Package: VRAP (via r-universe)

November 26, 2024

Type Package

Title VRAP for RER and Viability Computations

Version 1.3.11

Date 2018-01-11

Depends R (>= 2.15.0), stringr, shiny, shinyAce, doParallel, knitr, doSNOW, utils

Author Eli Holmes, Norma Sands, Howard Coleman NOAA, Seattle, USA

Maintainer Elizabeth Holmes - NOAA Federal <eli.holmes@noaa.gov>

Description This is an optionally parallel R version of the VRAP program.

License GPL-2

LazyData yes

BuildVignettes yes

ByteCompile TRUE

NeedsCompilation no

RoxygenNote 6.0.1

Config/pak/sysreqs make libicu-dev zlib1g-dev

Repository https://nmfs-opensci.r-universe.dev

RemoteUrl https://github.com/nwfsc-math-bio/VRAP

RemoteRef HEAD

RemoteSha 380e549866ba105218dee2d3e76894ae6ed75d8d

Contents

`AEQcalc` *SUB AEQcalc*

Description

Compute the AEQs for each age group. Original VB code:

Usage

```
AEQcalc(input)
```

Details

TAB: what are AEQs? Adult Equivalents! Sub AEQcalc() Dim TmpA As Double 'TAB: added line Dim TmpS As Double 'TAB: added line Dim Age As Integer 'TAB: added line TmpA = 0 TmpS = 0 For Age = MaxAge AEQ(Age) = MatRate(Age) + TmpS * (1 - MatRate(Age)) * TmpA TmpA = AEQ(Age) TmpS = 1 - NatMort(Age) Next Age

End Sub

Value

AEQ (scalar)

`Autocorrel` *Autocorrel*

Description

Compute autocorrelated variable, p = autocorrelation, lastx = last value of variable x

Usage

```
Autocorrel(p, lastx, x)
```

Arguments

`p`	autocorrelation
`lastx`	last value of variable x
`x`	A random (non-correlated) variable generated from the gamma function for the variable.

Value

New random autocorrelated variable (scalar)

`BufferInit` *BufferInit*

Description

ADJUST TARGET RATE FOR BUFFER EXPLOITATION

Usage

```
BufferInit(Buffer, inputs)
```

Arguments

`Buffer`	Scaling factor for Target ER
`inputs`	Inputs read from the .rav file

Details

If the StepFunc is ER, then the simulation steps through different exploitation rates. It does this by using the Target ER (in inputs$TargetU and multiplying that by a 'buffer' or scaling factor)

Value

list with new target rate for the simulation if StepFunc=ER or new capacity (b parameter in SR function) if StepFunc=Pop.

CompAgeCohort *CompAgeCohort*

Description

Compute age cohort

Usage

```
CompAgeCohort(TempCohort, Cohort, inputs)
```

Details

Original VB code: Sub CompAgeCohort(TempCohort() As Double) Dim Age As Integer 'counter for ages

For Age = MaxAge Cohort(Age) = TempCohort(Age - 1) Next Age

Cohort(MinAge

End Sub

Value

Cohort (scalar) EEH: this only works if MinAge=2, because Cohort[1] was set in CompRecruits (not Cohort[minage-1]) EEH: why isn't TempCohort[inputs$MinAge - 1] = Cohort[inputs$MinAge - 1]?

CompBetaVariate *CompBetaVariate*

Description

This subroutine generates a beta random variable.

Usage

```
CompBetaVariate(Alpha, Beta)
```

Arguments

Alpha	alpha parameter of gamma
Beta	beta parameter of gamma

Details

EEH for testing purposes the original VB code was duplicated. This could have been replaced with rbeta(alpha,beta).

Original VB codes creates beta r.v. by using the gamma distribution Mean = alpha/(alpha+beta) Variance = (a*b)/((a+b)^2*(a+b+1)) TAB: not sure this is correct way of getting beta distr since Beta(a,b) = (Gamma(a)*Gamma(b))/(Gamma(a+b)) ********** Function CompBetaVariate(Alpha As Double, Beta As Double) Dim G1 As Double Dim G2 As Double

If CONSTRUN = True Then CompBetaVariate = Alpha / (Alpha + Beta) 'expected value Else If Alpha <= 1 Then 'TAB: changed ALP to Alpha in this line G1 = Gam1(Alpha, 1) Else G1 = Gam2(Alpha, 1) End If

If Beta <= 1 Then 'TAB: changed BET to Beta in this line G2 = Gam1(Beta, 1) Else G2 = Gam2(Beta, 1) End If

