Shifts in the size of Chinook salmon from a culturally important recreational fishery

Supplementary material to accompany Quinn et al.

Mark Scheuerell

23 September 2021

Data

The original Tengu Derby 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 one worksheet of interest (`data in kg`) as `~/data/tengu_derby_data.csv`. The same Excel file also included a worksheet with information from WDFW on the number of natural- and hatchery-origin Chinook, and the mean mass of Chinook (`Losee Chiook data`), which I exported as `~/data/wdfw_data.csv`

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

```
##
## -- 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()
## )
```

```
## import raw WDFW data
wdfw_data <- readr::read_csv(file.path(datadir, "wdfw_data.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   year = col_double(),
##   NOR = col_double(),
##   HOR = col_double(),
##   total = col_double(),
##   size = col_double()
## )
```

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). Furthermore, the probability of catching a fish greater than 5 or 10 pounds clearly increases as the total number of fish caught also increases. Thus, we calculated the proportion of fish caught in a given year that were greater than the two size thresholds.

Proportions of large fish

```
## proportion of fish >5 lbs & >10 lbs
size_props <- tengu_data %>%
  mutate(
    ## proportion of fish >5 lbs
    p5 = n_over_5 / total_catch,
    ## proportion of fish >10 lbs
    p10 = n_over_10 / total_catch) %>%
  ## select cols of interest
  select(year, p5, p10) %>%
  ## remove years prior to 1950 with all NA's
  filter(!(is.na(p5) & is.na(p10)))
```

Here are plots of the two proportional size metrics over time.

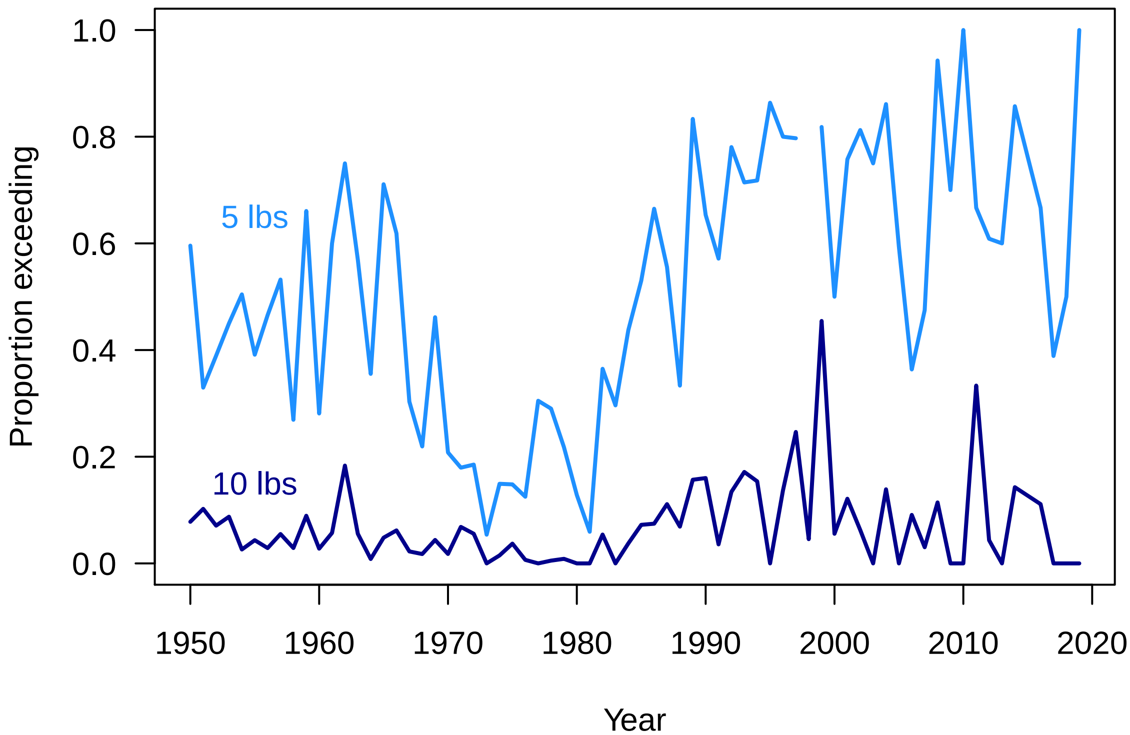


Figure 1. Time series of the proportion of fish caught in the Tengu Derby that exceeded 5 pounds (light blue) and 10 pounds (dark blue).

