

# INTERIM PCOD v1.1: A “HOW TO” GUIDE

---

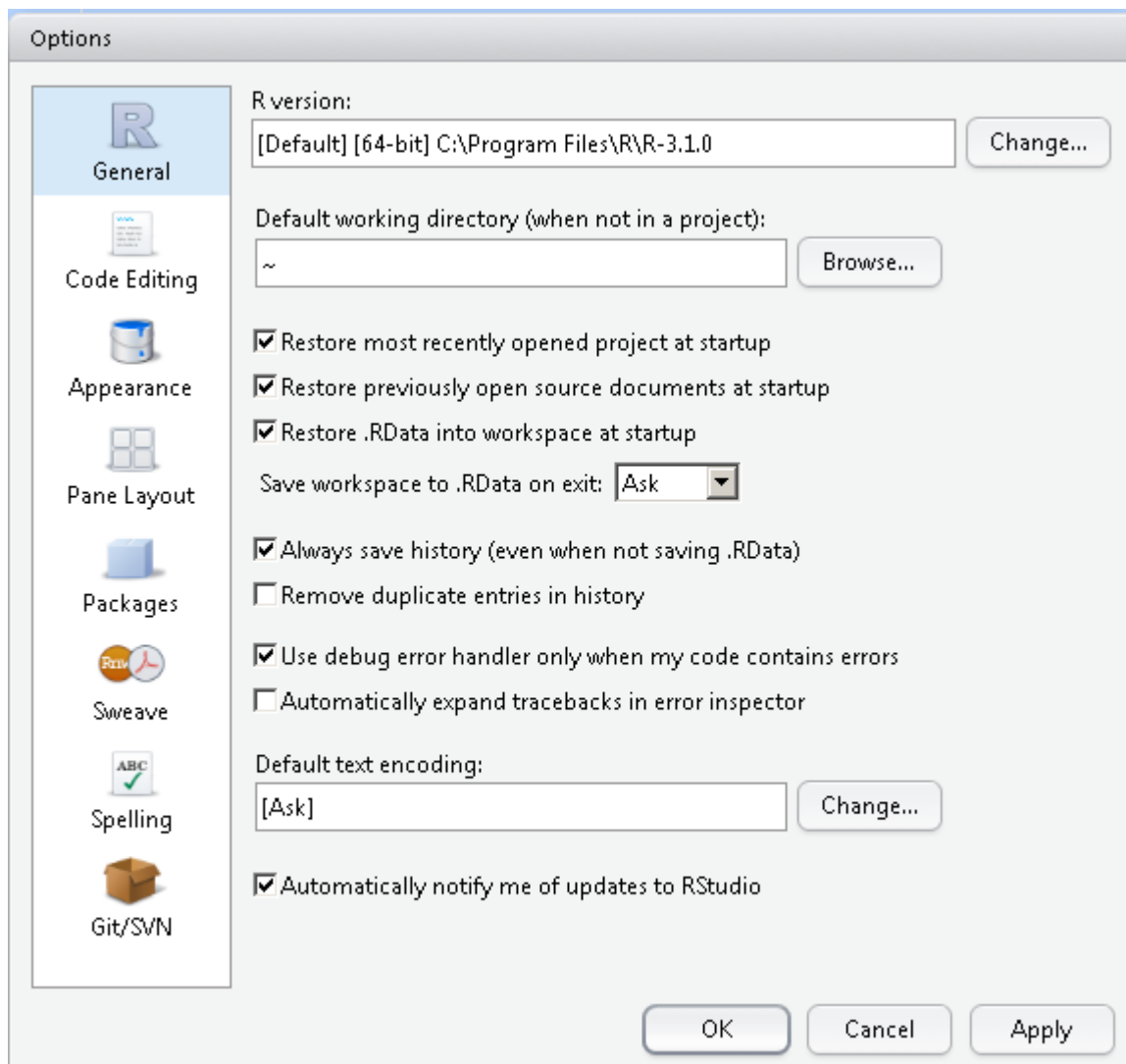
## 1. SETTING UP - THE COMPUTER

### Installing RStudio

1. If you do not have either R or R Studio please install the latest version of RStudio before installing the latest version of R. This can be done by going to the following website: <http://www.rstudio.com/products/rstudio/download/> and selecting the Mac or Windows downloads (e.g. [RStudio 0.98.994 - Windows XP/Vista/7/8](#)).

What follows is the installation for Windows users.

2. Open the RStudio-0.98.994.exe file and click Next
3. Set the install location to C:\Program Files\RStudio (this will require 267MB disk space), click Next.
4. Set the Start Menu folder to RStudio and click Install. RStudio will now install on your computer, when this is complete click Finish.
5. If you already have R installed, please install RStudio as above, but ensure that you have specified that RStudio is using the correct version of R (C:\Program Files\R\R-3.1.1 or 3.1.0):



## Installing R

(N.B. For FAQ on using R on mac OS or Windows OS, please see <http://cran.r-project.org/bin/macosx/RMacOSX-FAQ.html> or <http://cran.r-project.org/bin/windows/base/rw-FAQ.html> respectively).

1. You will then need to install the latest version of R. This can be done by going to the following website: <http://cran.r-project.org/bin/windows/base/> and clicking on [Download R 3.1.1 for Windows](#) (54 megabytes, 32/64 bit).
2. Open the R-3.1.1-win.exe file and click Run.
3. Set the language to English. This will open the Setup Wizard. Click Next.
4. Click Next at the bottom of the Information Ts & Cs window.
5. Ensure the destination location is C:\Program Files\R\R-3.1.1 (this will require at least 1.2MB of free disk space). Click Next.

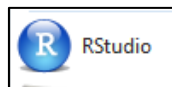
6. When prompted to select which components to install ensure the drop down menu is set to “User Installation” and all 4 boxes are ticked for “Core Files”, “32-bit Files”, “64-bit Files” and “Message Translations” (this will require at least 101MB of disk space), click Next.
7. You will be asked if you want to customize the startup options, select No (accept defaults), click Next.
8. The setup will create a shortcut in the Start Menu, make sure it is called R and click Next.
9. Tick all three boxes for additional tasks and click Next. R will now install on your computer, when this is complete click Finish.

## Setting up the InterimPCoD Folder

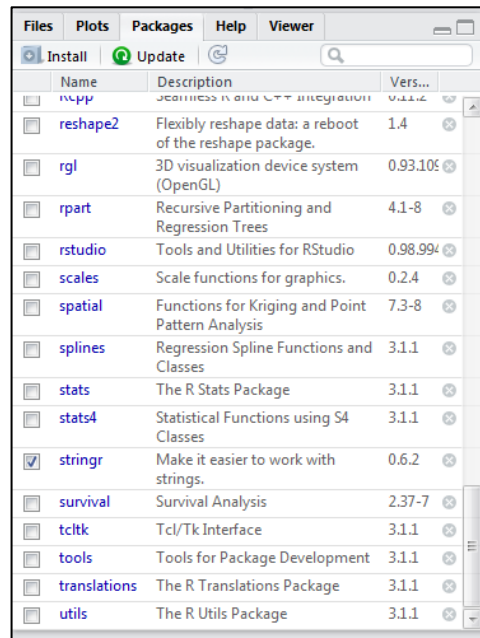
1. Copy the folder InterimPCoDCoDev1.1 to your C drive. This path can be whatever you choose – the important thing is that you know the route to the directory/folder which contains the relevant files from this zip folder. (see next point).
2. We’d suggest you create a new folder called “RRun” (please note you can name the path as you wish, but you will need to change any specified paths from what we outline below).
3. Open this new folder and create a new folder within it called “InterimPCoD”.
4. Copy all the files from the C:\InterimPCoDCoDev1.0 into this new folder – e.g. C:\RRun\InterimPCoD.

## Setting up RStudio

N.B. This is based on your machine running a 64-bit version of Windows OS. If you are unsure, please follow: <http://support.microsoft.com/kb/827218>.