CompBetaVariate = G1 / (G1 + G2) End If End Function

Value

Beta distributed random variable (scalar)

`CompEscpmnt` *CompEscpmnt*

Description

Compute escapement

Usage

```
CompEscpmnt(Regime, Year, inputs, BufTargetU, Cohort, AEQ, YearStats)
```

Arguments

`Regime`	Harvest regime.
`Year`	Year to compute the escapement for.
`inputs`	Inputs from .rav file
`BufTargetU`	Target ER to use for simulation. Adjusted if StepFunc=ER.
`Cohort`	Cohort
`AEQ`	Adult equivalents
`YearStats`	list of computed variables for each year: AEQMort, Escpmnt[Year,] = Escpmnt, TotAdultEscpmnt, TotAEQMort, TotEscpmnt,TempCohort.

Value

Updated YearStats list for value of variables in Year

CompNatMort *CompNatMort*

Description

Let the number of fish in each age class decrease according to the natural mortality in that age class.

Usage

```
CompNatMort(inputs, CohortBeforeNatMort)
```

Arguments

inputs	Inputs from .rav file
CohortBeforeNatMort	Cohort before natural mortality

Details

Original VB code: Sub CompNatMort() Dim Age

For Age Cohort(Age Next Age

End Sub

Value

Cohort after natural mortality

CompRecruits *CompRecruits*

Description

Compute recruits

Usage

```
CompRecruits(YearStats, Year, inputs, repvars, staticvars, BufSRb)
```

Arguments

YearStats	list of computed variables for each year: AEQMort, Escpmnt[Year,] = Escpmnt, TotAdultEscpmnt, TotAEQMort, TotEscpmnt,TempCohort.
Year	Year to compute the escapement for.
inputs	Inputs from .rav file
repvars	repvars
staticvars	Static variables
BufSRb	Capacity. Changes if StepFunc=Pop. Otherwise stays the same.

Value

updated repvars list with: Cohort[1]=CohortAge1, LastRanFlow, LastRanError, LastRanMarine.

`CompStatsEEH` *CompStats*

Description

Returns the statistics (calculated values) needed to produce the summary output file

Usage

```
CompStatsEEH(BufNum, inputs, BufSRb, YearStats, SummaryStats)
```

Arguments

`BufNum`	Which simulation (TargetER or Pop) is this for.
`inputs`	Inputs from .rav file
`BufSRb`	Capacity. Changes if StepFunc=Pop. Otherwise stays the same.
`YearStats`	list of computed variables for each year: AEQMort, Escpmnt[Year,] = Escpmnt, TotAdultEscpmnt, TotAEQMort, TotEscpmnt,TempCohort.
`SummaryStats`	list with the summary statistics to be updated

Details

This function similar but not identical to the original VB function

Value

Updated SummaryStats list

`CompStockStatus` *CompStockStatus*

Description

Compute stock status: is TotAdultEscpmnt > EscpmntBreakPoint[Break]

Usage

```
CompStockStatus(TotAdultEscpmnt, inputs)
```

Arguments

TotAdultEscpmnt	Total adult escapement
inputs	Inputs from .rav file

Details

Original VB code Sub CompStockStatus(TotAdultEscpmnt As Double, Status Dim Break Dim EscpmntCheck$

Break EscpmntCheck$ = "True" While Break If TotAdultEscpmnt > EscpmntBreakPoint(Break EscpmntCheck$ = "True" Break Else EscpmntCheck$ = "False" End If Wend

Status

End Sub

Value

Status ("True"/"False")

ComputeRER *ComputeRER*

Description

Computes the LEL and UEL RERs=

Usage

```
ComputeRER(VRAPList, UEL = 80, LEL = 5)
```

Arguments

VRAPList	output list from Main() or RunSims()
UEL	Upper target as percent of VRAP simulations that should be above recovery threshold
LEL	lower target as percent of VRAP simulations that should be not hit the lower threshold

Details

This fits a spline to the harvest versus UET and LET lines to get the ER at the LEL and UEL.

Value

a list with:

ERUEL The ER at the UEL

ERLEL The ER at the LEL

hr.vrap The harvest rates for each simulation in the VRAP output.

hr.smooth The harvest rates used to computed the smoothed UEL and LEL.

uel.vrap The UEL for each simulation in the VRAP output.

uel.smooth The smoothed UEL.

lel.vrap The LEL for each simulation in the VRAP output.

lel.smooth The smoothed LEL.

UEL The proportion to use for the upper threshold.

