SMRU Consulting

understand ◆ assess ◆ mitigate

Helpfile for the interim PCoD v5 model

Authors:	Rachael Sinclair, Cormac Booth, John Harwood, Carol Sparling
Date:	Wednesday, 06 March 2019

THIS REPORT IS TO BE CITED AS: SINCLAIR, R; BOOTH, C, HARWOOD, J. & SPARLING, C. (2019). HELPFILE FOR THE INTERIM PCOD V5 MODEL. MARCH 2019.

Contents

Forward from the Report: A Protocol for Implementing the Interim Population Consequences of Disturbance (PCoD) Approach

Our understanding of the effects of underwater noise and other impacting activities on marine mammals, and of how to assess and quantify potential impacts, is rapidly evolving. The Steering Group for the Interim PCoD (iPCoD) work emphasise that this framework is very much an interim measure, that is, the iPCoD approach has been developed to help developers, regulators and advisers working on offshore renewable energy projects now. It is expected that the framework will be further refined and built upon over time as more evidence becomes available. The iPCoD approach should be regarded as another tool among others already available for assessing potential impacts of PTS and disturbance on marine mammal populations. How appropriate it will be for use with particular projects and situations should be decided on a case by case basis. **It is important that developers considering using the iPCoD approach seek advice from the SNCBs and/or regulators at an early stage.**

The full report can be read here.

1 Introduction to this helpfile

The basic PCoD approach was developed by an international group of experts in the US Office of Naval Research Working Group on the Population Consequences of Acoustic Disturbance US National Research Council's Committee on Characterizing Biologically Significant Marine Mammal Behaviour in its 2005 report. Their work provided the foundation for the Interim PCoD framework, which was developed at a workshop on Assessing the Risks to Marine Mammal Populations from Renewable Energy Devices, which was jointly funded by Countryside Council for Wales (now Natural Resources Wales), the Joint Nature Conservation Committee and the Natural Environment Research Centre. The interim PCoD approach (iPCoD) was developed to deal with the current situation, where there are limited data on the way in which changes in behaviour and hearing sensitivity may affect the ability of individual marine mammals to survive and to reproduce.

The interim PCoD model (iPCoD) is therefore a protocol for implementing an interim version of the Population Consequences of Disturbance (PCoD) approach for assessing and quantifying the potential consequences for marine mammal populations from disturbance and/or injury that may result from offshore energy developments. It is set up to run models for five marine mammal species in UK waters (bottlenose dolphins, harbour porpoise, minke whale, harbour and grey seals). The iPCoD model has been designed to use the kinds of information that are likely to be provided by developers in their Environmental Statements and Habitats Regulations Assessments. For further information on the iPCoD approach please see Harwood et al., (2014)[1]. The code is written in R[2] which is a language and environment for statistical computing and graphics. R is a free, open source piece of software that runs on a variety of platforms including Linux, Windows and MacOS, which means that all potential users can access the software for free.

The purpose of this helpfile is to guide the user through how to set up the software required to run the iPCoD code (R and R studio), how to change parameters within the iPCoD code and how to interpret the outputs from the simulations.

We have included a series of internal hyperlinks to help you step back and forth through the helpfile as required. These are indicated by a blue box with bold text in it. We recommend that you open the bookmarks tab to see the section headings. You can click on the 'CLICK HERE' text to move within the document.

This is an example hyperlink box. Clicking the link will take you back to the start of the Introduction

CLICK HERE

There are number of key pieces of information in this guide. We have marked each of these in ***italic text and highlighted in grey***. They should not be ignored as each one may affect your ability to move through this helpfile successfully.

2 The iPCoD model

2.1 Model summary

iPCoD uses a stage structured model of population dynamics with nine age classes and one stage class (adults ten years and older). In the absence of empirical data on the extent to which disturbance and/or injury impacts affects individual survival and fecundity, the iPCoD framework uses the results of an expert elicitation process conducted according to the protocol described in

---

[1] Harwood, J., King, S., Schick, R., Donovan, C. & Booth, C. (2014) A protocol for implementing the interim population consequences of disturbance (PCoD) approach: quantifying and assessing the effects of UK offshore renewable energy developments on marine mammal populations. Report number SMRUL-TCE-2013-014. Scottish Marine and Freshwater Science, 5(2).

[2] R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Donovan *et al.,* (2016), to predict the effects of disturbance and a permanent threshold shift in hearing threshold (PTS) on individual survival and reproductive rates. The process generates a set of statistical distributions for these effects and then simulations for the impacted population are conducted using values randomly selected from these distributions that represent the opinions of a "virtual" expert. This process is repeated 1,000 times to capture the uncertainty among experts.

There are two principal stages involved in the iPCoD framework simulations – the first stage is a day-by-day simulation of up to 1,000 individual animals (the precise number is determined by the size of the population) across the period of predicted impact to calculate both the number of animals experiencing disturbance and/or PTS and also the amount of impact experienced by each of the individuals, by the end of each year. This is done using a combination of an estimate of: 1) the number of animals predicted to be impacted as a consequence of exposure to a single day of pile driving, 2) a schedule of the timing of the planned pile-driving and 3) the number of animals (i.e. as section of the population) thought to be vulnerable to the impacts. The first two of these are supplied by the developers from the impact assessment process, the last is a judgement based on the understanding of movements of individuals within the population in relation to the extent of the impact and the duration of construction activity in the simulation. With a smaller vulnerable sub-population, fewer animals would experience impacts but each animal experiencing impacts would be exposed to a relatively larger amount of impact, compared to if all animals were equally vulnerable. Other than the ability to compare different vulnerable subsections, the model is not spatially explicit.

The second stage scales these numbers up to the total population size to create a Leslie Matrix model that is used to calculate the future population growth of the impacted population using modified survival and birth rates for those animals that have experienced disturbance and PTS. In parallel, the baseline survival and birth rate values available for the population allow a Leslie Matrix model to project the future trajectory of the un-impacted population. This is repeated many times (1,000 times is the default, and is the minimum recommended by King et al. 2015, but this can be changed by the user) and each simulation draws parameter values from statistical distributions describing the uncertainty in the parameters. The distributions of the two trajectories can be compared to demonstrate the size of the long term effect of the predicted impact on the population as well as demonstrating the uncertainty in predictions.

Within the 1,000 simulations run for the iPCoD code, each simulation contains a matched pair of populations (an un-impacted population (baseline) and an impacted population).

Within 1 matched pair, the only difference is that impacts are allocated to the impacted population.

Between each of the 1,000 simulations of the model the differences between each matched pair are determined by:

- Environmental stochasticity – variation in the survival and fertility rates as a result of changes in environmental conditions (based on the results of the expert elicitation; see Harwood et al. 2014),
- Demographic stochasticity - variation among individuals in their realised vital rates as a result of random processes, and
- The effect of disturbance and PTS on vital rates – the model draws at random from statistical distributions derived from the results of the expert elicitation to realise the effect of the impact on vital rates of affected individuals (based on the expert elicitation from Harwood et al. 2014 and Booth et al. 2018 (disturbance) and Booth and Heinis, 2018 (PTS)).

The iPCoD code produces a .rdata file once it has run, which contains a matrix called dat.out where the results of each run is stored. The dat.out matrix is explained in detail in section 11.1.

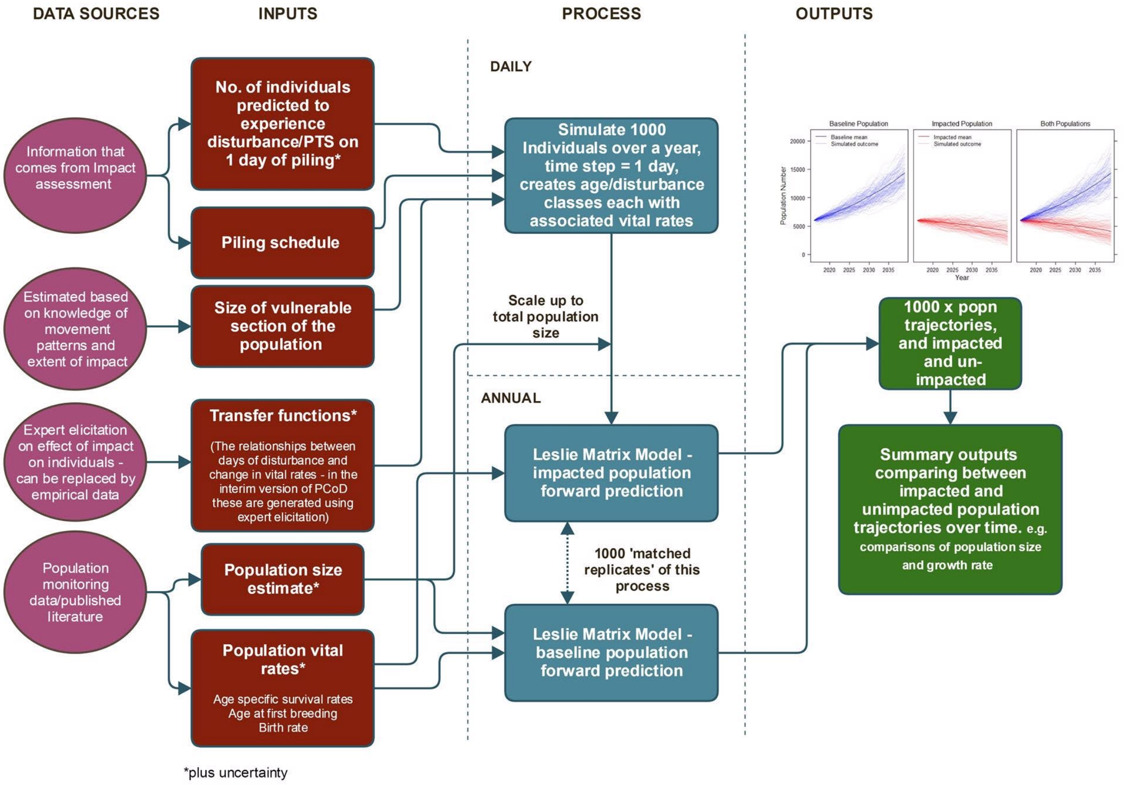

DATA SOURCES
INPUTS
PROCESS
OUTPUTS
DAILY
Information that comes from Impact assessment
No. of individuals predicted to experience disturbance/PTS on 1 day of piling*
Piling schedule
Simulate 1000 Individuals over a year, time step = 1 day, creates age/disturbance classes each with associated vital rates
Baseline Population
Impacted Population
Both Populations
Baseline mean
Simulated outcome
Impacted mean
Simulated outcome
Population Number
20000
15000
10000
5000
0
2020
2025
2030
2035
Year
Estimated based on knowledge of movement patterns and extent of impact
Size of vulnerable section of the population
Scale up to total population size
1000 x popn trajectories, and impacted and un-impacted
ANNUAL
Expert elicitation on effect of impact on individuals - can be replaced by empirical data
Transfer functions*
(The relationships between days of disturbance and change in vital rates - in the interim version of PCoD these are generated using expert elicitation)
Leslie Matrix Model - impacted population forward prediction
Summary outputs comparing between impacted and unimpacted population trajectories over time. e.g. comparisons of population size and growth rate
1000 'matched replicates' of this process
Population monitoring data/published literature
Population size estimate*
Population vital rates*
Age specific survival rates
Age at first breeding
Birth rate
Leslie Matrix Model - baseline population forward prediction
*plus uncertainty

Figure 1 Schematic representation of the Interim PCoD Framework.

2.2 Expert Elicitation

Because of the lack of data on the effects of disturbance and hearing damage on vital rates, a formal expert elicitation process was carried out in 2013-14 for all five UK priority species (harbour porpoise, grey seal, harbour seal, bottlenose dolphin and minke whale) in relation to impacts caused by pile driving. The expert elicitation process produced estimates of survival and birth rates for animals predicted to experience disturbance and hearing damage, each with their associated uncertainty.

Given advances in knowledge and updated expert elicitation methods, new expert elicitations were conducted in 2018 on the potential effects of disturbance and PTS as a consequence of exposure to low frequency broadband pulsed noise (e.g. pile driving, seismic airgun).

The results of the PTS expert elicitation can be found in the supporting documents folder and were incorporated into v4 of the code (and subsequently v5). The key points from the expert elicitation were:

1. PTS does not mean an animal is deaf. It means the animal experiences a reduction in hearing sensitivity in a specific frequency range.
2. The effect of PTS on vital rates will be driven by the magnitude and the frequency band of the PTS.
3. Reduced hearing ability does not necessarily mean a less fit animal.
4. A 6 dB PTS in the 2-10 kHz band was considered unlikely to have a large effect on survival or fertility and this is reflected in the updated (2018) model which uses the new elicited outputs.
5. The new expert elicitation means that the predicted effect size is much smaller than previously thought. This will result in smaller PTS effects on vital rates in the iPCoD model.

The only species for which a new expert elicitation was not conducted for was minke whales, so the effects of PTS on minke whale vital rates have remained unchanged since v3.

The results of the disturbance expert elicitation have now been processed and are incorporated in v5 of the code. The key points from the expert elicitation were:

1. For all species the effects of disturbance on vital rates were considered to be lower than estimated in the 2013 elicitation (this is reflected in the new code)
2. In general when comparing species, harbour seals vital rates were thought to be more sensitive to disturbance than grey seals.
3. For both seal species it was expected that disturbance would be most likely to impact on fertility and the survival of the ‘weaned of the year’ animals.
4. Experts agreed that harbour porpoise would be more susceptible to the effects of disturbance than either of the seal species.
5. For harbour porpoise it was expected that disturbance would be most likely to impact on post-weaning calf survival and fertility.

Note: The new disturbance expert elicitation was not conducted for minke whales or bottlenose dolphins, so the effects of disturbance on minke whale and bottlenose dolphin vital rates have remained unchanged since v3.

2.3 Version History

V4.0

- Revised demographic parameters to obtain desired population growth rates
- Bugs fixes relating to:
    - Piling schedule errors – amended PopDyn file.
    - Un-impacted populations of bottlenose dolphins fairing worse than impacted populations – amended PCoDFunctionsFinal file.

- Inclusion of revised PTS impacts from the 2018 expert elicitation (for harbour porpoise, bottlenose dolphin, harbour seal and grey seal – not minke whale)
- New output files including line plots, polygons and histograms of the un-impacted and impacted population sizes over time, calculation of the ratio of impacted to un-impacted population size, calculation of the ratio of impacted to un-impacted annual growth rate and calculation of the centile for un-impacted population which matches the 50th centile for the impacted population.

V4.1

- Bugs fix relating to how disturbance was read into the code – amended MakeInitScenario files.

V5.0

- Inclusion of revised disturbance impacts from the 2018 expert elicitation for harbour porpoise, harbour and grey seals.
- Bug fixes to prevent impacted populations doing better than un-impacted populations in small populations.

3 The “InterimPCoDCode_ver5.0” Folder

In order to run the iPCoD model you will require a copy of the “InterimPCoDCode_ver5.0” folder which contains the following files:

- Species specific .robj files (e.g. BND_mature_Days2018.robj)
- Species specific MakeInitScenario .R files (e.g. MakeInitScenarioGS_ver5b.R)
- MultPilingOpsMultYears.csv
- config_ver5b.R
- pcodControl_ver5d.R
- PCoDFunctionsFinal_ver5e.R
- PopDyn_ver5e.R
- summary_statistics.R

Navigate on your computer to the C drive. We recommend you create a folder called RRun. Copy the “InterimPCoDCode_ver5.0” folder and paste it in the RRun folder you just created. The folder is zipped so will require unzipping. In order to do this, right click on the zipped folder and select "Extract All".

Within the "InterimPCoDCode_ver5" folder there are some files that can be edited by the user and some files that should definitely NOT be edited by the user.

Details of which files can be edited by the user are outlined in section 8 The PCoD files

CLICK HERE

4 Installing R

Windows users: The latest version of R can be downloaded from: https://cran.r-project.org/bin/windows/base/ Click on the Download link at the top of the webpage to download the .exe file.

Mac users: The latest version of R can be downloaded from: https://cran.r-project.org/bin/macosx/ Click on the download link under the heading Files: to download the .exe file

Open the .exe file.

1. This will open a new window called **Setup – R for Windows**. Click Next > to start the installation process.
2. GNU General Public License information will then appear. Click Next >
3. When prompted to select the destination location it should automatically select the following: C:\Program Files\R\R-3.3.2. If not, browse to this location. Click Next >
4. When prompted to select components make sure all boxes are ticked (Core Files, 32-bit Files, 64-bit Files and Message translations). Click Next >
5. When prompted to customize startup options select No (accept defaults). Click Next >
6. When prompted to select the start menu folder accept the automatic R shortcut. Click Next >
7. When prompted to select additional tasks tick the following box: associate R with .RData files. Click Next >
8. R will then install on your computer
9. Click Finish to exit setup

5 Installing RStudio

RStudio[3] is an integrated development environment for R. For the novice or less experienced R user, Rstudio, which runs alongside R, provides a more user friendly interface than R itself. You cannot run RStudio without having first R installed on your computer.

The latest version of RStudio can be downloaded from:

https://www.rstudio.com/products/rstudio/download/

Click on the Installer link for Windows Vista/7/8/10 or Mac OS X 10.6+ (64-bit) to download the .exe file

Open the .exe file.

1. This will open a new window called Setup – RStudio Setup. Click Next > to start the installation process
2. When prompted to select the destination location it should automatically select the following: C:\Program Files\RStudio. If not, browse to this location. Click Next >
3. When prompted to select the start menu folder accept the automatic RStudio shortcut. Click Install
4. RStudio will then install on your computer
5. Click Finish to exit setup.

If for any reason you have multiple versions of R installed on your computer, you can specify which R version you wish to run in RStudio. This is done by opening RStudio and selecting the menu tab **Tools** > **Global Options** and selecting the most recent R version.

---

[3] RStudio Team (2016). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL http://www.rstudio.com/.

6 Installing R Packages

Packages are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them and sample data.