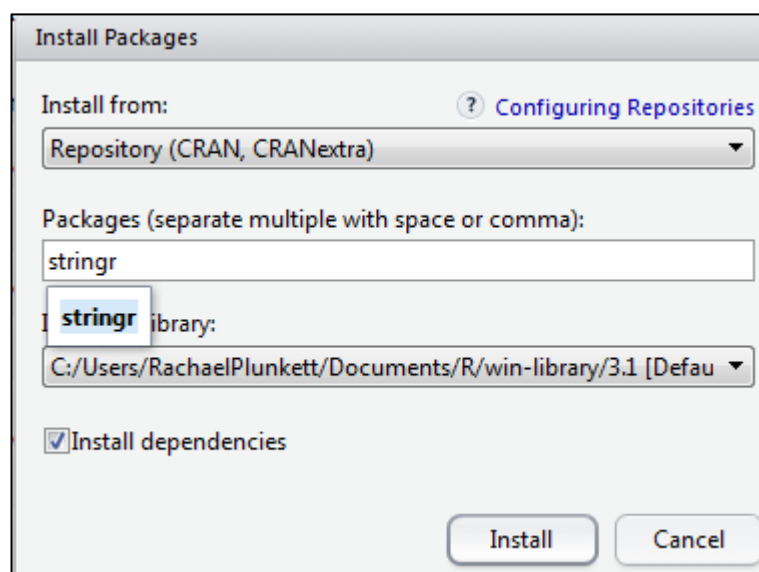
1. Open the Start Menu and open RStudio.
2. You will be prompted to select the version of R to use. Click “Use your machine’s default version of R64 (64bit)” and click “OK”. This will open RStudio.
3. You will now need to download the following libraries (see steps 4 & 5):
  - stringr
  - ggplot2
  - matrixStats
4. This can be done by clicking “Packages” in the bottom right hand box in RStudio.



5. Tick the boxes for `stringr`, `ggplot`, `matrixStats`. This will cause the following to appear in the RStudio Console:

```
> library("stringr", 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("ggplot2", lib.loc="~/R/R-3.1.1/library")
```

6. If you do not have these packages set up already in the user Library then you can click on the tab "Tools" at the top of the screen and selecting "Install Packages".
7. Install from: Repository (CRAN, CRANextra). Type `stringr` in the Packages box and click Install.



8. This will cause the following to appear in the RStudio Console:

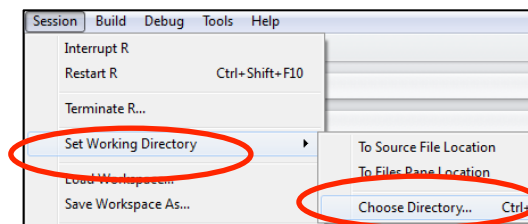
```
> install.packages("stringr")
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/stringr_0.6.2.
zip'
Content type 'application/zip' length 74918 bytes (73 Kb)
opened URL
downloaded 73 kb
```

package 'stringr' successfully unpacked and MD5 sums checked

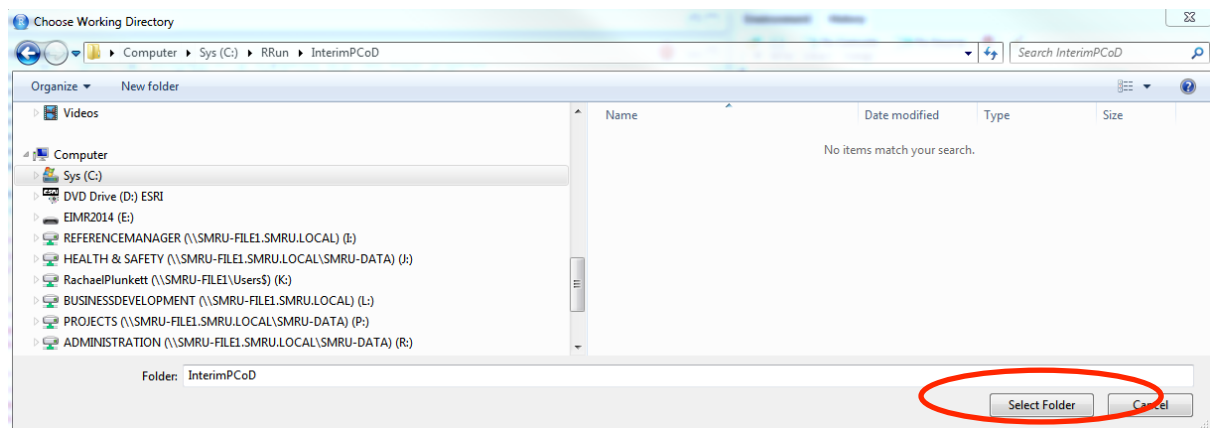
The downloaded binary packages are in  
C:\Users\Cormac Booth\AppData\Local\Temp\RtmpSAEfwh\downloaded\_packages

9. Repeat the process for the other libraries: ggplot2, matrixStats (you will also need grid, but this should be pre-installed and listed under System Library in the "Packages" window in the bottom right of RStudio)

10. Go to the tab "Session" at the top of the screen and click Set Working Directory then Choose Directory.



11. Navigate to C:\RRun\InterimPCoD, open this 'InterimPCoD' folder. Click 'Select Folder'.



12. This will cause the following to appear in the RStudio Console:

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

It is essential that you perform steps 10 and 11 at the start of every session.

You should now be ready to start running the interim PCOD files in R Studio.

## 2. SETTING UP THE PILING SCHEDULE

NORMALLY, YOU WOULD HAVE TO SET UP A FILE INDICATING ON WHAT DAYS PILING IS LIKELY TO OCCUR OVER THE ENTIRE COURSE OF CONSTRUCTION FOR A WIND FARM. THIS IS AN EASY, IF BORING, PROCESS FOR ANYONE FAMILIER WITH EXCEL. THE FOLLOWING INSTRUCTIONS TELL YOU HOW TO DO THIS. IF YOU ARE USING THE InterimPCoD SOFTWARE FOR THE FIRST TIME, WE HAVE PROVIDED YOU WITH A READYMADE PILING FILE THAT YOU CAN PRACTICE WITH. PLEASE GO TO SECTION 3.

**In order to run the Interim PCoD code it is necessary to prepare a schedule of information on the estimated days on which activity (e.g. piling or turbine operation) are expected to take place for each development.**

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.

**IMPORTANT: The start date for the timetable should be the beginning of the breeding season for the marine mammal species being modelled. This is 1 October for grey seals and 1 June for all other species. Remember to use the template for the species you are using (i.e. suffix 'GS' for grey seals, default for all others).**

Use this information to populate the Excel template `MultPilingOpsMultYears.csv` - this can be found in the `InterimPCoDCoDev1.0` folder you sourced from the MSS website.

1. Navigate to `C:\RRun\InterimPCoD` and open the csv file: **MultPilingOpsMultYears**

In this .csv document, there are at least 3 default columns:

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

**DayOfYear** – an integer which goes from 1 (the first row) to X depending on how many days you have in your calendar. The number of rows in your piling schedule must be at least a full year and be therefore divisible by 365 (i.e. last value in DayOfYear = 730, 1095, 1460, etc.)

**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 – where there is more than one development. Note: Two or more operations could be from a single development (e.g. if there are two piling vessels working on one site).

In this default template this gives a schedule for the piling activity for a development. The piling is set to start on 01/06/2050 and run every day for 2 years until 30/05/2052. Three development piling operation sites are going to occur during this 2 year time period hence the 3 Operations columns.

Each date for each Operation has either a 1 or a 0 in in, representing either piling or no piling scheduled that day.

	A	B	C	D	E
1	Date	DayOfYear	Operation1	Operation2	Operation3
2	01/06/2050	1	0	0	0
3	02/06/2050	2	0	0	0
4	03/06/2050	3	0	0	0
5	04/06/2050	4	0	0	0

Sometimes piling is only scheduled at one Operation site in one day. For example - on 15/08/2050 and 16/08/2050 piling is only scheduled at Operation site 3 - below.

