The Dendrometry Package

J. H. Gove

USDA Forest Service, Northern Research Station, 271 Mast Road, Durham, NH 03824 USA

(603) 868-7667; e-mail: jgove@fs.fed.us

Thursday 30th May, 2019

3:57pm

Contents

1 Introduction **1**

2 Example **2**

3 Use with sampSurf **4**

3.1 Creating "standingTree" objects 5

3.2 Creating "standingTrees" objects 7

4 Summary **8**

Bibliography **9**

1 Introduction

The dendrometry package is a very small set of routines that begin to bring some functionality to **R** for reading and parsing old STX (Grosenbaugh, 1974) dendrometry files. The package is quite incomplete as it only handles measurements taken using the Barr and Stroud (B&S) FP-12 or FP-15 dendrometers. It uses some old Fortran code that was developed to read in the dendrometer measurements, translate them to diameters, heights, etc., and subsequently fitted several splines (Natural Cubic, Hermite and Akima) to the individual trees. The code has been adapted so that it can be called from **R**, and the spline algorithms have been removed because they were from licensed Fortran libraries at the time. However, the **R** spline routines are now used for fitting several spline functions rather than the old Fortran libraries. All but the Akima spline are fit, and this can be added later using the `Akima` package if desired[1]

The old Fortran code is written in Fortran-90 and was not originally designed to be anything but a stand-alone program, but has been modified to interface with **R**. Please note that the actual Fortran routines that convert the dendrometry readings to range, diameter, etc. were taken directly from STXMOD (Space, 1974).[2]

[1] Akima splines may be the best fitting splines from past experience, so it might be worth considering. I have not added these since it requires another package to be available and I did not want this at the current time.

[2] The version is actually from 1990, and has contributions from Rennie (1977) and others as well.

The input file of dendrometry records should be in STXMOD format and should include the 10 control cards that are used in a normal run of STXMOD (Space, 1974; Rennie, 1977).[3]

As mentioned above, this is a *very* rudimentary package, and does not claim to do anything fancy other than get the data into **R** and allow you to plot some profiles. Suggestions for additions or modifications are welcome, as is code contributed from others to make this more useful.

Version 0.2-1 of the package (30-May-2019) adds the ability to convert the dendrometered tree measurements directly to `sampSurf` objects that can be used in subsequent simulations. This is covered in § 3.

2 Example

The main interface routine is called `SplineVolume` after the original Fortran routines at the core. The help files give more detail on each of the routines in the package. For now, assume that a file in valid STXMOD format resides in a `./data` subdirectory under the **R** working directory for this session.[4] In addition, there should be another subdirectory (e.g., `/Rwork/reports`) to hold the output files generated by the Fortran code. In this case, the file is named `odr_wp_1977.stx`, so...

```
R> args(SplineVolume)
```

```
function (fileName = "odr_wp_1977.stx", filePath = getwd(), dataPath = "data",
    reportPath = "reports", splineMethods = c("natural", "monoH"),
    ...)
NULL
```

```
R> dendro = SplineVolume()
R> names(dendro)
```

```
[1] "z"        "trees"    "segments"
```

```
R> head(dendro$trees)
```

```
  TreeNo Spp dbhob dbThick isForked nReadings totLen LenFlag conicVol natVol
1      1  WP  15.8     1.2    FALSE         7 68.644   FALSE   34.806 35.246
```

[3] Also please see the comments at the beginning of the STXMOD program code.
[4] I.e., if you are working out of `/Rwork` then the file would be in `/Rwork/data`.

```
2       2  WP  18.4     1.6    FALSE          7 81.058    FALSE    52.706 55.137
3       3  WP  16.9     1.3    FALSE          7 79.567    FALSE    49.287 51.235
4       4  WP  18.8     1.6    FALSE          7 81.078    FALSE    60.091 65.381
5       5  WP  15.0     1.3    FALSE          8 69.704    FALSE    37.256 37.798
6       6  WP  18.3     1.5    FALSE          7 76.003    FALSE    49.882 48.481
  monVol
1 34.877
2 53.471
3 49.906
4 62.603
5 37.652
6 49.889
```

