Package: WildLift (via r-universe)

November 18, 2026

Title An Open-source Tool to Guide Decisions for Wildlife Conservation

Date 2021-04-12

Version 0.3.1

Description Compare alternative recovery actions: Linear Feature Restoration, Linear Feature Deactivation, Maternal Penning, Conservation Breeding, Predator Exclosure, Wolf Reduction, and Moose Reduction. Quantifying the trade-offs associated with alternative recovery actions for threatened species. The tool consists of a generalized matrix population model that is parametrized based on information from the published literature or ongoing experiments. Users can input population parameters (e.g., population size and survival rates) or choose from pre-set caribou subpopulations to estimate changes to populations from implementing recovery actions as described in Nagy-Reis et al. (2020) <doi:10.3389/fevo.2020.564508>.

License Apache License (== 2.0) | file LICENSE

Encoding UTF-8

LazyData true

Depends popbio

Suggests knitr, rmarkdown, shiny, shinydashboard, plotly, openxlsx, shinyBS

VignetteBuilder knitr

Repository https://psolymos.r-universe.dev

Date/Publication 2021-04-12 19:19:12 UTC

RemoteUrl https://github.com/psolymos/WildLift

RemoteRef HEAD

RemoteSha d40032ee107de09a5ad93bf28a3877a0674be92a

Contents

breeding 2

linear 4
run_app 6
WildLift 6

Index **10**

`breeding` *Conservation Breeding*

Description

Functions evaluate different options for conservation breeding.

Usage

```
wildlift_matrix(settings, wild=TRUE,
    age.cens=3, age.1st.litter=3, age.calf.max=1)

wildlift_breeding(settings,
    in.inds=10, out.prop=1, f.surv.trans=1,
    j.surv.trans=1, j.surv.red=1,
    tmax=20, pop.start=100,
    breed.early=FALSE, f.preg.capt.2=0.57)

## S3 method for class 'wildlift_breeding'
print(x, ...)
## S3 method for class 'wildlift_breeding'
summary(object, ...)
## S3 method for class 'wildlift_breeding'
plot(x, plot = TRUE, ...)
```

Arguments

`settings`	a settings object returned by `wildlift_settings`.
`wild`	logical, whether to use the penned or wild vital `settings`.
`age.cens`	censoring age. The projection matrix will contain one-year age classes up to `age.cens`, all age classes above this are combined into a single age class.
`age.1st.litter`	integer, female age at 1st litter; lower limit of the age class, default is 3 years, i.e. the [3, Inf] interval.
`age.calf.max`	integer, maximum age to be considered as calf; upper limit of the age class, default is 1 year, i.e. the [0, 1) interval.
`in.inds`	integer vector, number of adult females put into pen each year. 0s are appended to the value meaning that no females are added after the last value (up until year `tmax`).

out.prop	numeric vector between 0 and 1, the proportion of juvenile females transferred from the penned to the recipient herd. The last value is repeated when length is less than tmax. Can be useful if juvenile female transfer should be delayed to build up stock first.
f.surv.trans	numeric between 0 and 1, adult female survival during capture and transportation into the penned population.
j.surv.trans	numeric between 0 and 1, juvenile female survival during capture and transportation from penned to recipient herd.
j.surv.red	numeric between 0 and 1, transported juvenile female survival reduction factor for 1 year following capture and traportation.
tmax	positive integer, number of years to forecast after initial year 0.
pop.start	positive integer, initial population size in year 0 for the recipient and status quo populations (wild not receiving females).
breed.early	logical, allow females in the facility (not in the wild) to reproduce at a younger age (2) if they are well fed. Defaults to FALSE which means reproduction at age 3.
f.preg.capt.2	numeric, fecundity rate for the 2 yrs old (default is 0.57 based on Adam et al. 2019). Only applies when breed.early = TRUE.
x, object	an object to print, summarize, plot, etc.
plot	logical, whether a plot is to be produced.
...	additional arguments to functions.

Details

The conservation breeding functionality is based on vital rates provided via the settings argument. These rates are turned into a projection matrix using wildlift_matrix. This projection matrix reflects survival and reproduction rates based on the settings, tracking females only based on a sex ratio of 0.5.

The wildlift_breeding function tracks changes in the conservation population starting with adding females to produce offspring.

Juvenile females are captured and transferred to the recipient herd. The proportion of juvenile females transferred can be changed through out.prop (0: none, 1: all).

The output object tracks the number of adult females added to the penned population, and the number of juvenile females transferred from the penned to the recipient herd.

Besides these, a status quo (wild without receiving juveniles) population trajectory is used as a reference. This status quo population uses the same vital rates as the recipient herd, but does not get the extra juvenile females from the penned population.

Value

wildlift_matrix returns a projection matrix.

wildlift_breeding returns a conservation breeding object. It contains age specific population sizes for the penned, recipient, and status quo populations (females only).

The print method returns the input object x invisibly.