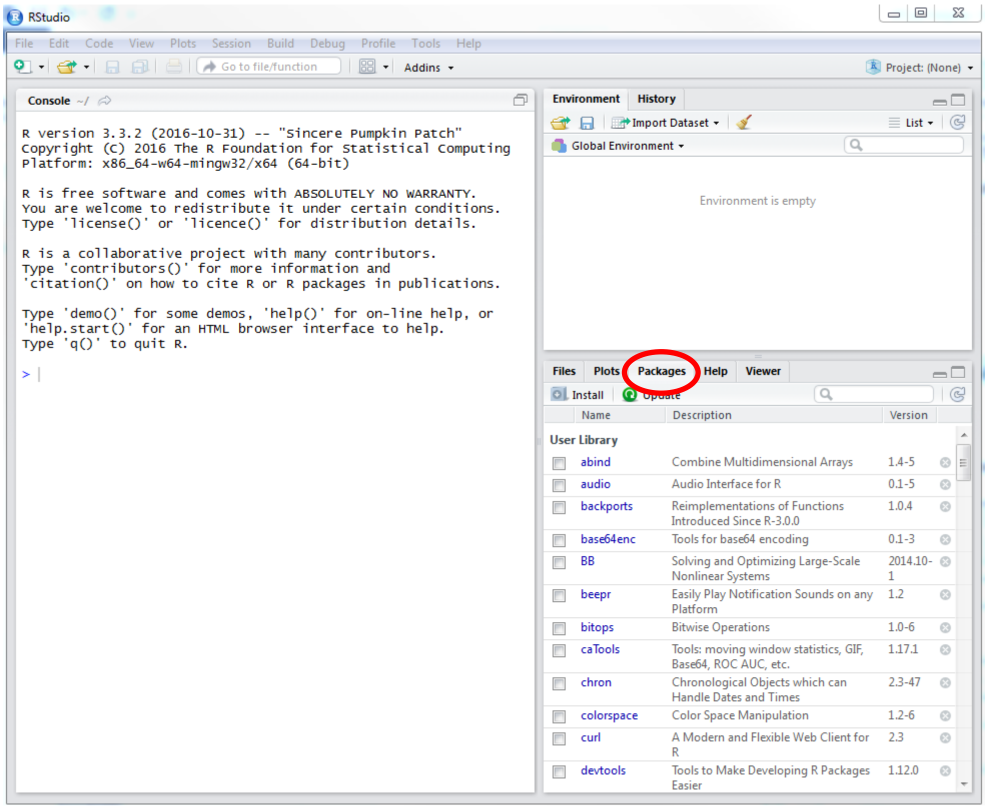
There are 4 packages that the PCoD code requires:

- `stringr`
- `ggplot2`
- `matrixStats`
- `grid`

6.1 Check to see if you have these packages installed

1) Open RStudio

2) Click on the tab “Packages” in the bottom right hand box in RStudio.



```
R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

	Name	Description	Version
User Library			
☐	abind	Combine Multidimensional Arrays	1.4-5
☐	audio	Audio Interface for R	0.1-5
☐	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.0.4
☐	base64enc	Tools for base64 encoding	0.1-3
☐	BB	Solving and Optimizing Large-Scale Nonlinear Systems	2014.10-1
☐	beepr	Easily Play Notification Sounds on any Platform	1.2
☐	bitops	Bitwise Operations	1.0-6
☐	caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
☐	chron	Chronological Objects which can Handle Dates and Times	2.3-47
☐	colorspace	Color Space Manipulation	1.2-6
☐	curl	A Modern and Flexible Web Client for R	2.3
☐	devtools	Tools to Make Developing R Packages Easier	1.12.0

3) If you have the packages installed then they will appear in the list here (note: stringr, ggplot2 and matrixStats will be listed in the first set of packages called “User Library”, grid is listed in the second set of packages called “System Library”).

4) If you have these packages installed then tick the boxes for stringr, ggplot2, matrixStats and grid. This will cause the following to appear in the RStudio Console:

```
> library("stringr", lib.loc="~/R/R-3.1.1/library")
> library("ggplot2", lib.loc="~/R/R-3.1.1/library")
> library("matrixStats", lib.loc="~/R/R-3.1.1/library")
matrixStats v0.10.0 (2014-06-07) successfully loaded. See ?matrixStats for
help.
> library("grid", lib.loc="C:/Program Files/R/R-3.3.2/library")
```

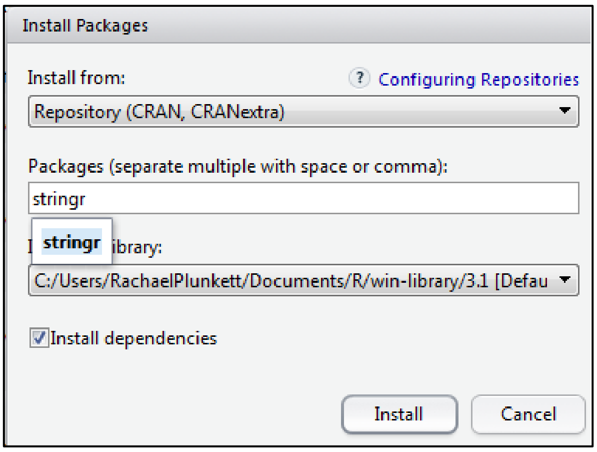
6.2 If you do NOT already have these packages installed

1) Make sure you are connected to the internet

2) Open RStudio

3) Click on the menu tab “Tools” at the top of the screen and selecting “Install Packages”.

4) Install from: Repository (CRAN, CRANextra). Type stringr in the Packages box and click Install.



Install from: Repository (CRAN, CRANextra)

Packages (separate multiple with space or comma): stringr

C:/Users/RachaelPlunkett/Documents/R/win-library/3.1 [Defau

☑ Install dependencies

5) This will cause something similar to the following to appear in the RStudio Console (depending on your computer name and the package version):

```
> install.packages("stringr")
Installing package into ‘C:/Users/COMPUTERNAME/Documents/R/win-library/3.3’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/stringr_1.2.0.zip'
Content type 'application/zip' length 148612 bytes (145 KB)
downloaded 145 KB
package ‘stringr’ successfully unpacked and MD5 sums checked
The downloaded binary packages are in
     C:\Users\COMPUTERNAME\AppData\Local\Temp\Rtmpg7tgbV\downloaded_packages
```

6) Repeat the process for the other libraries: `ggplot2, matrixStats` and `grid`

7 Organising folders

We highly recommend that you create a separate folder for each different simulation you run. This avoids the problem of R overwriting any previously saved files, and provides a record of how each simulation differs from the others you have run.

We recommend maintaining a pristine copy of the basic InterimPCoD_ver5.0 code. To create new folders, copy and paste the "InterimPCoDCode_ver5.0" folder and give it a name that relates to the simulation you are running e.g. HPsimulation1

In order to keep track of what simulations you have run and what parameters they contained, we have included a template .xls file called Simulation_Log.xls which can be populated by the user.

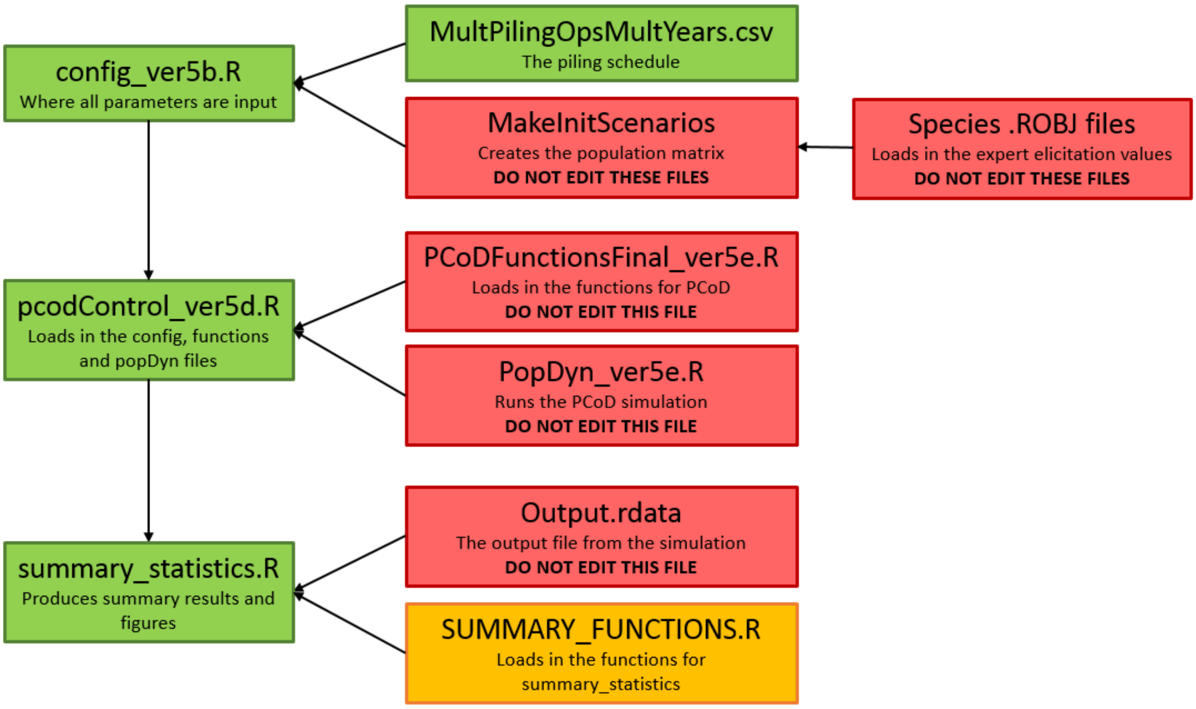
8 The iPCoD files

Within the "InterimPCoDCode_ver5.0" folder there are files that need to be edited by the user and some files that it is recommended aren't edited by the user (unless the user has sufficent knowledge of R and the code structure for iPCoD).

The MakeInitScenarios, PCoDFunctions, PopDyn and robj files should **never** be edited (even by an experienced R user) as they contain the code needed to run the simulations and include the results of the expert elicitations. Should a user wish to alter the way in which the population is modelled or model the effects of disturbance and PTS on vital rates in a different way, we recommend that you contact SMRU Consulting to discuss the development of a bespoke version of the code.

The Summary_functions file can be edited to create any combination of outputs.

The following diagram shows which files the user should edit (green boxes) and the files it is recommended that the user does not edit (red boxes), and the files that it is possible to edit if different outputs from the standard is requied (orange box). The arrows show the order of operations and which files are read in at each stage.


config_ver5b.R
Where all parameters are input
MultPilingOpsMultYears.csv
The piling schedule
MakeInitScenarios
Creates the population matrix
DO NOT EDIT THESE FILES
Species .ROBJ files
Loads in the expert elicitation values
DO NOT EDIT THESE FILES
PCoDFunctionsFinal_ver5e.R
Loads in the functions for PCoD
DO NOT EDIT THIS FILE
pcodControl_ver5d.R
Loads in the config, functions and popDyn files
PopDyn_ver5e.R
Runs the PCoD simulation
DO NOT EDIT THIS FILE
Output.rdata
The output file from the simulation
DO NOT EDIT THIS FILE
summary_statistics.R
Produces summary results and figures
SUMMARY_FUNCTIONS.R
Loads in the functions for summary_statistics

9 Piling Schedule

In order to model the impacts of disturbance and PTS as a result of pile driving activities, the iPCoD code requires a .csv file that indicates the days of the year on which piling is likely to occur. This is the piling schedule.

Note: if you are modelling the impact of **collisions only** (no disturbance or PTS from piling) then you do not need to edit the piling schedule file. Go to **section 10 config_ver5b.R**

CLICK HERE

We appreciate that developers will not be able to specify in advance the exact days on which construction work is likely to occur because this will depend on many factors, including weather and the availability of suitable equipment. However, they should be able to specify whether their preference is to carry out all construction work within the shortest period possible, which would result in many days of consecutive construction work, or for the work to be conducted sporadically and thus be spread over an extended period. They should also be able to specify single or multiple vessel scenarios.

IMPORTANT: The start date for the timetable should be 1 January, and the timetable must be made up of complete years, each of 365 days (because iPCoD does not account for leap years). However, the population model for each marine species operates from the beginning of the breeding season for that species. As a result, the forecast population size for the first year of piling will be the size of the population on 1 June of that year for all species except the grey seal. In the case of grey seals it will be the population size on 1 October of that year.

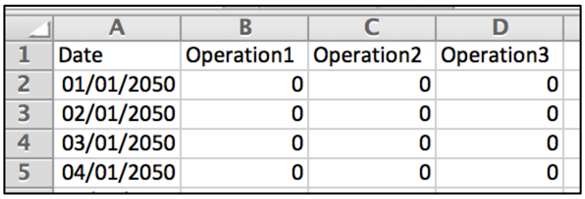
Use this information to populate the Excel template MultPilingOpsMultYears.csv in your simulation folder e.g. C:/RRun/simulation1. Open the csv file: MultPilingOpsMultYears

In this .csv document, there are at least two default columns:

Date – which is a calendar spanning the period of piling.

Operation1 – a column of 0s and 1s that specify the days upon which piling occurs (1) or not (0).

Operation2,3,4 etc. – multiple columns can be added and labelled as Operation2, Operation3 etc – if there is more than one development. Note: There may be two or more Operations for a single development site (e.g. if there are multiple piling vessels working on one site).



	A	B	C	D
1	Date	Operation1	Operation2	Operation3
2	01/01/2050	0	0	0
3	02/01/2050	0	0	0
4	03/01/2050	0	0	0
5	04/01/2050	0	0	0

IMPORTANT: The number of rows in your piling schedule <u>must</u> be at least a full year and must be divisible by 365 (i.e. last value in Date must be 31/12/yyyy). If you use the drag facility in Excel to create extra years, remember that Excel will automatically create an extra day for 29th February in all leap years. You will need to remove these days, otherwise the iPCoD code will give you an error when you run pcodControl_ver5b.r:

```
    Error in eval(expr, envir, enclos) :
  Number of days in piling file is not an exact multiple of 365
```

Note: every cell that contains a value of 1 means that piling will occur on that day. The user should ensure that the piling schedule is as realistic as possible and should (where possible) be informed by the developers/engineers working on the specific project in order to incorporate gaps in the piling schedule due to weather downtime, breakdown and other unavailability.

As a guide, the number of piling days should be related to the number of piles to be installed and dictated by how many piles can be installed in 1 day. For example, if the project envelope states that it takes up to 1 day to install a monopile foundation, and there will be 200 monopiles, then the piling schedule should contain 200 piling days.

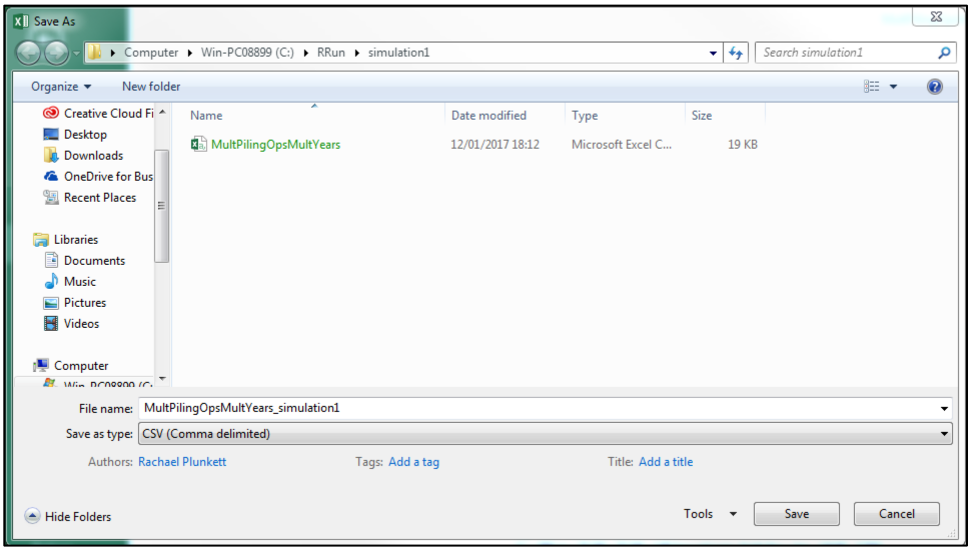
An example piling schedule may look something like this:

Date	**Operation1**
01/01/2022	1
02/01/2022	1
03/01/2022	0
04/01/2022	0
05/01/2022	0
06/01/2022	0
07/01/2022	0
08/01/2022	0

09/01/2022	1
10/01/2022	1
11/01/2022	0
12/01/2022	0
13/01/2022	0
14/01/2022	0
15/01/2022	0
16/01/2022	0
17/01/2022	0
18/01/2022	1
19/01/2022	1
20/01/2022	0
21/01/2022	1
22/04/2022	0

We recommend you save the file under a different name in the simulation folder e.g. MultPilingOpsMultYears_simulation1.csv

Ensure the file type is .csv not.xls.



Name	Date modified	Type	Size
MultPilingOpsMultYears	12/01/2017 18:12	Microsoft Excel C...	19 KB

File name: MultPilingOpsMultYears_simulation1
Save as type: CSV (Comma delimited)
Authors: Rachael Plunkett
Tags: Add a tag
Title: Add a title

You have now created your piling schedule. Please move to **section 10 config_ver5.R**

CLICK HERE

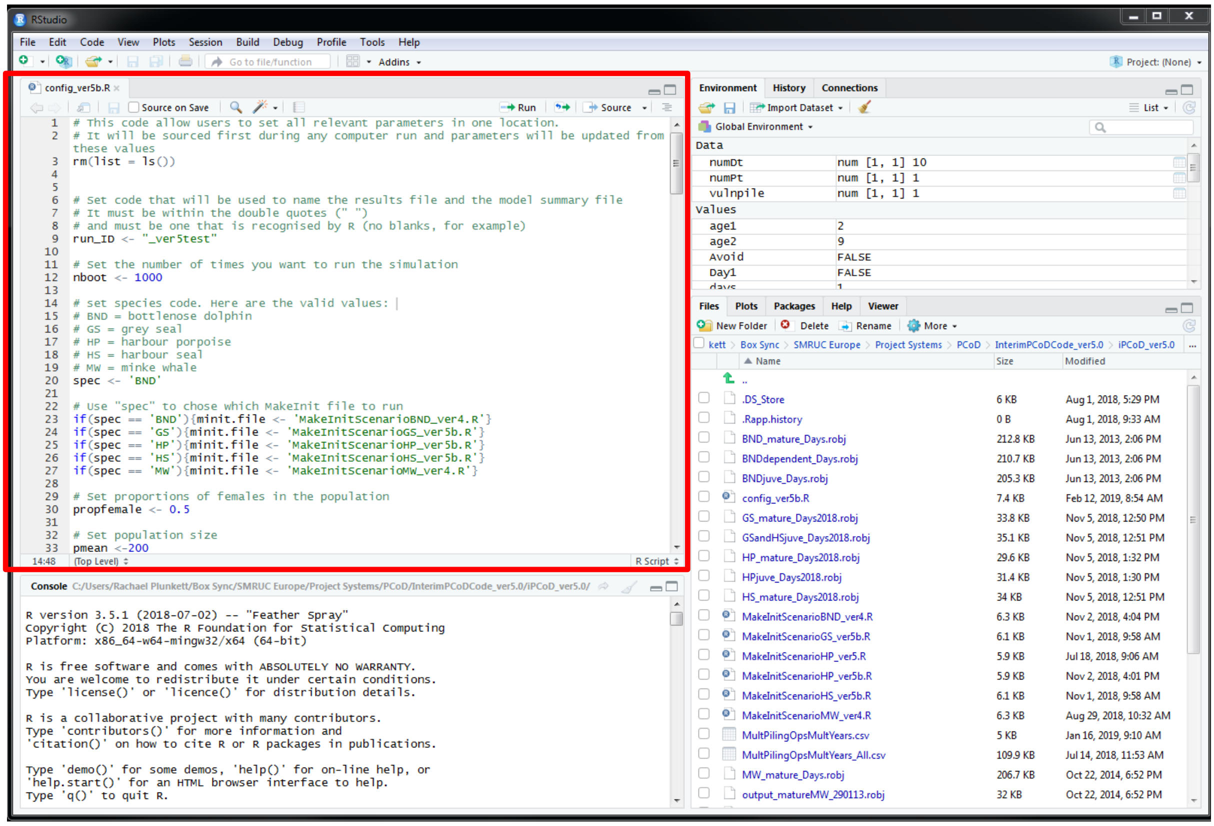
10config_ver5b.R

Navigate to C:\RRun.

Create a copy of the “InterimPCoDCode_ver5.0” folder and call it something new (e.g. simulation1).

Open the copied and renamed folder that you wish to work on (e.g. simulation1).

Double click on the R file called config_ver5b. This will open the R file in the script window in RStudio (framed red in the image below).



