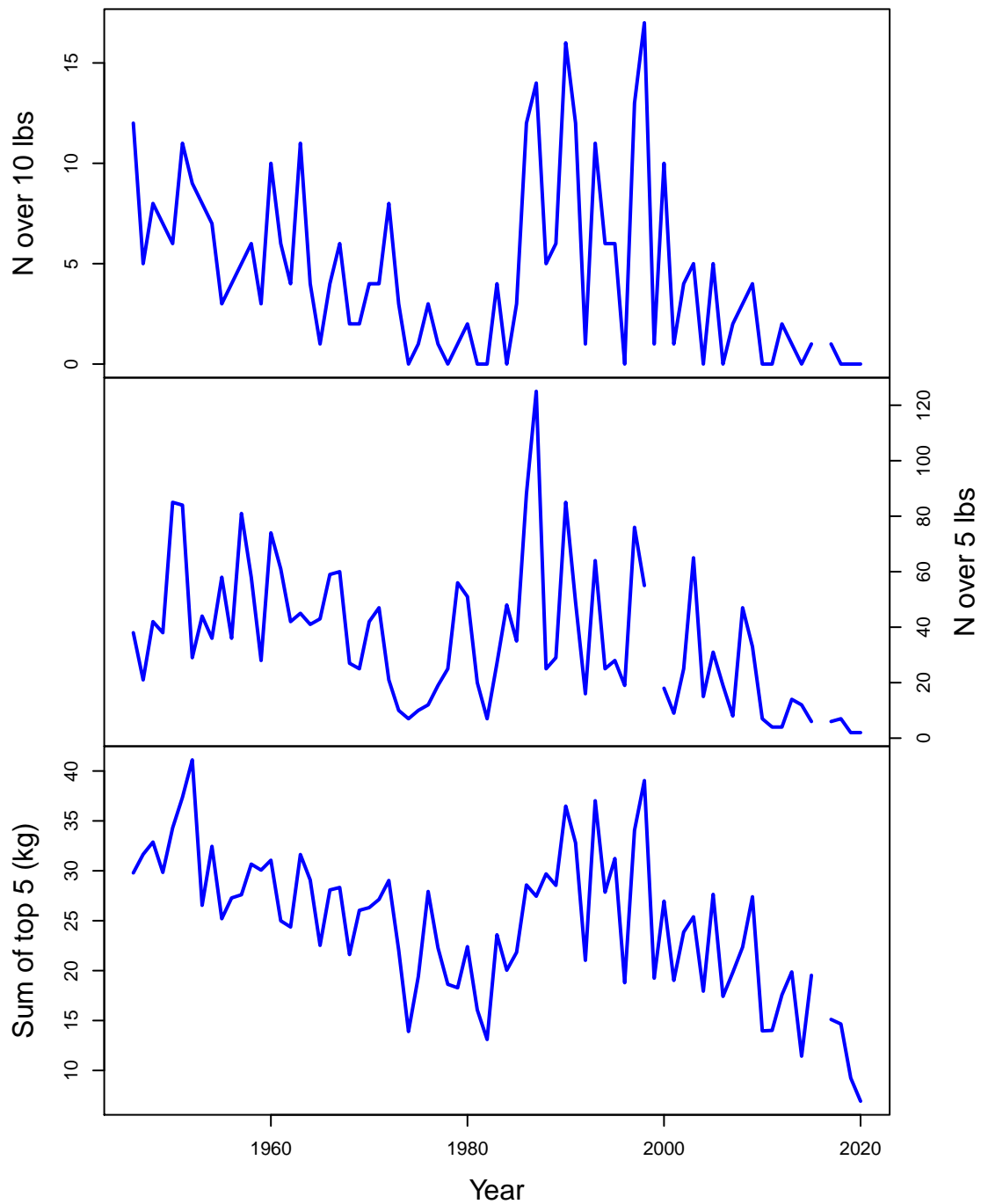


Figure 3. Time series of fish size over time, including the total number of fish over 10 pounds (top), the total number of fish over 5 pounds (middle), and the total mass (kg) of the five largest fish (bottom).

There appears to be an overall decline in fish size from the mid 1940s until the mid 1980s, when fish sizes increased rapidly before declining again until present.

Fit biased random walk

Just as I did with the CPUE data, I fit both biased and unbiased forms of random walk models to the fish size data. Here I use the log-transformed sum of the five largest fish.

```
## model setup
mod_list <- list(
  B = matrix(1),
  U = matrix("u"),
  Q = matrix("q"),
  Z = matrix(1),
  A = matrix(0),
  R = matrix(0)
)
## response
l_size <- matrix(log(sum_of_top_5), nrow = 1)

## fit model with bias (Eqn 1)
size_brw <- MARSS(l_size, model = mod_list)

## Success! algorithm run for 15 iterations. abstol and log-log tests passed.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Algorithm ran 15 (=minit) iterations and convergence was reached.
## Log-likelihood: -13.00375
## AIC: 32.00751   AICc: 32.35036
##
##      Estimate
## U.u      -0.0197
## Q.q       0.0824
## x0.x0     3.4139
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.

## 95% CI for bias
size_brw <- MARSSparamCIs(size_brw)
```

Fit unbiased random walk

```
## fit model without bias (Eqn 2)
mod_list$U <- matrix(0)
size_rw <- MARSS(l_size, model = mod_list)
```



```

## Success! algorithm run for 15 iterations. abstol and log-log tests passed.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Algorithm ran 15 (=minit) iterations and convergence was reached.
## Log-likelihood: -13.17801
## AIC: 30.35602   AICc: 30.52503
##
##           Estimate
## Q.q       0.0828
## x0.x0     3.3942
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.

```

The 95% confidence interval for the bias term (u) is $(-0.09, 0.05)$, and the model with a bias term has an AIC value ~ 1.7 units greater than the model without a bias term, suggesting there is no data support for a systematic downward trend in the log-transformed size over time.