



DỰ ÁN “HỖ TRỢ CHƯƠNG TRÌNH ĐÁNH GIÁ VÀ THEO DÕI LÂU DÀI
TÀI NGUYÊN RỪNG VÀ CÂY PHÂN TÁN TOÀN QUỐC TẠI VIỆT NAM”

PROJECT " SUPPORT TO NATIONAL ASSESSMENT AND LONG TERM
MONITORING OF THE FOREST AND TREE RESOURCES IN VIETNAM (NFA)"



OPEN FORIS CALC

System for data processing in FAO-Vietnam NFA Project,
Description of settings and scripts of Open Foris Calc

Lauri Vesa, Nguyen Dinh Hung, Cosimo Togna

Hanoi, January 2015
(v. 2)

Contents

| | |
|----------------------------------|----|
| Preface | 2 |
| 1. Settings | 2 |
| 1.1. Sampling design | 2 |
| 1.2. Aggregation functions | 3 |
| 2. Calculation | 3 |
| 2.1. List of modules | 3 |
| 2.2. Categories | 4 |
| 2.3. R scripts | 6 |
| 2.3.1. Tree | 6 |
| 2.3.2. Dead wood | 10 |
| 2.3.3. Stump | 11 |
| 2.3.4. Bamboo | 14 |
| 2.3.5. Liana (climbers) | 15 |
| 2.3.6. Regeneration | 16 |
| 2.3.7. Plot | 19 |
| 2.4. Error script | 22 |

PREFACE

Open Foris Calc is a robust, modular browser-based tool for data analysis and results calculation. It allows expert users to write custom R modules to perform country/inventory-specific calculations. This document contains brief description of these modules written for National Forest Inventory (NFI) by the FAO-Vietnam NFA Project in 2014.

Calc works for point sampling with a combination of the following: stratification/cluster/double sampling. The input metadata and data comes from Open Foris Collect, and it provides a flexible way to produce aggregated results for any defined area of Interest. The aggregated results can be analyzed and visualized through open source software Saiku.

The calculation chain and individual calculation modules will be further developed in Vietnam, so these will change. However, this document aims to show the current progress, and hopefully it also works as a model when tailoring Calc into new forest inventories and countries.

R code written for the NFA Project contains the following external R library:

- *LMFOR* library in module 2.3 “*Tree - Est. height*”. This R library is available at <http://cs.uef.fi/~lamehtat/rcodes.htm>

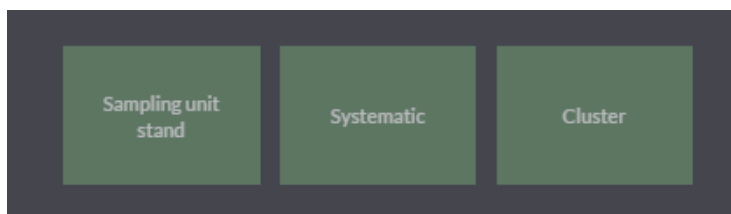
This Open Foris calc code contains some outputs that are written in CSV format files into a folder, see module 2.12 “*Tree - Total carbon & Tree species abundance*” because all results cannot be retrieved through Saiku, yet. Therefore if this code is applied as such, the output folder must exist.

How to cite this source code? As follows:

Vesa, L., Hung, N.D. and Togna, C. (2015). Open Foris Calc scripts. FAO-Vietnam NFA Project.

1. SETTINGS

1.1. Sampling design



Two options:

- a) Sampling unit weight script (Area of interest given without water):

```
stand$weight = ifelse ( stand$stand_id == 'A' & stand$forest_status != '630', 1 , 0 );
```

- b) Sampling unit weight script (Area of interest includes water):

```
. stand$weight = ifelse ( stand$stand_id == 'A', 1 , 0 );
```





1.2. Aggregation functions

Aggregation function represents the formula of the plot area for the entities which need to be aggregated following the sampling design.

