Beginner's Guide to Rdistance Line-Transect Analysis

No Covariates

Trent McDonald, Jason Carlisle, and Abigail Hoffman

2023-05-10

Introduction

This beginner's tutorial assumes familiarity with Program R, but not distance-sampling analysis. This tutorial focuses on input data requirements, fitting a detection function, and estimating abundance (or density) in `Rdistance`. We use one of the example data sets included in `Rdistance`; line transect surveys for sparrows. Beginner point-transect analyses are covered in a separate vignette.

1: Install and load Rdistance

First step is to install the latest version of `Rdistance`. In the R console, issue,

```
install.packages("Rdistance")
```

You can install the development version of `Rdistance` from GitHub using

```
devtools::install_github("tmcd82070/Rdistance")
```

After the package is installed, it is loaded into the current session using the following:

```
require(Rdistance)
```

The following statements bring the example data sets into R's global environment where other routines can see them.

```
data("sparrowDetectionData")
data("sparrowSiteData")
```

2: Input data

Estimation of abundance (or density) in `Rdistance` requires two data sets; the *distance* data set and the *transects* data set. These data sets are described below. Both data sets can be prepared using any method that results in an R `data.frame`. For example, one could format distance and transect information on separate sheets in `Excel`, export each sheet to CSV format, and read each into R using `read.csv`. In this tutorial, we make use of the sparrow data sets which are already contained within `Rdistance`.

Detection Data

The first data set required by `Rdistance` is the *detection* data frame. The detection data frame contains one row for each detected target group, while columns contain information on each detection. `Rdistance` looks for the following information in the *detection* data frame:

- **Detection Distance** = The perpendicular distance (also known as off-transect distance) from the transect to the detected group. Note that this column must have units assigned (e.g., "m" or "ft").

Measurement units can be assigned using the `units()<-` or `units::set_units` functions. For example, `units(sparrowDetectionData$dist) <- "m"` (see `help(dfuncEstim)` for details). Distances are required to estimate a distance function (function `dfuncEstim`) and to estimate abundance (function `abundEstim`).

- **Site ID** = The ID of the transect on which the detection was made. Transect IDs are required to estimate abundance (function `abundEstim`), but not to estimate a distance function (function `dfuncEstim`).
- **Group Sizes** = The number of individuals in the group associated with each detection. Group sizes are required to estimate abundance (function `abundEstim`), but not to estimate a distance function (function `dfuncEstim`).

The specific columns in the *detection* data frame that contain distances and group sizes are specified in the formula argument of function `dfuncEstim`. The column(s) containing transect IDs is specified in the `transectID` argument to function `dfuncEstim`. If `transectID` is NULL, the set of common columns from the *detection* and *site* data frames are assumed to form the transect IDs. See the **Input data frames** section of `help(dfuncEstim)` for additional details.

Line-transect distance-sampling analysis is performed on perpendicular off-transect distances, i.e., from detected groups to the transect, not from detected groups to the observer. Commonly, observers record straight-line sighting distance (from observer to group) and sighting angle instead of perpendicular distance. `Rdistance` provides a utility function named `perpDists` that computes perpendicular distances from sighting distance and angle. See `help(perpDists)` for details.

The first six rows of the sparrow *detection* data set are:

```
head(sparrowDetectionData)
```

```
##   siteID groupsize sightdist sightangle     dist
## 1     A1         1        65         15 16.8 [m]
## 2     A1         1        70         10 12.2 [m]
## 3     A1         1        25         75 24.1 [m]
## 4     A1         1        40          5  3.5 [m]
## 5     A1         1        70         85 69.7 [m]
## 6     A1         1        10         90 10.0 [m]
```

We will use `siteID`, `groupsize`, and `dist` in this tutorial. Column `dist` was computed using function `perpDist`. Details on the study and other columns are in `help(sparrowDetectionData)`.

