

# Competition Indices and Other Stuff in R

Greg Johnson (greg@nosnhoj.org)

Greg Johnson Biometrics LLC

2025-04-13



# Introduction

Three [GitHub](#) repositories with some things of potential interest:

- 1 [Biometrics Utilities](#)
- 2 [National Scale Volume and Biomass](#)
- 3 [Acadian Variant of FVS](#)

Solicit feedback and collaboration!

# Biometrics Utilities

The repository:

<https://github.com/gregjohnsonbiometrics/biometrics.utilities>

contains the source and R packages for a collection of metrics (mostly focused on competition indices) that are useful in describing tree and stand conditions. Several of them are somewhat complex and computationally expensive. The code translates these metrics into C++ for computational efficiency and hopefully sound implementations of the concepts.

## Distance Independent Metrics

- Basal Area (ba)
- Crown Closure at Tree Tip (cch)
- Basal Area in Larger Trees (bal)
- Crown Competition Factor in Larger Trees (ccfl)
- Dominant Height (dominant\_height)
- Quadratic Mean Diameter (qmd)
- Wilson's Relative Spacing (relative\_spacing)
- Crown Competition Factor (ccf)
- Curtis' Relative Density (curtis\_rd)
- Reineke's Stand Density Index (reineke\_sdi)
- Glover and Hool Index (Glover\_Hool)\*

\*The Glover and Hool index is the ratio of a tree's basal area to the basal area of the tree of mean diameter.

## Distance Dependent Metrics

- Clark-Evans R Aggregation Index (Clark\_Evans\_R)\*
- Hegyi's Distance-weighted size ratio (Hegyi)\*
- Arney's Competitive Stress Index (Arney\_CSI)
- Area Potentially Available (APA)

\*Clark-Evans and Hegyi have edge corrections if boundaries are provided.

## General Utilities

- Maximum Crown Width Estimate (mcw)
- Fit Height-DBH Curves (hd\_fit)
- Predict Height from DBH (hd\_predict)

## Design Philosophy

Keep the API simple (minimal dependencies) and consistent:

- tree attributes are passed as simple vectors
- imperial and metric options available where applicable
- flexibility
- return values usually simple vectors in original tree order if applicable

## Example: `dominant_height()`

```
library(biometrics.utilities)
data(treelist)
```

```
# compute the height of the 40 largest trees by dbh
dominant_height( treelist$height,
                 treelist$dbh,
                 treelist$tpa, 40, 0 )
```

```
[1] 129.1463
```

```
# compute the height of the 100 largest trees by height
dominant_height( treelist$height,
                 treelist$dbh,
                 treelist$tpa, 100, 1 )
```

```
[1] 122.0781
```

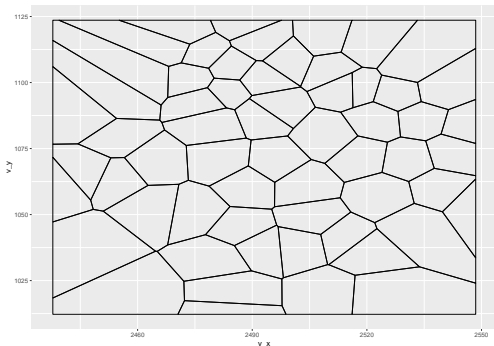
# Biometrics Utilities

## Example: APA()

```
library(biometrics.utilities)
library(ggplot2)
library(dplyr)

data("treelistxy")
min_x <- min(treelistxy$x)
min_y <- min(treelistxy$y)
max_x <- max(treelistxy$x)
max_y <- max(treelistxy$y)
poly_x <- c(min_x, max_x)
poly_y <- c(min_y, max_y)
p <- APA_Polygons( treelistxy$tree,
                  treelistxy$x,
                  treelistxy$y,
                  treelistxy$dbh,
                  poly_x, poly_y, F )
```

```
p %>% ggplot( aes( v_x, v_y, group=tree_id )) + geom_path()
```