	A	B	C	D	E
1	Date	DayOfYear	Operation1	Operation2	Operation3
77	15/08/2050	76	0	0	1
78	16/08/2050	77	0	0	1

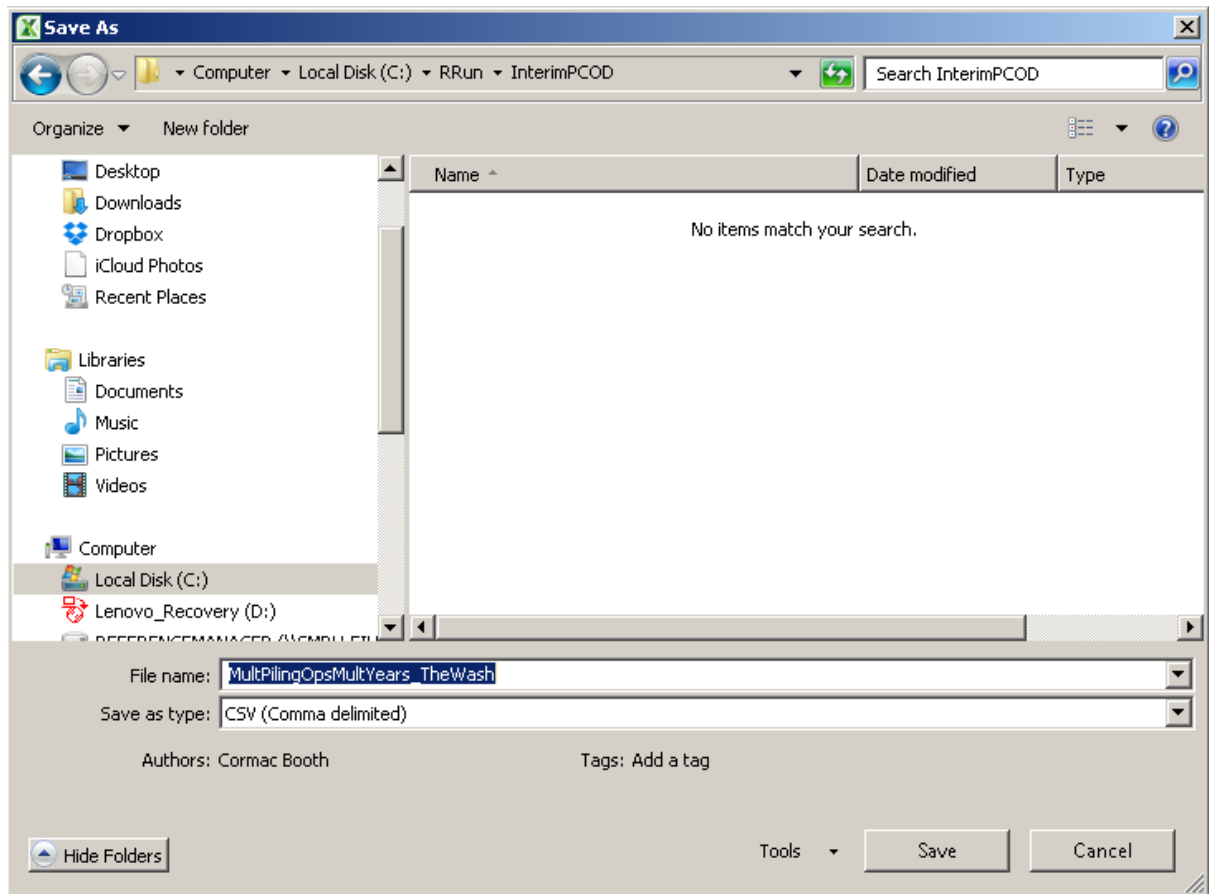
While at other times piling can be schedules to occur simultaneously at 2 or 3 operation sites in one day. For example:

	A	B	C	D	E
1	Date	DayOfYear	Operation1	Operation2	Operation3
154	31/10/2050	153	1	0	1
155	01/11/2050	154	1	1	1
156	02/11/2050	155	0	1	1
157	03/11/2050	156	0	1	1
158	04/11/2050	157	1	1	0

2. You can edit this file to reflect the number of piling operation sites and the scheduled days for your particular project.

**IMPORTANT: The number of rows in your piling schedule must be at least a full year and be therefore divisible by 365 (i.e. last value in DayOfYear = 730, 1095, 1460, etc.). If you use the drag facility to create extra years, remember that Excel will automatically create an extra day for 29 February in all leap years. You will need to remove these days, otherwise the Interim PCoD code will give you an error.**

3. We'd recommend you save this file under a different name – for example: MultPilingOpsMultiYears\_TheWash.csv and save it in the C:\RRun\InterimPCoD folder. Also ensure the file type is .csv not.xls.



4. Next we will set-up our MakeInit files. This is the file in which we specify the piling schedule to be used in the simulations.



### 3. SETTING UP A 'MakeInit....R' FILES

In order to carry out this step, you will need to compile (or calculate) the relevant information to run in your simulations. You will need to:

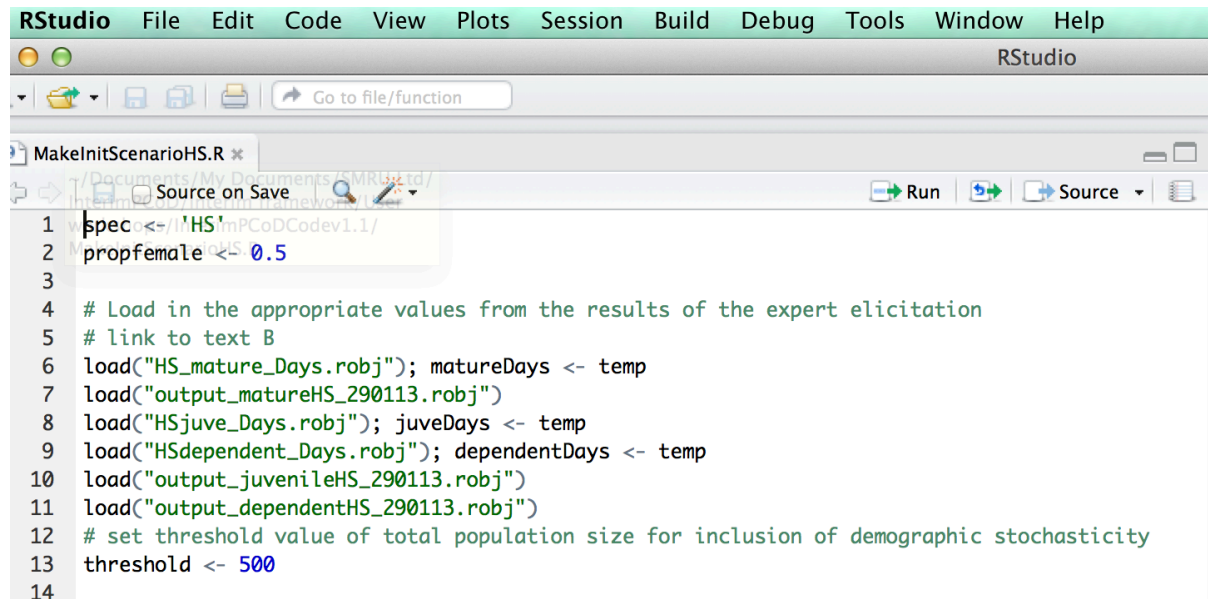
1. Identify the marine mammal Management Unit (MU) that may be affected by each development.
2. For each MU, look up the estimate for the current size of the population in that MU in the report prepared by Defra's Inter-Agency Marine Mammal Working Group (you can obtain a copy of this from Sonia Mendes in JNCC).
3. Look up the appropriate values for the key demographic rates for this population in Harwood & King 2013, which is included in the Directory you have downloaded:
  - 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. average age at which females give birth for the first time, and
  - e. fertility rate (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).
6. Decide on an appropriate range of values for the number of days of 'residual' disturbance associated with one day of actual disturbance

**For this worked example we will use:**

<b>Species</b>	<b>Harbour Seals</b>
<b>Management Unit</b>	<b>South East England (The Wash)</b>
<b>Population Size</b>	<b>4568</b>
<b>Pup Survival Rate</b>	<b>0.6</b>
<b>Juvenile Survival Rate</b>	<b>0.822</b>
<b>Adult Survival Rate</b>	<b>0.85</b>
<b>Age at first birth</b>	<b>4</b>
<b>Fertility Rate</b>	<b>0.95</b>
<b>Vulnerable population</b>	<b>1</b>
<b>Residual days of disturbance</b>	<b>1</b>

1. In RStudio click “File” then “Open File ..”. Select the folder \RRun\InterimPCoD and click on the R file MakeInitScenarioHS\_Concurrent. This will open a new window in RStudio with the contents of the MakeInitScenarioHS\_Concurrent R file.