There appears to be an overall decline in fish size from 1950 until the early 1980s, when fish sizes increased rapidly before declining again until present.

Mean of 5 largest fish

```
## get mean of top-5; code as NA if <5 fish were weighed
tengu_sizes <- tengu_data %>%
  #filter(year >= 1970 & year <= 2014) %>%
  select(starts_with(c("year", "size"))) %>%
  rowwise(year) %>%
  summarise(tengu_mean_kg = mean(c_across(everything())))
```

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

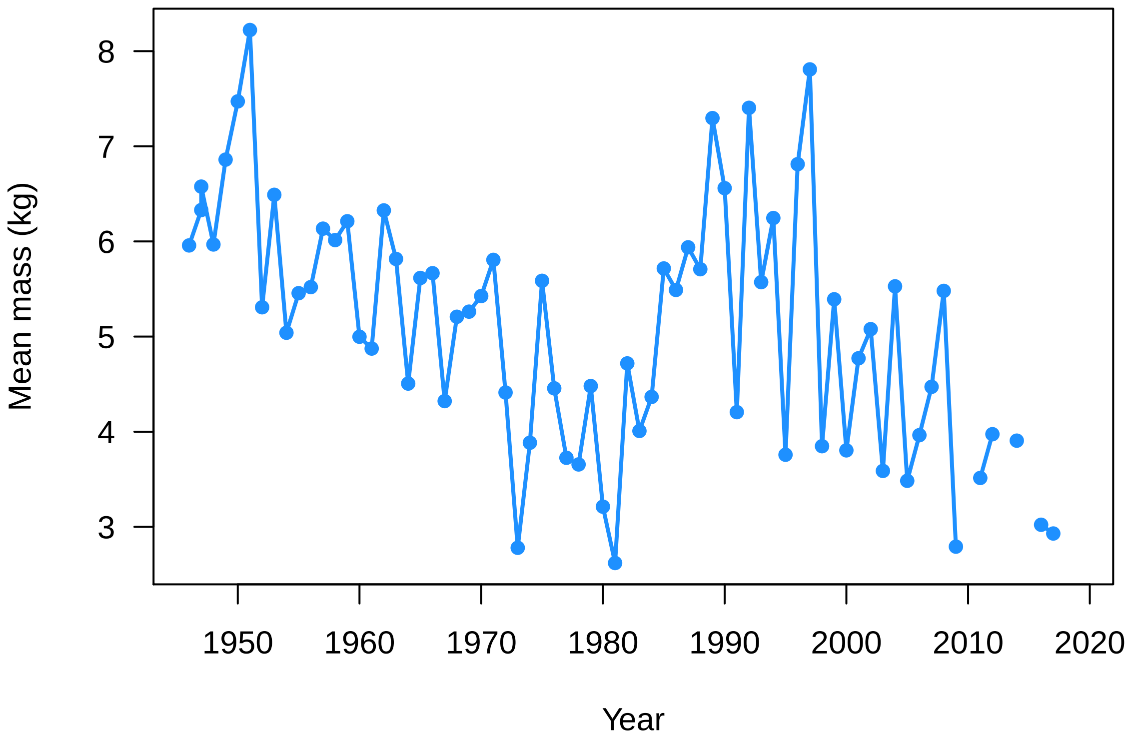


Figure 2. Time series of the mean mass of the five largest fish caught in the Tengu Derby.

Models for Tengu Derby fish mass

Unbiased random walk

```
## model defn
mod_list <- list(
  B = matrix(1),
  U = matrix(0),
  Q = matrix("q"),
  Z = matrix(1),
  R = matrix("r")
)

## data for {MARSS}
yy <- matrix(tengu_sizes$tengu_mean_kg, nrow = 1) %>%
  log()

## fit unbiased RW
size_tengu_urw <- MARSS(yy, model = mod_list)

## Success! abstol and log-log tests passed at 40 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 40 iterations.
## Log-likelihood: 6.656424
## AIC: -7.312848   AICc: -6.949212
##
##        Estimate
## R.r     0.03415
## Q.q     0.00434
## x0.x0   1.84001
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

Biased random walk

