

LionPopR – User Manual

Overview

LionPopR is a collaborative venture between Montana Fish, Wildlife and Parks, Panthera and the University of Montana. The purpose of the project is to develop population models that aid in the management of mountain lions in the state of Montana. In an effort to simplify model use a user interface, PopR, was developed for distribution. In the documentation that follows the functionality of the user interface is described in detail.

Each section in the document contains a short description followed by a more thorough description of software features. QD precedes quick descriptions.

Videos detailing installation, workflows and more are available at [this link](#).

Installation

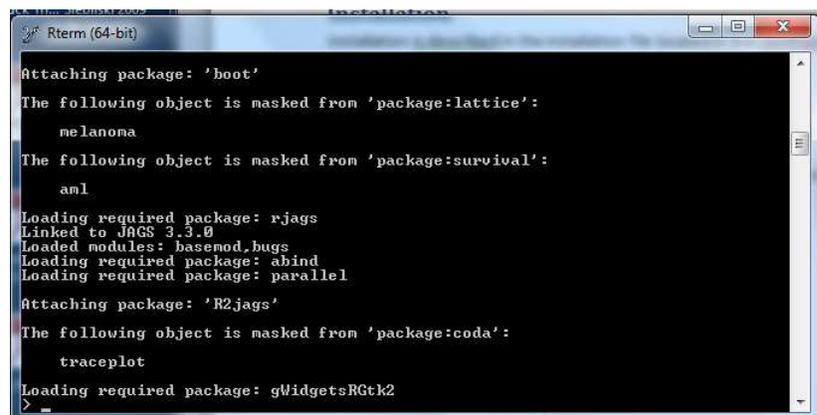
Installation is described in the PopR Installation Guide located in ~\LionPopR2\help.

Startup

QD: To start the program double-click on the  icon in the LionPopR2 folder.

A startup icon is provided with each distribution in the main directory (e.g. C:/.../LionPopR2), but may initially appear different. To start the program double-click on the  icon. Upon clicking on the startup icon a command prompt like window may appear (Figure1). This window simply lets you know that the program is working.

NOTE: If you do not wish to see this window in the future, you can edit the program's shortcut so that the program starts as a minimized window. To accomplish this right click on the  icon and select properties. Then in the shortcut properties window under the shortcut tab change the Run field to Minimized. Select OK to exit.



```
Rterm (64-bit)
Attaching package: 'boot'
The following object is masked from 'package:lattice':
  melanoma
The following object is masked from 'package:survival':
  aml
Loading required package: rjags
Linked to JAGS 3.3.0
Loaded modules: basemod,bugs
Loading required package: abind
Loading required package: parallel
Attaching package: 'R2jags'
The following object is masked from 'package:coda':
  traceplot
Loading required package: gWidgetsRGtk2
> _
```

Figure 1 Command prompt window seen at program initiation.

When PopR opens it should open in the LionPopR2 folder. If for some reason this does not happen a popup window will appear informing the user that the working directory must be set to the LionPopR2 folder. A window will appear after the initial warning that will allow the user to navigate through their folder system and select the LionPopR2 folder.

When the program starts it calls (.../Utility.Funs/start.gui.fun.r), which checks that several R packages are installed, installs them if necessary (requires an internet connection) and then loads the packages into the R workspace. With packages loaded, the user interface (.../User.Interface/LionPopR.3.0.0) is called. The user interface leverages the [tcl/tk](#) toolkit through wrappers written for the R package gWidgets.

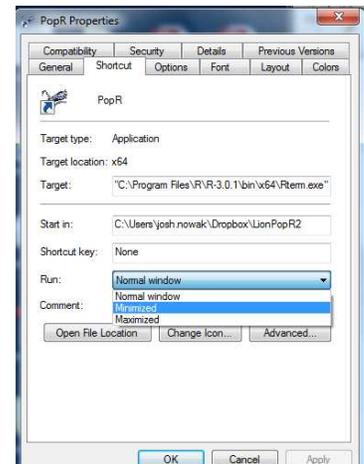


Figure 2 Setting command prompt window to open in a minimized state.

Toolbar

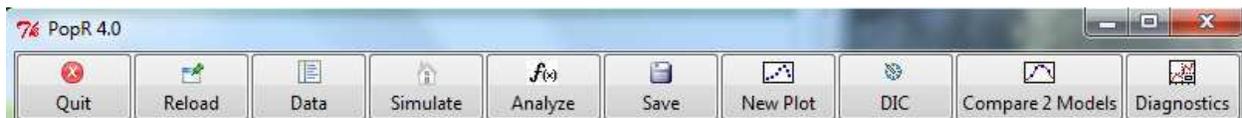


Figure 3 The toolbar at the top of the PopR user interface provides access to data, model fitting and plotting tools.

Quit

QD: Closes the program and R without saving.