```
# This code allow users to set all relevant parameters in one location.
# It will be sourced first during any computer run and parameters will be updated from
these values
rm(list = ls())


# Set code that will be used to name the results file and the model summary file
# It must be within the double quotes (" ")
# and must be one that is recognised by R (no blanks, for example)
run_ID <- "_ver5test"

# Set the number of times you want to run the simulation
nboot <- 1000

# set species code. Here are the valid values:
# BND = bottlenose dolphin
# GS = grey seal
# HP = harbour porpoise
# HS = harbour seal
# MW = minke whale
spec <- 'BND'

# Use "spec" to chose which MakeInit file to run
if(spec == 'BND'){minit.file <- 'MakeInitScenarioBND_ver4.R'}
if(spec == 'GS'){minit.file <- 'MakeInitScenarioGS_ver5b.R'}
if(spec == 'HP'){minit.file <- 'MakeInitScenarioHP_ver5b.R'}
if(spec == 'HS'){minit.file <- 'MakeInitScenarioHS_ver5b.R'}
if(spec == 'MW'){minit.file <- 'MakeInitScenarioMW_ver4.R'}

# Set proportions of females in the population
propfemale <- 0.5

# Set population size
pmean <-200
```

Console C:/Users/Rachael Plunkett/Box Sync/SMRUC Europe/Project Systems/PCoD/InterimPCoDCode_ver5.0/iPCoD_ver5.0/

```
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

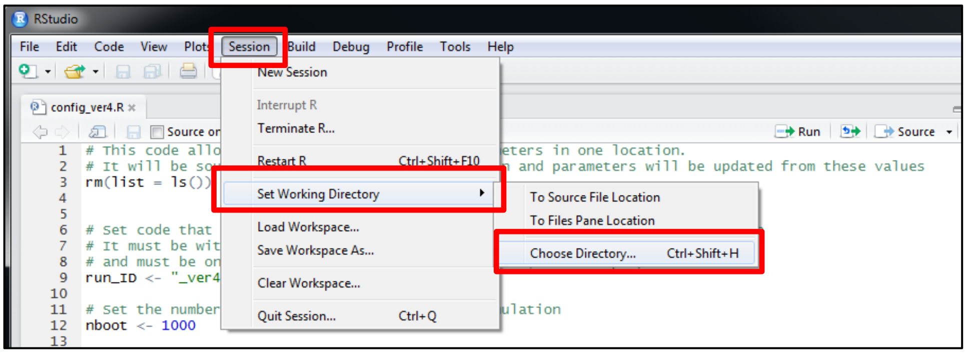
Data	
numDt	num [1, 1] 10
numPt	num [1, 1] 1
vulnpile	num [1, 1] 1
Values	
age1	2
age2	9
Avoid	FALSE
Day1	FALSE
days	1

kett > Box Sync > SMRUC Europe > Project Systems > PCoD > InterimPCoDCode_ver5.0 > iPCoD_ver5.0 ...

Name	Size	Modified
..		
.DS_Store	6 KB	Aug 1, 2018, 5:29 PM
.Rapp.history	0 B	Aug 1, 2018, 9:33 AM
BND_mature_Days.robj	212.8 KB	Jun 13, 2013, 2:06 PM
BNDdependent_Days.robj	210.7 KB	Jun 13, 2013, 2:06 PM
BNDjuve_Days.robj	205.3 KB	Jun 13, 2013, 2:06 PM
config_ver5b.R	7.4 KB	Feb 12, 2019, 8:54 AM
GS_mature_Days2018.robj	33.8 KB	Nov 5, 2018, 12:50 PM
GSandHSjuve_Days2018.robj	35.1 KB	Nov 5, 2018, 12:51 PM
HP_mature_Days2018.robj	29.6 KB	Nov 5, 2018, 1:32 PM
HPjuve_Days2018.robj	31.4 KB	Nov 5, 2018, 1:30 PM
HS_mature_Days2018.robj	34 KB	Nov 5, 2018, 12:51 PM
MakeInitScenarioBND_ver4.R	6.3 KB	Nov 2, 2018, 4:04 PM
MakeInitScenarioGS_ver5b.R	6.1 KB	Nov 1, 2018, 9:58 AM
MakeInitScenarioHP_ver5.R	5.9 KB	Jul 18, 2018, 9:06 AM
MakeInitScenarioHP_ver5b.R	5.9 KB	Nov 2, 2018, 4:01 PM
MakeInitScenarioHS_ver5b.R	6.1 KB	Nov 1, 2018, 9:58 AM
MakeInitScenarioMW_ver4.R	6.3 KB	Aug 29, 2018, 10:32 AM
MultPilingOpsMultYears.csv	5 KB	Jan 16, 2019, 9:10 AM
MultPilingOpsMultYears_All.csv	109.9 KB	Jul 14, 2018, 11:53 AM
MW_mature_Days.robj	206.7 KB	Oct 22, 2014, 6:52 PM
output_matureMW_290113.robj	32 KB	Oct 22, 2014, 6:52 PM

10.1 Set the working directory

Click on the menu tab **Session** then **Set Working Directory** then **Choose Directory** (see screenshot and red frames below) and browse to the **RRun** folder on the C drive. Select the folder you are working on (e.g. “simulation1”) and click Select Folder.



The following text will appear in the Console in RStudio:

```
> setwd("C:/RRun/simulation1")
```

This means that from now on, all files that R accesses will be from this simulation1 folder.

10.2 Setting user defined input parameters

In order to populate the config_ver5b.R file, you will need to collect (or calculate) the relevant information to run your simulations. You will need to:

1. Identify the marine mammal Management Unit(s) (MUs) that may be affected by each development (e.g. see IAWWMG 2015 or SCANS III for cetacean MUs and SCOS reports for seal MUs).
2. For each MU, look up the estimate for the current size of the population in that MU.
3. Look up the appropriate values for the key demographic rates for this population. The current recommended values are listed in Table 1:
    a. annual survival rate for pups or calves
    b. annual survival rate for juveniles (animals that are not yet able to give birth)
    c. annual survival rate for adults
    d. the average age at which females give birth for the first time, and
    e. fecundity rate (average probability of giving birth) for mature females for this population.
4. Decide on a range of values for the proportion (or proportions) of this population that is likely to be vulnerable to the effects of each development.
5. Compile estimates of the number of animals of the species under consideration that may be disturbed during each day of construction work and the number (if any) that are predicted to experience a permanent threshold shift (PTS) (these data are usually found in development specific EIAs).
6. Decide on an appropriate range of values for the number of days of 'residual' disturbance[4] associated with one day of actual disturbance.

---

[4] There is evidence that the effects of disturbance may persist for a number of days after an animal is actually disturbed. We refer to this as "residual" disturbance. The number of days of residual disturbance associated with 1 day of significant disturbance is set by the user. See Harwood et al., (2013) for more information.

Note: The values listed in Table 1 are recommended values at the time of writing. Individual users should check the validity of these values for their own situations. We recommend discussing and agreeing all of these inputs with SNCBs prior to carrying out any simulations. If users wish to model populations not specified in Table 1, we recommend you contact SMRU Consulting for advice.

Should the user wish to change any of the demographic parameters (e.g. if new information becomes available for a particular population) then they should check that the population growth rate that these edited parameters result in fits with their expected growth rate. This can be done using the command: `ev$val[1]` ***which can be run after the user has completed the config_ver5b.r file and run the first 2 lines in the PCoDControl_ver5d.r file to read in the config and the minit files.***

Table 1 Species demographic parameters recommended for the iPCoD v5.0 code (these remain the same as recommended in v4.0).

Species	MU	Age calf/pup becomes independent	Age of first birth	Calf/Pup Survival	Juvenile Survival	Adult Surv	Fertility	Growth Rate
		age1	age2	Surv[1]	Surv[7]	Surv[13]		
Harbour Porpoise	North Sea low adult surv.	1	5	0.6	0.85	0.85	0.958	1.0000
	North Sea high adult surv.	1	5	0.6	0.85	0.925	0.479	1.0000
Grey Seal	All	1	5	0.222	0.94	0.94	0.84	1.0100
Harbour Seal	Shetland modified using adult survival	1	4	0.55	0.61	0.8799	0.88	0.9450
	Shetland modified using fecundity	1	4	0.55	0.61	0.94	0.068	0.9450
	Orkney & north coast modified using survival	1	4	0.507	0.52	0.874	0.88	0.9150
	Moray Firth	1	4	0.55	0.61	0.9451	0.88	1.0000
	East Coast modified using survival	1	4	0.5	0.5	0.7701	0.88	0.8200
	South-west Scotland	1	4	0.55	0.61	0.9451	0.88	1.0000
	West Scotland	1	4	0.55	0.61	0.9451	0.88	1.0000
	Western Isles	1	4	0.55	0.61	0.9451	0.88	1.0000
	Northern Ireland	1	4	0.55	0.61	0.9451	0.88	1.0000
Minke Whale	European waters	1	9	0.72	0.77	0.96	0.9	1.0000
Bottlenose Dolphin	All other MUs	2	9	0.86	0.94	0.94	0.25	1.0000
	Coastal East Scotland	2	9	0.9	0.94	0.9497	0.3	1.0180

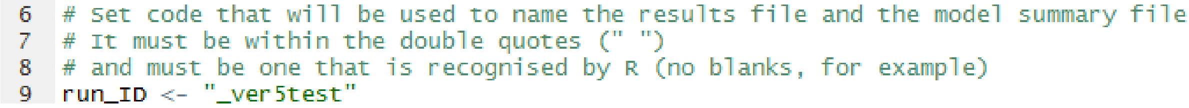
The following section lists the input parameters that need to be defined by the user.

10.2.1 Simulation parameters

10.2.1.1 run_ID

`run_ID <- "_ver5test"`

Choose a name for the output file that will be created for the particular scenario you are investigating and type it between the quotation marks in line 9. In this example below the scenario has been called “_ver5test”:

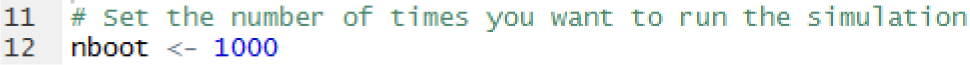


```
6  # Set code that will be used to name the results file and the model summary file
7  # It must be within the double quotes (" ")
8  # and must be one that is recognised by R (no blanks, for example)
9  run_ID <- "_ver5test"
```

10.2.1.2 nboot

`nboot <- 1000`

Set the number of times you want to run the simulation. We recommend that you set this parameter to 1000. Type the number of simulations you’d like to run in line 12.



```
11  # Set the number of times you want to run the simulation
12  nboot <- 1000
```

The higher the number specified here the longer the model will take to run. The user can test the model before running a full model by changing the nboot value to 10 (for example) checking that it runs, then changing it to 1000 to run a full model.

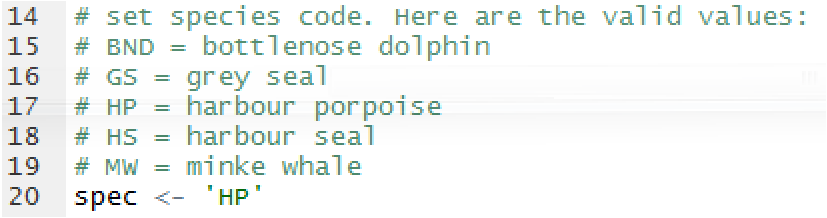
10.2.2 Species parameters

10.2.2.1 spec

`spec <- 'HP'`

Lines 14 to 19 provide the codes for the five species whose dynamics can be modelled. Choose the appropriate code for the species whose dynamics you will be modelling and type it between the quotation marks in line 20.

In this example, the species is harbour porpoise so the code reads **`spec <- 'HP'`**



```
# set species code. Here are the valid values:
# BND = bottlenose dolphin
# GS = grey seal
# HP = harbour porpoise
# HS = harbour seal
# MW = minke whale
spec <- 'HP'
```

Ensure that the species code is in capital letters or the code will not recognise it and when you run the pcodControl_ver5.r file you will get this error message:

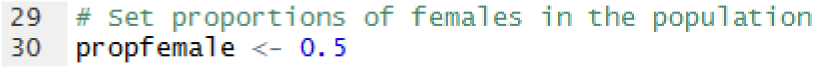
```
Error in source(minit.file) : object 'minit.file' not found
```

10.2.2.2propfemale

`propfemale <- 0.5`

You can change the proportion of females in the population, but we do not recommend doing this unless the sex ratio for the particular MU is known.

The default in the code is 0.5 which means that 50% of the population is female. You can change this ratio in line 30 by typing the percentage as decimal number <1.

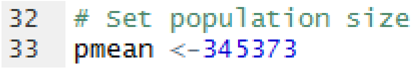


```
29 # Set proportions of females in the population
30 propfemale <- 0.5
```

10.2.2.3pmean

`pmean <-345373`

Set the initial size of the population. In this example the population size for harbour porpoise has been set to the North Sea abundance estimate from SCANS III which is 345,373:

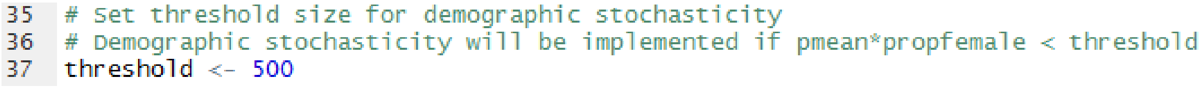


```
# Set population size
pmean <-345373
```

10.2.2.4threshold

`threshold <- 1000`

Demographic stochasticity will be implemented if the total number of females (pmean*propfemale) is less than the specified threshold. The default is 1000. We have increased this value from the 500 recommended with the previous version of iPCoD in order to avoid rounding problems that sometimes arise when small numbers of animals are predicted to be disturbed. We do not recommend changing this value.

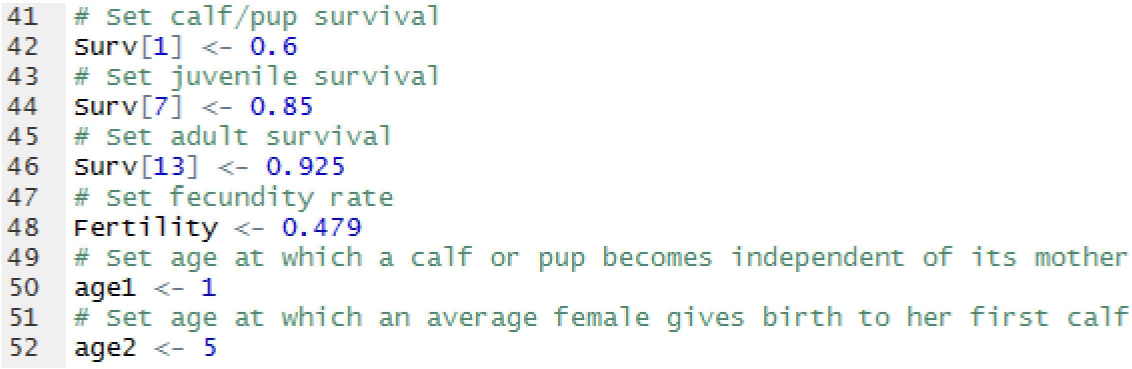


```
# Set threshold size for demographic stochasticity
# Demographic stochasticity will be implemented if pmean*propfemale < threshold
threshold <- 500
```

10.2.2.5 Surv, Fertility & age

```
Surv[1] <- 0.6
Surv[7] <- 0.85
Surv[13] <- 0.925
Fertility <- 0.479
age1 <- 1
age2 <- 5
```

Set appropriate values for the marine mammal MU that you are modelling. In the example below we are using values for harbour porpoise in the North Sea MU based on low adult survival:



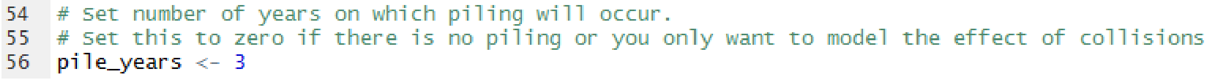
```
# Set calf/pup survival
Surv[1] <- 0.6
# Set juvenile survival
Surv[7] <- 0.85
# Set adult survival
Surv[13] <- 0.925
# Set fecundity rate
Fertility <- 0.479
# Set age at which a calf or pup becomes independent of its mother
age1 <- 1
# Set age at which an average female gives birth to her first calf
age2 <- 5
```

10.2.3 Piling parameters

10.2.3.1 pile_years

```
pile_years <- 3
```

Set the number of years that the total piling schedule covers (as set in the piling schedule detailed in section 9 Piling Schedule).



```
# Set number of years on which piling will occur.
# Set this to zero if there is no piling or you only want to model the effect of collisions
pile_years <- 3
```

You can set this to 0 if you wish to model collisions only with no pile driving.

