# Package: taigatools (via r-universe)

June 7, 2026

**Title** Tools for Boreal Caribou Demography

**Version** 0.0.0.9000

**Description** Spatial and demographic tools for caribou survival, recruitment, movement, and population growth rate estimation. Developed for Northwest Territories boreal caribou system as a part of CIMP247.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 8.0.0

**URL** https://github.com/chlobeau/taigatools

**BugReports** https://github.com/chlobeau/taigatools/issues

**Repository** https://chlobeau.r-universe.dev

**Date/Publication** 2026-06-04 13:40:18 UTC

**RemoteUrl** https://github.com/chlobeau/taigatools

**RemoteRef** HEAD

**RemoteSha** 02bb6f434cb3ee137724f5979d6d8dbd74828df4

## Contents


addBufferID . . . . . 2
addPropAssigned . . . . . 2
assignBounds . . . . . 3
calculateRecruitment . . . . . 3
combineMoveMort . . . . . 4
countPtsInPolygons . . . . . 5
estBetaParams . . . . . 5
estLogNormalParams . . . . . 6
getAnnualSurvival . . . . . 6
getMean . . . . . 7
getSurvTable . . . . . 7

resampleDaily . . . . . 8
runLambdaMCMC . . . . . 8
setupMonthlySurvival . . . . . 9

**Index** **10**

---

`addBufferID` *Assign buffer group IDs to point locations*

---

## Description

Buffers each point by half of buffer_dist, unions overlapping buffers within each biological year, and assigns a unique bufferID to each resulting polygon. Points are then spatially joined to their buffer group and returned as an sf with centroid coordinates. Used to group composition survey observations that were collected in the same area within a biological year.

## Usage

```
addBufferID(sf, buffer_dist = 500)
```

## Arguments

| | |
|---|---|
| `sf` | sf of point locations with column bioYear |
| `buffer_dist` | numeric; total buffer diameter in metres (default 500); halved internally so each point is buffered by a 250m radius |

## Value

sf with added columns bufferID, x, and y (buffer centroid coordinates)

---

`addPropAssigned` *Add proportion of time assigned to study area*

---

## Description

Add proportion of time assigned to study area

## Usage

```
addPropAssigned(df)
```

## Arguments

| | |
|---|---|
| `df` | one-row data frame (typically a single row from countPtsInPolygons output) with columns assigned_new and total_days |

## Value

df with added column prop_assigned

## assignBounds — *Assign boundary labels to individuals by lifetime and annual mean location*

### Description

Assign boundary labels to individuals by lifetime and annual mean location

### Usage

```
assignBounds(mean_sf, annual_mean_sf, bounds_sf, col_name)
```

### Arguments

| | |
|---|---|
| mean_sf | sf of lifetime mean locations with column ID |
| annual_mean_sf | sf of annual mean locations with columns ID and bioYear |
| bounds_sf | sf multipolygon boundary layer |
| col_name | unquoted name of the column in bounds_sf identifying unique polygons |

### Value

a list with two data frames:

| | |
|---|---|
| lifetime | ID and boundary assignment based on lifetime mean location |
| annual | ID, bioYear, and boundary assignment based on annual mean location |

## calculateRecruitment — *Calculate recruitment from composition survey data*

### Description

Calculates calf:cow ratios and recruited female rates from aerial composition surveys using the NWT method:

### Usage

```
calculateRecruitment(comp_sf, grouping_col = "StudyArea")
```

### Arguments

| | |
|---|---|
| comp_sf | sf of composition survey locations with columns bioYear, Cow, Bull, Calf, Yearling, Unk |
| bounds2_sf | sf multipolygon of study area boundaries with column StudyArea2 |

### Details

1. Totals observations by sex and age class per year
2. Apportions unknown adults to cows and bulls based on a 50/50 sex ratio
3. Apportions unknown age-sex animals among cows, bulls, and calves
4. Calculates calf:cow ratio using adjusted cow and calf counts
5. Calculates SE of calf:cow ratio using the binomial approximation (Cochran 1977 p. 52 Eq 3.11; Krebs 1999 Eq 8.13, excluding sampling)

Note: sex ratio apportionment follows Allicia Klapatiuk (GNWT) 50/50 method rather than the Rettie proportion-based approach.

### Value

data frame with one row per study area per biological year with columns Years, GroupsObs, Bulls, Cows, Calves, Total, CalfCow, CalfCow_SE, RecruitedFemales, RecruitedFem_SE, BullCow, Rrm

---

## `combineMoveMort` *Merge movement and survival data*

---

### Description

Merge movement and survival data

### Usage

```
combineMoveMort(movement_data, survival_data)
```

### Arguments

| | |
|---|---|
| `movement_data` | sf of GPS locations with columns ID, Time, Lon, Lat |
| `survival_data` | data frame with columns ID, startDate, endDate, Status |

### Value

sf of point locations trimmed to each animal's monitoring period, with added columns deathDate and Death (TRUE on the date of mortality or undetermined fate censoring)

## countPtsInPolygons *Count points in polygons per individual*

### Description

Counts how many locations each animal has within polygon boundaries. Useful for validating study area assignments or identifying where animals spend most of their time.