Reload

QD: This button allows the user to reload a previous model run back into the R workspace. To use the button simply click on it and then select the previous model run from the list. Then be patient some of the model objects are quite large and may take a few moments to load.

The reload feature is useful because it allows for models to be plotted side-by-side and compared using DIC without having to run each model during a single session. The models available for reload are those that are saved as R objects in the Saved.Outputs folder in the LionPopR2 directory. Upon completion of the reload procedure the model output is displayed in the PopR window and the model is again available for plotting, DIC calculations and plot comparisons.

Data

QD: This button allows the user to view, edit and add new survival, harvest, quota and spatial capture-recapture estimates to the suite of data available for model runs.

Work Flow

1. The first thing that happens is the user is asked if they want to view/edit or manually add (as in spreadsheet entry) data. If this is the desired path click yes. If you want the software to attempt to format your data and add it to an existing dataset or create a new dataset click no.

a. If you clicked Yes

i. **Data Type:** Select the type of data you wish to add

1. Harvest

- a. After clicking on continue a window will appear that allows the user to select the dataset of interest and the region which to view.
- b. After making selections and clicking Show Data a spreadsheet appears, this is the actual data used for fitting models
- c. If desired the user may now change the data and when finished close the window.
- d. A prompt will then appear asking the user if they wish to save the changes, click yes to save. These changes are permanent. A good practice might be to make a copy of the original data before making alterations.

2. Survival

- a. This method of editing data is not possible with survival data. To view harvest data navigate to ~/LionPopR2/Data/Data2Add, within this folder you will find summaries of collar deployments that you can edit and view. If changes are made be sure they conform to the standards for input data detailed in the metadata guide.

3. SCR – Spatially Explicit Capture Recapture Estimates

- a. After making selections and clicking continue a window will appear that allows the user to view and edit existing data.
- b. Users may also use this function to add new data to existing data. To accomplish this simply enter the year, mean (or median), sd and precision of the estimate. Precision is equal to $1/sd^2$.

4. Quota - Harvest Quota Data

- a. Upon clicking continue a window will appear that allows the user to select an existing quota dataset and the region of interest.
- b. After making selections, click Show Data, and edit or add new data as desired. If adding new data be sure to enter the year.

b. If you clicked No

i. **Data Type:** Select the type of data you wish to add

1. Harvest Data

- a. Select the new data that you want to add to an existing data set.
- b. Decide if missing values of animal sex should be sampled from regional data or from all regions.

- c. Decide if missing values of age should be sampled from regional data or from all regions. Currently regional sampling is not available due to age data not being available for some regions.
- d. Select the old data set that the new data should be appended to. This data is located in ../LionPopR2/Data/Harvest.Data.
- e. Name the new data set and click save. Do not use spaces or special characters.

2. Survival Data

- a. Select survival option
- b. Click Load Data button
- c. Navigate to new data and click open
- d. Answer the questions regarding censoring of individuals
- e. Name the new data set

3. SCR – Spatially Explicit Capture Recapture

- a. Enter first and last year of estimates to add. Include the century with the year. If only one year is being entered use it for the first and last year, do not leave anything blank.
- b. Click Set Years
- c. Click Ok
- d. Replace the 0's in the spreadsheet with the mean and standard deviation (standard error using frequentist methods) of the estimated population *density*. For years between estimates for which there is no data enter NA or leave as 0. Years with 0 or NA will be excluded from consideration when building the data.
- e. Close the window.
- f. Name the new data set.
- g. Click Save and Exit.

4. Quota – Harvest Quota Data

- a. Select quota option
- b. Click Continue
- c. Click Ok
- d. Enter the number of mountain lions of each sex that were allowed to be harvested in a given year.
- e. Close Data Editor Window
- f. Click Ok, the data is now changed

Harvest Data Notes

A folder called Data2Add is provided in the LionPopR2/data directory for the storage of new data. It is suggested that you store new data sets in this folder. The requirements for formatting of the new data are included in the Harvest.Data.Metadata.Formatting.xls file within the help folder. Missing values of sex and age at harvest are sampled randomly from the observed distributions of ages and age at harvest, respectively. That said, sampling can be accomplished using only a single regions data or the

entire states data. When the focal region has relatively little harvest then it is suggested that sampling be conducted at the state level. The way this tool works is to add the new data to an existing data set. Because of this, the user will need to select a base data set to add the new data to. LionPopR ships with a data set called regional.harvest.1997.2012, if we were adding data for 2013 we would select the ...1997.2012 data set. The following year when adding 2014 data we would select the data set ending in 2012 and so on. The name of the data set will be shown when using the run model tool, which suggests that a well thought out informative name be given to the new data. More control is gained by the user when using the view/edit option, but this option requires the user to summarize harvest records by sex, age, year and to guess at missing values of sex and age.