If you only want to model collisions (e.g. no disturbance or PTS from pile driving) then move to section 10.2.6 NCollisions

CLICK HERE

10.2.3.2 vulnmean

```
vulnmean <- c(0.5,0.5)
```

Within iPCoD it is possible to specify that some of the animals in the population will be completely unaffected by noise associated with piling operations. This is achieved by setting appropriate values for the vector vulnmean, which specifies what proportion of the population is actually vulnerable to the effects of piling from each of the operations being modelled (and, hence, the proportion that will not be effected at all by piling). For example, Thompson et al. (2013)[5] used telemetry data to estimate that 43% (511 out of 1183 animals) of the Moray Firth harbour seal Management Unit was likely to be disturbed by noise associated with wind farm development over the course of construction. This can be modelled within iPCoD by setting **`vulnmean <- c(0.43)`.**

It is possible to identify many vulnerable sub-populations**; the only constraint is that the sum of the proportions specified must be <u>less than or equal</u> to 1.0**.

Where multiple piling operations are being modelled (e.g. in a cumulative impact assessment), it is possible to specify the specific operations that each vulnerable subpopulation is vulnerable to, to do this use the command 'vulnpile' which is explained in more detail below.

Although King et al. (2015)[6] predicted that the effects of piling operations on harbour porpoises in the North Sea were likely to be greater if the population was divided into a number of sub-populations that were vulnerable to different operations, it does not necessarily follow that specifying a number of sub-populations will result in a larger predicted effect.

The decision to divide the population into a number of vulnerable sub-population should be based on the best available scientific evidence rather than an attempt to identify a "precautionary" scenario.

---

[5] Thompson, P.M., Hastie, G.D., Nedwell, J., Barham, R., Brookes, K.L., Cordes, L.S., Bailey, H. and McLean, N., 2013. Framework for assessing impacts of pile-driving noise from offshore wind farm construction on a harbour seal population. Environmental Impact Assessment Review, 43, pp.73-85.

[6] King, S.L., Schick, R.S., Donovan, C., Booth, C.G., Burgman, M., Thomas, L. and Harwood, J., 2015. An interim framework for assessing the population consequences of disturbance. Methods in Ecology and Evolution, 6(10), pp.1150-1158.

The type of evidence that should be considered here is the movement patterns of animals at different time scales and whether or not there is evidence for any sub-population within the MU. For example, the East Coast Scotland bottlenose dolphin MU is estimated to be 195 dolphins, however, dolphins are known to regularly travel outside of the Moray Firth and spend time elsewhere on the east coast of Scotland (Cheney et al., 2013), therefore on a short time frame (e.g. months) not all 195 dolphins will be expected to be present in the Moray Firth and therefore not all of the MU will be vulnerable to construction work within the Moray Firth within that timescale. Therefore it may be appropriate to set a vulnerable subpopulation based on the percentage of the MU that would be found within the Moray Firth at any one time. However, at a longer timescale over a period of years, there is more movement of animals between the Moray Firth and the Forth and Tay area, and therefore different animals may be impacted in different years. In this case it may not be appropriate to divide the MU into vulnerable subpopulations at larger time scales.

If the user is considering operations that are located both within the Moray Firth and within the Forth and Tay area (for example), they may wish to set the vulnerable sub-populations so that:

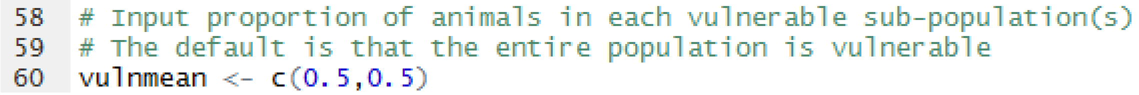
> ***x% of the MU is vulnerable to operations within the Moray Firth only,***
>
> ***y% of the MU is vulnerable to operations within the Forth and Tay area only and***
>
> ***z% is vulnerable to operations in both the Moray Firth and the Forth and Tay area.***

If the user is unsure of what (if any) vulnerable sub-populations may be most appropriate, we suggest they a) liaise with regulators and consultees and b) run a range of scenarios to assess the sensitivity of their inputs.

Each vulnerable subpopulation is separated in the code by a comma:

- If the entire population is vulnerable to disturbance, enter `c(1.0)`
- For 1 vulnerable subpopulation, enter `c(0.#) e.g. c(0.5)`
- For 2 vulnerable subpopulations, enter `c(0.#,0.#)` e.g. `c(0.7,0.3)`
- For 3 vulnerable subpopulations, enter `c(0.#,0.#,0.#)` e.g. `c(0.2,0.3,0.5)`

The following example specifies 2 sub-populations, each of which is half the size of the entire population:



```
# Input proportion of animals in each vulnerable sub-population(s)
# The default is that the entire population is vulnerable
vulnmean <- c(0.5,0.5)
```

If you enter a vulnmean value that sums to > 1.0 (e.g. vulnmean <- c(0.5,0.75)) then you will get this error message when you run pcodControl_ver5b.R:

```
Error in rmultinom(1, Ndist[ivuln, 1], newvulnmean) :
  negative probability
```

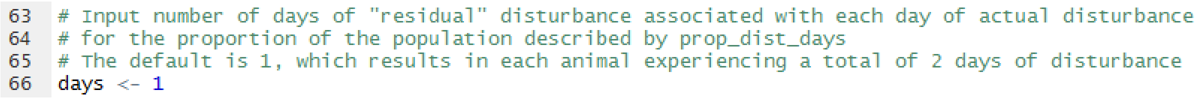
10.2.4 Disturbance parameters

10.2.4.1 days

`days <- 1`

There is evidence that the effects of disturbance may persist for a number of days after an animal is actually disturbed. We refer to this as “residual” disturbance, and the user can specify the number of days of “residual” disturbance these animals experience using the parameter **days**.

The default in the code is 1 which means that the animals experience 1 initial day of disturbance PLUS 1 day of residual disturbance – so a total of 2 days of disturbance.



```
# Input number of days of "residual" disturbance associated with each day of actual disturbance
# for the proportion of the population described by prop_dist_days
# The default is 1, which results in each animal experiencing a total of 2 days of disturbance
days <- 1
```

There is also evidence that some animals are only disturbed for a few hours when they are exposed to noise levels that are sufficient to cause disturbance. This can be modelled by setting a value for **days** that is less than 1. In these cases, the value of **days** indicates how long the period of disturbance lasts, as a fraction of 1 day.

days <- 1/2 means animals are disturbed for half a day (12 hours)
days <- 1/3 means animals are disturbed for a third of a day (8 hours)
days <- 1/4 means animals are disturbed for a quarter of a day (6 hours)

Note: a value of 0 means that there is no “residual” disturbance, so the animal experiences 1 initial day of disturbance and 0 days of residual disturbance – giving a total of 1 day of disturbance.

Note: it is currently not possible to set a total disturbance duration between 1 and 2 days. By entering days<-0 the animal gets 1 initial day of disturbance and no residual disturbance (total 1 day disturbance). By entering days<-1 the animal gets 1 day of initial disturbance plus 1 day residual disturbance (total 2 days disturbance). You cannot specify fractional days of residual disturbance.

10.2.4.2 prop_days_dist & other_days

```
prop_days_dist <- 1.0
other_days <- 0
```

A number of studies have looked at the duration of disturbance as a result of piling activities (e.g. for harbour porpoise: Brandt et al. 2011[7]; Dähne et al. 2013[8]; Brandt et al. 2016[9]; Rumes et al. 2017[10]; Nabe-Nielsen et al. 2018[11]). However it is not known if all disturbed animals will experience the same duration of disturbance. Therefore this section of code allows the user to set a 'dose-response' type approach to the duration of disturbance, where (for example) animals closer to the piling event can be disturbed for longer durations than animals further from the piling event.

The proportion of the animals that experience **days** of residual disturbance is determined by the value of the parameter **prop_days_dist**. If prop_days_dist is <1.0, then the length of disturbance experienced by the remaining proportion (1 minus "prop_days_dist") of disturbed animals is set by the parameter **other_days**.

The default value for this other_days is 0 (i.e. no "residual" disturbance). If **days** has been set at a value < 1.0, then **other_days** **must** be left at its default value of 0.

other_days <- 0 means no residual days in addition to the 1 day of disturbance – so a total of 1 day disturbance.

other_days <- 1 means 1 residual day in addition to the 1 day of disturbance – so a total of 2 day disturbance.

Using these three parameters, it is possible to simulate a wide range of disturbance scenarios:

---

[7] Brandt MJ, Diederichs A, Betke K, Nehls G (2011) Responses of harbour porpoises to pile driving at the Horns Rev II offshore wind farm in the Danish North Sea. Mar Ecol Prog Ser 421:205-216.

[8] Dähne, M., Gilles, A., Lucke, K., Peschko, V., Adler, S., Krügel, K., Sundermeyer, J. and Siebert, U., 2013. Effects of pile-driving on harbour porpoises (Phocoena phocoena) at the first offshore wind farm in Germany. Environmental Research Letters, 8(2), p.025002.

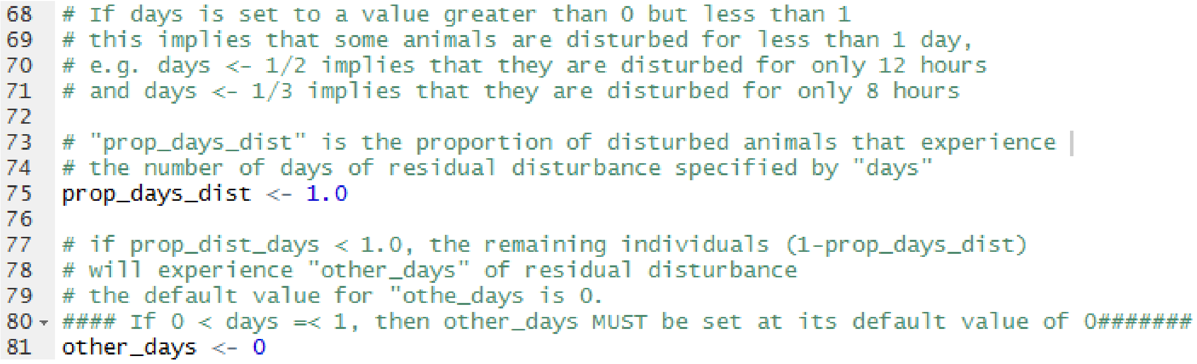
[9] Brandt, M.J., Dragon, A.C., Diederichs, A., Schubert, A., Kosarev, V., Nehls, G., Wahl, V., Michalik, A., Braasch, A., Hinz, C. and Ketzer, C., 2016. Effects of offshore pile driving on harbour porpoise abundance in the German Bight. Assessment of Noise effects. Final report. IBL Umweltplanung GmbH, Institut für Angewandte Ökosystemforschung & BioConsult SH, p.262.

[10] Rumes, B.; Debusschere, E.; Reubens, J.; Norro, A.; Haelters, J.; Deneudt, K.; Degraer, S. (2017). Determining the spatial and temporal extent of the influence of pile driving sound on harbour porpoises, in: Degraer, S. et al. Environmental impacts of offshore wind farms in the Belgian part of the North Sea: A continued move towards integration and quantification. pp. 129-141.

[11] Nabe-Nielsen, J., van Beest, F.M., Grimm, V., Sibly, R.M., Teilmann, J. and Thompson, P.M., 2018. Predicting the impacts of anthropogenic disturbances on marine populations. Conservation Letters, p.e12563.

`days <- 1/3`, `prop_dist_days <- 0.5`, `other_days <- 0` will result in 50% of the disturbed animals being disturbed for 8 hours and 50% being disturbed for 1 day.

`days <- 2`, `prop_dist_days <- 0.25`, `other_days <- 1` will result in 25% of the disturbed animals being disturbed for 3 days and 75% being disturbed for 2 days.

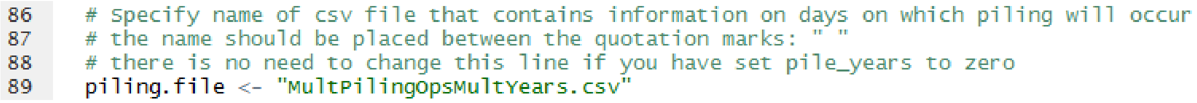


```
# If days is set to a value greater than 0 but less than 1
# this implies that some animals are disturbed for less than 1 day,
# e.g. days <- 1/2 implies that they are disturbed for only 12 hours
# and days <- 1/3 implies that they are disturbed for only 8 hours

# "prop_days_dist" is the proportion of disturbed animals that experience
# the number of days of residual disturbance specified by "days"
prop_days_dist <- 1.0

# if prop_dist_days < 1.0, the remaining individuals (1-prop_days_dist)
# will experience "other_days" of residual disturbance
# the default value for "othe_days is 0.
#### If 0 < days =< 1, then other_days MUST be set at its default value of 0########
other_days <- 0
```

10.2.4.3piling.file

`piling.file <- "MultPilingOpsMultYears.csv"`

This identifies the Piling Schedule to use in the simulation. Type in the name of the piling schedule .csv file in the quotation marks:

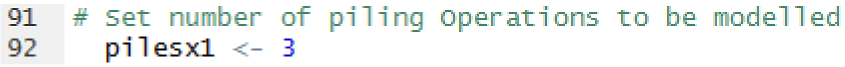


```
# Specify name of csv file that contains information on days on which piling will occur
# the name should be placed between the quotation marks: " "
# there is no need to change this line if you have set pile_years to zero
piling.file <- "MultPilingOpsMultYears.csv"
```

10.2.4.4pilesx1

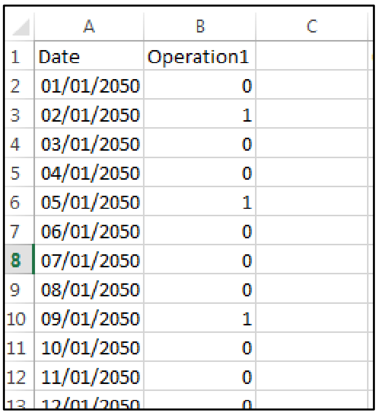
`pilesx1 <- 3`

This specifies how many piling operations are being modelled. In this example 3 piling Operation are being modelled:



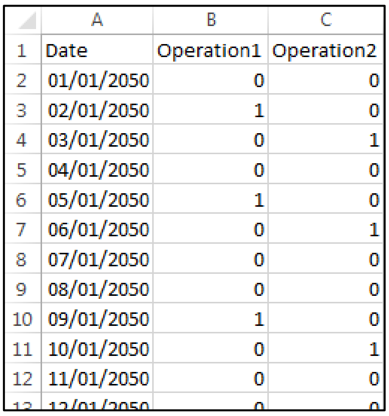
```
# Set number of piling Operations to be modelled
  pilesx1 <- 3
```

`pilesx1 <-1` would need a piling schedule that looks like this:



	A	B	C
1	Date	Operation1	
2	01/01/2050	0	
3	02/01/2050	1	
4	03/01/2050	0	
5	04/01/2050	0	
6	05/01/2050	1	
7	06/01/2050	0	
8	07/01/2050	0	
9	08/01/2050	0	
10	09/01/2050	1	
11	10/01/2050	0	
12	11/01/2050	0	

`pilesx1 <-2` would need a piling schedule that looks like this:



	A	B	C
1	Date	Operation1	Operation2
2	01/01/2050	0	0
3	02/01/2050	1	0
4	03/01/2050	0	1
5	04/01/2050	0	0
6	05/01/2050	1	0
7	06/01/2050	0	1
8	07/01/2050	0	0
9	08/01/2050	0	0
10	09/01/2050	1	0
11	10/01/2050	0	1
12	11/01/2050	0	0

If the number of piling operations in your piling schedule does not match the value specified in pilesx1 then you will get this error message when you run the pcodControl file:

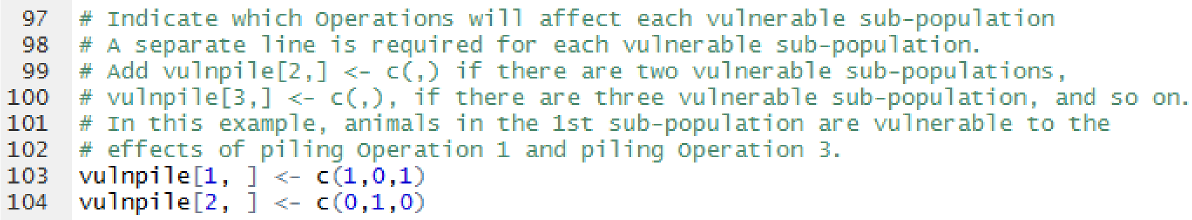
```
Error in eval(expr, envir, enclos) :
  Number of Piling Operations do not match
```

10.2.4.5vulnpile

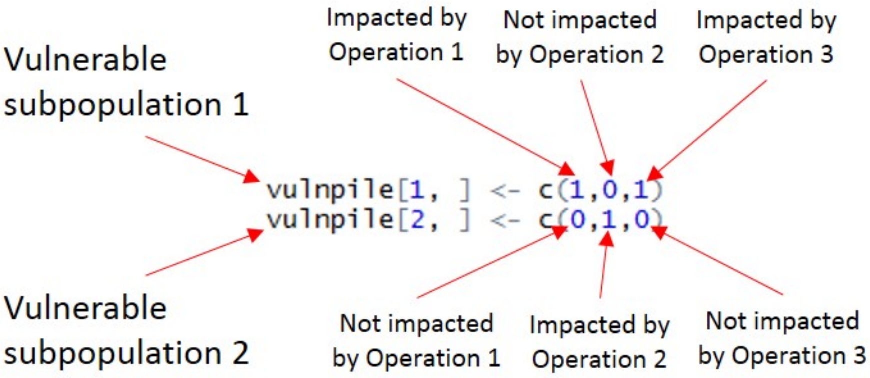
```
vulnpile[1, ] <- c(1,0,1)
vulnpile[2, ] <- c(0,1,0)
```

Now you must indicate which piling Operations will affect each vulnerable sub-population. This requires a separate line for each sub-population that indicates whether or not it will be affected by each of the Operations.

The following example is for the 2 sub-populations identified above, and specifies that sub-population1 is affected by Operation1 and Operation3 of the piling schedule, while sub-population2 is only affected by Operation2.



```
# Indicate which Operations will affect each vulnerable sub-population
# A separate line is required for each vulnerable sub-population.
# Add vulnpile[2,] <- c(,) if there are two vulnerable sub-populations,
# vulnpile[3,] <- c(,), if there are three vulnerable sub-population, and so on.
# In this example, animals in the 1st sub-population are vulnerable to the
# effects of piling Operation 1 and piling Operation 3.
vulnpile[1, ] <- c(1,0,1)
vulnpile[2, ] <- c(0,1,0)
```


Impacted by Operation 1
Not impacted by Operation 2
Impacted by Operation 3
Vulnerable subpopulation 1
vulnpile[1,] <- c(1,0,1)
vulnpile[2,] <- c(0,1,0)
Vulnerable subpopulation 2
Not impacted by Operation 1
Impacted by Operation 2
Not impacted by Operation 3

If you have specified 2 vulnerable sub-populations then you must have 2 lines for the parameter vulnpile, one for each vulnerable sub population.