```
## update model defn
mod_list$U <- matrix("u")

## fit biased RW
size_tengu_brw <- MARSS(yy, model = mod_list)
```

```
## Success! abstol and log-log tests passed at 45 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 45 iterations.
## Log-likelihood: 7.467964
## AIC: -6.935929   AICc: -6.320544
##
##        Estimate
## R.r     0.03468
## U.u    -0.00952
## Q.q     0.00352
## x0.x0   1.87554
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

Compare models

```
## AICc for unbiased RW
size_tengu_urw$AICc

## [1] -6.949212

## AICc for biased RW
size_tengu_brw$AICc

## [1] -6.320544
```

Plot model fit

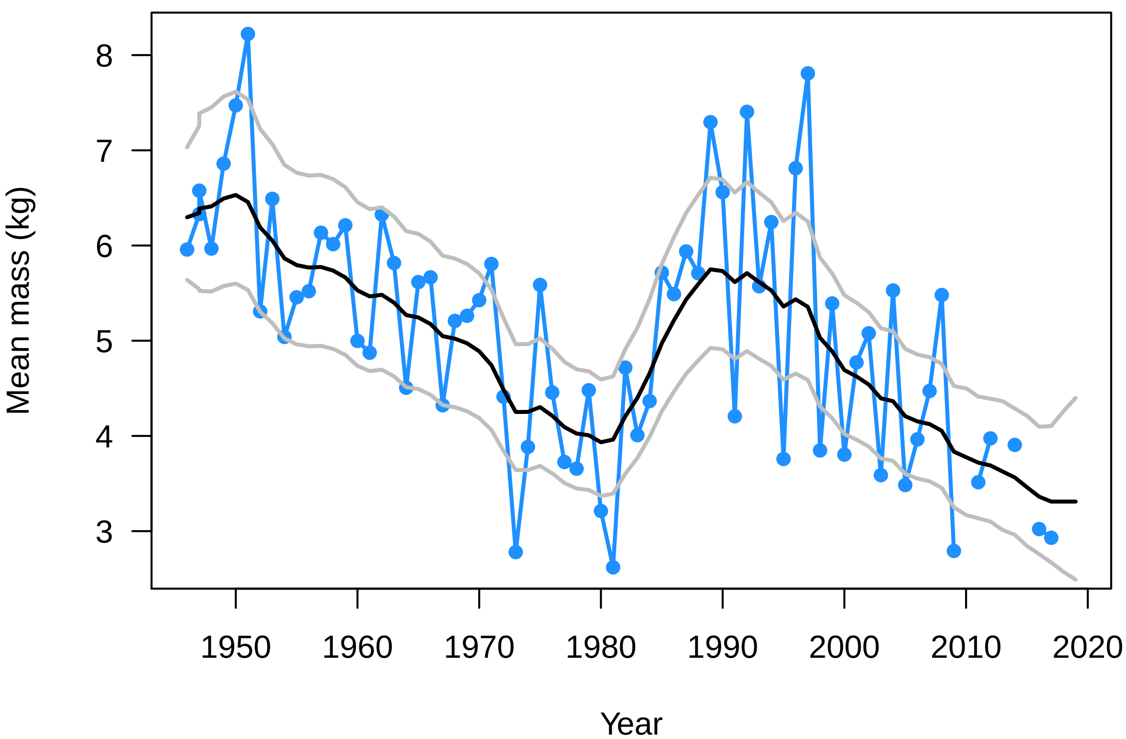


Figure 3. Time series of the mean mass of the five largest fish caught in the Tengu Derby (solid) and the model fitted values (dashed).

Comparisons between the Tengu Derby & WDFW

Combined data sets

The time series from WDFW begins in 1970 and runs through 2019, so we restricted our analysis to the 45 years from 1970-2019. Again, we fit models to the log-transformed size data.

```
## get corresponding size data from WDFW
wdfw_sizes <- wdfw_data %>%
  ## drop first 2 years with NA's for size
  drop_na(size) %>%
  ## keep year & size
  select(year, size) %>%
  ## rename size
  rename(wdfw_mean_kg = size)

## combine data sets for {MARSS}
y2 <- tengu_sizes %>%
  left_join(wdfw_sizes, by = "year") %>%
  ungroup() %>%
  ## drop non-overlapping years
  filter(year >= 1970 & year <= 2019) %>%
  ## drop year
  select(-year) %>%
  ## log-transform
  log() %>%
  t()
```