Survival Data Notes

When making a new survival data set follow the format defined in Survival.Data.Metadata.Formatting.xls *exactly*. This file can be found in the help directory. After selecting the survival option from the add data pop-up window you will be prompted to select the new data to add. Use ensuing browser to navigate to and select the data that you wish to format (e.g. ../Data/Data2Add/in.surv.test.csv). Once the user selects the new data to format a few checks are made and the user is asked to make a few decisions about how to proceed. First, if any animals in the data set have less than one month between the time of capture and death the user will be asked if they want to censor these animals from consideration. Second, when animals were monitored before 1997 these animals can be censored. The number of animals censored will be calculated and presented in its own window. If it is desirable to run, say a constant model, using survival data from before 1997 then it is recommended that the user manually change the dates of monitoring to reflect some later date. When doing this make sure the name given to the data is very informative and these data are not used for any model that varies survival by year (denoted by .yr in the model name). Enter a new and informative name for the data set and then click save to complete the addition.

SCR Data Notes

The Add SCR tool allows for manual entry of population estimates derived from spatially explicit capture recapture methods. The mean and standard deviation of density estimates are required inputs. Do not input abundance estimates.

Quota Data Notes

The add quota data option allows manual alteration of the quota data used as a model covariate. The quota data values represent the total number of lions, of each sex, harvestable in a given year at the regional level. Some discretion is required to obtain this number. In general, when special permits allowed for harvest above and beyond the quota values they were added to the quota. The final report for this project details exactly how this covariate data was created.

Simulate

QD: Clicking on this button allows the user to run a population simulation.

Workflow

1. **Years to simulate** – Enter the number of years to simulate

2. **Repeat simulations** – If desired the simulation can be repeated so that bootstrapped confidence intervals can be created. Enter the number of times to repeat simulations in this box. Using values > 1,000 is good practice.
3. **Specify Vital Rates** – The series of boxes provided here allows the user to enter the mean vital rate value and variation. This is hard to do mentally and so whenever a value is changed the resulting distribution is plotted. If process error (i.e. stochastic variation) is not desired simply change the value of SD to something small (e.g. 0.01).
4. **Click next when finished** – Click on the Next button when satisfied with the entries.
5. **Initial Population Size** – Click on the Set Initial Population button to set the starting population size.
 - a. After clicking on Set Initial Population the user is prompted to decide if they want to use the stable stage distribution (SSD) as the initial population size.
 - i. *Click Yes to manually select the population starting values*
 1. Area – Enter the area of the region under consideration. This area is used to calculate density within this window.
 2. Define proportion in each sex age class – In order to start the model we require a chosen density, but density is the sum of abundances divided by area, which does not help with starting the model. To fix these, the user is asked to set the proportion of the total population that is made up of each sex by age class combination.
 - a. The sum of the proportions must equal 1. A green arrow appears when this is true and is red otherwise.
 3. Adjust density using slider – A starting density is required to start the model, use the slider to set the density
 4. Starting abundance and vital rates show in table below – After setting the proportions and density the table under this heading displays starting abundance for each of the sex by age combinations. The table also displays recruitment and survival terms as solved by dividing age class $n+1$ by age class n . Similarly, the table displays recruitment. Recruitment is calculated by dividing the number of kittens by the number of adults.
 5. Close – When finished click Close.
 - ii. *Click No to use the SSD to proportion animals into sex and age classes*
 1. The reason for using a stable stage distribution to start the simulations is that matrix models will go through transitory dynamics in the first years (generally reaching SSD within 15-20 years). Transitory dynamics are nothing more than fluctuations induced by a population composition different than SSD. Because vital rates are stochastic the proposed SSD is calculated from 500 matrices created from the chosen vital rate values.
 2. After clicking No a window appears that allows the user to set the initial density. The SSD is used to divide the total density into sex and

ageclass. The results of the proportioning are displayed as a table. Click close after using the slider to set the desired initial density.

6. **Model Choice** – Select the model you wish to run. At the time of this writing a single model is available for consideration.
 - a. **Simulation Model Structure** – `sim.fun.4stage.2sex.r` – This model considers 2 sexes and 4 age classes. Reproduction maturity is reached at 18 months of age. A 2 year birth interval is imposed along with an equal sex ratio at birth by dividing litter size by 4. Harvest mortality is subtracted from annual abundance. In this way harvest is additive on abundance, but has no effect on survival.
7. **Name the model run** – Give a name to the model run. Names should not contain spaces, special characters or begin with a number.
8. **Load Harvest Data** – Click on Click to Load Harvest Data, this allows you to navigate to a .csv file containing harvest data that is used in the simulation. A harvest dataset is provided with the distribution. The data are located in `~/LionPopR2/Data/Harvest.Data` and named `sim.test.harv.csv`. Formatting instructions for harvest data used in *simulation models* can be found in the help directory (`Simulation.Harvest.Metadata.Formatting.xls`).
9. **View Harvest Data** – This button allows users to view and edit harvest data before running a model. This is an optional button.
10. **Check Data** – A few simple checks of the data will be made when this button is clicked. In addition, by checking the data the Run It button is activated.
11. **Run It** – Click this button to begin model runs. When the runs finish popup windows will close and the results of the simulation will be placed in the main GUI window.