**Note: HS stands for Harbour Seal.**



```

RStudio File Edit Code View Plots Session Build Debug Tools Window Help
Go to file/function
MakeInitScenarioHS.R
Source on Save Run Source
1 spec <- 'HS'
2 propfemale <- 0.5
3
4 # Load in the appropriate values from the results of the expert elicitation
5 # link to text B
6 load("HS_mature_Days.robj"); matureDays <- temp
7 load("output_matureHS_290113.robj")
8 load("HSjuve_Days.robj"); juveDays <- temp
9 load("HSdependent_Days.robj"); dependentDays <- temp
10 load("output_juvenileHS_290113.robj")
11 load("output_dependentHS_290113.robj")
12 # set threshold value of total population size for inclusion of demographic stochasticity
13 threshold <- 500
14

```

2. Ensure that the variable spec is set to HS, for harbour seal.

```

1 spec <- 'HS'
2 propfemale <- 0.5

```

Line 2 sets the proportion of the population that is female to 0.5.

Next, set the size of the population. Use the appropriate value from the IAMMAWG Management Unit (MU). For this example, we will use the South East England (The Wash) harbour seal MU, which has a population size of 4568.

3. Type 4568 into the code to right of ‘pmean <- ‘

```

15 # iPCoD PROTOCOL STEP 2
16
17 pmean <- 4568 # population size value from the IAMMAWG MU report should be used for the MU

```

So far, we have set the population mean size to 4568 and half of that population to be female..

4. Next we enter values for survival and fertility. Suggested values for each MU can be found in Harwood & King (2013]. **You will need to change the survival rates, fertility rates, age1 and age2 to match the code below for our South East England (The Wash) Population:**

```

27 # iPCoD PROTOCOL STEP 2
28 # INPUT DEMOGRAPHIC RATES FROM HARWOOD & KING (2014) HERE
29
30 Surv[c(1, 7, 13)] <- c(0.6, 0.82, 0.85)
31 Fertility <- 0.95
32
33 # age1 = age at which a calf becomes independent from its mother, default is 1
34 age1 <- 1
35
36 # age2 = age at which a female give birth to her first calf
37 age2 <- 4

```

**Surv[1]**, **Surv[7]** and **Surv[13]** contain the mean survival rates for pups or calves (i.e. animals that are still dependent on their mothers), juveniles and adults, respectively.

5. In the **Surv** line, edit **<- c(0.6, 0.822, 0.85)**, if necessary, to set the pup survival to 0.6, the juvenile survival to 0.822 and the adult survival to 0.85.
6. **Fertility <- 0.95** sets the adult female fertility to 0.95
7. **age1 <- 1** sets the age at which a pup becomes independent from its mother to 1 year (these is the default for HS and won't need to be changed).
8. **age2 <- 4** sets the age at which a female give birth to her first calf to 4 years (these is the default for HS and won't need to be changed).

Users can specify that only a proportion of the population within an MU is likely to spend time in the region around a particular development where sound exposure levels are sufficiently high that they will cause a behavioural response or injury. We refer to these animals as being members of a **vulnerable sub-population**. We set the vulnerable sub-populations with the following code:

```

79 # iPCoD PROTOCOL STEP 3
80
81 # INPUT PROPORTIONS OF POPULATION IN VULNERABLE SUB-POPULATIONS
82 # DEFAULT: entire population in one sub-population)
83
84 vulnmean <- c(1.0)

```

**vulnmean** is the proportion of the total population in each vulnerable sub-population. It is possible to identify many vulnerable sub-populations; **the only constraint is that the sum of the values specified must be less than or equal to 1.0**. The following example specifies 3 sub-populations

```
vulnmean <- c(0.2, 0.2, 0.3)
```

This indicates that the first and second sub-populations each contain 20% of the total population, and the third sub-population contains 30% of that population. The remaining 30% of the population are assumed to be unaffected by any of the developments being modelled.

**NOTE: If you are using the Interim PCoD code to model the potential effects of collision or entanglement with tidal turbines or wave energy devices, you cannot divide the population into vulnerable sub-populations, so leave vulnmean set to c(1.0). You will also need to change this line:**

```

91 # Set number of years on which piling will occur.
92 # Set this to zero if there is no piling or you only want to model the effect of collisions
93 pile_years <- 1

```

```
to
pile_years <- 0
```

and then go to section 19.

- Next we need to specify the piling schedule we want to use. This is a timetable of the estimated days on which different activities (e.g. piling or turbine operation) are expected to take place for each development. We will use `MultPilingOpsMultYears_Concurrent.csv`, which is supplied with the Interim PCoD code and should be visible if you click on “File”, then “Open File ..”.

**Note: If you have created your own Piling Schedule csv file, change the name between the quotation marks so that it matches the name of your file.**

```
97 # iPCoD PROTOCOL STEP 4
98
99 # Specify name of csv file that contains information on days on which piling will occur
100 # the name should be placed between the quotation marks: " "
101 # there is no need to change this line if you have set pile_years to zero
102 pile <- read.csv(file = 'MultPilingOpsMultYears.csv', header = TRUE) ## XXX strip out date
...
```

- Highlight line 102, shown in the screen-shot above, and hit “Run”. This will enter the highlighted code into the bottom left box in R Studio, which is known as the R “Console”. We can now check if the new piling data has been imported correctly by typing `head(pile)` at the `>` prompt in the R Console and pressing Return. This will give the headings and first 6 rows of the data in the Piling Schedule file. This is also a good way to check that the piling schedule starts at the correct date (1 June for harbour seals). **Be careful to un-highlight line 102 when you return to the main RStudio window, otherwise you may accidentally delete it!**

```
> head(pile)
      Date DayOfYear Operation1 Operation2 Operation3
1 01/06/2050         1         0         0         0
2 02/06/2050         2         0         0         0
3 03/06/2050         3         0         0         0
4 04/06/2050         4         0         0         0
5 05/06/2050         5         0         0         0
6 06/06/2050         6         0         0         0
```

- The file `MultPilingOpsMultYears_Concurrent.csv` contains information on 3 piling Operations, so the code in line 123 should read: `pilesx1 <- 3`. If you’ve created a piling schedule with a different number of Operations, please specify the correct number here.

```
119 # iPCoD PROTOCOL STEP 5
120
121 # input number of piling operations to be modelled (3 in this case)
122
123 pilesx1 <- 3
...
```

`vulnpile` is a matrix of index values (1’s or 0’s) indicating which Operations will affect each vulnerable sub-populations.

```

129 # iPCoD PROTOCOL STEP 6
130
131 # Indicate which Operations will affect each vulnerable sub-population
132 # A separate line is required for each vulnerable sub-population.
133 # Add vulnpile[2,] <- c(,,) if there are two vulnerable sub-populations,
134 # vulnpile[3,] <- c(,,), if there are three vulnerable sub-population, and so on.
135 # In this example, animals in the 1st sub-population are vulnerable to the
136 # effects of piling Operation 1 and piling Operations 2.
137
138 vulnpile[1, ] <- c(1, 1, 1)

```

12. Indicate which operations will affect each vulnerable sub-population. In our worked example, using the `MultPilingOpsMultYears_Concurrent.csv` file we assume that the vulnerable sub-population is affected by all three piling operations, so the code should read:

```
vulnpile[1, ] <- c(1, 1, 1).
```

If you think that only piling Operations 1 and 3 will impact the seals in this population, you should change this line to:

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

If you specify more than one vulnerable sub-population in Step 8., you will need to enter additional lines of code for each of these sub-populations. For example, if you decided that there are actually two vulnerable sub-populations, and that sub-population 1 is impacted by piling Operations 1 and 2 but not Operation 3, while sub-population 2 is impacted by piling Operations 2 and 3 but not Operation 1, you would need the following code:

```

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

```

It is possible to allow the number of animals that are predicted to be disturbed or suffer PTS as a result of the same piling Operation to vary between seasons. the following section explains how to do this. However, such information is rarely available. Please check with John Harwood ([jh17@st-andrews.ac.uk](mailto:jh17@st-andrews.ac.uk)) if you want to use this capability.