| Entity | Plot area script |
|------------------|---|
| bamboo | <code>bamboo\$plot_area <- 0.01</code> |
| dead_wood | <code>dead_wood\$plot_area <- 0.05</code> |
| liana | <code>liana\$plot_area <- 0.05</code> |
| regen | <code>regen\$plot_area <- (2 * pi* 1.5*1.5) / 10000</code> |
| stump | <code>stump\$plot_area <- 0.05</code> |
| tree | <code>tree\$plot_area <- with (tree , ifelse(dbh >= 40, 0.1 , ifelse(dbh >= 20 , 0.05 , 0.01)));</code> |

2. CALCULATION

2.1. List of modules

| | | | | | | |
|--|--|--|---|---|--|---|
|  |  | <input type="checkbox"/> Stand - Major forest status | <input type="checkbox"/> Stand - FRA class | <input type="checkbox"/> Stand - IPCC class | <input type="checkbox"/> Tree - DBH class (04) | <input type="checkbox"/> Tree - DBH class (20) |
|  |  | <input type="checkbox"/> Plot - Major Forest Status | <input type="checkbox"/> Tree - Count | <input type="checkbox"/> Tree - Basal area | <input type="checkbox"/> Tree - Est. height | <input type="checkbox"/> Tree - Height class (02) |
| <input type="checkbox"/> Tree - Bole height | <input type="checkbox"/> Tree - Volume | <input type="checkbox"/> Tree - Bole volume | <input type="checkbox"/> Tree - AG Biomass | <input type="checkbox"/> Tree - BG Biomass | <input type="checkbox"/> Tree - Total biomass | |
| <input type="checkbox"/> Tree - AG Carbon | <input type="checkbox"/> Tree - BG Carbon | <input type="checkbox"/> Tree - Total C & Sp. abund. | <input type="checkbox"/> DW - Volume | <input type="checkbox"/> DW - Biomass | <input type="checkbox"/> DW - Carbon | |
| <input type="checkbox"/> Stump - Count | <input type="checkbox"/> Stump - DBH estimate | <input type="checkbox"/> Stump - D0.15 | <input type="checkbox"/> Stump - Volume | <input type="checkbox"/> Stump - AG Biomass | <input type="checkbox"/> Stump - Height before felling | |
| <input type="checkbox"/> Stump - Tree AG biomass | <input type="checkbox"/> Stump - BG Biomass | <input type="checkbox"/> Stump - Total biomass | <input type="checkbox"/> Stump - Removal volume | <input type="checkbox"/> Stump - Carbon | <input type="checkbox"/> Stump - Removal biomass | |
| <input type="checkbox"/> Bamboo - Count | <input type="checkbox"/> Bamboo - AG Biomass | <input type="checkbox"/> Bamboo - BG Biomass | <input type="checkbox"/> Bamboo - Biomass | <input type="checkbox"/> Bamboo - Carbon | <input type="checkbox"/> Liana - Count | |
| <input type="checkbox"/> Liana - Volume | <input type="checkbox"/> Liana - Biomass | <input type="checkbox"/> Liana - Carbon | <input type="checkbox"/> Regen - Count total | <input type="checkbox"/> Regen - Basal area | <input type="checkbox"/> Regen - Volume | |
| <input type="checkbox"/> Regen - AG Biomass | <input type="checkbox"/> Regen - BG Biomass | <input type="checkbox"/> Regen - Biomass | <input type="checkbox"/> Regen - AG Carbon | <input type="checkbox"/> Regen - BG Carbon | <input type="checkbox"/> Regen - Carbon | |
| <input type="checkbox"/> Plot - Count | <input type="checkbox"/> Plot - Count (Accessible) | <input type="checkbox"/> Plot - Canopy closure | <input type="checkbox"/> Plot - Basal area | <input type="checkbox"/> Plot Volume | <input type="checkbox"/> Plot - Biodiv. Sp. count | |
| <input type="checkbox"/> Plot - Biodiv. Shannon | <input type="checkbox"/> Plot - Biodiv. Simpson | | | | | |