Simulation Notes

The simulation model requires that the user set many parameters. To get an idea of what survival values are reasonable the user might consider fitting survival model detailed under the Analyze button. Litter size estimates from the field are routinely reported between about 2.2 and 3.0 with more values occurring near 3. Simulation modeling is part art and much trial and error will be required to find a set of values that does what is expected of the model.

Analyze

QD: Click on this button to initiate the model setup window for survival and population models. Click on the question marks next to the fields to find out what they do. Click set selections and then follow the prompts. Once the prompts have been completed click run model to initiate the model run.

Work Flow

1. The first option asks whether the user wants to run a survival only model or an integrated population model (IPM). The IPM is the full population model that considers all sources of data simultaneously.
 - a. **Survival**
 - i. **Choose Survival Data Set** – From this dropdown choose a dataset to analyze.

- ii. **Model Choice** – At the time of writing 5 models of different structures are available for analysis. The models are all multi-state model with 3 fates possible. The naming convention suggest how survival is structured or varies in time. Model definitions are provided to the user after clicking on the book icon to the right of the box.
- iii. **Name this model run** – In this box the user must provide a name for the model run.
- iv. **Censor Harvested Individuals** – Checking this box will invoke a routine that censors all harvested individuals from the dataset before calling the models. This is a necessary step if estimation is being done for later incorporation in the simulation model described elsewhere. Censor occurs at the time step prior to harvest. Harvest mortality must be recorded as a 2 for this to work.
- v. **Model Control** - The Model control box extends control over MCMC sampling to the user. Each feature is described below and in more detail along with a not so technical first introduction to Bayesian statistics is provided in Appendix A.
 1. *Automatic/Manual* – This option allows the user to control the burnin, number of iterations and thinning rate of the MCMC sampling.
 2. *Run until converged* – Checking this box will make the model update repeatedly until convergence is reached according to the Brooks-Gelman diagnostic (i.e. R_{hat} in the model output is < 1.1 for all parameters).
- vi. **Set Selections** – Clicking on this button invokes sparse data checking and makes available the Run Model button.
- vii. **Run Model** – When available and you are satisfied with the rest of the form click on Run Model to call the model. A progress bar will appear shortly letting you know the model is running.

b. IPM

- i. **Region** – Select the region that should be modeled.
- ii. **Years to Model** – Set the first and last year of consideration. Data does not exist prior to 1997. The models rely heavily on the auto-correlation of the time-series and consequently leaving the beginning year at 1997 is generally recommended because it maximizes sample size.
- iii. **Model Choice** – Five model structures are available to run with the option to add SCR estimates to each. Select a model from the dropdown menu. The models are named such that the spatial extent is referenced first (single.region). The next elements of the name suggest how survival and mortality vary in time. For example, the model single.region.sex.age.quota , estimates different values of survival for each sex, age class and considers harvest quota values through time as a covariate on harvest mortality. Clicking on the book icon to the right of the dropbox will display definitions of the models. The model named Batch.Run allows the user to run all models on a consistent set of data.

- iv. **Choose Harvest Data Set** – Use the dropbox to choose a harvest data set. During all model runs the user will have the opportunity alter harvest data or data can be permanently altered using the Data button.
- v. **Choose Survival Data Set** – Select a survival data set to use for the model run. Two options exist initially, a Montana specific dataset and a dataset that covers and longer time span and larger geographic area (WA, WY, MT).
- vi. **Choose SCR Data Set** – If you told the model to use SCR data then a dataset needs to be selected from this dropbox.
- vii. **Name this model run** – In this box provide a name for the model run. The name should not start with a number, include spaces or special characters.
- viii. **Model Control** – The Model control box extends control over MCMC sampling to the user. Each feature is described below and in more detail along with a not so technical first introduction to Bayesian statistics is provided in Appendix A.
 - 1. *Automatic/Manual* – This option allows the user to control the burnin, number of iterations and thinning rate of the MCMC sampling.
 - 2. *Run until converged* – Checking this box will make the model update repeatedly until convergence is reached according to the Brooks-Gelman diagnostic (i.e. R_{hat} in the model output is < 1.1 for all parameters).
- ix. **Email me at address below when finished** – Model runs can take a long time. Because of this an option is provided to email the user when the model run finishes. To use the feature simply enter your email address in the space provided.
- x. **Set Selections** – Click on this button after filling out the rest of the form. The software will run a few checks on the data and produce error messages if problems are noticed. Error checking is not exhaustive and ultimately it is up to the user to ensure accuracy of the results. After clicking on Set Selections a series of windows will appear:
 - 1. Typically the first window asks the user to fill in values for quota data where there are NA's. Missing data are not allowed in covariate data.
 - 2. Second the user will have the opportunity to view and edit harvest data. NA's in the harvest data are perfectly fine. However, it is also possible to put in values and predict as if future harvest were known.
 - 3. Third, the user will have the opportunity to change starting values. The initial population density may be set by abundance or density. Note that if the model runs until convergence that the initial values should be forgotten. However, if models runs are short the initial values may have very real effects on the model output.
- xi. **Run Model** – After working through the set selections windows the run model button will become available. Click on this button to initiate the model run. You

should see a progress bar shortly after clicking on this button. The progress bar may appear stuck from time to time, be very patient it is working hard.