Site Data

The second data set required by `Rdistance` is the **site** data frame. We use the term 'site' because it covers both continuous transects and point transects. In this tutorial, a 'site' is one transect. A key feature of the *site* data frame is that it contains one row for each surveyed site regardless of whether the site was "positive" (with detections) or "zero" (without detections).

`Rdistance` looks for the following information in the *transect* data frame:

- **Site IDs** = The ID of every surveyed transect in the study area.
- **Length** = The length of each transect. The length column in the *transect* data frame must have assigned measurement units (e.g., "m" or "ft"). Again, measurement units can be assigned using `units()<-` or `units::set_units`.

Note that a `site` data frame is not required if users only wish to estimate a distance function but not abundance. In addition to site IDs and length, the *site* data set can contain transect level covariates (i.e., covariates that are constant over detections on the same transect).

The first six rows of the sparrow *site* data set are:

```
head(sparrowSiteData)
```

```
##   siteID  length observer bare herb shrub height shrubclass
## 1     A1 500 [m]     obs4 36.7 15.9  20.1   26.4       High
## 2     A2 500 [m]     obs4 38.7 16.1  19.3   25.0       High
## 3     A3 500 [m]     obs5 37.7 18.8  19.8   27.0       High
## 4     A4 500 [m]     obs5 37.7 17.9  19.9   27.1       High
## 5     B1 500 [m]     obs3 58.5 17.6   5.2   19.6        Low
## 6     B2 500 [m]     obs3 56.6 18.1   5.2   19.0        Low
```

We will use `siteID` and `length` in this tutorial. See vignette 'Rdistance_BeginnerLineTransectCovar' for an analysis that includes covariates in the distance function.

3: Fit a detection function

After input data are prepared, it is generally a good idea to inspect a histogram of the perpendicular distances. We suggest checking the minimum, maximum, and measurement units of the distance measurements in the following histogram. Note the automatic inclusion of measurement units in the x-axis label.

```
hist(sparrowDetectionData$dist
     , col="grey"
     , main=""
     , xlab = "Distance")
rug(sparrowDetectionData$dist,quiet = TRUE)
```

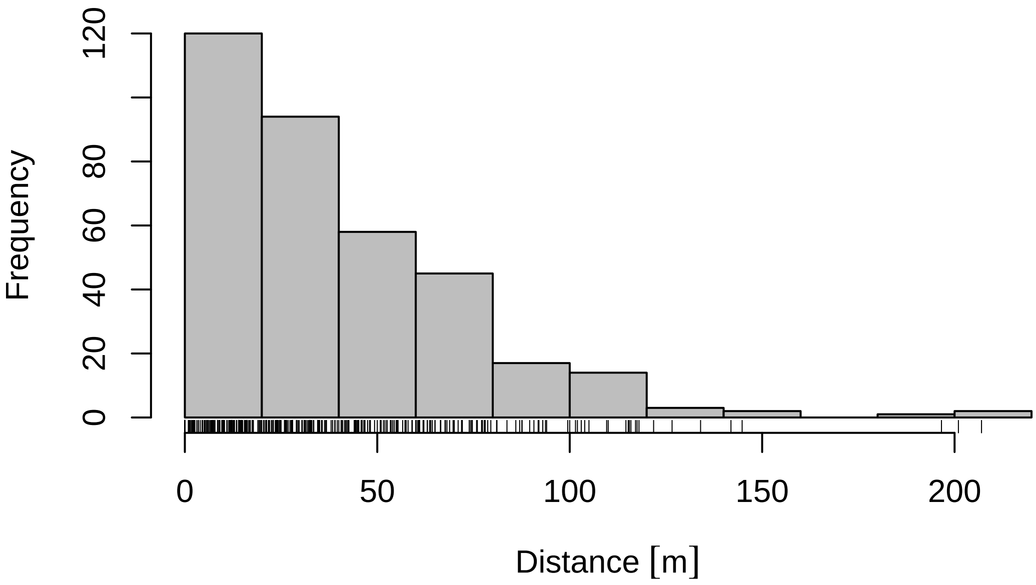

Figure 1: Histogram of sparrow off-transect detection distances.

```
summary(sparrowDetectionData$dist)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00   14.15   30.75   39.64   57.35  207.00
```