Note: if your number of vulnerable subpopulations does not equal the number of vulnpile lines, e.g. here we have specified 2 vulnerable sub-populations in vulnmean, but have specified 3 sub-populations in vulnpile:

```
vulnmean <- c(0.5,0.5)
vulnpile[1, ] <- c(1,1,1)
vulnpile[2, ] <- c(0,1,0)
vulnpile[3, ] <- c(1,0,1)
```

then you will get the following error message when you run pcodControl_ver3.r:

```
Error in `[<-`(`*tmp*`, 3, , value = c(1, 1)) : subscript out of bounds
```

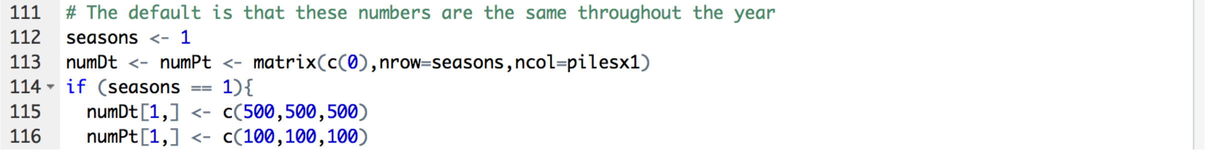
10.2.4.6 seasons, numDt & numPt

```
Seasons <- 1
numDt[1]<- c(622,622,622)
numPt[1]<- c(3,3,3)
```

Seasons

In version 5 of iPCoD it is possible to specify that the number of animals that are disturbed by, and experience PTS as a result of, a particular operation varies through the year because of changes in density. This is done by specifying a value of 4 for the variable “seasons”. This allows you to specify different values for numDt and numPt in Winter (December, January and February), Spring (March, April and May), Summer (June, July and August) and Autumn (September, October and November). It is, of course, possible to specify the same values of numDt and numPt in some seasons if densities are expected to be the same.

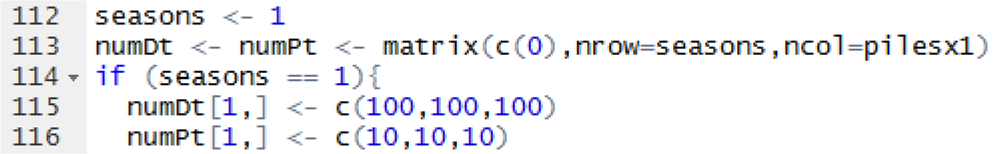
The default value of “seasons” is 1, which implies that numDt and numPt are the same throughout the year.



```
# The default is that these numbers are the same throughout the year
seasons <- 1
numDt <- numPt <- matrix(c(0),nrow=seasons,ncol=pilesx1)
if (seasons == 1){
  numDt[1,] <- c(500,500,500)
  numPt[1,] <- c(100,100,100)
```

<u>**numDt and numPt**</u>

Input the number of animals predicted to experience disturbance (**`numDt`**) and PTS (**`numPt`**) on one day of piling for each Operation. These are typically estimated by combining impact zones from noise modelling with species density surfaces to estimate the number of animals impacted. There must be as many values as the number of piling Operations (**`pilesx1`**), so for an example with 3 piling Operations, there are 3 values listed in the example shown below.



```
seasons <- 1
numDt <- numPt <- matrix(c(0),nrow=seasons,ncol=pilesx1)
if (seasons == 1){
  numDt[1,] <- c(100,100,100)
  numPt[1,] <- c(10,10,10)
```

Note: Do not enter a numPT value that is greater than the numDT value, because this implies that more animals get PTS than the number that get disturbed. Although the code will still run under this situation, it is not biologically realistic.

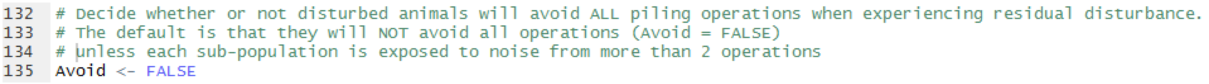
The **`numDT`** and the **`numPT`** values can be different for each Operation. In the following example, Operation 1 disturbs 622 animals and causes PTS to 3 animals, Operation 2 disturbs 500 animals and 2 animals get PTS and Operation 3 disturbs 140 animals and 1 animal gets PTS:

```
numDt[1] <- c(622,500,140)
numPt[1] <- c(3,2,1)
```

10.2.4.7 Avoid

`Avoid <- FALSE`

By setting Avoid <- FALSE disturbed animals will NOT avoid other piling operations while it is experiencing residual disturbance. **This is the default recommended setting.**



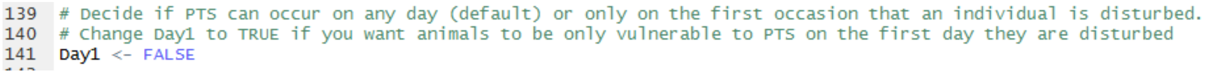
```
# Decide whether or not disturbed animals will avoid ALL piling operations when experiencing residual disturbance.
# The default is that they will NOT avoid all operations (Avoid = FALSE)
# unless each sub-population is exposed to noise from more than 2 operations
Avoid <- FALSE
```

If you want an animal to only avoid the operation that disturbed it, set **`Avoid <- TRUE`**.

10.2.4.1 Day1

`Day1<-FALSE`

Decide if PTS can occur on any day or only on the first occasion that an individual is disturbed. The default setting is Day1<-FALSE which means that an animal can get PTS on any day – not just the first time it is disturbed.



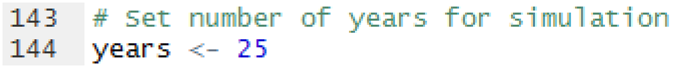
```
# Decide if PTS can occur on any day (default) or only on the first occasion that an individual is disturbed.
# Change Day1 to TRUE if you want animals to be only vulnerable to PTS on the first day they are disturbed
Day1 <- FALSE
```

If you want animals to only be vulnerable to PTS on the first day they are disturbed then set Day1<-TRUE.

10.2.5 Years

`years <- 25`

Specify the number of years to be simulated. Users should be aware that the predictions of the model become increasingly uncertain as this value is increased; values in excess of 25 years are not usually recommended.

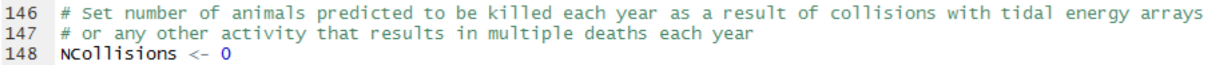


```
# Set number of years for simulation
years <- 25
```

10.2.6 NCollisions

`NCollisions <- 0`

The user can also specify that a certain number of animals are killed **each year** as a result of collisions with renewable energy devices or any other activity that results in multiple deaths.



```
# Set number of animals predicted to be killed each year as a result of collisions with tidal energy arrays
# or any other activity that results in multiple deaths each year
NCollisions <- 0
```

If you are NOT modelling density dependence then you are done with the config_ver4.r file. Save the config_ver5b.r file and move to **section 11 pcodControl** to run the simulation.

CLICK HERE

10.2.7 Density Dependence

The iPCoD code allows the user to simulate the potential effects of density dependence, if there is information on the form of this relationship for the MU that is being modelled. It uses the same density dependent relationship that Thompson et al. (2013) used to model the effect of wind farm construction on the Moray Firth harbour seal population.

This relationship was originally proposed by Taylor & DeMaster (1993)[12] and takes the form:

$$Fert_t = Fert_K + (Fert_0 - Fert_K)*(1 - (N_t/K)^z)$$

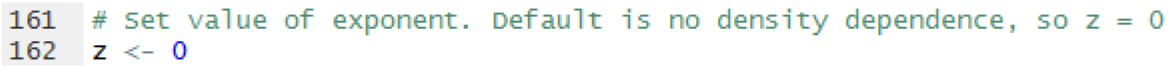
where

- $Fert_t$ is the fecundity rate at population size N_t
- $Fert_0$ is the maximum possible value for fecundity
- K is the carrying capacity for the population
- $Fert_K$ is the fecundity rate when the population is at carrying capacity
- z determines the shape of the density dependent relationship, and Taylor & DeMaster (1993) provide more details of this.

NOTE: this density dependence code is not fully developed and tested. It was created for harbour seals in the Moray Firth MU as part of a separate project. Selection of density dependent values and testing have not been developed for any other MUs or for any other species. It is recommended that this section of code is NOT used unless the user has the necessary data to support the selection of values for the shape of the density dependent relationship, the carrying capacity and the maximum value for fecundity under density dependence.

10.2.7.1 z (Density dependence)

The default in the code is that density dependence is not modelled, and **z** is set to 0.



```
# Set value of exponent. Default is no density dependence, so z = 0
z <- 0
```

---

[12] Taylor, B.L. and Demaster, D.P., 1993. Implications of non-linear density dependence. Marine Mammal Science, 9(4), pp.360-371.

This value determines the shape of the density dependent relationship. This is calculated based on the following relationship:

$$z = \frac{\ln(0.5)}{\ln(\frac{N}{K})}$$

where

- z is the shape parameter
- ln(0.5) is the natural logarithm of 0.5 if we want a given parameter to have changed 50% of its range
- ln(N/K) is the natural logarithm of the population size (N) divided by the carry capacity (K)

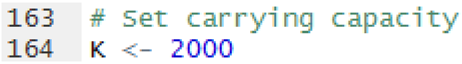
- If z = 1 the density dependent relationship is a straight line
- If z <1 most density dependence occurs when the population size < K/2
- if z > 1, most density dependence occurs when the population is close to K

Here are some example z value calculations as presented in Taylor & DeMaster (1993):

proportion of parameter range changed	N/K	z
0.5	0.9	6.5788
0.5	0.5	1
0.5	0.95	13.513

10.2.7.2K (Carrying capacity)

K is the carrying capacity = maximum number of individuals of a given species that an area's resources can sustain indefinitely without significantly depleting or degrading those resources.

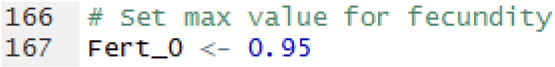


```
# Set carrying capacity
K <- 2000
```

It is possible for `K` ***to be less than*** `pmean` ***(which means that the starting population size is above its carrying capacity). However this means that both the baseline (un-impacted) and impacted population will decrease from the*** `pmean` ***value to a population size at or below the value of K.***

10.2.7.3Fert_0 (Fecundity)

This sets the maximum value for fecundity under density dependence.



```
# Set max value for fecundity
Fert_0 <- 0.95
```

You have now finished setting config_ver4.r. Save the config_ver5b.r file (without changing the name) and move to section 11 pcodControl to run

CLICK HERE

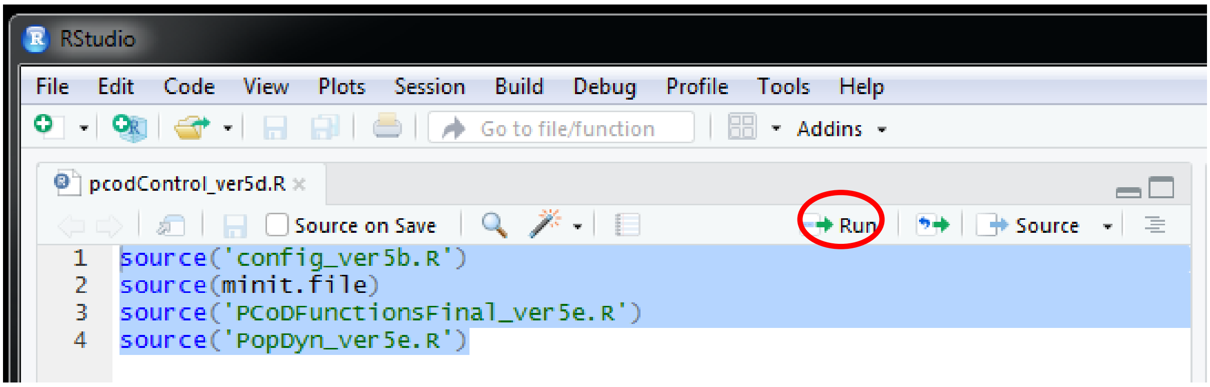
11 pcodControl_ver5d.R

Only work on the pcodControl_ver5.R file if you have set all parameters in config_ver5b.R

Open the pcodControl_ver5d.R file in your simulation folder.

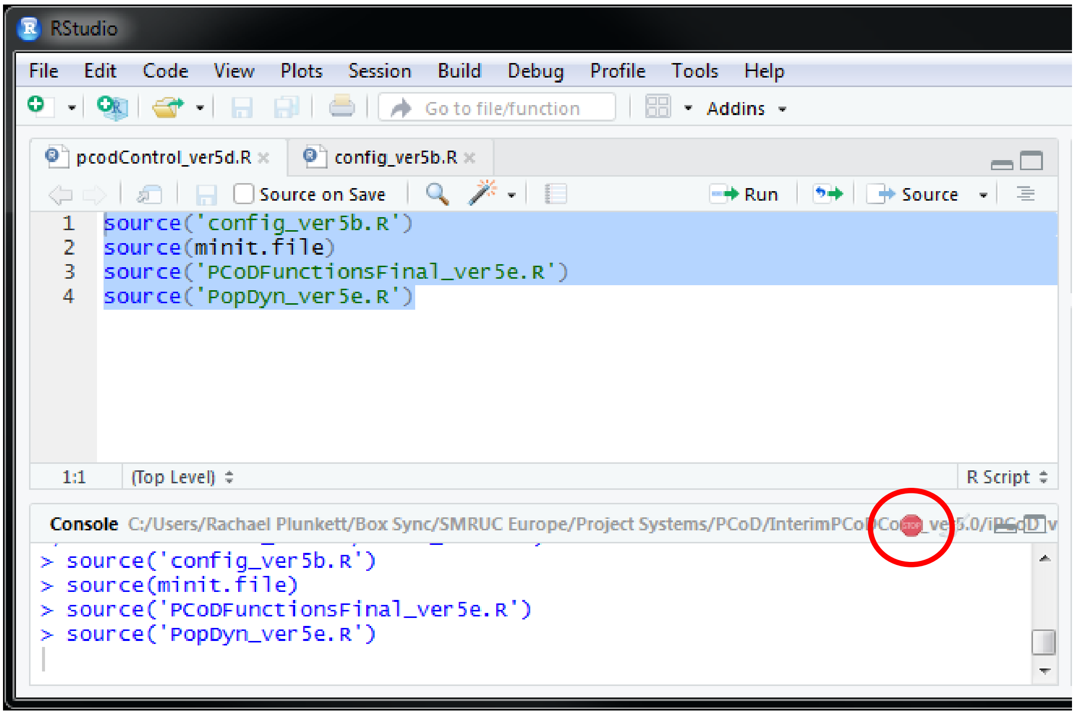
If you have changed the name of the config_ver5b.R file then you will need to type the new name in line 1 so that it calls the correct config file.

Then highlight all four lines and click run (encircled in red in the screenshot below):



```
1  source('config_ver5b.R')
2  source(minit.file)
3  source('PCoDFunctionsFinal_ver5e.R')
4  source('PopDyn_ver5e.R')
```

When the code is running the screen will look as if it is frozen:



```
1  source('config_ver5b.R')
2  source(minit.file)
3  source('PCoDFunctionsFinal_ver5e.R')
4  source('PopDyn_ver5e.R')
```

```
Console C:/Users/Rachael Plunkett/Box Sync/SMRUC Europe/Project Systems/PCoD/InterimPCo
> source('config_ver5b.R')
> source(minit.file)
> source('PCoDFunctionsFinal_ver5e.R')
> source('PopDyn_ver5e.R')
```

The red stop sign lets you know that the code is running. If you want to end the simulation prematurely, you can hit the red stop sign at the top of the console window (encircled red in the above screen shot). However, no output will be produced.

Once the code has finished running the stop sign will appear greyed out.

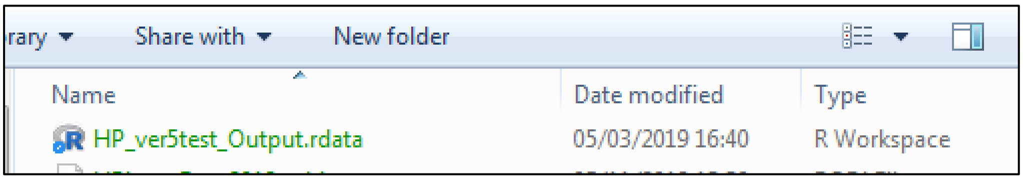
If the code does not run correctly then an error message will be generated in the console window. A list of common error messages and their meanings can be found in **section 13 Possible error messages and their meanings** to help the user solve the problem.

CLICK HERE

11.1 Outputs

The iPCoD code will have created a new file in your simulation folder called **`"spec""run_ID"Output.rdata`** (where "spec" is the species you entered in the config.R file and "run_ID" is the simulation name you entered in the config.R file). This contains the results.

For example: HP_ver5test_Output.rdata



Name	Date modified	Type
HP_ver5test_Output.rdata	05/03/2019 16:40	R Workspace

The .rdata file contains the outputs from iPCoD in a file called "**dat.out**".

dat.out is a matrix called where the results of each of the 1,000 paired population trajectories are stored. The matrix dat.out contains 5 columns and 26 rows for each of the 1,000 simulations.

The columns are:

- dat.out [,1,] "Impacted" is the total size of the impacted population
- dat.out [,2,] "Unimpacted" is the total size of the un-impacted population
- dat.out [,3,] "Unimpacted-Impacted" is the difference in total population size between the two paired populations (un-impacted minus impacted)
- dat.out [,4,] "average pa% change Impacted" is the average % annual change in the size of the impacted population from time.point 1

- dat.out [,5,] “average pa% change Unimpacted” is the average % annual change in the size of the un-impacted population from time.point 1.

The rows represent the population information for each year in the simulation. When running this code it is important to understand that for a 25 year simulation, the dat.out file will have 26 time points for each of the 1,000 paired simulations:

time.point=1 the start of year 1 (ie: before any time has passed or any impact has occurred)

time.point=2 is the start of year 2 (ie: after 1 year of impact)

time.point=3 is the start of year 3 (ie: after 2 years of impact has occurred)

… etc …

time.point=26 is the start of year 26 (ie: after the full 25 simulated years).

You have now run the iPCoD code. Please move to section 12 to get the results in table and graph formats.