Save

QD: Launches a new window within which one can choose to save a given model run as summary statistics (.csv), summary plots (.pdf) and/or an R object (.RData). It is important to remember that no saving is done automatically.

Work Flow

1. **Choose model to save** – Using the dropbox select the model you wish to save. Simulation, survival and IPM models can be saved using this function.
2. **Choose output type** – The next element allows the user to select the type of output to save. The CSV option makes tabular data and saves this data as .csv file in `~/LionPopR2/Saved.Outputs`. Summary plots will create all of the available plots and save them as a .pdf in the same directory. Last, the R object save type allows users to save the R object of the model. This last option is very important as saving R objects allows the user to reload the model at a later date. Only R objects may be reloaded into the workspace!
3. **Name the file to save** – A default name is chosen for the user, but the name is long and ugly. The purpose of the name is to ensure that the name is unique and other saved items are not overwritten. This is accomplished by appending the date and time to the model name. The time format is YYYYMMDD.HHMMSS. Finally, it is reasonable to simply provide an informative customized name of your liking. The appropriate file extension will be added to the file name if none is supplied and it is recommended that the user simply omit extensions when saving objects.
4. **Save** – Click save when ready and the object will be saved. A check is done internally to ensure that the file was in fact created and saved. When problems occur the user will be notified by popup window.

New Plot

QD: First choose which model run to plot and then a series of plots will appear along with a window so the user can toggle among them. This button only works for simulation runs and IPM's. If it is called on a survival model the user will be provided instructions on how to plot.

Work Flow

1. **Pick a model** – Using the dropbox choose the IPM or simulation that you want to plot.
2. Once a model is selected all plots are created and a second popup window appears that allows the user to scroll through available plots. The window sometimes hides behind other windows, placing your mouse over the PopR icon in the task bar will allow you to select it and bring it to the foreground.
3. If desired the user can also click on the New Plot button more than once and open several plot selectors for side by side viewing.

DIC

QD: DIC is just like AIC and is used to compare models. This button creates a table of DIC values for each model that is currently in the R workspace. One can add models from previous sessions to the R workspace using the reload button if necessary. Note that survival and IPM models will be compared within this window if both are in the workspace. This will cause deltaDIC to be wrong. To correct this issue subtract the minimum DIC from every other DIC estimate of a particular variety.

Compare 2 Models

QD: If two or more models are in the workspace it is possible to compare their plots. Clicking on this button will produce a new window that allows the user to select the two models to compare and then the parameter to compare. Clicking on plot comparison will produce the plot if possible.

When comparing two plots it is necessary that the dimensions of the parameter remain the same. For example, if one model holds survival constant through time while the other allows survival to vary annually a comparison of survival is not possible. However, several parameters will always have the same dimension (e.g. N).

Diagnostics

QD: Clicking on this button opens a window that allows the user to select the model run and parameters of interest. The user then clicks on Plot Diagnostics. This operation may take a moment, but when completed the default web browser will open and a series of diagnostics are available to study the behavior and convergence of the model. A copy of the plots is always saved to ...LionPopR2/Saved.Outputs/Diagnostics.

Diagnosing MCMC runs is part art, but is very useful when deciding whether a model run should be trusted. Appendix A and one of the instructional videos provides details on interpreting the output of MCMC runs.

Appendix A

Markov Chain Monte Carlo (MCMC) methods provide a means of finding solution to very difficult problems. In what follows I provide a very brief, and hopefully non-technical, introduction to MCMC methods and Bayesian modeling in general. The language used here is purposely loose to increase accessibility, more formal treatments of the matter can be found elsewhere. For further details I would suggest Marc Kery's amazingly accessible book [Introduction to WinBUGS for Ecologists](#) and for the more adventurous perhaps Link and Barker's 2006 text [Bayesian Inference with Ecological Applications](#).