One-state models

Unbiased random walk

```
## model defn
mod_list <- list(
  B = matrix(1),
  U = matrix(0),
  Q = matrix("q"),
  Z = matrix(1, nrow = 2, ncol = 1),
  A = matrix(c("T", "W"), nrow = 2, ncol = 1),
  R = matrix(list("r", 0, 0, "r"), 2, 2)
)

## fit unbiased RW (Eqns 5 & 6)
size_both_1u <- MARSS(y2, model = mod_list)
```

```
## Success! abstol and log-log tests passed at 43 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 43 iterations.
## Log-likelihood: 17.78598
## AIC: -25.57196   AICc: -24.89781
##
##       Estimate
## A.T   -0.24495
## A.W    0.23233
## R.r    0.03512
## Q.q    0.00145
## x0.x0  1.82155
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

Biased random walk

```
## update model list
mod_list$U <- matrix("u")

## fit biased RW (Eqns ? & ?)
size_both_1b <- MARSS(y2, model = mod_list)
```

```
## Success! abstol and log-log tests passed at 50 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 50 iterations.
## Log-likelihood: 18.57263
## AIC: -25.14526   AICc: -24.19072
##
##       Estimate
## A.T   -0.24470
## A.W    0.23348
## R.r    0.03524
## U.u   -0.00662
## Q.q    0.00108
## x0.x0  1.84343
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
```

```
## Use MARSSparamCIs to compute CIs and bias estimates.
```

Two-state models

Unbiased random walk

```
## model defn
mod_list <- list(
  B = diag(2),
  U = matrix(0, nrow = 2, ncol = 1),
  Q = matrix(list("T", 0, 0, "W"), 2, 2),
  # Q = matrix(list("q", 0, 0, "q"), 2, 2),
  Z = diag(2),
  # A = matrix(c("T", "W"), nrow = 2, ncol = 1),
  A = matrix(c(0, 0), nrow = 2, ncol = 1),
  # R = matrix(list("r", 0, 0, "r"), 2, 2)
  R = matrix(list("T", 0, 0, "W"), 2, 2)
)

## fit biased RW (Eqns ? & ?)
size_both_2u <- MARSS(y2, model = mod_list)
```

```
## Success! abstol and log-log tests passed at 50 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 50 iterations.
## Log-likelihood: 36.2337
## AIC: -60.46739   AICc: -59.51285
##
##                      Estimate
## R.T                   0.04668
## R.W                   0.00886
## Q.T                   0.00516
## Q.W                   0.00120
## x0.X.tengu_mean_kg    1.54080
## x0.X.wdfw_mean_kg     2.10603
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

Unbiased random walk

```
## update model defn
mod_list$U <- matrix(c("T", "W"), nrow = 2, ncol = 1)

## fit biased RW
size_both_2b <- MARSS(y2, model = mod_list)
```

```
## Success! abstol and log-log tests passed at 106 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 106 iterations.
## Log-likelihood: 38.01458
## AIC: -60.02917   AICc: -58.35475
##
##                       Estimate
## R.T                   0.046693
## R.W                   0.009873
## U.T                  -0.007889
## U.W                  -0.006955
## Q.T                   0.004720
## Q.W                   0.000388
## x0.X.tengu_mean_kg  1.564683
## x0.X.wdfw_mean_kg   2.131286
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

Compare models