2.2. Categories

| | |
|----------------|---|
| # | 1.1 |
| Caption | Stand - Major forest status |
| Type | Category |
| Entity | stand |
| Purpose | 'Major forest status' class, bases on 'Forest status' code |
| Code | <pre>stand\$stand major forest status <- with(stand, ifelse(forest_status < 200 , 1, # Primary forest ifelse(forest_status < 300, 2, # Secondary forest ifelse(forest_status < 400, 3, # Plantation ifelse(forest_status < 500, 4, # Bushes, grasses land ifelse(forest_status < 600, 5, # Agriculture land ifelse(forest_status == 630, 6, 7) # Water, Other land)))))));</pre> |

| | |
|----------------|---|
| # | 1.2 |
| Caption | Stand - FRA class |
| Type | Category |
| Entity | stand |
| Purpose | FRA class, bases on 'Forest status' code ('1' Forest land, '2' Other wooded land, ['3' Vietnam OLWTC class], '4' Other land, '5' Water) |
| Code | <pre>stand\$fra class <- with(stand, ifelse(forest_status < 420, 1, # forest land ifelse(forest_status < 440, 2, # OWL ifelse(forest_status == 520 forest_status == 640, 2, # Vietnamese class OLWTC (other land with trees has reserved code 3, but this goes into OWL in this calculation ifelse(forest_status == 630, 5, 4) # Water; Other land))));</pre> |

| | |
|----------------|---|
| # | 1.3 |
| Caption | Stand - IPCC class |
| Type | Category |
| Entity | stand |
| Purpose | IPCC class, bases on 'Forest status' code ('1' Forest land, '2' Grass land, '3' Cropland, '4' Settlements, '5' Wetland) |
| Code | <pre>stand\$ipcc_class = with (stand, ifelse(forest_status ==160 forest_status == 630, 5, ifelse(forest_status < 440, 1, ifelse(forest_status == 440, 2, ifelse(forest_status < 600, 3, 4)))));</pre> |

| | |
|----------------|---|
| # | 1.4 |
| Caption | Tree - DBH class (04) |
| Type | Category |
| Entity | Tree |
| Purpose | Tree DBH classes in 4 cm intervals. '1' 6 - 9.9 cm, '2' 10 - 13.9 cm, '3' 14 - 17.9 cm, '4' 18 - 21.9 cm, '5' 22 - 25.9 cm, '6' 26 - 29.9 cm, '7' 30 - 33.9 cm, '8' 34 - 37.9 cm, '9' 38 - 41.9 cm, '10' 42 - 45.9 cm, '11' 46 - 49.9 cm, '12' 50 - 53.9 cm, '13' 54 - 57.9 cm, '14' 58+ |
| Code | <pre>tree\$dbh_4 <- trunc(((tree\$dbh - 6.0)+4)/4 ,0) tree\$dbh_4 <- ifelse(tree\$dbh_4 >14, 14, tree\$dbh_4) tree\$dbh_4 <- ifelse(tree\$dbh_4 <1, 1, tree\$dbh_4) tree\$dbh_class04 <- as.integer(tree\$dbh_4)</pre> |

| | |
|---------|--|
| # | 1.5 |
| Caption | Tree - DBH class (20) |
| Type | Category |
| Entity | tree |
| Purpose | Tree DBH classes ('1' < 20 cm, '2' 20 - 39.9 cm, '3' >= 40 cm) |
| Code | <pre>tree\$dbh_class20 <- with(tree, ifelse(dbh < 20, 1, ifelse(dbh < 40, 2, 3)));</pre> |

| | |
|---------|--|
| # | 1.6 |
| Caption | Tree – Height class (02) |
| Type | Category |
| Entity | Tree |
| Purpose | Tree height classes in 2 m intervals. Note: this module must be after 'Tree - Est. height'. '1' < 4 m, '2' 4 - 5.9 m, '3' 6 - 7.9 m, '4' 8 - 9.9 m, '5' 10 - 11.9 m, '6' 12 - 13.9 m, '7' 14 - 15.9 m, '8' 16 - 17.9 m, '9' 18 - 19.9 m, '10' 20 - 21.9 m, '11' 22 - 23.9 m, '12' 24+ m |
| Code | <pre>tree\$h_2 <- trunc(((tree\$est_height - 2) + 2)/2 ,0) tree\$h_2 <- ifelse(tree\$h_2 > 12, 12, tree\$h_2) tree\$height_class02 <- as.integer(tree\$h_2)</pre> |