Bayesian Inference

In short, the Bayesian view of the world is a little different and certainly not new given that the original work of the reverend Bayes was published in 1763. The recent explosion of Bayesian methods is likely the result of computers because before computers many problems quickly became too difficult to solve. The basic formulation for any Bayesian problem is stated such that the posterior is proportional to the prior times the likelihood. The posterior is the answer and it is a true probability distribution. The prior is the information known before the study is conducted. Priors can contain lots of information or very little. The likelihood is a statement about the plausibility of data given the model. Ok, so what? Well, consider this great example from the forums of [StatExchange](#):

I have misplaced my phone somewhere in the home. I can use the phone locator on the base of the instrument to locate the phone and when I press the phone locator the phone starts beeping.

Problem: Which area of my home should I search?

Frequentist Reasoning:

I can hear the phone beeping. I also have a mental model which helps me identify the area from which the sound is coming from. Therefore, upon hearing the beep, I infer the area of my home I must search to locate the phone.

Bayesian Reasoning:

I can hear the phone beeping. Now, apart from a mental model which helps me identify the area from which the sound is coming from, I also know the locations where I have misplaced the phone in the past. So, I combine my inferences using the beeps and my prior information about the locations I have misplaced the phone in the past to identify an area I must search to locate the phone.

Under the heading Bayesian Reasoning we see that the individual used not only the sound of the phone beeping, but also previous information about where the phone was previously found. This is Bayesian inference. So how does one conduct Bayesian inference and what do computers have to do with it?

MCMC is the answer. The difficulty associated with solving Bayesian problems and many other types of problems lead researchers to look for new methods of “looking for an answer”. One technique that works particular well is to take a walk. Random walks are a type of algorithm that are used to model population growth, animal movements and as it turns out can help us find an answer. The walk is implemented as a type of simulation where values are evaluated sequentially and “kept” if they are better than the last value. Imagine some data, say 55, 54, 80, 65 and 19. If we want to solve the mean by simulation we could choose a starting value, say 30. Now we can calculate the likelihood of our data given that our model says the mean is 30. Ok, one down, now select a new value, how about 40. We now calculate the likelihood again and ask if the second value is more likely than the first. If 40 is more likely then we move to 40. Now select another, this time something big like 100, the likelihood of 100 being the mean of our data is really small, so this time we stay at 40 instead of moving. This sort of simulation results in a chain of values that can be plotted and summarized. The chain is thought to be samples from the true posterior distribution or the answer.

Controlling a Bayesian Model

From the above description I hope it is clear that a Bayesian model is fit by simulation, by iteratively selecting values. The first question is where should the model start. PopR handles this for you by randomly selecting values at which to start each chain. How many chains should we run? PopR runs 3 chains in all models. This is an important point for several reasons. First, because initial values or starting points are selected randomly each chain starts in a different place. The advantage is that when our model finishes we can see if each chain ended up at the same place. Second, running 3 chains allows us to test if the model has converged. Convergence is a fancy word for describing when a model has found a reliable answer. This is a difficult thing to do, but some diagnostics help us with this problem. You might notice that in the model outputs there is a value for Rhat. This is a convergence diagnostic. When the values of Rhat are less than 1.1 we generally take that as evidence that the model has converged. Back to running the model, the next question is how long should we run the chains, how many simulations or iterations should be conducted? The answer is variable, but let’s start by saying that we first want the model to find an answer before we start recording values. The initial answers that a chain moves through are likely to be related to the starting values and may not reflect where the chain ends up. Graphically we might see the chain moving up or down, but it won’t be bouncing around a value. To make sure we don’t consider these values when making inference we omit what is called a burnin period. Another words we throw out all the values before the model reaches convergence. Then we run the model a while longer and record the values it chooses. So the length of a run has two components, the burnin and the iterations. The burnin period is a sort of initialization while the iterations are the part we want to save and summarize as our answer. I will make a few risky recommendations here and say that for the purposes of decision making burnin’s should exceed 100,000 with at least 20,000 more iterations after the burnin. The last thing we need to discuss is thinning. Thinning is a way to save computer memory. If we run a model for a burnin of 10 and then do 20,000 more iterations we have the option of keeping every value or every other or every 10th. I think you get the point, a thinning value of 1 means keep every value while setting the thin parameter to 5 keeps every 5th value.

Convergence

Knowing whether an answer is reliable is one of the most difficult tasks when conducting an analysis. A first criteria might be to ask if the answer is reasonable. This is a good first question, but be careful to not throw out a disagreeable result simply because it was unexpected or did not confirm a prior belief. If confirmation of what you thought before fitting the model is the goal, you don't need a model or data. A second step for evaluating models is to check the Rhat statistic reported in the model output. As stated above we like to see that all parameters receive Rhat values below 1.1. With a gut check and Rhat values that do not exceed the threshold we can use the diagnostic button provide in PopR. Clicking on this button allows the user to create a series of plots. These plots are quite informative, see the example below.

