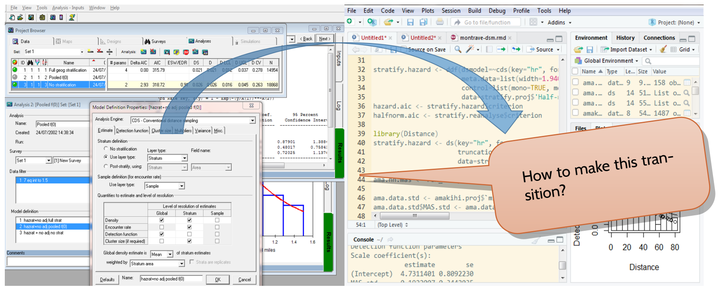
Migrating distance sampling projects from Distance for Windows to the Distance R package

Eric Rexstad, David L. Miller, Laura Marshall and Len Thomas

Centre for Research into Ecological and Environmental Modelling University of St Andrews

Introduction

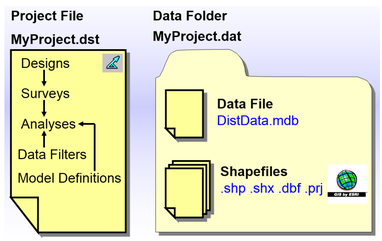
The Distance software (Thomas et al., 2010) has been downloaded >40,000 times in its 20-year history. Much of the underlying machinery is written in R. For some users, there may be benefits to performing the analysis with the underlying R code, rather than working with the graphical user interface (GUI).



Challenges hindering the transition between analysis with the GUI and analyses in R are two-fold:

- Legacy data reside in Distance (GUI) projects, unavailable for importing into R, and
- Analyses that are easily described using the GUI may be difficult to specify, particularly if analyst is not proficient in R.

How to bridge between the two?



Distance sampling analysis are defined in Distance project files (an Access database).

- Data
- Model definitions
- Analysis results

`readdst` uses `RODBC` (Windows) or `mdb-tools` (Mac) to mine the above information out of the database, then:

1. Translate model definitions into R code
2. Save data into an R `environment`
3. Extract results from the database for completed analyses

Converting Distance projects to R

Workhorse function in `readdst` package is `convert_projects()`. Its single argument is simply the path to an existing Distance GUI project (created by Distance 6 or 7). `convert_projects()` interrogates the Access database and translates the contents of database tables into an R object of class `converted_distance_analyses`, a list of lists of class `converted_distance_analysis`.

Object produced by `convert_project()`

The list returned by `convert_project()`, for a given entry (analysis) contains all the salient information of a Distance analysis extracted from the Access database. We point out several of the list elements critical to subsequent processing of the now-extracted data.

call R code call to the function `ddf()` in package `mrds` to duplicate the analyses done in the Distance GUI.

env An *environment* consisting of a set of data frames. These data frames include the original data extracted from the Distance project, along with the observation, sample and region tables describing the hierarchial nature of the data base.

The nature of the analysis conducted by the Distance GUI, translated into a call to `ddf()` in package `mrds`, as well as the data used in the analysis is contained within this object and available for further analysis within the R environment.
The `converted_distance_analysis` list can be passed as an argument to `run_analysis()`. This function will perform an `mrds` analysis *behind-the-scenes* , without the user seeing how it was performed.

Learning distance sampling in R

Distance GUI users can use `converted_distance_analysis` objects to learn how to perform corresponding analyses using the `mrds` R package:

```
> library(readdst)
> stratify.proj
  ID                                                Name            Status
1 13 Half-normal cosine no stratification exact             Ran OK
2 16            Half-normal cosine pooled detfn             Ran OK
3 15    Half-normal cosine strat-specific detfn Ran with warnings
> stratify.proj[[1]]
Model name  : Half-normal cosine no stratification exact
ID          : 13
Data filter :
mrds call   : mrds::ddf(dsmodel=~cds(key="hn", formula=~1,
              adj.series="cos", adj.order=NULL),
              meta.data=list(width=NA, left=0),
              control=list(mono=TRUE, mono.strict=TRUE),
              method="ds", data=data)
```

Comparative analysis of difficult data

The `test_stats()` function in `readdst` performs the analysis residing in the call element of the `converted_distance_analysis`, producing estimates of parameters of distance sampling analyses (*e.g.* AIC scores, $\hat{P}_a$, $\hat{D}$, $\hat{D}$ and associated measures of precision). These results are contrasted with estimates of the same parameters stored in the Access data base, as computed by the Distance GUI.
The resulting comparison is produced in tabular form with the difference between estimates presented as proportions. Those comparative estimates within a specified tolerance are highlighted.

Statistic	Distance_value	mrds_value	Difference	Pass
n	90	90	0	<U+2713>
parameters	1	1	0	<U+2713>
AIC	63.880100250244	63.879819152981	0.000004396471	<U+2713>
Chi^2 p	0.1707298	0.9906514	4.802452	
P_a	0.451959013939	0.451956754149	0.000004969147	<U+2713>
CV(P_a)	0.07679188997	0.07679084338	0.00001362931	<U+2713>
log-likelihood	-30.94005	-30.93991	0	<U+2713>
C-vM p	1	0.9	0.1	
density	0.05059615	0.02768243	0.4528748	
CV(density)	0.2693232	0.3070729	0.1401652	
individuals	34658	18962.4623604	0.4528691	
CV(individuals)	0.2693232	0.3070729	0.1401652	

This can be used to investigate situations in which the FORTRAN optimisation engine in the Distance GUI performs differently from the optimiser in `mrds()`.

Caveats

`readdst` is not able to translate all GUI analyses into R code. Current limitations are inability to translate

- analyses using the `dsm`, `mads` and `Dssim` engines,
- analyses using post-stratification and
- bootstraps for variance estimation.

Additional information

References

Laake, J., D. Borchers, D. Miller, and J. Bishop. 2018. *package mrds*.

Miller, D.. 2018. *package Distance*.

Miller, D. L.. 2017. *Package readdst*.

Miller, D. L., E. Rexstad, L. Thomas, L. Marshall, and J. Laake. 2016. Distance Sampling in R. *bioRxiv*.

Thomas, L., S. T. Buckland, E. A. Rexstad, J. L. Laake, S. Strindberg, S. L. Hedley, J. R. Bishop, T. A. Marques, and K. P. Burnham. 2010. Distance software: design and analysis of distance sampling surveys for estimating population size. *Journal of Applied Ecology*, 47(1):5–14.

QR codes to package/website/bioRxiv