# Biometrics Utilities

## Example: example code in `cch()` help

```
library(biometrics.utilities)
library(dplyr)

# use Hann's (1998) largest crown width equations Douglas-fir,
# western hemlock, bigleaf maple, and red alder
lcw.parms <- data.frame( species=c(202,263,312,351),
                        b0=c(0.0,      0.0, 0.0,      0.3227140 ),
                        b1=c(0.004363240, 0.0, 0.0,      0.0      ),
                        b2=c(0.6020020,  0.0, 1.470180, 0.0      ))

# estimate distance above crown base: use Hann 1999 for Douglas-fir,
# Marshall et al. 2003 for western hemlock, and assuming 0.0 for bigleaf maple and red alder
dacb.parms <- data.frame( species=c(202,263,312,351),
                        d0=c(0.06200, 0.355270, 0.0, 0.0 ) )

# use crown profile parameters for Douglas-fir
cch.parms <- data.frame( species=202, b0=0.929973, b1=-0.1352120, b2=-0.0157579 )

temp <- treelist %>% mutate( mcw.hat=mcw( species, dbh ) ) %>%
  inner_join( lcw.parms, by="species" ) %>% inner_join( dacb.parms, by="species" ) %>%
  mutate( crown_length = height - htlc, cr = crown_length/height,
         lcw = mcw.hat * cr^(b0+b1*crown_length+b2*(dbh/height)), dacb = d0 * crown_length )

cch( temp$species, temp$dbh, temp$height, temp$crown_length, temp$dacb, temp$lcw, temp$tpa, cch.parms )

[1] 0.000000000 1.109380717 0.000000000 0.153726623 0.005873616 0.021481633
[7] 0.012059208 0.483895072 0.236987791 0.002627645 0.042973879 0.000000000
[13] 0.005873616 0.309891279 0.284608394 0.858692077 0.062852671 0.088125682
[19] 0.102715755 0.021481633 0.000000000 0.913079088 0.827632497 0.001786221
```

## Documentation

- Each function has a full `man` page of documentation available in R (e.g., `?dominant_height` or `help(dominant_height)`).
- The package description (under `biometrics.utilities-package` in `help`) contains a list of references for all functions.
- The function help gives a brief description of the metric, a literature citation, and a working example.

## Arney\_CSI() compute Arney's competitive stress index.

### Description

Compute Arney's Competitive Stress Index as described in Arney (1973). CSI is the sum of percentage competing trees crown area overlapping a subject tree to the subject tree's crown area.

$$CSI_i = 100 \sum \frac{AO_j}{CA_i}$$

where  $AO_j$  is the area of overlap of tree  $j$  on subject tree  $i$ ,  $CA_i$  is the crown area of the subject tree  $i$ , and  $CSI_i$  is the competitive stress index for tree  $i$ .

This version currently does not adjust for edge effects.

### Usage

```
Arney_CSI(x, y, dbh, mcw)
```

### Arguments

- `x` : double | vector of x coordinates of trees on plot (in same units as mcw)
- `y` : double | vector of y coordinates of trees on plot (in same units as mcw)
- `dbh` : double | vector of diameter at breast height
- `mcw` : double | vector of maximum crown width for the tree record
- `imperial_units` : bool | TRUE = imperial, FALSE = metric (default)

### Value

Returns the Arney's CSI for each tree.

### Examples

# Biometrics Utilities

pc is a data.frame containing data from the Panther Creek ground-truth measurement plots.

Example of computing plot-level statistics and metrics:

```
plot.summary <- pc %>% group_by( PLOTID, RefYear ) %>%
  summarise( ba      = ba( DBH, tpa ),
             Dq      = qmd(DBH, tpa ),
             dht.40.dbh = dominant_height(HT, DBH, tpa, 40, 0 ),
             dht.100  = dominant_height(HT, DBH, tpa, 100, 1 ),
             lorey.ht  = dominant_height(HT, DBH, tpa, 40, 2 ),
             rel.spacing = relative_spacing( tpa, dht.40.dbh ),
             ccf      = ccf( lcw, tpa ),
             Rd      = curtis_rd( DBH, tpa ),
             SDI     = reineke_sdi( DBH, tpa))

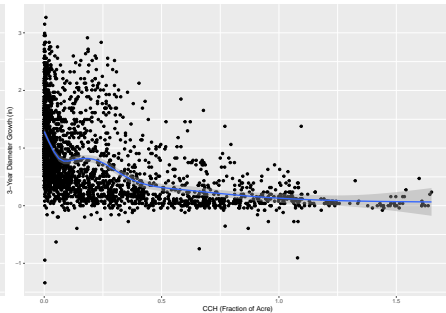
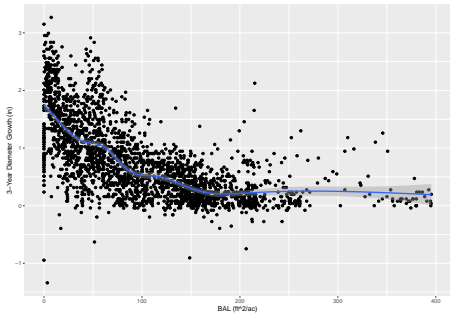
knitr::kable( plot.summary[1:10,], digits = 2 )
```