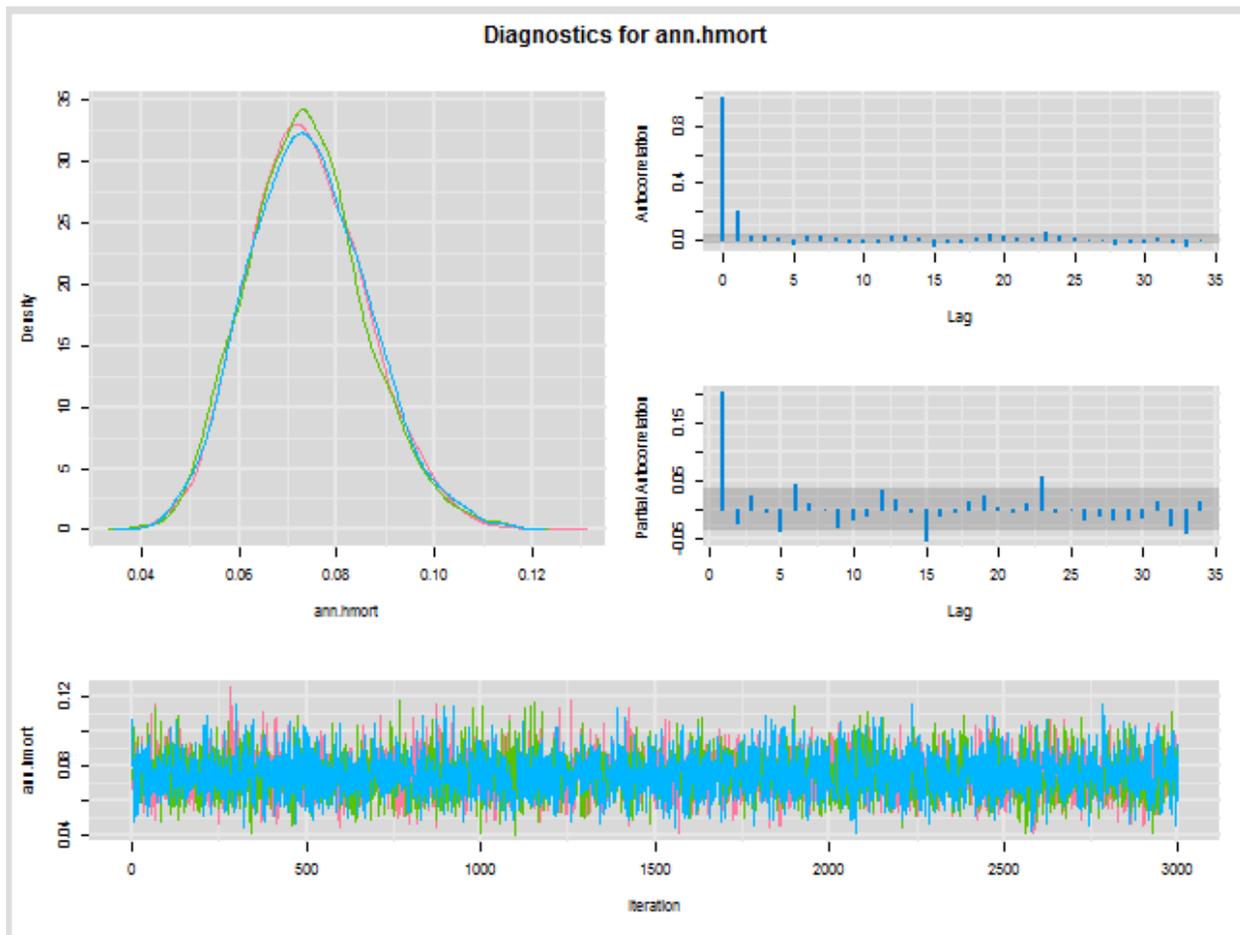


Figure 4 Diagnostic plots for a single parameter from a survival model. The plot in the top left is a posterior density plot of the parameter. The top right plot shows the within chain autocorrelation. The next plot down shows the partial autocorrelation. Finally, the long plot at the bottom shows the actual chains. It is clear that there are 3 chains (green, blue and red) that overlap nicely covering many of the same areas. The x-axis of this last plot also shows us that 3,000 iterations were saved.

Figure 4 shows a typical suite of diagnostic plots. Here I will highlight some of the big things to look for when diagnosing convergence. The top left plot is a density plot. That is it shows the density of each value on the x-axis. The things to look for here are that all three chains overlap, the distribution is

smooth and only has one peak (unimodal). The last thing to look for in the density plot is symmetry, it would be nice if all the plots typified the bell curve of the normal distribution, but this is a lofty goal. The top right plot shows us the autocorrelation within each chain. We want the walk to be random, not a correlated random walk, so that values are chosen randomly and we can rest assured that many plausible values were evaluated. The thing to look for in this plot of autocorrelation is that the values drop towards 0 very quickly. The plot above is nearly ideal as the correlation is near zero within 1-2 steps. The partial autocorrelation plot is useful when chains are highly correlated to diagnosis the lag at which the correlation occurs. The pretty plot on the bottom shows the chains through time. If you look at the x-axis you can see the iterations labeled. The y-axis labels the values the chain took at that iteration. What we want to see ideally is that the chains move in the same space, they are not sticky or get stuck in any place away from the mean and the mix (or overlap) well. The chains above might be described as grassy, this is ideal. Random movements away from the mean are part of the sampling, they ensure that the chain does not get stuck on a value and that many solutions are evaluated.