12 summary_statistics.R

The summary_statistics.R file was new to v4 of the iPCoD code (and is included in the subsequent v5 code). This file provides outputs from the iPCoD code that we have found useful or that we have generated in discussion with consultees.

Some of these outputs were based on the recommendations of Marine Scotland, following the analyses carried out by Jitlal et al. (2017). Marine Scotland have previously specified (for a number of offshore wind farm projects in Scotland) that the following metrics should be reported as they were considered to be the least sensitive to mis-specification of demographic parameters, therefore enabling more robust assessment of offshore renewable effects:

- Median of the ratio of impacted to un-impacted population size (section 12.7)
- Median of the ratio of impacted to un-impacted annual growth rate (section 12.8)
- Centile for un-impacted population which matches the 50th centile for the impacted population (section 12.9).

However, the exact outputs that are presented in EIAs should be discussed and agreed with consultees in advance. It should be noted that the Jitlal et al (2017) analysis was based on seabird population viability analyses and did not specifically carry out any of these sensitivity analyses with iPCoD specifically or for any marine mammal populations.

If the users are confident with R and fully understands dat.out then they should be able to go into the code and pull out any outputs they require.

When running this code it is important to understand that for a 25 year simulation, the dat.out file will have 26 time points for each of the 1,000 paired simulations:

time.point=1 the start of year 1 (ie: before any time has passed or impact has occurred)

time.point=2 is the start of year 2 (ie: after 1 year of impact)

time.point=26 is the start of year 26 (ie: after 25 simulated years).

This is important to keep in mind when running the code for: CUSTOM.TABLE, CENTILES, HISTOGRAMS and RATIOS below.

You can only perform the following operations once the iPCoD simulation has run and has produced an .rdata output file.

12.1 Load the output file

Open the summary_statistics.R file in your simulation folder.

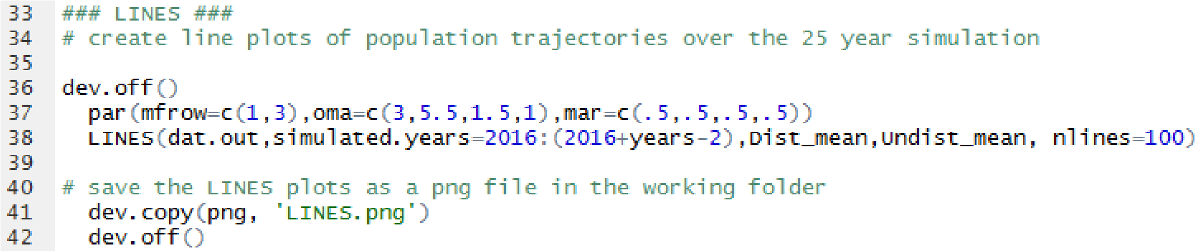
It is essential that you change the code on line 11 so that it loads in the correct output.rdata file.

For example: `HP_ver5test_Output.rdata`

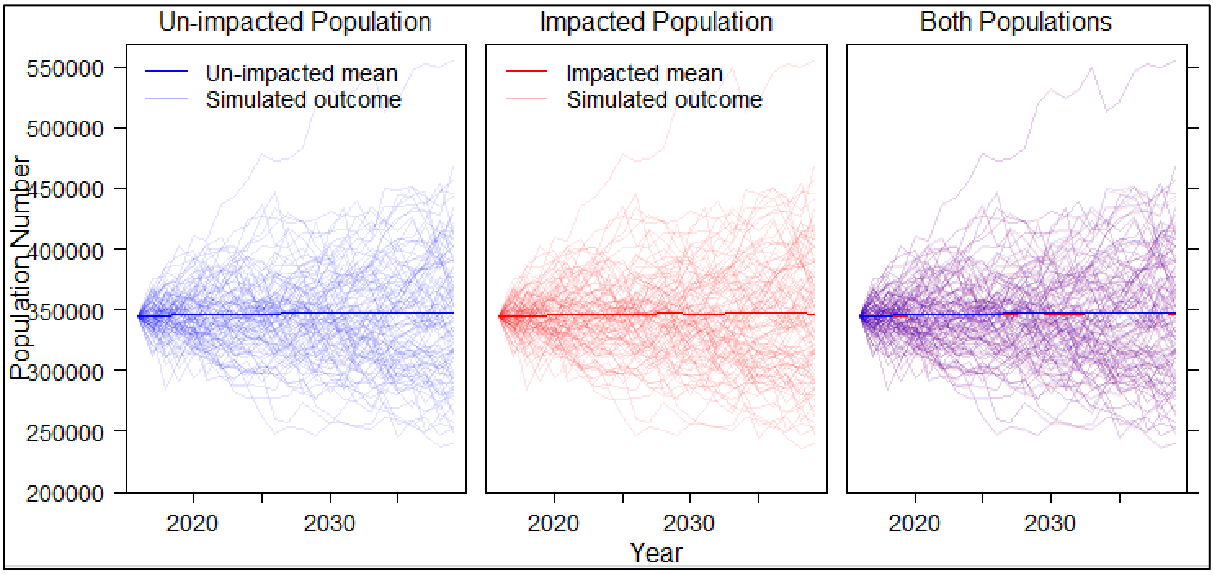
Select and run all lines between Line 1 and Line 30 to load in the code required to produce the summary statistics.

12.2 LINES

Select and run the following lines of code to create a panel of 3 line plots showing the mean population trajectories and the 1000 simulated populations for the un-impacted population, the impacted population and both overlain. The last part of the code then saves this plot to the working folder as a .png file.



```
### LINES ###
# create line plots of population trajectories over the 25 year simulation

dev.off()
  par(mfrow=c(1,3),oma=c(3,5.5,1.5,1),mar=c(.5,.5,.5,.5))
  LINES(dat.out,simulated.years=2016:(2016+years-2),Dist_mean,Undist_mean, nlines=100)

# save the LINES plots as a png file in the working folder
  dev.copy(png, 'LINES.png')
  dev.off()
```

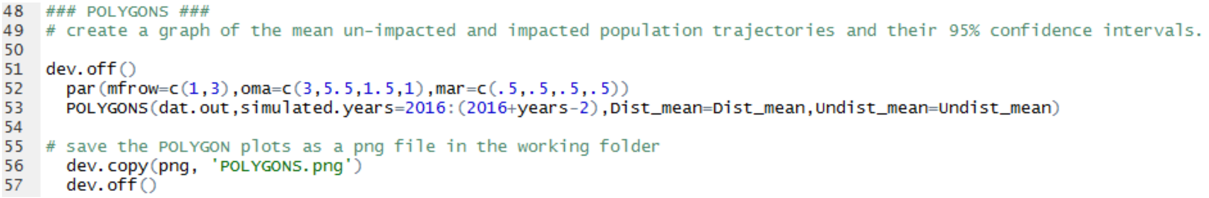


The user can change the start year in the simulation. In the code above the start year is 2016 but this can be changed to any year, for example 2018:

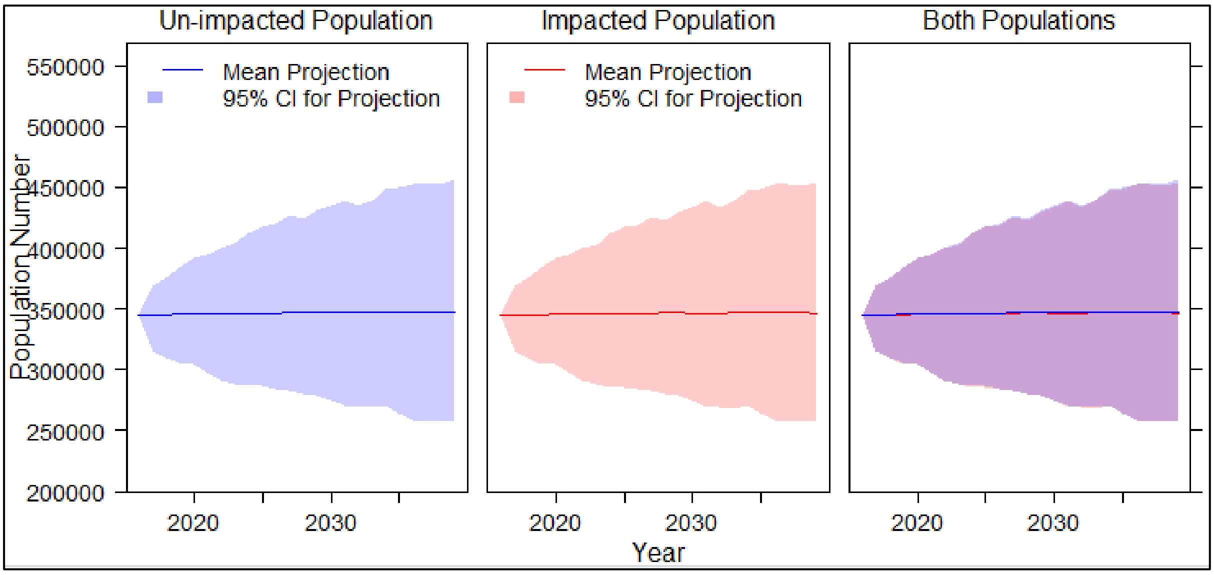
```
LINES(dat.out,simulated.years=2018:(2018+years-2),Dist_mean,Undist_mean,
nlines=100)
```

12.3 POLYGONS

Select and run the following lines of code to create a visualisation of the mean un-impacted and impacted population trajectories and their 95% confidence intervals. The last part of the code then saves this plot to the working folder as a .png file.



```
### POLYGONS ###
# create a graph of the mean un-impacted and impacted population trajectories and their 95% confidence intervals.

dev.off()
  par(mfrow=c(1,3),oma=c(3,5.5,1.5,1),mar=c(.5,.5,.5,.5))
  POLYGONS(dat.out,simulated.years=2016:(2016+years-2),Dist_mean=Dist_mean,Undist_mean=Undist_mean)

# save the POLYGON plots as a png file in the working folder
  dev.copy(png, 'POLYGONS.png')
  dev.off()
```

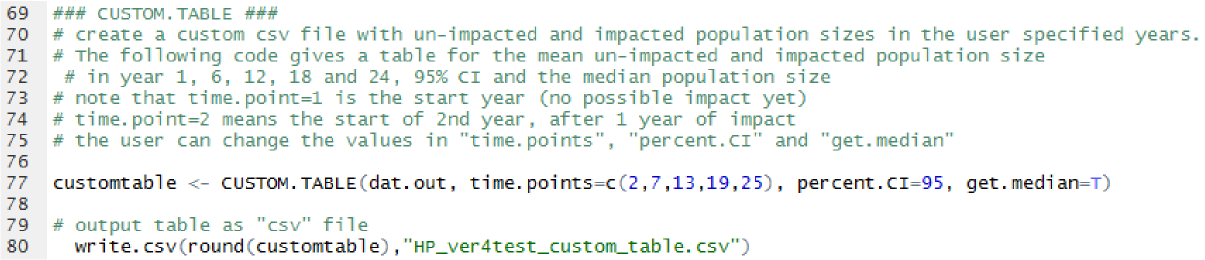


The user can change the start year in the simulation. In the code above the start year is 2016 but this can be changed to any year, for example 2018:

```
POLYGONS(dat.out,simulated.years=2018:(2018+years-2),
Dist_mean=Dist_mean,Undist_mean=Undist_mean)
```

12.5 CUSTOM.TABLE

Select and run the following lines of code to create a .csv file that contains the mean (and median) undisturbed and disturbed population size in user specified years of the simulation and with user specified % Confidence Intervals for the mean.



```
### CUSTOM.TABLE ###
# create a custom csv file with un-impacted and impacted population sizes in the user specified years.
# The following code gives a table for the mean un-impacted and impacted population size
 # in year 1, 6, 12, 18 and 24, 95% CI and the median population size
# note that time.point=1 is the start year (no possible impact yet)
# time.point=2 means the start of 2nd year, after 1 year of impact
# the user can change the values in "time.points", "percent.CI" and "get.median"

customtable <- CUSTOM.TABLE(dat.out, time.points=c(2,7,13,19,25), percent.CI=95, get.median=T)

# output table as "csv" file
  write.csv(round(customtable),"HP_ver4test_custom_table.csv")
```

The user must specify the following:

- time.points (between 1 and 26 if running a 25 year simulation)
- percentage confidence intervals (eg: 90, 95, 99 etc)
- whether or not to present the median population size (`get.median=T`) or only the default mean population size (`get.median=F`)

Remember:

time.point=1 the start of year 1 (ie: before any time has passed or impact has occurred)

time.point=2 is the start of year 2 (ie: after 1 year of impact)

time.point=26 is the start of year 26 (ie: after 25 simulated years).

Here are some examples of code to generate different output tables:

```
(customtable <- CUSTOM.TABLE(dat.out, time.points=c(2,7,12,19,25),
percent.CI=95, get.median=F))
```

The code above will output the mean un-impacted and impacted population size in years 2 (after 1 year of impact), 7 (after 6 years of impact), 13 (after 12 years of impact), 19 (after 18 years of impact) and 25 (after 24 years of impact) and their associated 95% confidence intervals. It will not output the median un-impacted and impacted population size for each time.point.

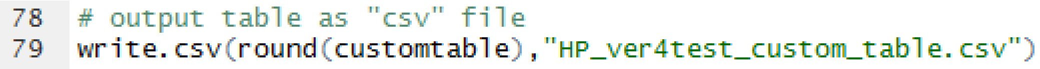
```
(customtable <- CUSTOM.TABLE(dat.out, time.points=c(1,26), percent.CI=c(90,95,99), get.median=T))
```

The code above will output the mean and the median undisturbed and disturbed population size in years 1 (before any impact) and 26 (after 25 years of impact) and their associated 90%, 95% and 99% confidence intervals.

```
(customtable <- CUSTOM.TABLE(dat.out,
time.points=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
22,23,24,25,26), percent.CI=95, get.median=T))
```

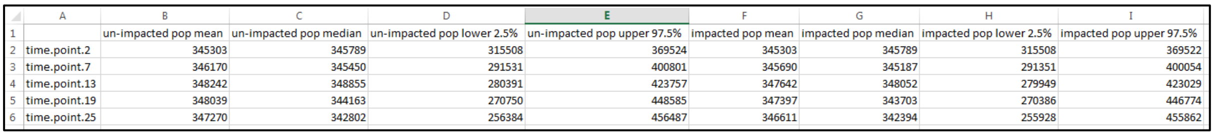
The code above will output the mean and the median undisturbed and disturbed population size in every year between year 1 and 26 (the start and end of a 25 year simulation) and their associated 95% confidence intervals:

The user can change the name of the .csv file produced, by editing the green text between the " " marks on line 79. For example: "`HP_ver4test_custom_table.csv`"



```
78  # output table as "csv" file
79  write.csv(round(customtable),"HP_ver4test_custom_table.csv")
```

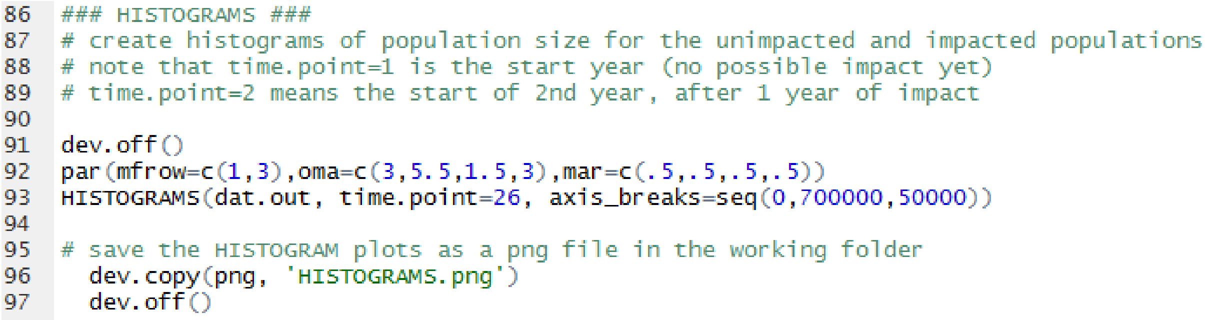
The resulting table in the .csv file looks something like this (depending on the custom table settings):



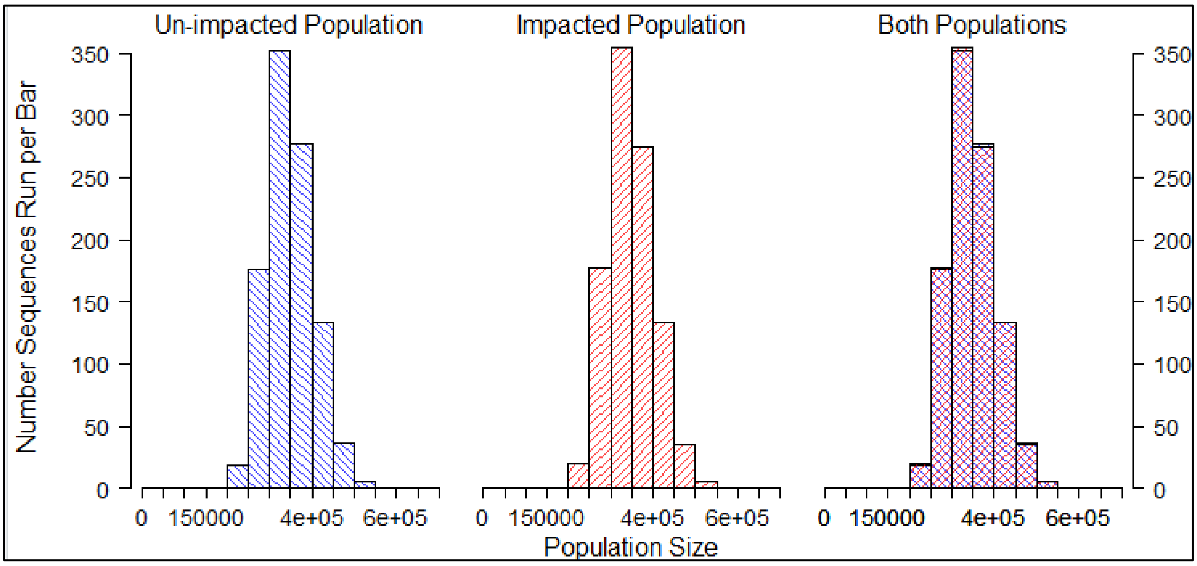
	un-impacted pop mean	un-impacted pop median	un-impacted pop lower 2.5%	un-impacted pop upper 97.5%	impacted pop mean	impacted pop median	impacted pop lower 2.5%	impacted pop upper 97.5%
time.point.2	345303	345789	315508	369524	345303	345789	315508	369522
time.point.7	346170	345450	291531	400801	345690	345187	291351	400054
time.point.13	348242	348855	280391	423757	347642	348052	279949	423029
time.point.19	348039	344163	270750	448585	347397	343703	270386	446774
time.point.25	347270	342802	256384	456487	346611	342394	255928	455862

12.6 HISTOGRAMS

Select and run the following lines of code to create a histogram of predicted population size in the user specified year for each of the un-impacted and impacted populations. The last part of the code then saves this plot to the working folder as a .png file.



```
### HISTOGRAMS ###
# create histograms of population size for the unimpacted and impacted populations
# note that time.point=1 is the start year (no possible impact yet)
# time.point=2 means the start of 2nd year, after 1 year of impact

dev.off()
par(mfrow=c(1,3),oma=c(3,5.5,1.5,3),mar=c(.5,.5,.5,.5))
HISTOGRAMS(dat.out, time.point=26, axis_breaks=seq(0,700000,50000))

# save the HISTOGRAM plots as a png file in the working folder
  dev.copy(png, 'HISTOGRAMS.png')
  dev.off()
```



Remember:

time.point=1 the start of year 1 (ie: before any time has passed or impact has occurred)

time.point=2 is the start of year 2 (ie: after 1 year of impact)

time.point=26 is the start of year 26 (ie: after 25 simulated years).