PLOTID	RefYear	ba	Dq	dht.40.dbh	dht.100	lorey.ht	rel.spacing	ccf	Rd	SDI
100401	2009	56.67	5.92	42.23	39.74	37.77	0.29	91.39	23.30	127.85
100401	2012	83.97	7.33	46.57	46.97	43.86	0.26	100.59	31.02	174.08
100402	2009	34.30	5.66	43.49	37.33	43.85	0.34	55.43	14.42	78.74
100402	2012	49.73	7.09	47.93	45.40	46.94	0.32	65.70	18.67	104.41
100405	2009	24.06	4.15	36.10	35.17	31.88	0.36	53.87	11.82	62.46
100405	2012	39.44	5.36	41.63	43.06	39.25	0.32	60.10	17.03	92.50
109101	2009	209.27	13.98	117.66	106.26	109.60	0.13	117.54	55.97	336.11
109101	2012	213.05	14.48	124.55	113.87	117.60	0.12	110.50	55.98	337.44
109102	2009	215.63	15.43	131.05	110.82	119.74	0.12	95.76	54.89	333.10
109102	2012	226.17	15.80	133.64	113.54	122.67	0.12	97.01	56.89	346.10

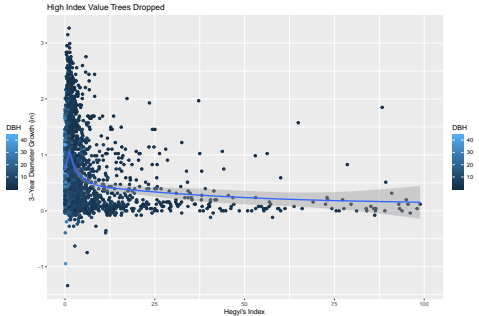
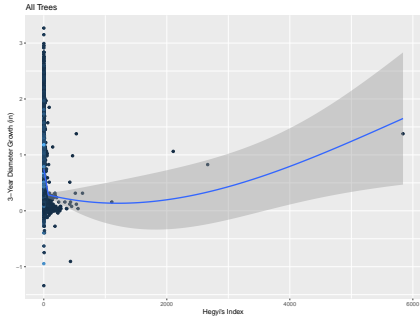
Example of computing tree-level statistics and metrics:

```
pc.tree.level <- pc %>% reframe(  
  bal = bal( DBH, tpa ),  
  cch = cch( fia, DBH, HT, HT*cr, dacb, lcv, tpa, cch.parms ),  
  ccfl = ccfl( DBH, mcw( fia, DBH ), tpa),  
  CSI = Arney_CSI( X*3.28084, Y*3.28084, DBH, mcw( fia, DBH )),  
  CE = Clark_Evans_R_circle( X, Y, 0.08042477, UTMXpc1[1], UTMYPc1[1], 16.0 ),  
  Hegyi = Hegyi(X, Y, DBH, 0, 0, F ),  
  GH = Glover_Hool( DBH, tpa ),  
  .by=c(PLOTID, RefYear) )
```

# Biometrics Utilities



# Biometrics Utilities



Correlation coefficients with 3-Year diameter growth:

DG	bal	cch	ccfl	CSI	Hegy	CE	GH	DBH
1.00	-0.64	-0.50	-0.64	-0.29	-0.05	0.34	0.02	-0.11

# National Scale Volume and Biomass (NSVB)

The repository:

<https://github.com/gregjohnsonbiometrics/nsvb>

contains the National Scale Volume and Biomass Estimators (NSVB) that were implemented in R by Virginia Tech. The performance of the code is slowed by the interpretive nature of R. To provide a higher performance API the Virginia Tech code is implemented in C++23.

# National Scale Volume and Biomass (NSVB)

## Biomass Components Estimated

The dry weight in pounds is estimated for the following tree components:

- Wood
- Bark
- Branches
- Foliage
- Total

Additionally, above ground biomass (AGB) is computed.

## C++ Application Programming Interface

The following C++ functions are exposed:

- `biomass_components()`
- `compute_green_tons()`
- `compute_volib()`
- `compute_volob()`

`biomass_components()` returns a `BIOMASS_COMP` structure defined in `nsvb.hpp`. If an FIA numeric species code is not found, the system uses 999 (other or unknown live tree).

# National Scale Volume and Biomass (NSVB)

## R Package (`nsbvR`)

An R package accessing the NSVB API is provided in the repository, and the most recent Windows Binary and tarballs are in the root directory of the repository. `Rcpp` is required to install and run the package.

The R package provides the following R equivalents to the C++ functions:

- `biomass_components()`
- `green_tons()`
- `volib()`
- `volob()`

`biomass_components()` returns a `data.frame` with `wood`, `bark`, `branch`, `foliage`, `total`, and `above_ground_biomass` estimates.

# Acadian Variant of FVS (ACD7)

Since last year, ACD7, a variant of FVS for the northeast (especially Maine, Vermont, New Hampshire, New Brunswick, and Nova Scotia) that I reported on, has been refined (with updates to mortality and crown recession). The repository is at:

<https://github.com/gregjohnsonbiometrics/acd7>

and includes the C++ source code as well as a working R package (acd7).

# Thanks

All of the repositories are free to download and are covered under the GNU General Public License Version 2.

Comments, suggestions, edits, additions are most welcome!