The output from the `SplineVolume` function is returned in a list. The first component holds the information passed in the call to the Fortran in list format, and will not be of much interest. The second component is a tree level summary with one record per tree, as shown above. The final component holds all of the individual dendrometer and segment or section values for each tree. For example,

```
R> nrs = dendro$trees[ dendro$trees$TreeNo==1, 'nReadings']
R> dendro$segments[1:nrs,]
```

```
  TreeNo Reading Spp    dob    dib prod tgrad fgrad sinelv instRange hgtHoriz
1      1       7  WP  0.100  0.096   TW -99.9 -99.9 0.6310     0.001   62.191
2      1       6  WP  6.388  6.084   TW  48.4  61.5 0.4830    86.758   41.915
3      1       5  WP  8.685  8.230   ST  47.7  65.8 0.4149    83.538   34.669
4      1       4  WP 11.723 11.015   ST  45.9  72.0 0.2220    76.400   16.965
5      1       3  WP 14.941 13.864   ST  45.5  78.4 0.0073    75.004    0.548
6      1       2  WP 15.800 14.600   ST   0.0  15.8 4.0000     0.001   -3.452
7      1       1  WP 18.600 16.883 <NA> -99.9  18.6 3.0000     0.001   -6.452
  segLen height conicVol natVol monVol
1 20.276 68.644    1.364  1.349  1.386
2  7.246 48.368    2.040  2.064  2.064
3 17.704 41.122    9.004  9.190  9.166
4 16.418 23.418   13.917 14.304 13.877
5  4.000  7.000    4.420  4.327  4.359
6  3.000  3.000    4.062  4.013  4.025
7  0.000  0.000    0.000  0.000  0.000
```

In the above listing for the first tree we see that each segment is listed in order from top to bottom within the tree. Diameter, height, instrument information and volumes are listed.

The same information listed in the tree and segments data frames above is also available in separate files in the report subdirectory (`*.tre` and `*.seg` files). In addition, a nicely formatted report (`*.rpt`) is also generated for each tree that is reminiscent of some of the STX output.

Finally, the package enables some rudimentary plotting of tree profiles.

```
R> taperPlot(dendro, grid=TRUE, TreeNos = c(1,2,7,8,9,10), diamType='xsection',
+            standUp=TRUE, method='mono')
```

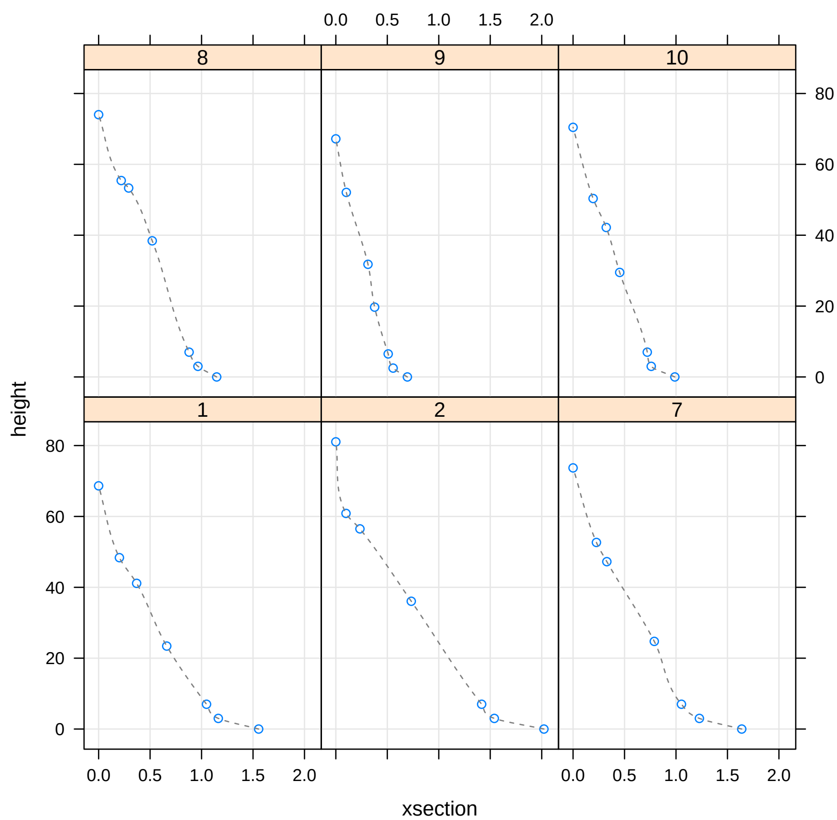

Figure 1: Some tree profiles in cross-sectional area for dendrometered trees.