Next, we fit a detection function to perpendicular distances function `dfuncEstim` and plot it. Function `dfuncEstim` uses maximum likelihood to estimate parameters of the distance function fitted to individual distances. Histogram bin sizes in this and subsequent plots are for visual purposes and do not matter for estimation. In this tutorial, we proceed using the half-normal distance function. See `help(autoDistSamp)` for a function that automates the process of fitting different detection functions, assessing their fit (by AICc), and estimating abundance using the best-fitting function.

If group sizes are not specified, `Rdistance` assumes all detections consisted of one individual. Our example data contain detected group sizes in the `groupsize` column of the detection data set. We specify group sizes as an `offset` term in the equation for the distance function in our call to `dfuncEstim`. Group sizes are not used in `dfuncEstim`, but will be used later in function `abundEstim` to estimate density and abundance.

Many analysts advocate dropping a small proportion of large distances to improve distance function estimation stability and thereby reduce variance. This is called right-truncation. Large but rare distances add variance to final estimates and can unduly influence (i.e., bias) distance functions, especially in small (e.g., n < 100) data sets. Analysts typically drop between 1% and 5% of the largest distances depending on histogram shape and personal preference. The 95-th quantile of the sparrow detection distances is 104.18, while the 99-th quantile is 143.2. We will right-truncate the sparrow detection distances at 150 meters, which is a nice round number just above the 99-th quantile. In `Rdistance`, we right-truncate by specifying a value for parameter `w.hi`. Parameter `w.hi` must have measurement units because it is a distance.

```r
rightTruncDistance <- units::set_units(150, "m")
dfuncSparrow<- dfuncEstim(formula = dist ~ 1 + offset(groupsize)
                          , detectionData = sparrowDetectionData
                          , likelihood = "halfnorm"
                          , w.hi = rightTruncDistance)
```

```r
dfuncSparrow
```

```
## Call: dfuncEstim(formula = dist ~ 1 + offset(groupsize), detectionData
##    = sparrowDetectionData, likelihood = "halfnorm", w.hi =
##    rightTruncDistance)
## Coefficients:
##        Estimate  SE        z         p(>|z|)      
## Sigma  49.87369  2.014173  24.76138  2.338191e-135
## 
## Convergence: Success
## Function: HALFNORM  
## Strip: 0 [m] to 150 [m] 
## Effective strip width (ESW): 62.34277 [m] 
## Probability of detection: 0.4156185
## Scaling: g(0 [m]) = 1
## Negative log likelihood: 1630.716 
## AICc: 3263.443
```

In this printout, the estimated parameter of the half-normal distance function is 49.87. When a half-normal distance function is estimated, this parameter is the standard deviation of a normal distribution if it were fitted to positive data only and if we required the mean to be zero. Approximately 68% of distances will be between 0 and this parameter. Approximately 95% of distances will be between 0 and twice this parameter. In the sparrow detection data, 69.4% of observed distances are between 0 and 49.87 meters. 93.5% of observed distances are between 0 and 99.75 meters.

Effective strip width (ESW) is the key piece of information we need to estimate abundance in cases when the distance function does not contain covariates. In `Rdistance`, ESW appears in the default printouts or it can be calculated separately using the `ESW` function.

```
ESW(dfuncSparrow)
```

```
## 62.34277 [m]
```

We interpret ESW as the distance at which the same number of targets are missed between 0 and the ESW as were sighted between ESW and infinity. A survey with imperfect detection and ESW equal to *X* effectively covers the same area as a study with perfect detection out to a distance of *X*. See the help documentation for `ESW` for details.