The user must change the axis breaks. The axis breaks must be informed by the range of the population size in each of the 1,000 simulations. For example:

```
axis_breaks=c(0,700000,50000))
```

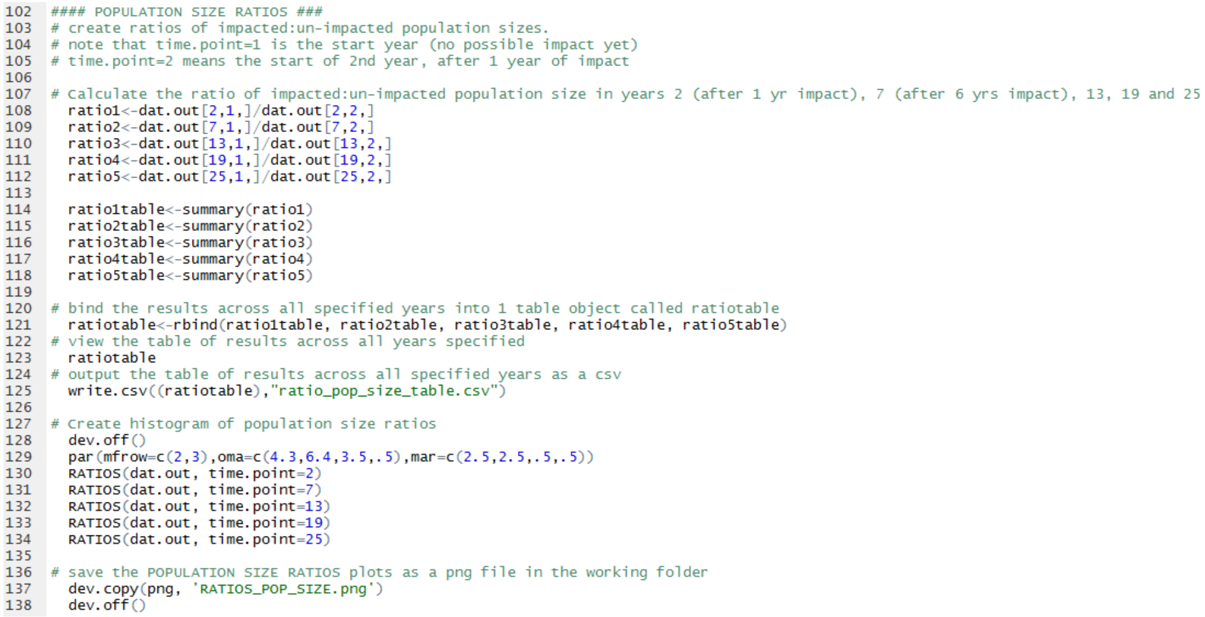
provides a histogram for a population size range between 0 and 700,000 animals with bars set every 50,000 animals.

This requires the user to know the population size range at the user specified time point. The minimum and maximum population size can be quickly obtained by viewing the LINES plot.

12.7 POPULATION SIZE RATIOS

Select and run the following lines of code to output the ratios of impacted:un-impacted population sizes (often termed counterfactual of population size). The ratios are then exported as a .csv file and then the last part of the code creates and saves a histogram plot to the working folder as a .png file.

Each section of this code is explained below:



```
#### POPULATION SIZE RATIOS ###
# create ratios of impacted:un-impacted population sizes.
# note that time.point=1 is the start year (no possible impact yet)
# time.point=2 means the start of 2nd year, after 1 year of impact

# Calculate the ratio of impacted:un-impacted population size in years 2 (after 1 yr impact), 7 (after 6 yrs impact), 13, 19 and 25
  ratio1<-dat.out[2,1,]/dat.out[2,2,]
  ratio2<-dat.out[7,1,]/dat.out[7,2,]
  ratio3<-dat.out[13,1,]/dat.out[13,2,]
  ratio4<-dat.out[19,1,]/dat.out[19,2,]
  ratio5<-dat.out[25,1,]/dat.out[25,2,]

  ratio1table<-summary(ratio1)
  ratio2table<-summary(ratio2)
  ratio3table<-summary(ratio3)
  ratio4table<-summary(ratio4)
  ratio5table<-summary(ratio5)

# bind the results across all specified years into 1 table object called ratiotable
  ratiotable<-rbind(ratio1table, ratio2table, ratio3table, ratio4table, ratio5table)
# view the table of results across all years specified
  ratiotable
# output the table of results across all specified years as a csv
  write.csv((ratiotable),"ratio_pop_size_table.csv")

# Create histogram of population size ratios
  dev.off()
  par(mfrow=c(2,3),oma=c(4.3,6.4,3.5,.5),mar=c(2.5,2.5,.5,.5))
  RATIOS(dat.out, time.point=2)
  RATIOS(dat.out, time.point=7)
  RATIOS(dat.out, time.point=13)
  RATIOS(dat.out, time.point=19)
  RATIOS(dat.out, time.point=25)

# save the POPULATION SIZE RATIOS plots as a png file in the working folder
  dev.copy(png, 'RATIOS_POP_SIZE.png')
  dev.off()
```

Remember:

time.point=1 the start of year 1 (ie: before any time has passed or impact has occurred)

time.point=2 is the start of year 2 (ie: after 1 year of impact)

time.point=26 is the start of year 26 (ie: after 25 simulated years).

12.7.1 POPULATION SIZE RATIOS Part 1

The following code calculates the population size ratio in each user specified time point. In the example below, the user specified time points are every 6 years (specified in the code as 2, 7, 13, 19 and 25). The highlighted numbers in the code below are the time points that can be changed by the user.

```
# Calculate the ratio of impacted:un-impacted population size in years 2 (after 1 yr impact), 7 (after 6 yrs impact), 13, 19 and 25

    ratio1<-dat.out[2,1,]/dat.out[2,2,]
```

```
ratio2<-dat.out[7,1,]/dat.out[7,2,]
ratio3<-dat.out[13,1,]/dat.out[13,2,]
ratio4<-dat.out[19,1,]/dat.out[19,2,]
ratio5<-dat.out[25,1,]/dat.out[25,2,]
```

If, for example, the user wanted the population size ratio for time points 2, 3, 4, 5 and 6 the code would be changed to the following:

```
ratio1<-dat.out[2,1,]/dat.out[2,2,]
ratio2<-dat.out[3,1,]/dat.out[3,2,]
ratio3<-dat.out[4,1,]/dat.out[4,2,]
ratio4<-dat.out[5,1,]/dat.out[5,2,]
ratio5<-dat.out[6,1,]/dat.out[6,2,]
```

12.7.2 POPULATION SIZE RATIOS Part 2

The following lines of code simply creates tables for each of the population size ratios created in Part 1 above.

```
ratio1table<-summary(ratio1)
ratio2table<-summary(ratio2)
ratio3table<-summary(ratio3)
ratio4table<-summary(ratio4)
ratio5table<-summary(ratio5)
```

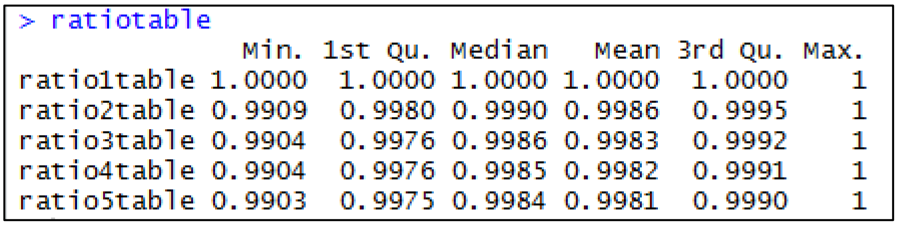
The next part of code then binds the tables into 1 table:

```
# bind the results across all specified years into 1 table object called ratiotable
ratiotable<-rbind(ratio1table, ratio2table, ratio3table, ratio4table, ratio5table)
```

Then the table can be viewed by the user in the console and then exported as a csv file in the working folder:

```
# view the table of results across all years specified
ratiotable
# output the table of results across all specified years as a csv
write.csv((ratiotable),"ratio_pop_size_table.csv")
```

The ratiotable looks like this in R:



```
> ratiotable
                 Min.  1st Qu. Median   Mean 3rd Qu. Max.
ratio1table 1.0000  1.0000 1.0000 1.0000  1.0000    1
ratio2table 0.9909  0.9980 0.9990 0.9986  0.9995    1
ratio3table 0.9904  0.9976 0.9986 0.9983  0.9992    1
ratio4table 0.9904  0.9976 0.9985 0.9982  0.9991    1
ratio5table 0.9903  0.9975 0.9984 0.9981  0.9990    1
```

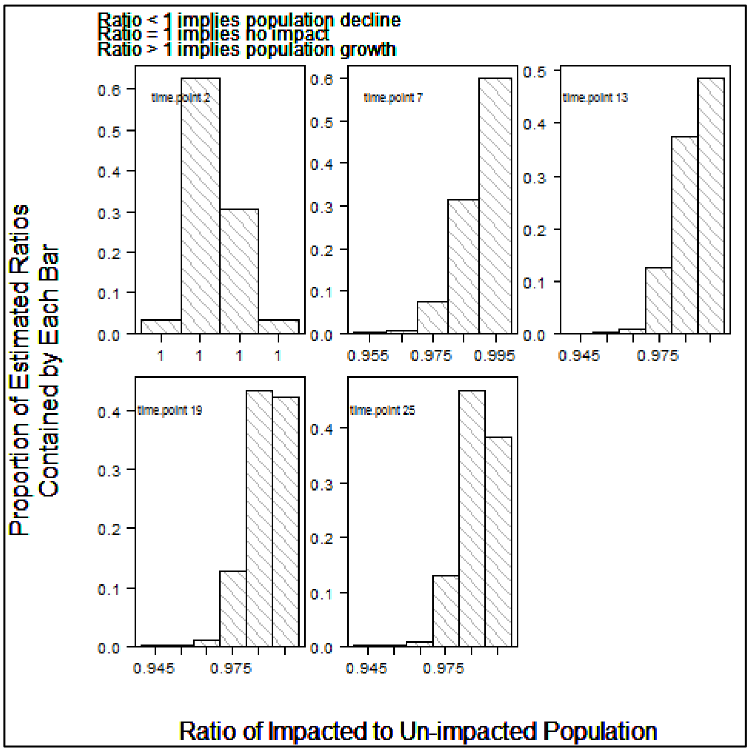
It gives the minimum, 1st quartile, median, mean, 3rd quartile and maximum ratios of impacted:un-impacted population sizes across all 1,000 simulations for each of the user specified time points.

12.7.3 POPULATION SIZE RATIOS Part 3

The following code creates a histogram of the population size ratios at each user specified time point and saves the histogram plot as a png file in the working folder:

```
# Create histogram of population size ratios
dev.off()
        par(mfrow=c(2,3),oma=c(4.3,6.4,3.5,.5),mar=c(2.5,2.5,.5,.5))
        RATIOS(dat.out, time.point=2)
        RATIOS(dat.out, time.point=7)
        RATIOS(dat.out, time.point=13)
        RATIOS(dat.out, time.point=19)
        RATIOS(dat.out, time.point=25)
# save the POPULATION SIZE RATIOS plots as a png file in the working folder
        dev.copy(png, 'RATIOS_POP_SIZE.png')
        dev.off()
```

In this section of code, the user can change the time points they want to plot in the histograms. In the example above, the histograms are created for time points 2, 7, 13, 19 and 25.


Ratio < 1 implies population decline
Ratio = 1 implies no impact
Ratio > 1 implies population growth
Proportion of Estimated Ratios Contained by Each Bar
time.point 2
time.point 7
time.point 13
time.point 19
time.point 25
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1
1
1
1
0.955
0.975
0.995
0.945
0.975
Ratio of Impacted to Un-impacted Population

12.8 GROWTH RATE RATIOS

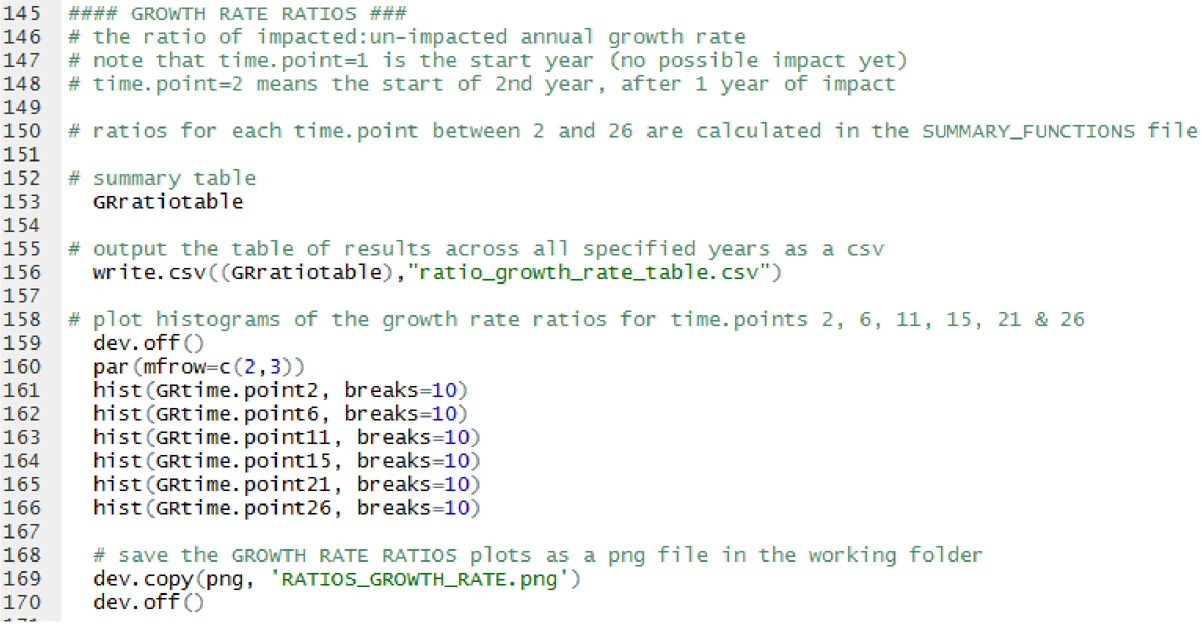
Select and run the following lines of code to create ratios of impacted:un-impacted growth rates.

This is calculated as the ratio of impacted:un-impacted average annual growth rate from time.point1 to each other time.point in the data. For example:

$$\text{GRtime.point10} = \frac{\text{Impacted Population average growth rate between time.point1 and time.point10}}{\text{Un}-\text{impacted Population average growth rate between time.point1 and time.point10}}$$

The growth rate ratios for each time.point between 2 and 26 are then exported as a .csv file.