LEL The proportion to use for the lower threshold.

`Cycle` *Cycle*

Description

Compute cyclic variable, a = amplitude, p = period, s = start, y = year, x = mean value of variable

Usage

```
Cycle(a, p, s, y)
```

Arguments

`a`	amplitude
`p`	period
`s`	starting point
`y`	time period

Details

NJS: created 7/9/02 corrected 9/16/03

Function Cycle(a As Double, p As Double, s As Double, y, x As Double) As Double a is amplitude, p is period, s is starting point, y time period what is x doing here? It is average value and is not needed here. Dim cy As Double cy = Sin(2# * 3.141592654 * (y + s - 1) / p) 'in good survival, cycle ranges from 1 to a (amplitude) in bad survival, cycle ranges from 1/a to 1 (this might be lower than expected) If cy >= 0 Then cy = (cy * (a - 1)) + 1 Else cy = (cy * (1 - (1 / a))) + 1 End If Cycle = cy + x ' use if x is changed to scalar Cycle = cy

End Function

Value

cyclic variable (scalar or vector)

GammaSample *GammmaSample*

Description

Function generates a random gamma deviate with shape paramater alpha and scale parameter beta

Usage

```
GammaSample(Alpha, Beta)
```

Arguments

Alpha	alpha parameter of gamma
Beta	beta parameter of gamma

Value

Gamma distributed random variable (scalar)

GetInput *GetInput*

Description

Read in a .rav file and assign all the variables

Usage

```
GetInput(InFile)
```

Arguments

InFile	the name of the .rav file

Value

Returns the list of all inputs

Main *Main*

Description

Runs VRAP. This function is largely specific to the R version of VRAP.

Usage

```
Main(InFile = NULL, OutFileBase = NULL, NRuns = -1, NYears = -1,
  Title = -1, TargetStart = -1, TargetEnd = -1, TargetStep = -1,
  ERecovery = -1, QET = -1, ECrit = -1, NewRavFileName = "tmprav.rav",
  forceNewRav = NULL, silent = FALSE, lcores = 1,
  parallel.backend = "doParallel", save.output.as.files = TRUE)
```

Arguments

InFile	The name of the .rav file
OutFileBase	The basename for the .sum, .byr, and .esc output files
NRuns	Number of runs to use in the simulations if the user wants to use something different than what is in the .rav file
NYears	Number of years to project forward in the simulations if the user wants to use something different than what is in the .rav file
Title	Title to use for the report if the user wants to use something different than what is in the .rav file
TargetStart	Target ER to start simulations at if the user wants to use something different than what is in the .rav file
TargetEnd	Target ER to end simulations at if the user wants to use something different than what is in the .rav file
TargetStep	Target ER step sizes if the user wants to use something different than what is in the .rav file
ERecovery	Recovery target if the user wants to use something different than what is in the .rav file
QET	if the user wants to use something different than what is in the .rav file
ECrit	if the user wants to use something different than what is in the .rav file
NewRavFileName	A new .rav file is saved in case the user has changed any values from what is in the .rav file.
forceNewRav	Force use of new rav file. Needed for shiny app.
silent	Whether to show progress bar.
lcores	Number of cores to use. Default is non-parallel so lcores=1
parallel.backend	doParallel or doSNOW. The latter allows the progress bar to appear.
save.output.as.files	If TRUE (default), then .sum, .byr, .esc and .rav files are saved using OutFileBase. If FALSE, no files are saved and only the list is output.

Value

list with output list from RunSims() and output time

`progressBar` *progressBar*

Description

if silent=FALSE, then a progress bar is shown

Usage

```
progressBar(prop = 0, prev = 0)
```

`Recruits` *Recruits*

Usage

```
Recruits(inputs)
```

Arguments

`inputs`	Inputs from .rav file

Details

Function Recruits() As Double Compute factor to convert calculated spawner equivalent production to age cohort (source is PSC Chinook Model).