| | |
|---------|--|
| # | 1.7 |
| Caption | Plot - Major Forest Status |
| Type | Category |
| Entity | Plot |
| Purpose | Major forest/land status class for each plot. Note: <i>Forest_status</i> read only from plot center point (stand ='A'). |
| Code | <pre>standData <- dbGetQuery(conn=connection, statement="SELECT stand_id, plot_id_, forest_status from stand view WHERE stand id ='A' "); standData\$plot_forest_status <- with(standData, ifelse(forest_status < 200 , 1, # Primary forest ifelse(forest_status < 300, 2, # Secondary forest ifelse(forest_status < 400, 3, # Plantation ifelse(forest_status < 500, 4, # Bushes, grasses land ifelse(forest_status < 600, 5, # Agriculture land ifelse(forest_status == 630, 6, 7) # Water, Other land))))))); plot\$plot_forest_status <- NULL ; plot <- merge(x=plot , y=standData , by="plot_id_" , all=TRUE);</pre> |

2.3. R scripts

2.3.1. Tree

| | |
|----------------|---------------------|
| # | 2.1 |
| Caption | Tree – Count |
| Type | R Script |
| Entity | tree |
| Purpose | Number of trees |
| Code | tree\$stocking <- 1 |

| | |
|----------------|---|
| # | 2.2 |
| Caption | Tree - Basal area |
| Type | R Script |
| Entity | tree |
| Purpose | Basal area of tree (m ²) |
| Code | tree\$basal_area <- with(tree, pi * (0.01*dbh/2)^2) |

| | |
|----------------|---|
| # | 2.3 |
| Caption | Tree - Est. height |
| Type | R Script |
| Entity | tree |
| Purpose | Height of tree (m) |
| Code | <pre> library('lmfor'); tree <- subset(tree, !is.na(dbh)); tree\$cluster_plot_temp <- paste(tree\$cluster_name , tree\$plot_id , sep = '_'); tree\$H <- tree\$hvn; # SELECT ONLY trees where total h >= 1.35 tree\$H[tree\$H < 1.35] <- NA; # DEAD, but probably broken trees dropped out from analysis tree\$H[tree\$quality == 4 & tree\$dbh/tree\$H > 5] <- NA # living, probably broken trees dropped out from analysis. these are rare cases tree\$H[tree\$quality != 4 & tree\$dbh/tree\$H > 10] <- NA # use models calibrated for plots im1 <- ImputeHeights(tree\$dbh, tree\$H, tree\$cluster_plot_temp, makeplot=FALSE, varf=FALSE, level= 1); # use models calibrated for clusters im2 <- ImputeHeights(tree\$dbh, tree\$H, tree\$cluster_name, makeplot=FALSE, varf=FALSE, level=1); # use fixed part of the model only im3 <- ImputeHeights(tree\$dbh, tree\$H, tree\$cluster_name, nranp=0, makeplot=FALSE, lev el=0); hpred <- im1\$h hpred[im1\$predType==2] <- im2\$h[im1\$predType==2] hpred[im2\$predType==2] <- im3\$h[im2\$predType==2] # Add imputed heights into the data (hpred) and a column indicating the type of prediction (hpredType): # 0: tree has been measured, hpred includes the measured height # 1: tree height has been predicted either using a plot-level or cluster lever random effect # 2: tree height has been predicted using the fixed part of the model # selected CLUSTER calibration </pre> |