A visual picture of the distance function is obtained using the `plot` method.

```
plot(dfuncSparrow, nbins =40, col="grey")
```

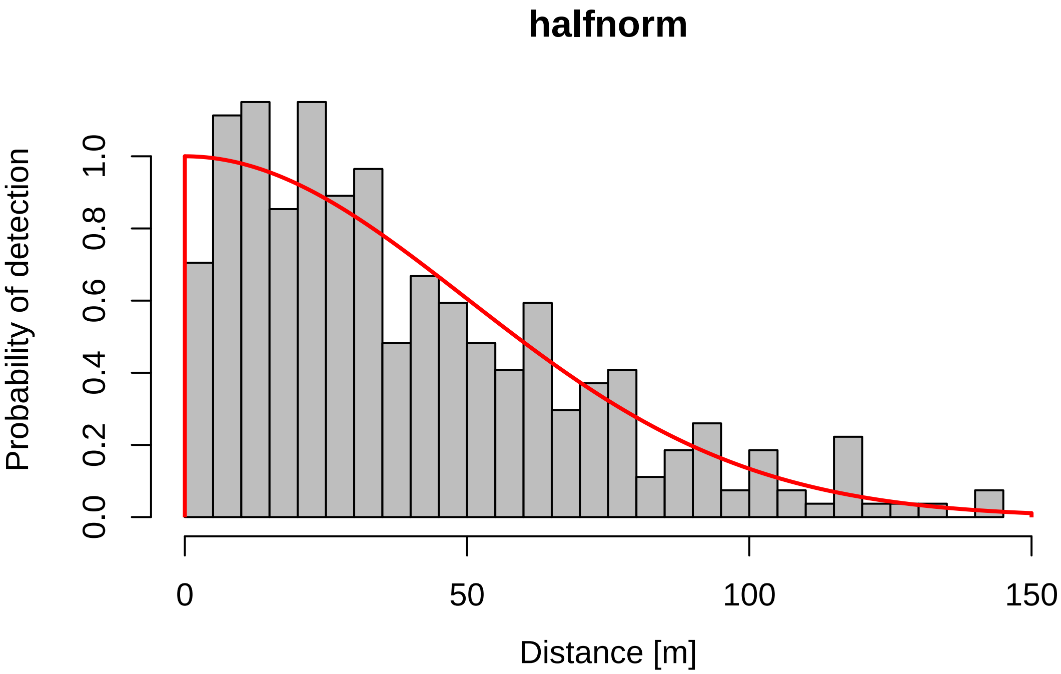

Figure 2: The half-normal distance function fitted to sparrow off-transect detection distances.

In this plot, the histogram of distances is blue and the estimated distance function is red. ESW is area under the red curve. See `help(plot.dfunc)` for ways to modify the plot (e.g., more bins, change colors, line types, etc).

4: Estimate abundance from the detection function

Abundance estimation requires additional information contained in the *site* data set, described earlier. Note that we did not use the *site* data set to estimate the half-normal sparrow detection function because we are not including covariates. Here, we estimate abundance on the 4105 km, km study area using function `abundEstim`.

In `Rdistance`, confidence intervals for true abundance are calculated using a bias-corrected bootstrapping method (see `help(abundEstim)`). As a result, confidence intervals will vary slightly between runs due to simulation error.

```
fit <- abundEstim(dfuncSparrow
                , detectionData = sparrowDetectionData
                , siteData = sparrowSiteData
                , area = saSize
                , ci = 0.95
                )
```

```
fit
```

```
## Call: dfuncEstim(formula = dist ~ 1 + offset(groupsize), detectionData
##    = sparrowDetectionData, likelihood = "halfnorm", w.hi =
##    rightTruncDistance)
## Coefficients:
##        Estimate  SE        z         p(>|z|)
## Sigma  49.87369  2.014173  24.76138  2.338191e-135
##
## Convergence: Success
## Function: HALFNORM
## Strip: 0 [m] to 150 [m]
## Effective strip width (ESW): 62.34277 [m]
##                        95% CI: 56.66586 [m] to 68.72937 [m]
## Probability of detection: 0.4156185
## Scaling: g(0 [m]) = 1
## Negative log likelihood: 1630.716
## AICc: 3263.443
##
## Average group size: 1.050992
##             Range: 1 to 3
##
## Density in sampled area: 8.265237e-05 [1/m^2]
##                   95% CI: 6.544368e-05 [1/m^2] to 0.0001003868 [1/m^2]
##
## Abundance in 4.105e+09 [m^2] study area: 339288
##                                 95% CI: 268646.3 to 412088
```