Tmp = 0 X9 = 1 - NatMort(1) For Age X9 = X9 * (1 - NatMort(Age Tmp = Tmp + X9 * MatRate(Age X9 = X9 * (1 - MatRate(Age Next Age Recruits = Tmp

End Function The SR function gets us the AEQRecruits from spawners in year t. We needs to then translate that to age 1 indiv in pop (Cohort[1]) We know AEQRecruit. How many Age 1 individuals does that translate to? Age1 * (1- total fraction lost) = AEQRecruits So Age1 = AEQRecruits/(1-total fraction lost) Tmp here is total fraction of age 1 ind that eventually return AEQRecruits/Tmp = Age 1 or Cohort[1]

Value

Recruits at age 1

RepInit *RepInit*

Usage

```
RepInit(inputs)
```

Arguments

inputs	Inputs from .rav file

RunSims *RunSims*

Description

RunSims takes the input list and runs the VRAP functions

Usage

```
RunSims(inputs, silent, parallel.backend = "doParallel")
```

Arguments

inputs	Inputs from .rav file
silent	Whether to show progress bar

Value

list with inputs, SummaryStats, staticvars, comp.time.

SaveBYrData *SaveBYrData*

Description

Write the .byr output file

Usage

```
SaveBYrData(inputs, SummaryStats)
```

Arguments

inputs	Inputs from .rav file
SummaryStats	list with the summary statistics to be updated

Value

Nothing. Writes file.

SaveEscpmntData *SaveEscpmntData*

Description

Write the .esc output file

Usage

```
SaveEscpmntData(inputs, SummaryStats)
```

Arguments

inputs	Inputs from .rav file
SummaryStats	list with the summary statistics to be updated

Value

Nothing. Writes file.

SaveSummary *SaveSummary*

Description

Write the .esc output file

Usage

```
SaveSummary(inputs, SummaryStats, staticvars)
```

Arguments

inputs	Inputs from .rav file
SummaryStats	list with the summary statistics to be updated
staticvars	Static variables

Value

Nothing. Writes file.

SaveYearData *SaveYearData*

Description

Update CalendarHR[Year] in YearStats

Usage

```
SaveYearData(Year, YearStats)
```

Arguments

Year	year
YearStats	list of computed variables for each year: AEQMort, Escpmnt[Year,] = Escpmnt, TotAdultEscpmnt, TotAEQMort, TotEscpmnt,TempCohort.

Value

Updated YearStats list

SetOutFileNames *SetOutFileNames*

Description

Set output file names

Usage

```
SetOutFileNames(BaseName, inputs, PathName = NULL)
```

Arguments

BaseName	The basename for the .sum, .byr, and .esc output files
inputs	Inputs from .rav file
PathName	Path for the files

Details

Used by GetOutFiles and also by GetCommandLine

Value

Updated inputs list with full names for the output files

SetupSummaryStats *SetupSummaryStats*

Description

Create an list set up for all the summary stats. Each list itme is length of BufMax and is all 0.

Usage

```
SetupSummaryStats(inputs)
```

Arguments

inputs	Inputs from .rav file

Value

SummaryStats list

Trend *Trend*

Description

Compute variable x with trend t for year y

Usage

```
Trend(t, y, x, z)
```

Arguments

t	trend rate
y	time increment
x	first value
z	type of trend 0 or 1

Details

NJS: created 7/9/02. Original VB code: Function Trend(t As Double, y, x As Double, z) As Double t is trend rate, y is time increment, x is first value, z is type of trend

If z = 0 Then Trend = x * (1 + t) ^ y ElseIf z = 1 Then Trend = x + (y * t) Else If Trend < 0 Then Trend = 0

Print "Unknown trend/cycle function" Stop End If

End Function

Value

trend variable (scalar or vector) EEH: changed to work with vectors of y

---

VRAP *VRAP*

Description

Function to call the shiny app

Usage

```
VRAP()
```

WriteRavFile *WriteRavFile*

Description

This function takes inputs list and creates a RAV file

Usage

```
WriteRavFile(inputs, filename)
```

Arguments

inputs	Inputs from .rav file.
filename	Name of the .rav file to write.

Details

This function is new to the R VRAP. Needed to create record of the .rav file used for the VRAP run in case the user changed .rav values in the shiny app.

Value

Nothing. .rav file is written.

`WriteReport` *Write a report*

Description

Create a pdf with basic information about the VRAP output.

Usage

```
WriteReport(InFile = NULL, OutFileBase = NULL, show.file = FALSE)
```

Arguments

`InFile`	the .rav input file.
`OutFileBase`	If OutFileBase is NULL, the VRAP output files are assumed to be in the same directory as InFile and named InFile.sum, InFile.byr, InFile.esc. Thus they have the same basename. If this is not the case, then OutFileBase can be passed in.
`show.file`	Whether to open the pdf after it is produced.

Details

knit2pdf is used to create the pdf using Report-knitr-ER.xRnw or Report-knitr-Pop.xRnw (Sweave files) in inst/doc.

Value

Nothing. The pdf is made and saved.

Index