| | |
|-------------|--|
| | <pre>tree <- cbind(tree, est height = hpred, est height prediction type = im2\$predType); tree\$est_height <- tree\$hpred ; height model fixed <- c(im2\$model\$coefficients\$fixed[[1]], im2\$model\$coefficients\$fixed[[2]])</pre> |
| Note | heigh_model_fixed for calcuting stump results (tree height before felling) |

| | |
|----------------|--|
| # | 2.4 |
| Caption | Tree - Bole height |
| Type | R Script |
| Entity | tree |
| Purpose | Bole height (m) |
| Code | <pre>tree\$bole_height <- 0 # model for Bac Giang only: tree\$bole_height <- 0.548 * tree\$est_height + 0.1923 # if bole height is recorded, use recorded value tree\$bole_height[!is.na(tree\$hdc) & tree\$hdc>0] <-tree\$hdc[!is.na(tree\$hdc) & tree\$hdc>0] # correct possible cases where bole volume height is greater than stem top height: tree\$bole_height<- with(tree, ifelse(bole_height > est_height & !is.na(bole_height), est_height, bole_height))</pre> |

| | |
|----------------|--|
| # | 2.5 |
| Caption | Tree – Volume |
| Type | R Script |
| Entity | tree |
| Purpose | Stem volume (m ³) |
| Code | <pre># Basic form factor volume model (FIPI) ff <- 0.515; tree\$volume <- with(tree, (0.1291+1.5984 * ff) * pi * (0.01 * dbh / 2)^2 * est_height^0.764);</pre> |

| | |
|----------------|--|
| # | 2.6 |
| Caption | Tree - Bole volume |
| Type | R Script |
| Entity | tree |
| Purpose | Bole height (m ³) |
| Code | <pre># Basic bole volume model, Form Factor 0.65 tree\$bole volume <- 0 tree\$bole_volume <- with(tree, 0.65 *pi * (0.01 * dbh / 2)^2 * bole_height); # correct cases where bole volume greater than stem volume: tree\$bole_volume <- with(tree, ifelse(bole_volume > volume & !is.na(bole_height), volume, bole_volume))</pre> |

| | |
|----------------|--|
| # | 2.7 |
| Caption | Tree - AG Biomass |
| Type | R Script |
| Entity | tree |
| Purpose | Above-ground biomass (tons) |
| Code | <pre>BEF pinus <- 1.3; tree\$genus_code <- substr(tree\$species_code, 1, 3); # compute AGB in kg tree\$aboveground_biomass <- with (tree, ifelse(genus_code=='PIN', BEF_pinus * volume * 500, #Pinus, wood density =500 kg/m3 269.63396 * (((dbh/100)^2*est_height)^0.95193) # Evergreen forest))</pre> |

```
# convert kg -> tons
tree$aboveground_biomass <- tree$aboveground_biomass / 1000
```

| | |
|----------------|--|
| # | 2.8 |
| Caption | Tree - BG Biomass |
| Type | R Script |
| Entity | tree |
| Purpose | Below-ground biomass (tons) |
| Code | # conversion factor source: in IPCC/UNFCCC (TABLE 4.4 RATIO OF BELOW-GROUND BIOMASS TO ABOVE-GROUND BIOMASS) tree\$belowground_biomass <- tree\$aboveground_biomass * 0.265 ; |

| | |
|----------------|--|
| # | 2.9 |
| Caption | Tree - Biomass |
| Type | R Script |
| Entity | tree |
| Purpose | Total tree biomass (tons) |
| Code | tree\$total_biomass <- tree\$aboveground_biomass + tree\$belowground_biomass ; |

| | |
|----------------|---|
| # | 2.10 |
| Caption | Tree - AG Carbon |
| Type | R Script |
| Entity | tree |
| Purpose | Above-ground carbon (tons) |
| Code | tree\$ag_carbon <- tree\$aboveground_biomass * 0.50 ; |

| | |
|----------------|---|
| # | 2.11 |
| Caption | Tree - BG Carbon |
| Type | R Script |
| Entity | tree |
| Purpose | Below-ground carbon (tons) |
| Code | tree\$bg_carbon <- tree\$belowground_biomass * 0.50 ; |