An example of a model that has not converged is shown in Figure 5.

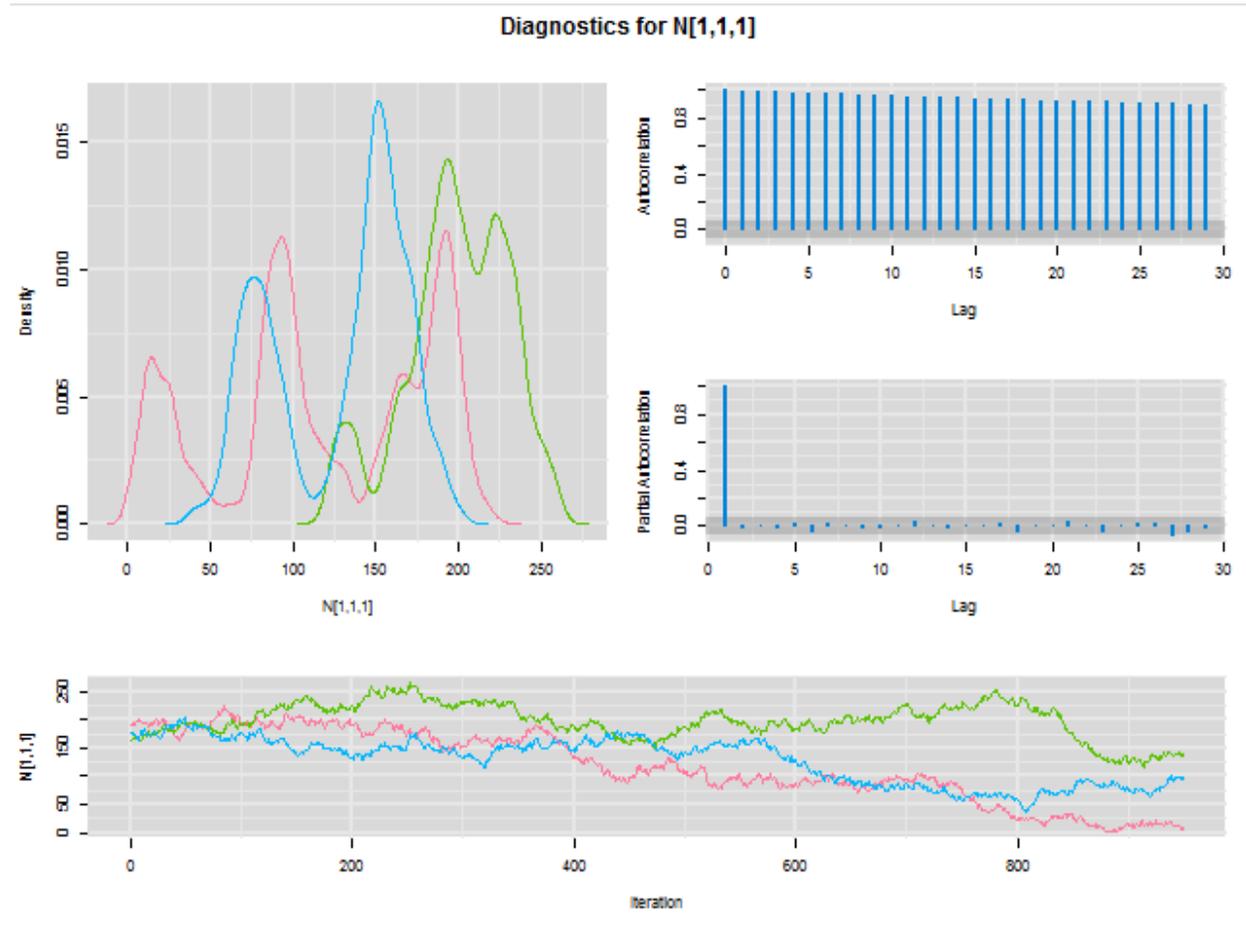


Figure 5 Diagnostic plot of an IPM run with a burnin of 50 and only 1,000 iterations.

Hopefully it is clear that bad things are happening in the Figure 5. The density plots are not smooth with a single peak. The individual chains in the density plot do not overlap. The autocorrelation plot

illustrates highly correlated chains with a correlation of near 1 out to a lag of 30. The plot of the chains tells us that mixing not occurring and the chains are clearly moving from a high point to one much lower. This is a rather extreme example, but it illustrates a poorly mixing model from which inference should not be made.