3 Use with sampSurf

Dendrometered trees can be used for simulation in package **sampSurf**. Doing so allows for real taper data to be used in the simulations. The construction of a "standingTree" object in **sampSurf** is straightforward because there is a class constructor method that allows creating a "standingTree" object directly from the taper data. Dendrometry measurements provide a perfect source of accessible taper data for this purpose. When this option is used to create a "standingTree" object, a (cubic) spline is fitted to the taper measurements in order to determine the tree's volume. It is also used to interpolate diameter or heights and determine volumes in cases where Monte Carlo subsampling methods are used in simulation.

There are two new[5] routines that are used to accomplish this as described below. Both of them take a list object as output from `SplineVolume` as their first argument.

3.1 Creating "standingTree" objects

First, the function `stx2sT` is used to create one "standingTree" object. Note that one can choose the input and output units, so measurements taken in "English" may be left as is, or translated to "metric".[6] More options can be found in the documentation using "`?stx2sT`." The function name itself is a little bit of a misnomer since this routine does not translate raw STX files into "standingTree" objects (the "sT" component of the function name); `SplineVolume` does that, but the genesis is still the original STX format on disk—hence the name.

An example converting one tree from the above results of the `SplineVolume` run follows...

```
R> args(stx2sT)
```

```
function (sv.list, treeNo = 1, centerOffset = c(x = 0, y = 0),
    units.in = "English", units.out = "metric", prefix = "odr77:tree",
    runQuiet = TRUE, ...)
NULL
```

```
R> odr77.t1 = stx2sT(dendro, 1, units.out = 'English', runQuiet = FALSE)
```

```
Bark deductions present: STX volumes are inside bark.
Tree volumes...
  Spline volume from standingTree = 35.781055
  Average tree volume (3 methods below) = 34.976333
  SplineVolume conicVol = 34.806
  SplineVolume natVol = 35.246
  SplineVolume monVol = 34.877
Tree's pith location centerOffset: x = 0 y = 0
```

[5]These have been added to the original package (Version 0.1-1) in May-2019.
[6]These are the names of the two systems of units as defined in `sampSurf`.

Note in the above output that the spline fit from `sampSurf` gives a volume close to the spline and conic section volumes from `SplineVolume`. The "standingTree" object is then summarized as usual via...

```
R> odr77.t1$stree
```

```
Object of class: standingTree
------------------------------------------------------------

------------------------------------------------------------

Stem...
  Species:  WP
  units of measurement:  English
  spatial units:  NA
  location...
    x coord:  0
    y coord:  0
    (Above coordinates are for dbh center)
  Spatial ID: odr77:tree.1

standingTree...
  Butt diameter = 1.4069167 feet (16.883 in)
  Top diameter = 0.008 feet (0.096 in)
  DBH = 1.2166667 feet (14.6 in)
  Basal area =  1.0909016 square feet
  Height = 68.644 feet
  Tree volume = 35.781055 cubic feet
  Tree surface area = 155.17461 square feet
  Taper parameter =  NULL

Taper (in part)...
    diameter height
7 1.40691667  0.000
6 1.21666667  3.000
5 1.15533333  7.000
4 0.91791667 23.418
3 0.68583333 41.122
2 0.50700000 48.368
```

The output above shows the tree as converted to a "standingTree" object. The `taper` slot above shows part of the section measurements from the first tree illustrated in § 2. Note the differences here, which are important. A “standingTree” object stores the taper in reverse order from the returned `SplineVolume` object. And most importantly, diameter is in *feet* for “English” units. `sampSurf` always converts diameters to the same units as height in order to have everything to scale for graphical display.