| | |
|----------------|--|
| # | 2.12 |
| Caption | Tree - Total carbon & Tree species abundance |
| Type | R Script |
| Entity | tree |
| Purpose | 1) Total carbon (tons), 2) Tree species abundance (%) of stocking & volume into CSV files. |
| Code | # 1) Carbon tree\$carbon <- tree\$ag_carbon + tree\$bg_carbon ##### # 2) Tree species abundance analysis ##### treeCount <- sqldf(" SELECT forest_status,SUM(1/plot_area) AS total_tree_count, SUM(volume * 1/plot_area) AS total volume FROM tree GROUP BY forest_status ") ; treeData <- sqldf(" SELECT forest_status,species_code,SUM(1/plot_area) AS tree_count,SUM(volume * 1/plot_area) AS t_volume FROM tree GROUP BY forest_status, species code ORDER BY 1,3 DESC") ; treeData <- merge(x=treeData , y=treeCount , by="forest_status" , all=TRUE); treeData <- sqldf("SELECT forest_status,species code,tree count/total tree count AS sp_prop_N,t_volume/total_volume AS sp_prop_Vol FROM treeData") ; write.csv(treeData, "c:/temp/treeAbundance_ForestStatus.csv") ; |



```
treeCount <- sqldf(" SELECT stand_major_forest_status AS major_status,
    SUM(1/plot area) AS total tree count,SUM(volume * 1/plot area) AS total volume
    FROM tree GROUP BY major_status " ) ;
treeData <- sqldf(" SELECT stand_major_forest_status AS
major_status,species_code,SUM(1/plot_area) AS tree_count, SUM(volume * 1/plot_area)
    AS t volume FROM tree GROUP BY major status, species code ORDER BY 1,3 DESC" ) ;
treeData <- merge( x=treeData , y=treeCount , by="major status" , all=TRUE);
treeData <- sqldf(" SELECT major_status,species_code,tree_count/total_tree_count
    AS sp_prop_N,t_volume/total_volume AS sp_prop_Vol FROM treeData" ) ;

write.csv(treeData, "c:/temp/treeAbundance MajorForestStatus.csv" ) ;
```

2.3.2. Dead wood

| | |
|----------------|--|
| # | 3.1 |
| Caption | DW - Volume |
| Type | R Script |
| Entity | dead_wood |
| Purpose | Dead wood volume (m ³) |
| Code | <pre> dead_wood\$dwlength <- dead_wood\$length dead_wood\$dk2 <- with(dead_wood,ifelse(dk1 >= 10 & is.na(dk2) & dwlength >0,10,dk2)) dead_wood\$dk1 <- with(dead_wood,ifelse(dk2 >= 10 & is.na(dk1) & dwlength >0,10, dk1)) # missing length: set 1 m # max. length: set 30 m dead_wood\$dwlength <- with(dead_wood, ifelse(dk1 >= 10 & dk2 >= 10 & is.na(dwlength), 1, ifelse(dk1 >= 10 & dk2 >= 10 & dwlength ==0, 1, ifelse(dk1 >= 10 & dk2 >= 10 & dwlength> 30, 30, dwlength)))); dead_wood\$dk2 <- with(dead_wood, ifelse(dk1 >= 10 & is.na(dk2) & dwlength > 0, 10, ifelse(dk1 >= 10 & dk2 < 10 & dwlength > 0, 10, dk2)); dead_wood\$dk1 <- with(dead_wood,ifelse(is.na(dk1) & dk2 >= 10 & dwlength > 0, 10, ifelse(dk1 < 10 & dk2 >= 10 & dwlength > 0, 10, dk1)); # default number of dw particles (socay): 1; max. number: 40 dead_wood\$socay <- with(dead_wood,ifelse(is.na(socay), 1, ifelse(socay == 0, 1, ifelse(socay > 40, 40, socay)))); dead_wood\$rotten[is.na(dead_wood\$rotten)] <- 'S' # default: solid dead_wood\$rotten[dead_wood\$rotten ==''] <- 'S' # default: solid dead_wood\$volume_dw <- with(dead_wood, ((pi*(dk1/200)^2 + pi*(dk2/200)^2) /2) * dwlength); # result must be multiplied with number of similar dw particles dead_wood\$volume_dw <- dead_wood\$volume_dw * dead_wood\$socay </pre> |