The last part of the code creates and saves a histogram plot to the working folder as a .png file.



```
#### GROWTH RATE RATIOS ###
# the ratio of impacted:un-impacted annual growth rate
# note that time.point=1 is the start year (no possible impact yet)
# time.point=2 means the start of 2nd year, after 1 year of impact

# ratios for each time.point between 2 and 26 are calculated in the SUMMARY_FUNCTIONS file

# summary table
  GRratiotable

# output the table of results across all specified years as a csv
  write.csv((GRratiotable),"ratio_growth_rate_table.csv")

# plot histograms of the growth rate ratios for time.points 2, 6, 11, 15, 21 & 26
  dev.off()
  par(mfrow=c(2,3))
  hist(GRtime.point2, breaks=10)
  hist(GRtime.point6, breaks=10)
  hist(GRtime.point11, breaks=10)
  hist(GRtime.point15, breaks=10)
  hist(GRtime.point21, breaks=10)
  hist(GRtime.point26, breaks=10)

  # save the GROWTH RATE RATIOS plots as a png file in the working folder
  dev.copy(png, 'RATIOS_GROWTH_RATE.png')
  dev.off()
```

Each section of this code is explained below:

Remember:

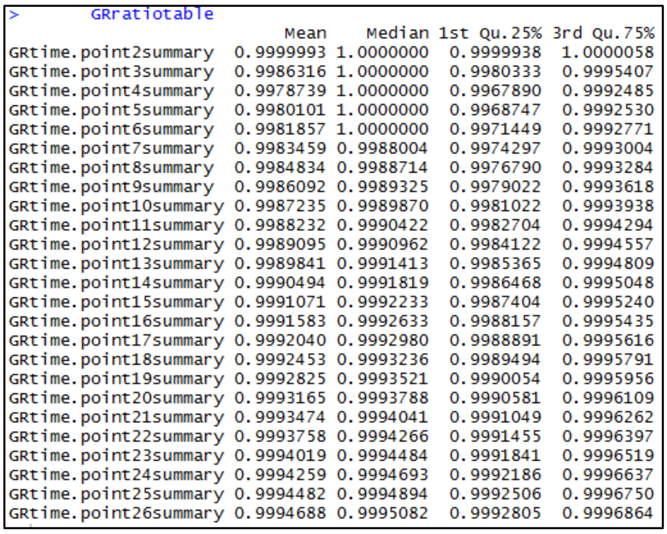
> ***time.point=1 the start of year 1 (ie: before any time has passed or impact has occurred)***
>
> ***time.point=2 is the start of year 2 (ie: after 1 year of impact)***
>
> ***time.point=26 is the start of year 26 (ie: after 25 simulated years).***

12.8.1 GROWTH RATE RATIOS Part 1: ratio table

The growth rate ratios are calculated in the SUMMARY_FUNCTIONS.R file which runs in the background. The resulting growth rate ratios for each time.point are read into a summary table called “GRratiotable” and then output into the working folder as a csv file.

```
# summary table
GRratiotable
# output the table of results across all specified years as a csv
write.csv((GRratiotable),"ratio_growth_rate_table.csv")
```

The GRratiotable looks like this in R:



```
>       GRratiotable
                         Mean    Median 1st Qu.25% 3rd Qu.75%
GRtime.point2summary  0.9999993 1.0000000  0.9999938  1.0000058
GRtime.point3summary  0.9986316 1.0000000  0.9980333  0.9995407
GRtime.point4summary  0.9978739 1.0000000  0.9967890  0.9992485
GRtime.point5summary  0.9980101 1.0000000  0.9968747  0.9992530
GRtime.point6summary  0.9981857 1.0000000  0.9971449  0.9992771
GRtime.point7summary  0.9983459 0.9988004  0.9974297  0.9993004
GRtime.point8summary  0.9984834 0.9988714  0.9976790  0.9993284
GRtime.point9summary  0.9986092 0.9989325  0.9979022  0.9993618
GRtime.point10summary 0.9987235 0.9989870  0.9981022  0.9993938
GRtime.point11summary 0.9988232 0.9990422  0.9982704  0.9994294
GRtime.point12summary 0.9989095 0.9990962  0.9984122  0.9994557
GRtime.point13summary 0.9989841 0.9991413  0.9985365  0.9994809
GRtime.point14summary 0.9990494 0.9991819  0.9986468  0.9995048
GRtime.point15summary 0.9991071 0.9992233  0.9987404  0.9995240
GRtime.point16summary 0.9991583 0.9992633  0.9988157  0.9995435
GRtime.point17summary 0.9992040 0.9992980  0.9988891  0.9995616
GRtime.point18summary 0.9992453 0.9993236  0.9989494  0.9995791
GRtime.point19summary 0.9992825 0.9993521  0.9990054  0.9995956
GRtime.point20summary 0.9993165 0.9993788  0.9990581  0.9996109
GRtime.point21summary 0.9993474 0.9994041  0.9991049  0.9996262
GRtime.point22summary 0.9993758 0.9994266  0.9991455  0.9996397
GRtime.point23summary 0.9994019 0.9994484  0.9991841  0.9996519
GRtime.point24summary 0.9994259 0.9994693  0.9992186  0.9996637
GRtime.point25summary 0.9994482 0.9994894  0.9992506  0.9996750
GRtime.point26summary 0.9994688 0.9995082  0.9992805  0.9996864
```

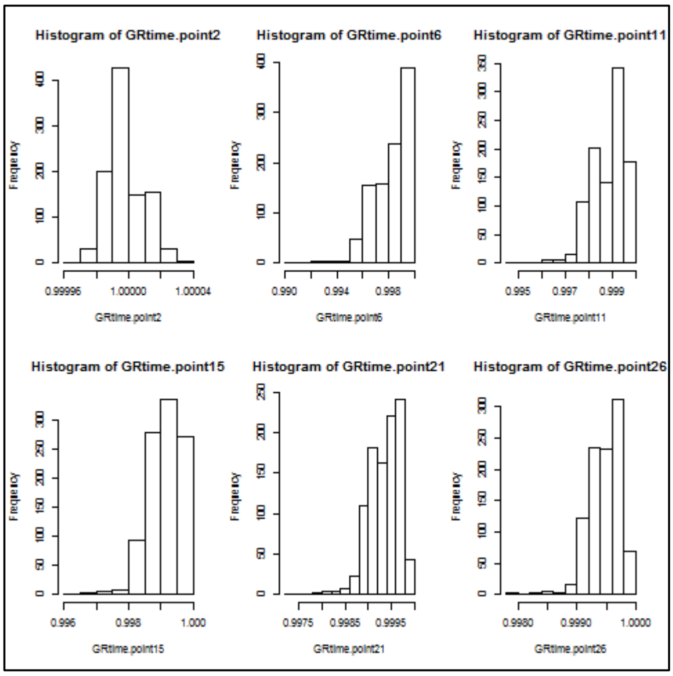
It gives the mean, median, 1st quartile and 3rd quartile ratios of impacted:un-impacted annual growth rate across all 1,000 simulations for each of the time.points between time.point2 and time.point16.

12.8.2 GROWTH RATE RATIOS Part 2: outputting the ratios as histograms

The last part of the code creates histograms for six user specified time points and saves them as a png file. In the example below this exports the growth rate ratios every five years (time.points 2, 6, 11, 16, 21 & 26):

```
# plot histograms of the growth rate ratios for time.points 2, 6,
11, 15, 21 & 26
    dev.off()
    par(mfrow=c(2,3))
```

```
        hist(GRtime.point2, breaks=10)
        hist(GRtime.point6, breaks=10)
        hist(GRtime.point11, breaks=10)
        hist(GRtime.point16, breaks=10)
        hist(GRtime.point21, breaks=10)
        hist(GRtime.point26, breaks=10)
    # save the GROWTH RATE RATIOS plots as a png file in the working folder
        dev.copy(png, 'RATIOS_GROWTH_RATE.png')
        dev.off()
```

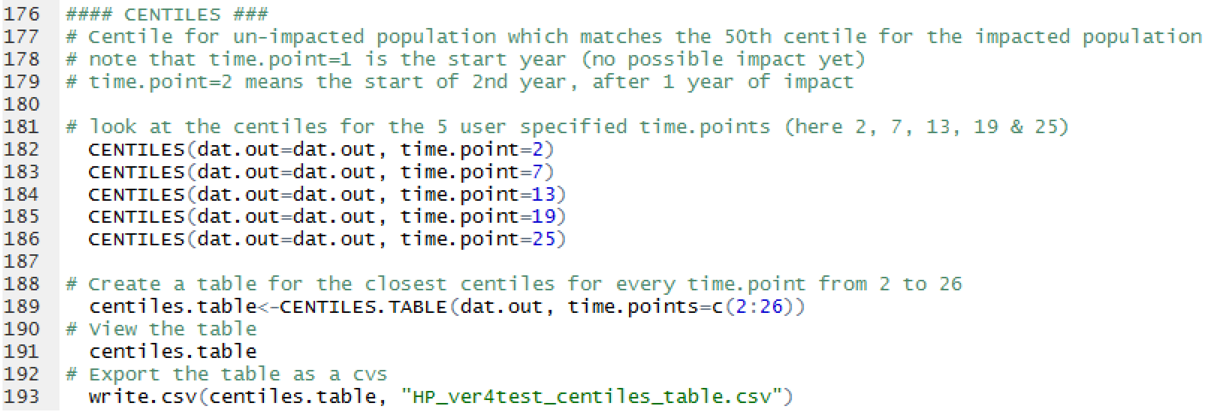


The user can change which time.points are included in the histograms. For example, if the user wants to present time.points 2-7 then the code would be changed to:

```
dev.off()
par(mfrow=c(2,3))
hist(GRtime.point2, breaks=10)
hist(GRtime.point3, breaks=10)
hist(GRtime.point4, breaks=10)
hist(GRtime.point5, breaks=10)
hist(GRtime.point6, breaks=10)
hist(GRtime.point7, breaks=10)
```

12.9 CENTILES

Select and run the following lines of code to obtain the centile for un-impacted population which matches the 50th centile for the impacted population in the user specified year of the simulation.



```
#### CENTILES ###
# Centile for un-impacted population which matches the 50th centile for the impacted population
# note that time.point=1 is the start year (no possible impact yet)
# time.point=2 means the start of 2nd year, after 1 year of impact

# look at the centiles for the 5 user specified time.points (here 2, 7, 13, 19 & 25)
  CENTILES(dat.out=dat.out, time.point=2)
  CENTILES(dat.out=dat.out, time.point=7)
  CENTILES(dat.out=dat.out, time.point=13)
  CENTILES(dat.out=dat.out, time.point=19)
  CENTILES(dat.out=dat.out, time.point=25)

# Create a table for the closest centiles for every time.point from 2 to 26
  centiles.table<-CENTILES.TABLE(dat.out, time.points=c(2:26))
# View the table
  centiles.table
# Export the table as a cvs
  write.csv(centiles.table, "HP_ver4test_centiles_table.csv")
```

Remember:

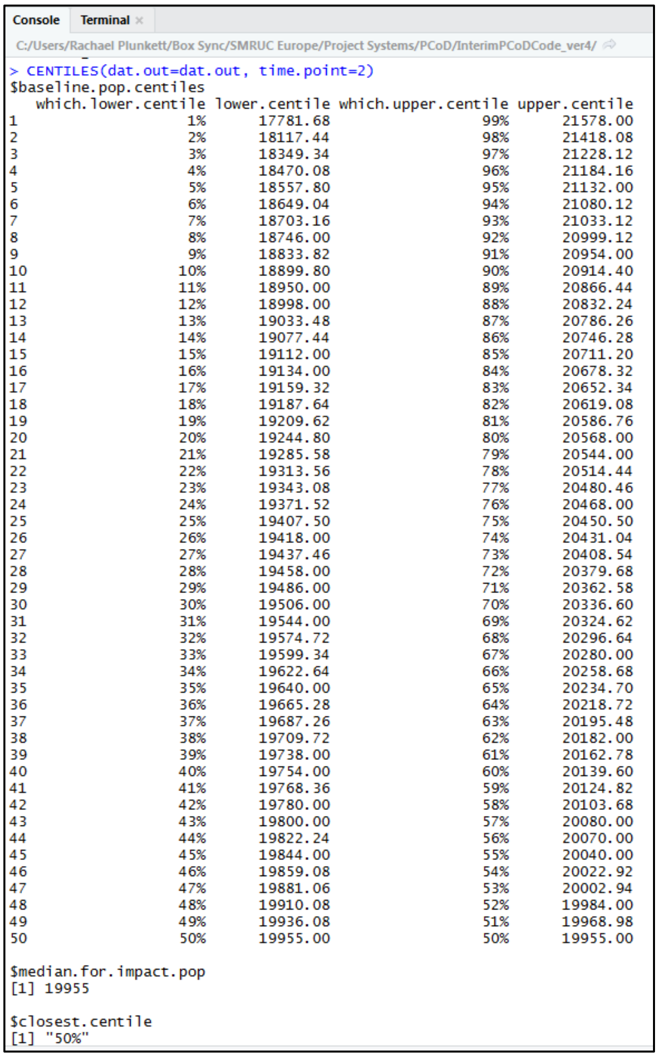
time.point=1 the start of year 1 (ie: before any time has passed or impact has occurred)

time.point=2 is the start of year 2 (ie: after 1 year of impact)

time.point=26 is the start of year 26 (ie: after 25 simulated years).

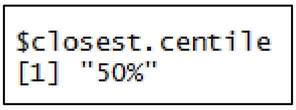
# LOOK AT THE CENTILES FOR THE USER SPECIFIED TIME.POINTS

This first set of code there will produce a series of centile tables in the console window for each user specified time point. The tables will look like this in R:


Console
Terminal
C:/Users/Rachael Plunkett/Box Sync/SMRUC Europe/Project Systems/PCoD/InterimPCoDCode_ver4/

```
> CENTILES(dat.out=dat.out, time.point=2)
$baseline.pop.centiles
   which.lower.centile lower.centile which.upper.centile upper.centile
1                   1%      17781.68                 99%      21578.00
2                   2%      18117.44                 98%      21418.08
3                   3%      18349.34                 97%      21228.12
4                   4%      18470.08                 96%      21184.16
5                   5%      18557.80                 95%      21132.00
6                   6%      18649.04                 94%      21080.12
7                   7%      18703.16                 93%      21033.12
8                   8%      18746.00                 92%      20999.12
9                   9%      18833.82                 91%      20954.00
10                 10%      18899.80                 90%      20914.40
11                 11%      18950.00                 89%      20866.44
12                 12%      18998.00                 88%      20832.24
13                 13%      19033.48                 87%      20786.26
14                 14%      19077.44                 86%      20746.28
15                 15%      19112.00                 85%      20711.20
16                 16%      19134.00                 84%      20678.32
17                 17%      19159.32                 83%      20652.34
18                 18%      19187.64                 82%      20619.08
19                 19%      19209.62                 81%      20586.76
20                 20%      19244.80                 80%      20568.00
21                 21%      19285.58                 79%      20544.00
22                 22%      19313.56                 78%      20514.44
23                 23%      19343.08                 77%      20480.46
24                 24%      19371.52                 76%      20468.00
25                 25%      19407.50                 75%      20450.50
26                 26%      19418.00                 74%      20431.04
27                 27%      19437.46                 73%      20408.54
28                 28%      19458.00                 72%      20379.68
29                 29%      19486.00                 71%      20362.58
30                 30%      19506.00                 70%      20336.60
31                 31%      19544.00                 69%      20324.62
32                 32%      19574.72                 68%      20296.64
33                 33%      19599.34                 67%      20280.00
34                 34%      19622.64                 66%      20258.68
35                 35%      19640.00                 65%      20234.70
36                 36%      19665.28                 64%      20218.72
37                 37%      19687.26                 63%      20195.48
38                 38%      19709.72                 62%      20182.00
39                 39%      19738.00                 61%      20162.78
40                 40%      19754.00                 60%      20139.60
41                 41%      19768.36                 59%      20124.82
42                 42%      19780.00                 58%      20103.68
43                 43%      19800.00                 57%      20080.00
44                 44%      19822.24                 56%      20070.00
45                 45%      19844.00                 55%      20040.00
46                 46%      19859.08                 54%      20022.92
47                 47%      19881.06                 53%      20002.94
48                 48%      19910.08                 52%      19984.00
49                 49%      19936.08                 51%      19968.98
50                 50%      19955.00                 50%      19955.00

$median.for.impact.pop
[1] 19955

$closest.centile
[1] "50%"
```

The import result to note down is the value provided for the closest centile for the un-impacted population which matches the 50th centile for the impacted population:

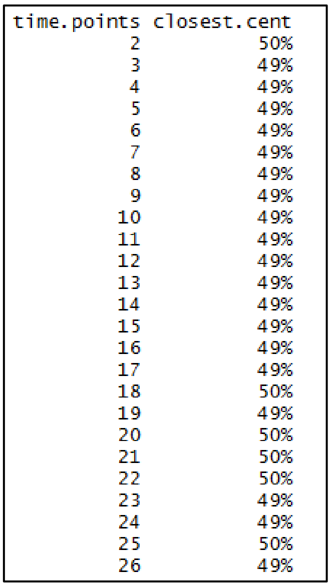


```
$closest.centile
[1] "50%"
```

If the closest centile is 50% (as per the example above) then the paired un-impacted and the impacted populations do not differ in the specified year. If the closest centile is <50% then the impacted population is smaller than the un-impacted population in the specified year. If the closest centile is >50% then the impacted population is larger than the un-impacted population in the specified year.

# CREATE A TABLE FOR THE CLOSEST CENTILES FOR EVERY TIME.POINT FROM 2 TO 26

This second set of code there will produce a table that lists the closest centile for the un-impacted population which matches the 50th centile for the impacted population over all years in the 25 year simulation. The final part of the code will export this table as a csv file in the working folder.



```
time.points closest.cent
          2          50%
          3          49%
          4          49%
          5          49%
          6          49%
          7          49%
          8          49%
          9          49%
         10          49%
         11          49%
         12          49%
         13          49%
         14          49%
         15          49%
         16          49%
         17          49%
         18          50%
         19          49%
         20          50%
         21          50%
         22          50%
         23          49%
         24          49%
         25          50%
         26          49%
```

13 Possible error messages and their meanings

Here is a list of some of the common error message that R produces when trying to run the iPCoD code:

```
Error in source("config_ver5.R") : config_ver4.R:22:8: unexpected symbol
21:
22: # Use "spec
           ^
```

You have most likely entered a **run_ID** that is not acceptable. Most often this is because of the “” symbols on either side of the runID text.

```
Error in source(minit.file) : object 'minit.file' not found
```

You have incorrectly specified the species type under **spec** in the config_ver5.R file. Often this has to do with the “” marks on either side of the species code, or that the letters are not capitals.

```
Error in array(c((age1 + 2):(age2 + 1), (age1 + 1):age2), dim = c((age2 -:
  negative length vectors are not allowed
```

The values for **age1** and **age2** in the config_ver4.R file are not valid. This can happen if you set age2 (age of first birth) to a higher age than age1 (age at independence).

```
Error in eval(expr, envir, enclos) :
Number of Piling Operations do not match
```

The number of piling Operations listed for **pilesx1** in the config_ver4.R file does not match the number of piling Operations listed in the piling schedule .csv file.

```
Error in eval(expr, envir, enclos) :
Number of days in piling file is not an exact multiple of 365
```

The number of days in the piling schedule csv is not an exact multiple of 365. This is most often caused when the piling schedule runs over a leap year which has 366 days, in which case you need to delete the row for 29th February.

```
Error in rmultinom(1, Ndist[ivuln, 1], newvulnmean) :
negative probability
```

The **vulnmean** value in the config_ver5.R file is >1 which is not an acceptable value.

```
Error in `[<-`(`*tmp*`, 3, , value = c(1, 1)) : subscript out of bounds
```

The number of **`vulnpile`** lines in the config_ver5.R file does not match the number of vulnerable subpopulations.

```
Error in file(filename, "r", encoding = encoding) :
  cannot open the connection
In addition: Warning message:
In file(filename, "r", encoding = encoding) :
  cannot open file 'config_ver5.R': No such file or directory
```

You have incorrectly specified the config_ver5.R file name in the pcodControl_ver5.R file **and/or** you have not set the working directory to the correct folder which means that the code cannot find the named config_ver5.R file.

```
Error in readChar(con, 5L, useBytes = TRUE) : cannot open the connection
In addition: Warning message:
In readChar(con, 5L, useBytes = TRUE) :
  cannot open compressed file 'HPsimulation2_Output.rdata', probable reaso
n 'No such file or directory'
```

You have incorrectly specified the output.rdata file name in the RiskCurvePlotting.r file **and/or** you have not set the working directory to the correct folder which means that the code cannot find the named output.rdata file.

```
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
 cannot open file 'MultPilingOpsMultYears3.csv': No such file or directory
```

You have incorrectly specified the piling schedule file name in the config_ver4.R file and/or you have not set the working directory to the correct folder which means that the code cannot find the named piling schedule file.

```
Error in hist.default(data.out[time.point, 2, ], plot = F, breaks =
axis_breaks) : some 'x' not counted; maybe 'breaks' do not span range of
'x'
```

The value that you have set for the minimum and maximum population size range is not correct when running HISTOGRAMS in the summary_statistics.r file.