The variable **seasons** determines whether or not the number of animals that are likely to be disturbed each day (NDt) and the number that may experience PTS (NPt) **by the same piling Operation** vary by season (spring, summer, etc.). If

- seasons = 1, there is no seasonal variation, this is the default value;
- seasons = 2, there are 2 seasons in the year “summer” = May - October (ie the months when water is warmest), and “winter”.
- seasons = 4, there are 4 seasons in a year. summer = June, July, August; autumn = Sept, Oct, Nov; winter = Dec, Jan, Feb; spring = March, April, May
- NOTE: the simulated year for harbour seals and all cetaceans starts in June, for grey seals it starts in October.

For our worked example we have set `seasons <- 1`, indicating that we are assuming there is no seasonal variation in the number of animals that are likely to be disturbed or experience PTS.

```

152 seasons <- 1
153 inputindex <- seasons
154 if(seasons == 2) {
155     seasons <- seasons + 1
156     breaks1 <- c(152,182,31)
157 } else {
158     breaks1 <- c(91,92,91,91)
159 }
160
161 if(seasons == 1){s_index <- c(365)} else {s_index <- breaks1}

```

If you wanted the number of animals **affected by a particular piling Operation** to vary between all 4 seasons in the year the code would look like this:

```
seasons <- 4
```

But please note, you need to ensure you have breaks specified correctly.

**daily\_NDt** stores estimates of the number of individuals of the species being modelled that may experience disturbance that is likely to impair their ability to survive, breed, reproduce, or raise young during one day of construction for each operation.

**daily\_NPt** stores the estimates of the number of animals that may experience a permanent shift in the threshold for hearing (PTS) during one day of construction for each operation.

```

170 # iPCoD PROTOCOL STEP 6
171 # input number of animals predicted to experience significant disturbance on one day of piling
172 # the first 1:inputindex values in the string give the number of animals that will be affected
173 # for piling operation 1, the next (inputindex+1):(2*inputindex) values give the number of ani
174 # that will be affected in each season for piling operation 2, and so on
175
176 daily_NDt[1:inputindex,] <- c(60, 60, 60)
177
178 # now do the same for the number of animals predicted to experience PTS on one day of piling
179
180 daily_NPt[1:inputindex,] <- c(2, 2, 2)

```

In the above code, you will need to change the values in line 176 and 180 to the number of animals predicted to experiencing disturbance and PTS for each operation (we have three operations in this example).

So, for example, if 60 seals are predicted to experience disturbance and 2 seals are predicted to experience PTS as a result of one day of activity at each operation the code would look like:

```

daily_NDt[1:inputindex,] <- c(60, 60, 60)
daily_NPt[1:inputindex,] <- c(2, 2, 2)

```

Different operations can be coded to have different impacts. For example, if Operation 3 is predicted to disturb 80 seals and cause PTS for 3 seals the code would look like:

```

daily_NDt[1:inputindex,] <- c(60, 60, 80)
daily_NPt[1:inputindex,] <- c(2, 2, 3)

```

15. Input the number of animals to experience disturbance and PTS. In this example, we will assume that each of the 3 Operations causes disturbance to 690 seals and PTS to 1 seal:

```
daily_NDt[1:inputindex,] <- c(690, 690, 690)
daily_NPt[1:inputindex,] <- c(1, 1, 1)
```

16. Next we enter values for the number of days a seal is likely to be excluded from the area around a development after it has been disturbed.

```
215 # iPCoD PROTOCOL STEP 7
216
217 # input number of days of "residual" disturbance. DEFAULT IS 1
218
219 days <- 1
```

In the above code, seals are assumed to be displaced from the area for an additional day after being disturbed.

17. The interim PCoD model allows for a scenario in which an animal will avoid the area immediately around a development after it has been disturbed for the first time. As a result it will not experience sound levels likely to cause PTS after that occasion. The default setting is:

```
222 # iPCoD PROTOCOL STEP 8
223
224 # decide if PTS can occur on any day (default) or only on the first occasion that an individual
225 # change Day1 to TRUE if you want animals to be only vulnerable to PTS on the first day they are
226
227 Day1 = TRUE # Flag for the different PTS model
```

which assumes that animals are at risk of PTS every time they enter the region around a development where they may be disturbed by construction noise.

If this is changed to:

Day1 = TRUE

animals are predicted to avoid the immediate vicinity of an operation after they have been disturbed once.

Next we set the number of animals that may be killed by collisions with devices **during one year of operation**:

```
231 # iPCoD PROTOCOL STEP 9
232
233 # input number of animals predicted to be killed each year as a result of collisions
234 NCollisions <- 0
```

18. Replace the value for NCollisions with the number of animals you estimate may be killed by collisions with wave/tidal devices over the course of one year.

The final step is to set the number of years over which the simulation will run. The default is 25 years, but this can be changed as long as the new value is greater than the number of years over which piling is estimated to occur:

```
239 #number of years for simulation
240 years <- 25
```



## 4. SETTING UP – ‘PopDyn.R’ FILES

Before starting this section, please ensure that the `MakeInitScenarioHS` and piling schedule (`MultiPilingOpsMultiYears_TheWash.csv`) have been edited as outlined in the above section and the edited version has been saved in the same Working Directory. We will now edit the main processing code `PopDyn.R`.

1. Open RStudio. Go to File, Open File. Navigate to `C:\RRRun\InterimPCoD` (or where you have stored the code and files) and select the R file ‘PopDyn’ and click Open.
2. for our worked example, change the name of the file specified in line 7 to ‘`MakeInitScenarioHS.r`’, or whatever name you have chosen for your MakeInit file:

```
1 rm(list = ls())
2
3 # iPCoD PROTOCOL STEP 10
4 # Set the name of the MakeInit file you want to use
5
6 source('PCoDFunctionsFinal.r')
7 source('MakeInitScenarioHP_NL_Spring_622.r')
8 # Set the number of times you want to run the simulation
9 nboot <- 500
10
```

**Note:** You should never alter line 7; `PopDyn.R` must always source the `PCoDFunctionsFinal.r` file.

3. Change the value `nboot` to the number of times you wish to run the simulation. The default is 1000, and we would not advocate a lower number for an impact assessment. **For the purposes of this training session you should set `nboot` to 10, as it will take too long to run 1000 simulations.**

**Note:** The more simulations you choose to run, the longer the program will take to complete. The time for the simulations to run will also depend on the number of Operations and the number of vulnerable sub-populations you have specified in your MakeInit files.