The summary method returns population summaries for the conservation breeding object.

The plot method return the plotted data invisibly and produces a plot as a side effect.

See Also

wildlift_settings, wildlift_forecast

Examples

```
## projection matrix
s <- wildlift_settings()

wildlift_matrix(s, wild=TRUE)  # wild
wildlift_matrix(s, wild=FALSE) # captive

## out.prop = 0.5: move half of the juveniles
x0 <- wildlift_breeding(s,
    tmax = 20,        # projection horizon
    in.inds = rep(10, 5),
    out.prop = 0.5)
x0
summary(x0)

## out.prop = 1: move all juveniles
x1 <- update(x0, out.prop = 1)
x1

op <- par(mfrow=c(1, 2))
plot(x0, main="out.prop = 0")
plot(x1, main="out.prop = 1")
par(op)
```

---

linear *Linear feature effects on lambda*

Description

Linear feature restoration (i.e. forest regeneration) and linear feature deactivation (e.g. blocking entrances) effects on lambda.

Usage

```
wildlift_linear(tmax=20, pop.start=100,
    area=10000, lin=0, seism=0, young=0,
    cost=12, yr_deact=5, yr_restor=15)
```

Arguments

tmax	positive integer, number of years to forecast after initial year 0.
pop.start	positive integer, initial population size in year 0.
area	positive numeric, area of range in km^2.
lin	non-negative numeric, linear features (total length in km) including seismic lines (to be restored or deactivated) and permanent linear featured (roads etc.).
seism	non-negative numeric, seismic lines (total length in km) to be restored or deactivated.
young	numeric between 0 and 100, percent of young (<30 yrs) forest.
cost	non-negative numeric, cost of management action (in canadian dollars) per km linear feature restored/deactivated.
yr_deact	non-negative numeric, years it takes for deactivation to reach a level where it begins to affect lambda.
yr_restor	non-negative numeric, years it takes for restoration to reach a level where it begins to affect lambda.

Details

Lambda is calculated as 1.0184 - 0.0234 * (lin / area) - 0.0021 * young.

Value

A list with population table and costs.

See Also

wildlift_forecast

Examples

```
## nothing to restore/deactivate
wildlift_linear()

## all linear restored/deactivated
wildlift_linear(lin=1000, seism=1000, young=10)

## half of linear is to be restored/deactivated
wildlift_linear(lin=1000, seism=500, young=10)
```

run_app *Run Shiny apps*

Description

Run the Shiny apps that are included in the package.

Usage

```
run_app(app = "WildLift")
```

Arguments

app	character, which app to run.

Details

"WildLift": the default app.

WildLift *Population Forecasting*

Description

Functions to set demographic and cost parameter, to perform forecasting, and to inspect the results.

Usage

```
wildlift_settings(pen.type = c("mat.pen",  "pred.excl",
    "moose.red", "wolf.red", "cons.breed"), herd = NULL, ...)

wildlift_forecast(settings, tmax = 20, pop.start = 100,
    fpen.prop, fpen.inds)

wildlift_breakeven(forecast, lambda=1,
    type=c("prop", "inds"), max=10^4, tol=0.01)

## S3 method for class 'wildlift_settings'
print(x, ...)
## S3 method for class 'wildlift_forecast'
print(x, ...)
## S3 method for class 'wildlift_forecast'
plot(x, plot = TRUE, ...)
## S3 method for class 'wildlift_forecast'
lines(x, pen = TRUE, plot = TRUE, ...)
## S3 method for class 'wildlift_forecast'
```

```
summary(object, ...)
## S3 method for class 'summary.wildlift_forecast'
print(x, ...)
```

Arguments

pen.type	character: maternity penning, predator exclusion, moose reduction, wolf reduction, or conservation breeding. It can also be a settings object.
herd	NULL (default, East Side Athabasca) or character. The following herds are available: "EastSideAthabasca" (Boreal ecotype) "ColumbiaNorth", "ColumbiaSouth", "FrisbyQueest", "WellsGreySouth", "Groundhog", "Parsnip", "KennedySiding", "KlinsezaMoberly", "Quintette" (Southern Mountain ecotype). Note: only "KennedySiding", "KlinsezaMoberly", or "Quintette" can be selected when pen.type = "wolf.red", beside a generic herd setting when herd is NULL that is specific to the 'wolf removal' treatment type (wild, i.e. 'no wolf removal' settings for these herds are available under the default pen.type = "mat.pen" treatment but no separate generic herd settings are provided for the 'no wolf removal' treatment).
settings	a settings object returned by wildlift_settings.
tmax	positive integer, number of years to forecast after initial year 0.
pop.start	positive integer, initial population size in year 0.
fpen.prop, fpen.inds	fpen.prop is a single numeric value between 0 and 1 for the proportion of females penned; fpen.inds is a non-negative integer for the number of females penned, can be a vector giving number of females penned in each year to be accumulated. Only one of fpen.prop or fpen.inds can be provided at a time. Any of the two arguments can be NULL or missing, but not both at the same time.
forecast	a forecast object returned by wildlift_forecast.
lambda	numeric, annual intrinsic growth rate, must be >0.
type	what to provide as output: proportion ("prop") or numbers ("inds") of females penned at breakeven point.
max	numeric (>0), maximum value for breakeven optimization when type = "inds".
tol	numeric, tolerance limit, i.e. the maximum acceptable deviation from lambda when solving for 'breakeven' proportion of females penned.
x, object	an object to print, summarize, plot, etc.
plot	logical, whether a plot is to be produced.
pen	logical, whether to display the pen or no-pen (baseline) results.
...	additional arguments to functions. For wildlift_settings see Details.