```
## AICc for unbiased RW
size_both_2u$AICc
```

```
## [1] -59.51285
```

```
## AICc for biased RW
size_both_2b$AICc
```

```
## [1] -58.35475
```

The unbiased random walk has an AICc score that is slightly greater than the biased random walk, but it's more simple and should be favored here.

Plot model fits

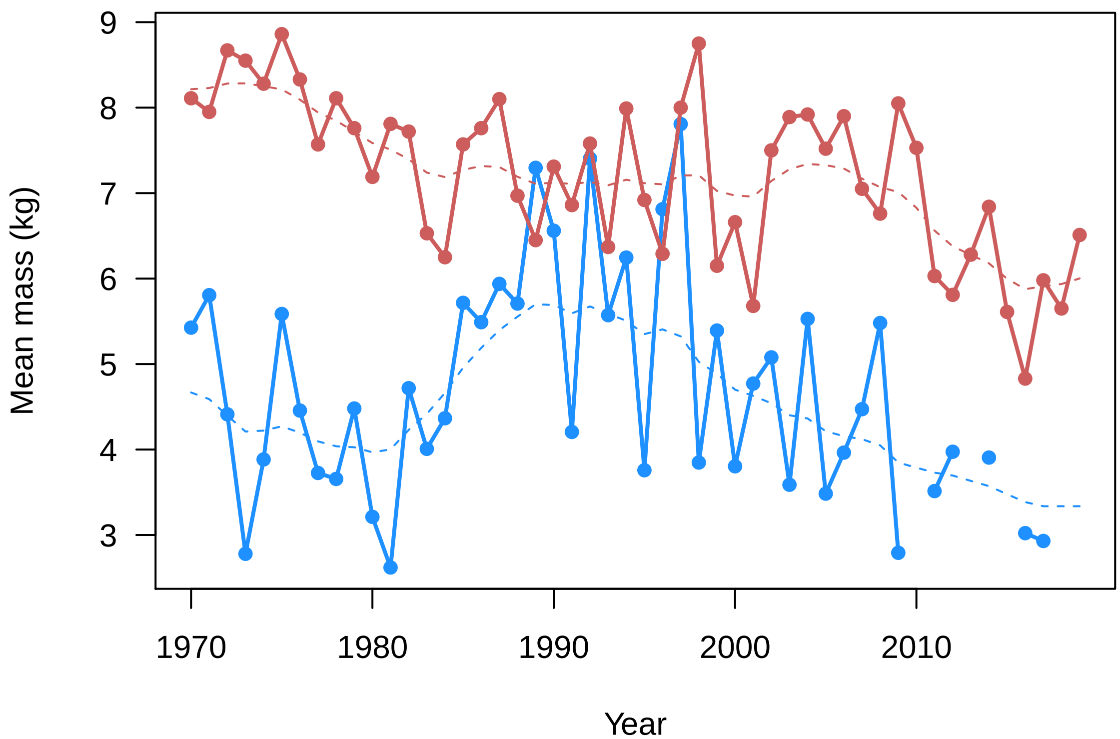


Figure 4. Time series of observed fish size from the Tengu derby (blue) and WDFW surveys (red), including fits from the multivariate random walk model for both series (dashed lines).

Summary of size comparison

The model with one common state has an AICc value of -24.9 and the model with two unique states has an AICc value of -59.5, which indicates rather strong support for two unique temporal patterns in the data.

Additional figures for paper

Plot fitted values for one-state model

Figure 5. Time series of observed fish size from the Tengu derby (blue) and WDFW surveys (red), including fits from the multivariate random walk model (Eqns 6 & 7; dashed).

Other size metrics

Proportions of large fish