4. Chose a name for the output file and type it between the quotation marks and before `Output_file` in line 253 .

```
253 save(vulnmean, days, nboot, Ndist_out_1, Ndist_out_2, dds, Surv, Fert, dat.out, file = paste(spec, '_Output_file', rdata', sep = ''))
```

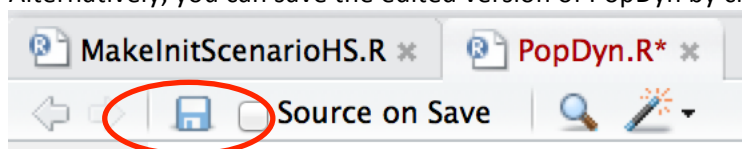
5. Now highlight the entire script you just edited and hit “Run”

```

1 rm(list = ls())
2
3
4 source('PCoDFunctionsFinal.r')
5 source('MakeInitScenarioHS_Sequential.R')
6
7 # set threshold value of total population size (pmean) for inclusion of demographic stochasticity
8 threshold <- 3000
9
10 # Initiate the Bootstrap
11 nboot <- 500#0
12
13 # this will only store the final values for Ndistsum and NNotdistsum, we want the values for each year,
14 # so data.out needs to be an array with dimensions nboot x x x 2
15
16 dat.out <- array(NA, dim = c(years, 4, nboot))
17
18 # Ndistsum is the total population size for the disturbed population each year, NNotdistsum is the
19 # equivalent for the matching undisturbed population
20 colnames(dat.out) <- c('Ndistsum', 'NNotdistsum', 'decline 1st', '% decline 1')
21
22 # store row/element subscripts for undisturbed age classes
23 dist_ages <- seq(1, 55, 6)
24
25 # matrices to store numbers affected in different ways in a more accessible format than Ndistsum for last
26 # iteration of bootstrap.
27 Ndists_out_1 <- Ndists_out_2 <- array(0, dim = c(10, 6, pile_years))

```

6. Alternatively, you can save the edited version of PopDyn by clicking on the Save symbol



typing

`source('PopDyn.R')`

in the R Console and then hitting Return

```

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.

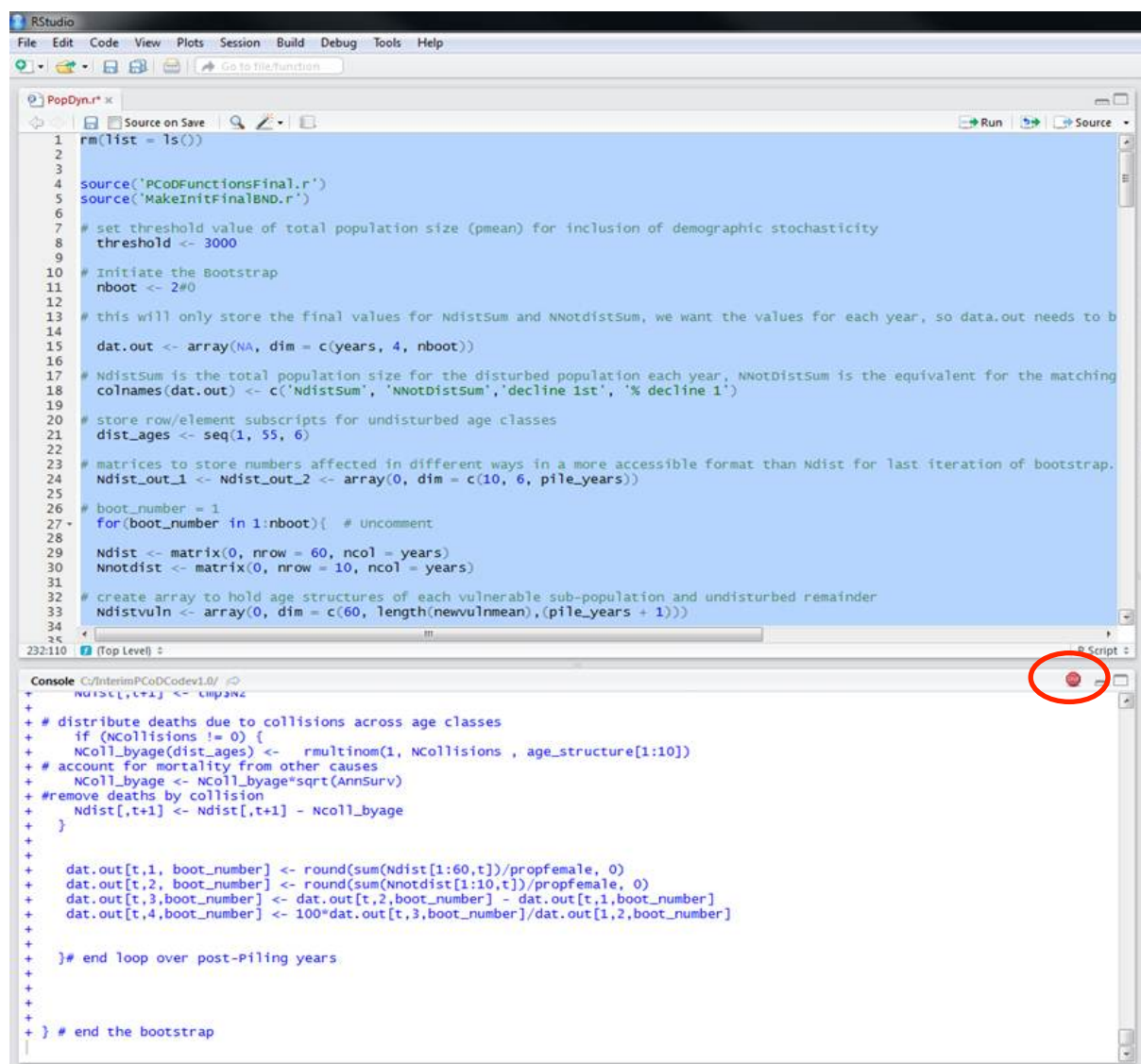
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

[workspace loaded from ~/.RData]
> source('PopDyn_H5.R')

```

While the code is running the screen will freeze, and the “Stop” sign (see below) will glow red:



If you want to end the simulation prematurely, you can hit the Stop sign. However, no output will be produced.

Once the PopDyn code has finished running the Stop sign will disappear. The PopDyn code will have created a new file in your C:\RRRun\InterimPCoD folder called HS\_??????OutputFile.rdata.

where ?????? is the name you chose in Step 4. This OutputFile can be used to create plots of the results in the next section.

## 1. PRODUCING OUTPUTS PLOTS

Once you have run the PopDyn simulation for a particular scenario, R will have generated an output file and deposited it into your working directory. It will have the format 'spec'\_?????OutputFile.Rdata' (spec = the species code, e.g. HS or HP, ????? indicates the name you chose for the file).

We have provided two plotting options that summarize the results of the simulations.

### MedPopDiff

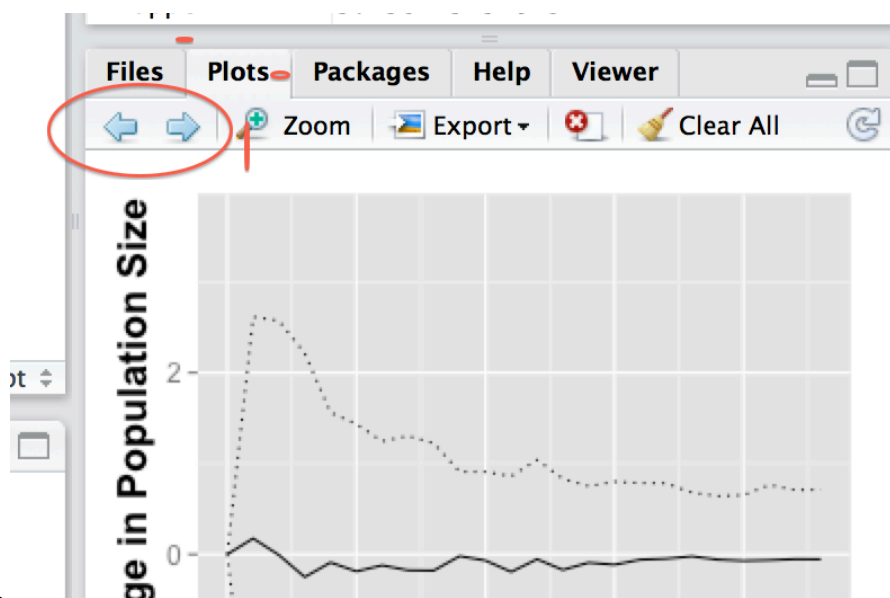
This will calculate and display the median difference between the disturbed and undisturbed populations each year, the annual percentage change in the size of the disturbed population, and the annual percentage change in the undisturbed population, with an indication of the range in each case.

1. Open the file in the InterimPCoD folder called MedPopDiff.R in R Studio.
2. Type in the name of the Output.rdata file you wish to analyse: e.g.