### Usage

```
countPtsInPolygons(
  pts_sf,
  bounds,
  bounds_col,
  time_col = "Time",
  assigned_col = "study_site"
)
```

### Arguments

| | |
|---|---|
| pts_sf | sf of point locations; must contain an ID column plus the columns named by time_col and assigned_col |
| bounds | sf multipolygon boundary layer |
| bounds_col | character; name of the column in bounds identifying unique polygons |
| time_col | character; name of the POSIXct datetime column (default "Time") |
| assigned_col | character; name of the study area assignment column (default "study_site") |

### Value

data frame with columns: ID, start, end, assigned, total_pts, polygon_pts for each boundary, total_days, and polygon_days for each boundary

## estBetaParams *Estimate beta distribution parameters from mean and variance*

### Description

Estimate beta distribution parameters from mean and variance

### Usage

```
estBetaParams(mu, var)
```

### Arguments

| | |
|---|---|
| `mu` | numeric; mean of the distribution |
| `var` | numeric; variance of the distribution |

### Value

list with elements alpha and beta

---

## `estLogNormalParams` *Estimate lognormal distribution parameters from mean and variance*

---

### Description

Estimate lognormal distribution parameters from mean and variance

### Usage

```
estLogNormalParams(mean, var)
```

### Arguments

| | |
|---|---|
| `mean` | numeric; mean of the distribution |
| `var` | numeric; variance of the distribution |

### Value

list with elements mu and sigma2

---

## `getAnnualSurvival` *Calculate annual survival from monthly Kaplan-Meier estimates*

---

### Description

Multiplies monthly survival estimates across the biological year (April-March) to yield an annual product-limit estimate, with variance approximated by the sum of monthly variances weighted by the squared annual survival.

### Usage

```
getAnnualSurvival(monthly_survival, grouping_col = "StudyArea")
```

### Arguments

| | |
|---|---|
| `monthly_survival` | data frame from getSurvTable with columns survival, survival_var, n.total, month, Year, and a grouping column |
| `grouping_col` | character; name of the grouping column (default "StudyArea") |

### Value

data frame with one row per group per biological year with columns Years, km_survival, km_var, km_se, survival_LCL, survival_UCL, n_survival, n_survival_max, km_95ci_finite

---

## getMean — *Get mean location and attributes per individual*

---

### Description

Get mean location and attributes per individual

### Usage

```
getMean(x, id.col = "ID")
```

### Arguments

| | |
|---|---|
| x | sf or data frame with an ID column |
| id.col | character; name of the ID column (default "ID") |

### Value

sf or data frame with one row per individual containing mean numeric values, first value for character/factor columns, and mean timestamp for POSIXct columns

---

## getSurvTable — *Fit Kaplan-Meier survival model and return monthly estimates*

---

### Description

Fit Kaplan-Meier survival model and return monthly estimates

### Usage

```
getSurvTable(df)
```

### Arguments

| | |
|---|---|
| df | data frame from setupMonthlySurvival with columns month_start, month_end, DeadInMonth |

### Value

data frame with columns n.total, n.dead, survival, survival_se, survival_var, survival_LCL, survival_UCL

## resampleDaily — *Resample to one random location per individual per day*

### Description

Resample to one random location per individual per day

### Usage

```
resampleDaily(move_mort_sf)
```

### Arguments

| | |
|---|---|
| move_mort_sf | sf of point locations with columns ID and Time |

### Value

sf with one randomly selected location per individual per day

## runLambdaMCMC — *Estimate lambda and uncertainty via Monte Carlo simulation*

### Description

Approximates uncertainty in population growth rate (lambda) by drawing 10,000 times from annual survival and recruitment distributions following Hood (2009) and Morris and Doak (2002). Survival is drawn from a beta distribution (bounded 0-1) using estBetaParams; recruited females from a lognormal distribution (bounded at 0) using estLogNormalParams. Lambda is calculated as S / (1 - Rrm) where Rrm is the recruited female rate. Uncertainty is summarized as the SD of the MCMC draws.

### Usage

```
runLambdaMCMC(df, reps = 10000)
```

### Arguments

| | |
|---|---|
| df | one-row data frame with columns km_survival, km_var, RecruitedFemales, RecruitedFem_SE, and lambda |
| reps | integer; number of Monte Carlo draws (default 10,000) |

### Value

df with added columns lambda.mcmc, lambda.sd, lambda.se, lambda.var, lambda_LCL, lambda_UCL

`setupMonthlySurvival` *Expand individual survival records to monthly intervals*

## Description

Expand individual survival records to monthly intervals

## Usage

```
setupMonthlySurvival(mort.df)
```

## Arguments

| | |
|---|---|
| `mort.df` | one-row data frame for a single individual with columns ID, Status, startDate, endDate |

## Value

data frame with one row per month of monitoring, with columns month_start, month_end, Animal_ID, Fate, startDate, endDate, end, StartInMonth, DeadInMonth, CensInMonth, month, Year

# Index

addBufferID, 2
addPropAssigned, 2
assignBounds, 3

calculateRecruitment, 3
combineMoveMort, 4
countPtsInPolygons, 5

estBetaParams, 5
estLogNormalParams, 6

getAnnualSurvival, 6
getMean, 7
getSurvTable, 7

polygon, *5*

resampleDaily, 8
runLambdaMCMC, 8

setupMonthlySurvival, 9