```r
## proportion of fish >5 lbs & >10 lbs
big_per_angler <- tengu_data %>%
  mutate(
    ## proportion of fish >5 lbs
    b5 = n_over_5 / members,
    ## proportion of fish >10 lbs
    b10 = n_over_10 / members) %>%
  ## select cols of interest
  select(year, b5, b10)
```

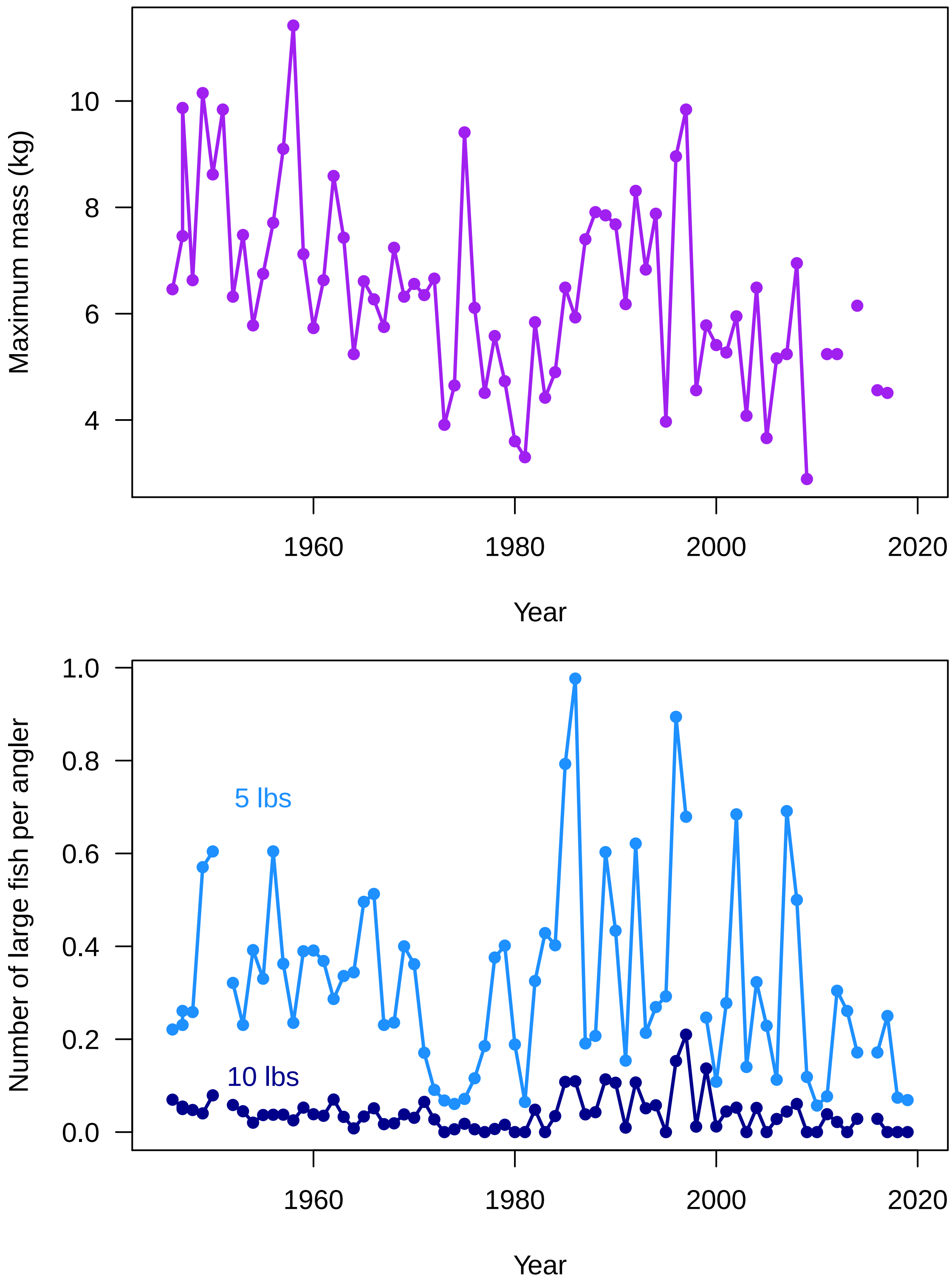
Here are some additional plots for the supplemental material.

```r
tengu_max <- tengu_data %>%
  #filter(year >= 1970 & year <= 2019) %>%
  select(starts_with(c("year", "size"))) %>%
  rowwise(year) %>%
  summarise(tengu_max_kg = max(c_across(everything())))

## plot other size metrics
par(mfrow = c(2,1),
    mai = c(0.9, 0.9, 0.1, 0.1))

## largest fish
plot(tengu_max$year, tengu_max$tengu_max_kg,
     las = 1, type = "o", pch = 16, lwd = 2,
     col = "purple", xlim = c(1945, 2020),
     xlab = "Year",
     ylab = "Maximum mass (kg)")

## proportion exceeding
matplot(big_per_angler[,1], big_per_angler[,(2:3)],
        las = 1, pch = 16, xlim = c(1945, 2020),
        lwd = 2, type = "o", lty = "solid",
        col = c("dodgerblue", "darkblue"),
        xlab = "Year", ylab = "Number of large fish per angler")
text(1955, 0.72, "5 lbs", col = "dodgerblue")
text(1955, 0.12, "10 lbs", col = "darkblue")
```


Maximum mass (kg)
10
8
6
4
1960
1980
2000
2020
Year
Number of large fish per angler
1.0
0.8
0.6
0.4
0.2
0.0
5 lbs
10 lbs
1960
1980
2000
2020
Year