3.2 Creating “standingTrees” objects

The second additional routine is closely related to the first, it is `stx2sTs`. This routine converts an entire list of dendrometered trees as output from `SplineVolume` to a `sampSurf` “standingTrees” container object. It will simulate tree locations within a `sampSurf` “bufferedTract” object such that all trees are located within the buffer area; a spatial inhibition process is used to determine the locations. More information on the function may be found in the help system by `?stx2sTs`. The upshot of all this is that one must also pass a “bufferedTract” object as an argument to this routine in order to be able to calculate the locations. An example follows. . .

```
R> bufftr = initTract()
R> args(stx2sTs)
```

```
function (sv.list, buffTr, units.in = "English", units.out = "metric",
    prefix = "odr77:tree", inhibitDist = 3, runQuiet = TRUE,
    ...)
NULL
```

```
R> odr77.sts = stx2sTs(dendro, bufftr, units.out = 'metric',
+                       prefix = 'odr77:tree', runQuiet = FALSE)
```

```
Processing tree number...1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
```

```
R> sapply(odr77.sts, class)
```

```
    strees.list          strees          prefix
         "list" "standingTrees"     "character"
```

Note that the entire set of trees that were output from `SplineVolume` have been processed and converted. Also note that output includes both a “standingTrees” object and a list of “standingTree” objects. The latter is included in case one wants to concatenate or sample from the population of dendrometered trees. In this case, do what needs to be done, then create a new “standingTrees” container object from the results; this will insure that the bounding box and other spatial attributes are correctly computed in the new container object.

The "standingTrees" object has the following summary information...

```
R> odr77.sts$strees
```

```
Object of class: standingTrees
------------------------------------------------------------

------------------------------------------------------------
Container class object...
  Units of measurement:  metric

  Encapulating bounding box...
        min       max
x 53.733418 132.38422
y 33.409474 128.51346

  There are 12 trees in the population
  Population tree volume =  14.139768 cubic meters
  Population tree surface area =  200.54094 square meters
  Average volume/tree =  1.178314 cubic meters
  Average surface area/tree =  16.711745 square meters
  Average height/tree =  24.301272 meters
(**All statistics exclude NAs)
```

Lastly, we can plot the set of "standingTrees" on the "bufferedTract" as...

```
R> plot(bufftr, gridColor = transparentColorBase('lightsteelblue4', .5))
R> plot(odr77.sts$strees, add = TRUE)
```

The result is displayed in Figure 2.

4 Summary

The `Dendrometry` package is a start to getting B&S dendrometry information compatible with STX into R. It provides a rudimentary set of routines for translating and plotting B&S measurements.

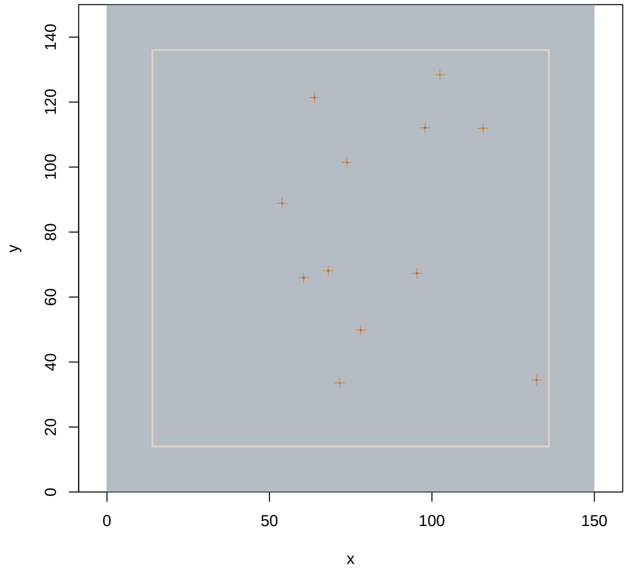

Figure 2: A small population of "standingTrees" objects on a buffered tract.

At present, the package has only been tested in Linux machines. The package builds fine using the `Makefile` for the Fortran code. However, compiling on other platforms will be different, specifically for Microsoft.[7]

Finally, please note that this package is not a substitute for STX. It does not do any sampling calculations, etc., it only imports the data into **R** and calculates a few volume related quantities.

References

L. R. Grosenbaugh. STX 3-3-73: Tree content and value estimation using various sample designs, dendrometry methods, and v-s-l conversion coefficients. Research Paper SE-117, U.S. Forest Service, 1974. 1

J. C. Rennie. STX: a computer system for processing timber inventory data. Technical report, 1977. Unpublish manuscript. 1, 2

J. C. Space. 3-P forest inventory: Design, procedures, data processing. Technical report, U.S. Forest Service State and Private Forestry, Southeastern Area, 1974. 1, 2

[7]It would be great if someone who uses this platform could contribute the `Makefile.win` file that would assure the Fortran compiles correctly for Windows.