The estimated number of sparrows on the 4105 [km^2] study area is 339,288 with 95% confidence interval from 268,646 to 412,088 individuals. Estimated density is 8.265237e-05 [1/m^2], which converts to 0.8265 [1/ha] (95% CI: 0.6544 [1/ha] to 1.0039 [1/ha]). The observer's effective strip width was 62.3 [m] (95% CI: 56.67 [m] to 68.73 [m]).

Advanced Remarks

Boostrap Distribution

`Rdistance` stores density and ESW values from all bootstrap iterations in the `$B` component of abundance objects. This facilitates a couple more advanced analyses, i.e.,

- Users can append bootstrap iterations from multiple abundance objects by 'rbind-ing' the `$B` components.
- Users can compute confidence intervals using different methods (e.g., percentile)
- Users can compute the variance of other quantities that depend on density or ESW.
- Users can plot the full bootstrap distribution of density (or ESW) to inspect statistical properties.

The first six lines of the bootstrap iteration results are:

```
# Convert to hectares for readability
units(fit$B$density) <- "1/ha"
```

```
head(fit$B)
```

```
##             density  effDistance
## 1 0.7411862 [1/ha] 65.96033 [m]
## 2 0.8697759 [1/ha] 64.51214 [m]
## 3 0.8209953 [1/ha] 62.59340 [m]
## 4 0.8248241 [1/ha] 60.78737 [m]
## 5 0.9833776 [1/ha] 57.05958 [m]
## 6 0.9260318 [1/ha] 58.79328 [m]
```

For example, the bootstrap distribution of sparrow density is,

```
hist(fit$B$density
     , n = 30
     , xlab = "Density"
     , main = NULL)
# Show final density estimates, after converting to 1/ha
d <- fit$density
d.ci <- fit$density.ci
units(d) <- "1/ha"
units(d.ci) <- "1/ha"
abline(v = c(d, d.ci), col="blue")
```