Details

The following cost parameters can be set by wildlift_settings:

pen.cap: how many adult females can live in a single maternity pen.

pen.cost.setup: initial cost in thousands to set up pen.

pen.cost.proj: annual costs in thousands for project manager.

pen.cost.maint: annual cost in thousands for patrolling and repairing fence.

pen.cost.capt: annual cost in thousands to capture cows, monitor, survey, calf collar.

pen.cost.pred: annual cost in thousands for removing predators.

The following demographic parameters can be set by wildlift_settings:

c.surv.wild: calf survival rate in the wild, annual.

c.surv.capt: calf survival rate when captive, annual.

f.surv.wild: adult female survival when wild, annual.

f.surv.capt: adult female survival when captive, annual.

f.preg.wild: pregnancy rate when wild.

f.preg.capt: pregnancy rate when captive.

Value

wildlift_settings returns a settings object.

wildlift_forecast returns a forecast object.

wildlift_breakeven returns a numeric value representing the 'breakeven' proportion of females penned where lambda is within tolerance. It returns NA when proportion satisfying the lambda criterion cannot be found, alongside a warning.

The print method returns the input object x invisibly.

The summary method returns population and cost summaries for the forecast object (cost is in million dollars).

The plot and lines methods return the plotted data invisibly (years, pen and no-pen population size for plot; years, pen or no-pen population size for lines). Both methods produce plots as a side effect.

See Also

wildlift_matrix, wildlift_breeding

Examples

```
## Predefined settings
(s1 <- wildlift_settings("mat.pen"))
(s2 <- wildlift_settings("pred.excl"))

## Modifying predefined settings
wildlift_settings("mat.pen", c.surv.capt=0.65, pen.cap=30)
wildlift_settings(s1, c.surv.capt=0.65, pen.cap=30)

## Forecast based on settings for 75% females penned
(f1 <- wildlift_forecast(s1, fpen.prop = 0.75))
(f2 <- wildlift_forecast(s2, fpen.prop = 0.75))
```

```
## Get population and cost summaries
summary(f1)
summary(f2)

## Plot the results
plot(f2)
lines(f1, col = 2)
legend("topleft", col = c(1,1,2), lty = c(2,1,1),
    legend = c("No pen", "Mat pen", "Pred excl"))

## Find 'breakeven' proportion of females penned where lambda=1
(b1 <- wildlift_breakeven(f1, lambda = 1))
(b2 <- wildlift_breakeven(f2, lambda = 1))
f3 <- wildlift_forecast(s1, fpen.prop = b1)
f4 <- wildlift_forecast(s2, fpen.prop = b2)
## See that lines are truly flat
op <- par(mfrow = c(1, 2))
plot(f3, main = "Mat pen")
plot(f4, main = "Pred excl")
par(op)
```

Index

∗ **methods**
    breeding, 2
    WildLift, 6
∗ **misc**
    run_app, 6
∗ **ts**
    breeding, 2
    linear, 4
    WildLift, 6
∗ **utilities**
    breeding, 2
    WildLift, 6

breeding, 2

linear, 4
lines.wildlift_forecast *(*WildLift*)*, 6

plot.wildlift_breeding *(*breeding*)*, 2
plot.wildlift_forecast *(*WildLift*)*, 6
print.summary.wildlift_forecast
        *(*WildLift*)*, 6
print.wildlift_breeding *(*breeding*)*, 2
print.wildlift_forecast *(*WildLift*)*, 6
print.wildlift_settings *(*WildLift*)*, 6

run_app, 6

summary.wildlift_breeding *(*breeding*)*, 2
summary.wildlift_forecast *(*WildLift*)*, 6

WildLift, 6
WildLift-package *(*WildLift*)*, 6
wildlift_breakeven *(*WildLift*)*, 6
wildlift_breeding, *8*
wildlift_breeding *(*breeding*)*, 2
wildlift_forecast, *4*, *5*
wildlift_forecast *(*WildLift*)*, 6
wildlift_linear *(*linear*)*, 4
wildlift_matrix, *8*
wildlift_matrix *(*breeding*)*, 2
wildlift_settings, *2*, *4*
wildlift_settings *(*WildLift*)*, 6