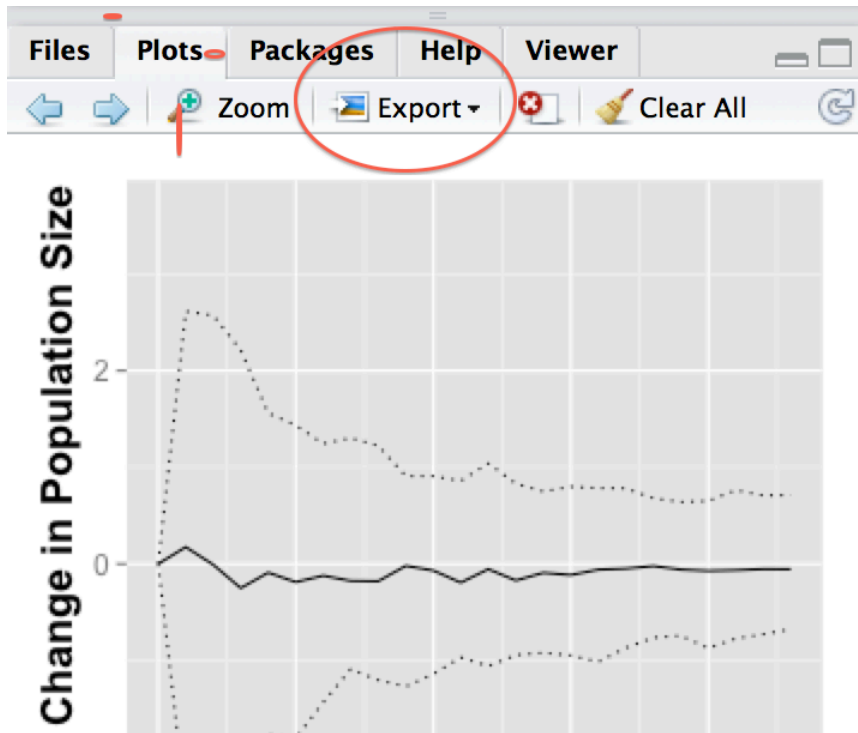
```
7 load('HP_Output_NL_Spring_622.rdata')
```

Highlight the rest of the code in the file and click on "Run"

The output will be a series of figures that will appear in the "Plots" window at the bottom right of your screen. You can view each of them by clicking on the left and right arrows in the top left hand corner of this screen.



You can right click on each figure, and then choose how you want to save it by clicking on the "Export" button. **Note that R does not support file names that contain blanks, so use the underscore character instead**



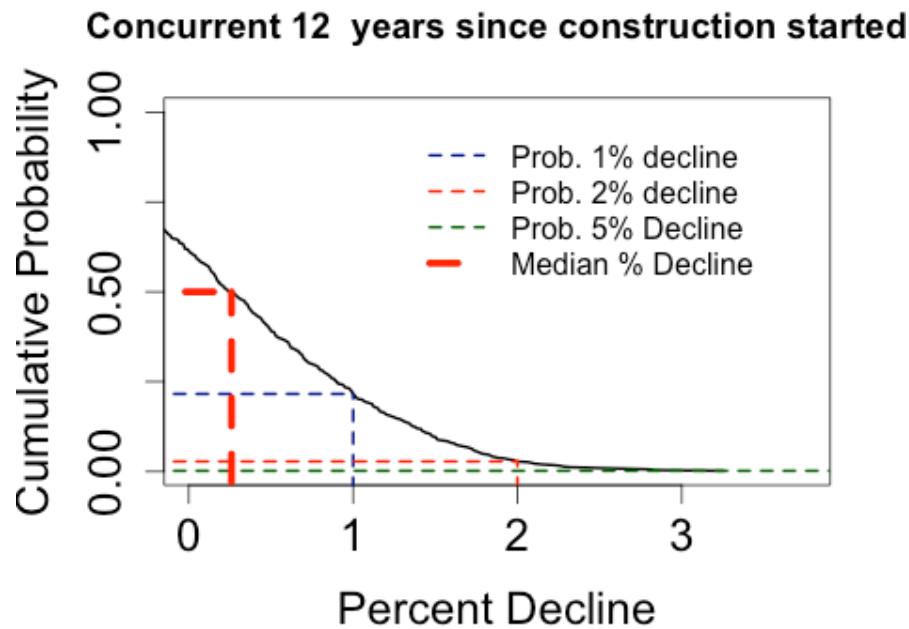
### RiskCurve Plotting

This program will calculate the cumulative probability of particular percentage **annual** declines in the population 1 year, 6 years, 12 years and 18 years after the start of piling.

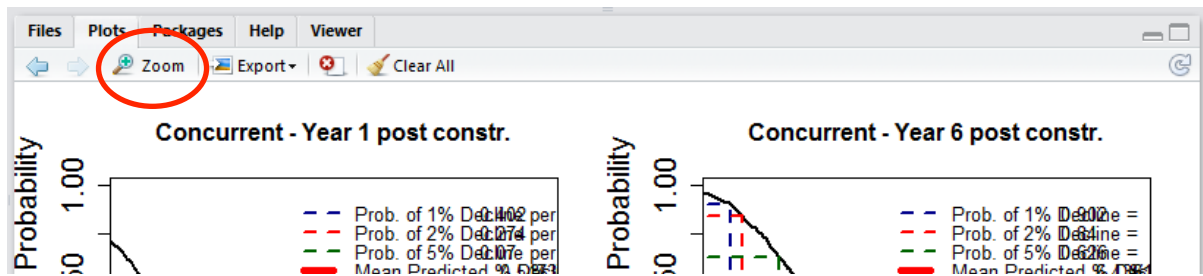
1. Open the file called `RiskCurve.R` in R Studio.
2. Type in the name of the `Output.rdata` file you want to summarise. e.g.:

```
3 par(mfrow=c(1,1))
4 load('HP_Output_NL_Spring_622.rdata')
```

3. Highlight all the code and click “Run”. This will create a series of graphs in the “Plots” window that will look like this:



If you are not happy with the way the figure is formatted, you can modify it by clicking on the “Zoom” button.



4. This opens “Plots” in a new window. Adjust the size of the window until you are happy with the layout and none of the text overlaps. Now you can take a screen shot or save the plot as an image file in your working directory.
5. You can also flip through the different graphs using the arrow buttons and save individual graphs using the “Export” button.
6. The code also calculates the exact risks of 1%, 2% and 5% annual declines for the disturbed and undisturbed populations. You can view these by typing `risk_dist` or `risk_notdist` in the R CONsole. However, these values are also saved into a .csv whose name you can chose in line 122

```
write.table(MainOutput,col.names=F,row.names=F, sep="," , file = 'HP_Output_NL_Spring_13736.csv'
```