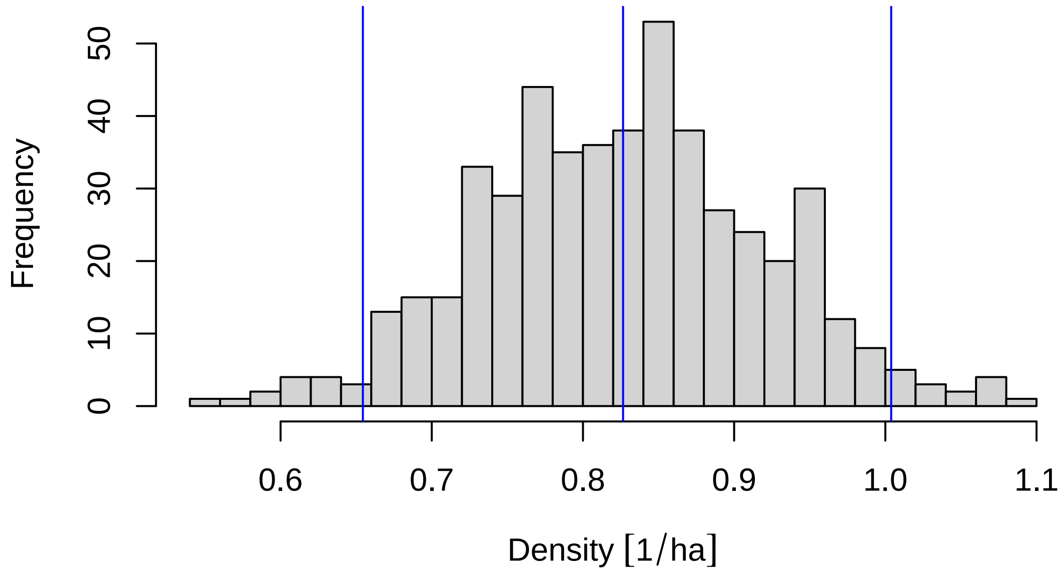

Figure 3: Bootstrap distribution of sparrow density computed in the `abundEstim` routine.

For our example data (Figure 3), the density point estimate (0.8265237 [1/ha]) and confidence limits (0.6544368 [1/ha] and 1.0038680 [1/ha]) visually plot where we expect and encompass the center of the bootstrap distribution. For other data sets, the final density estimate and confidence limits may be 'shifted' relative to the density bootstrap distribution because the bias-corrected bootstrap method attempts to remove estimation bias (distributional 'shift').

Output Measurment Units

It is possible to change the output measurement units on ESW and density by setting the `outputUnits =` parameter in the original call to `dfuncEstim`. Setting `outputUnits` only effects reporting units (output). There is no need to change units of inputs, such as distances, truncation distance, and study area size. All conversions to `outputUnits` are handled internally and automatically. For example, to output linear distances in kilometers, and density in square kilometers, issue the following:

```
dfuncSparrow <- dfuncEstim(formula = dist~1
                           , detectionData = sparrowDetectionData
                           , likelihood = "halfnorm"
                           , w.hi = rightTruncDistance
                           , outputUnits = "km")
dfuncSparrow
```

```
## Call: dfuncEstim(formula = dist ~ 1, detectionData =
##    sparrowDetectionData, likelihood = "halfnorm", w.hi =
##    rightTruncDistance, outputUnits = "km")
## Coefficients:
##        Estimate    SE           z         p(>|z|)
## Sigma  0.04987416  0.002014229  24.76092  2.365001e-135
##
## Convergence: Success
## Function: HALFNORM
## Strip: 0 [km] to 0.15 [km]
## Effective strip width (ESW): 0.06234334 [km]
## Probability of detection: 0.4156223
## Scaling: g(0 [km]) = 1
## Negative log likelihood: -807.7218
## AICc: -1613.432
```

```
fit <- abundEstim(dfuncSparrow
                , detectionData = sparrowDetectionData
                , siteData = sparrowSiteData
                , area = saSize
                , ci = 0.95
                )
```

```
fit
```

```
## Call: dfuncEstim(formula = dist ~ 1, detectionData =
##    sparrowDetectionData, likelihood = "halfnorm", w.hi =
##    rightTruncDistance, outputUnits = "km")
## Coefficients:
##        Estimate    SE           z         p(>|z|)
## Sigma  0.04987416  0.002014229  24.76092  2.365001e-135
##
## Convergence: Success
## Function: HALFNORM
## Strip: 0 [km] to 0.15 [km]
## Effective strip width (ESW): 0.06234334 [km]
##                      95% CI: 0.05674188 [km] to 0.06864928 [km]
## Probability of detection: 0.4156223
## Scaling: g(0 [km]) = 1
## Negative log likelihood: -807.7218
## AICc: -1613.432
##
## Average group size: 1
##
## Density in sampled area: 78.64156 [1/km^2]
##                 95% CI: 60.93035 [1/km^2] to 95.15893 [1/km^2]
##
## Abundance in 4105 [km^2] study area: 322823.6
##                            95% CI: 250119.1 to 390627.4
```

Additional Reading

View the full set of vignettes, as well as other teaching and tutorial materials, on the `Rdistance` wiki: https://github.com/tmcd82070/Rdistance/wiki.