## Appendix 1 - MakeInitScenario Example Code

Here is the entire R code for the MakeInitScenarioHS file. **The variables highlighted in yellow or green are the ONLY ones the user should change . The variables highlighted in green are the ones you are most likely to want to vary.** It is essential that you use the correct 'MakeInitScenarioXX.R' file for the species you are modelling as a starting template for your own 'MakeInit...R' files.

```
spec <- 'HS'
propfemale <- 0.5

# Load in the appropriate values from the results of the expert elicitation
# link to text B
load("HS_mature_Days.robj"); matureDays <- temp
load("output_matureHS_290113.robj")
load("HSjuve_Days.robj"); juveDays <- temp
load("HSdependent_Days.robj"); dependentDays <- temp
load("output_juvenileHS_290113.robj")
load("output_dependentHS_290113.robj")
# set threshold value of total population size for inclusion of demographic
stochasticity
threshold <- 500

# iPCoD PROTOCOL STEP 2

pmean <- 4568 # population size value from the IAMMAWG MU report should be
used for the MU being modelled
pmean <- round(pmean*propfemale)

Surv <- rep(0, 18)

# iPCoD PROTOCOL STEP 2
# INPUT DEMOGRAPHIC RATES FROM HARWOOD & KING (2014) HERE

Surv[c(1, 7, 13)] <- c(0.6, 0.82, 0.85)
Fertility <- 0.95

# age1 = age at which a calf becomes independent from its mother, default
is 1
age1 <- 1

# age2 = age at which a female give birth to her first calf
age2 <- 4

Fert <- rep(0, 6); Fert[1] <- Fertility/2.0

# determine initial stable age structure for population from Leslie matrix
```

```

L          <- array(0, dim = c(10, 10))
L[1, age2] <- Fert[1] * Surv[7]
L[1, (age2 + 1):10] <- Fert[1] * Surv[13]
index3     <- array(c(2:(age1+1), 1:age1), dim = c( age1 ,2) )
L[index3]   <- Surv[1]
mature     <- ifelse(age2 == 9, 9, age2 + 1)
index1     <- array(c((mature + 1):10, (mature):9), dim = c((10 -
mature), 2))
L[index1]   <- Surv[13]
L[10, 10]  <- Surv[13]
index2     <- array(c((age1+2):(age2 + 1), (age1+1):age2), dim =
c((age2 - age1), 2))
L[index2]   <- Surv[7]

ev          <- eigen(L)
# ev$val[1] contains growth rate of the undisturbed population

evCmplx     <- ev # in case we need this later on
# age_structure holds the proportion of animals in each age class for a
stable age structure
age_structure <- as.numeric(ev$vec[1:10, 1] / (sum(ev$vec[1:10, 1])))

# set % level of environmental variation for calf survival, juvenile
survival and fertility from expert elicitation results
EnvStoch <- c(30, 30, 25)
# calculate standard deviations that will result in lower 99% CL matching
environmental variation
SDs <- EnvStoch/(3*100)
mu <- Surv[c(1,7)]
mu[3] <- Fertility
# calculate parameters of Beta distribution to match these values for mu
and SD
a <- 0
b <- 0
for (ibeta in 1:3){
  b[ibeta] <- mu[ibeta]*(1-mu[ibeta])^2/SDs[ibeta]^2 + mu[ibeta] - 1
  a[ibeta] <- mu[ibeta]*b[ibeta]/(1-mu[ibeta])
}

# iPCoD PROTOCOL STEP 3

# INPUT PROPORTIONS OF POPULATION IN VULNERABLE SUB-POPULATIONS
# DEFAULT: entire population in one sub-population)

vulnmean <- c(1.0)

nvulnmean <- length(vulnmean)

```



```

# newvulnmean includes proportion of animals in undisturbed remainder of
population, if there is one!
newvulnmean <- vulnmean
if(sum(vulnmean)== 1){newvulnmean <- newvulnmean} else
{newvulnmean[nvulnmean+1] <- 1 - sum(newvulnmean)}

# 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 <- 1

if (pile_years > 0) {

# iPCoD PROTOCOL STEP 4

# 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
pile <- read.csv(file = 'MultPilingOpsMultYears.csv', header = TRUE) ##
XXX strip out date column

# removes day labels from first column of csv file
widx <- which(colnames(pile) %in% c('Date', 'DayOfYear'))
if(length(widx) > 0){pile <- pile[,-widx]}
# In case we want to get it programmatically, this works:
library(stringr)
yvec <- str_match(colnames(pile), 'Operation')
npiles <- length(yvec[!is.na(yvec)])
pilesx <- which(str_match(colnames(pile), 'Operation') %in% 'Operation')

# template for MultPilingOpsMultYears.csv should ensure that number of
rows is an exact multiple of 365
pile_years <- nrow(pile) / 365
# yearvec indicates year number for each day in pile
yearvec <- rep(1:pile_years, each = 365)

# iPCoD PROTOCOL STEP 5

# input number of piling operations to be modelled (3 in this case)

pilesx1 <- 3

if(pilesx1 != ncol(pile)){stop('Number of Piling Operations do not match')}
# vulnpile is a matrix indicating which columns of pile are to be combined
to provide piling information for each vulnerable sub-population
vulnpile <- matrix(0, nrow = nvulnmean, ncol = pilesx1)

# iPCoD PROTOCOL STEP 6

# 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 Operations 2.
```

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

```
# repeat this for each sub-population.
```

```
if(ncol(vulnpile) != ncol(pile)){stop('Number of Piling Operations do not
match')}
```

```
# "seasons" determines whether or not the number of animals that are likely
# to be disturbed each day (NDt)
# and the number that may experience PTS (NPt) vary by season. seasons = 1,
# no seasonal variation is the default value;
# if seasons > 1, remember that the simulated year starts in June, for all
# species except GS, when it starts in October!
# if seasons = 4, 4 there are seasons in a year. summer = June, July,
# August; autumn = Sept, Oct, Nov; winter = Dec, Jan, Feb; spring = March,
# April, May
# if seasons = 2 (ie just summer and winter), we will actually require 3
# breaks, assuming "summer" = May - October (ie months when water is warmest)
```

```
seasons <- 1
```

```
inputindex <- seasons
if(seasons == 2) {
  seasons <- seasons + 1
  breaks1 <- c(152,182,31)
} else {
  breaks1 <- c(91,92,91,91)
}
```

```
if(seasons == 1){s_index <- c(365)} else {s_index <- breaks1}
```

```
# these matrices will hold the number of animals predicted to experience
# PTS and disturbance during one day for each piling operation
# each row contains the values for a particular season
# each column contains the values for a particular piling operation
daily_NPt <- daily_NDt <- matrix(0, nrow = seasons, ncol = npiles)
```

```
# iPCoD PROTOCOL STEP 6
```

```
# input number of animals predicted to experience significant disturbance
# on one day of piling for each operation
# the first 1:inputindex values in the string give the number of animals
# that will be affected in each season
```

```
# for piling operation 1, the next (inputindex+1):(2*inputindex) values
give the number of animals
# that will be affected in each season for piling operation 2, and so on
```

```
daily_NDt[1:inputindex,] <- c(60, 60, 60)
```

```
# now do the same for the number of animals predicted to experience PTS on
one day of piling
```

```
daily_NPt[1:inputindex,] <- c(2, 2, 2)
```

```
if (inputindex == 2) {
  daily_NPt[3,] <- daily_NPt[1,]
  daily_NDt[3,] <- daily_NDt[1,]
}
```

```
daily_NDt <- daily_NDt*propfemale
daily_NPt <- daily_NPt*propfemale
```

```
pile <- data.frame(pile, pbdays = rowSums(pile), pvec = rowSums(pile))
# pile$pbdays indicates number of piling events on a particular day,
pile$pvec is a flag indicating whether or not piling has occurred
pile$pvec[pile$pvec > 1] <- 1
```

```
NDt <- matrix(0,nrow(pile), npiles, byrow = TRUE)
NPt <- matrix(0,nrow(pile), npiles, byrow = TRUE)
```

```
for (j in 1:npiles){
  NDt[,j]<-
rep(rep(c(daily_NDt[1:seasons,j]),c(s_index[1:seasons]))),pile_years)
  NPt[,j]<-
rep(rep(c(daily_NPt[1:seasons,j]),c(s_index[1:seasons]))),pile_years)
}
```

```
#creates a column for each vulnerable sub-population indicating the days on
which there is piling that will affect it
for(i in 1:nrow(vulnpile)){
  pvec <- as.matrix(pile[, pilesx]) %*% as.matrix(vulnpile[i, ])
  pvec[pvec > 1] <- 1
  colnames(pvec) <- paste('vuln', i, 'pvec', sep = '')
  pile <- cbind(pile, pvec)
}
```

```
# iPCoD PROTOCOL STEP 7
```

```
# input number of days of "residual" disturbance. DEFAULT IS 1
```

```
days <- 1
```

```
# iPCoD PROTOCOL STEP 8
```

```
# decide if PTS can occur on any day (default) or only on the first  
occasion that an individual is disturbed by  
# change Day1 to TRUE if you want animals to be only vulnerable to PTS on  
the first day they are disturbed
```

```
Day1 = TRUE # Flag for the different PTS model
```

```
}
```

```
# iPCoD PROTOCOL STEP 9
```

```
# input number of animals predicted to be killed each year as a result of  
collisions with tidal energy arrays
```

```
NCollisions <- 0
```

```
NCollisions <- NCollisions*propfemale
```

```
#number of years for simulation
```

```
years <- 25
```