| | |
|----------------|---|
| # | 3.2 |
| Caption | DW - Biomass |
| Type | R Script |
| Entity | dead_wood |
| Purpose | Dead wood biomass (tons) |
| Code | <pre> dead_wood\$biomass_dw <- with(dead_wood, ifelse(rotten =='S', 0.9 * volume_dw * 0.6, 0.50 * volume_dw * 0.6)) </pre> |

| | |
|----------------|--|
| # | 3.3 |
| Caption | DW – Carbon |
| Type | R Script |
| Entity | dead_wood |
| Purpose | Dead wood carbon (tons) |
| Code | <pre> dead_wood\$carbon_dw <- 0.50 * dead_wood\$biomass_dw </pre> |

2.3.3. Stump

| | |
|---------|---|
| # | 4.1 |
| Caption | Stump - Count |
| Type | R Script |
| Entity | stump |
| Purpose | Number of stumps |
| Code | <pre>stump\$quantity[is.na(stump\$quantity)] <- 1 stump\$quantity[stump\$quantity == ''] <- 1 stump\$count_stump <- stump\$quantity</pre> |

| | |
|---------|---|
| # | 4.2 |
| Caption | Stump - DBH estimate |
| Type | R Script |
| Entity | stump |
| Purpose | DBH of tree before felling (cm) |
| Code | <pre># MODEL TO ESTIMATE DBH for tree before felling # Model by Bac Giang & Bac Kan combined data, by L. Vesa 16.6.2014 # set max. diameter 400 cm for calculations stump\$diameter[stump\$diameter > 400] <- 400 stump\$height[is.na(stump\$height)] <- 15 stump\$est_dbh <- stump\$diameter - 0.00173 * (130 - stump\$height) * stump\$diameter</pre> |

| | |
|---------|--|
| # | 4.3 |
| Caption | Stump - D0.15 |
| Type | R Script |
| Entity | stump |
| Purpose | Diameter at 0.15 height (cm) |
| Code | <pre># set max. diameter 400 cm for calculations stump\$diameter[stump\$diameter > 400] <- 400 stump\$height[is.na(stump\$height)] <- 15 # diameter at the ground level (0.15 m) k <- (stump\$est_dbh - stump\$diameter) / (1.3 - stump\$height / 100) # slope in linear equation stump\$d0_15 <- ifelse(stump\$height / 100 != 1.3, stump\$est_dbh + k * (0.15 - 1.3), stump\$diameter) # but if diameter at ground level (0.15 m) is less than recorded diameter, then use recorded diameter stump\$d0_15[stump\$d0_15 < stump\$diameter] <- stump\$diameter[stump\$d0_15 < stump\$diameter]</pre> |

| | |
|---------|---|
| # | 4.4 |
| Caption | Stump - Volume |
| Type | R Script |
| Entity | stump |
| Purpose | Stump volume, above 0.15 cm ground level (m ³) |
| Code | <pre># set max. diameter 400 cm for calculations stump\$diameter[stump\$diameter > 400] <- 400 stump\$height[is.na(stump\$height)] <- 15 stump\$volume_stump <- 0 stump\$volume_stump <- with(stump, (pi*(d0_15/200)^2 + pi*(diameter/200)^2) / 2) * (height/100 - 0.15)) # if volume gets negative due to that stump height <15cm, set volume to zero</pre> |

| | |
|--|---|
| | <pre>stump\$volume stump[stump\$volume stump<0] <- 0 stump\$volume stump <- stump\$volume stump * stump\$count stump</pre> |
|--|---|

| | |
|----------------|---|
| # | 4.5 |
| Caption | Stump - AG Biomass |
| Type | R Script |
| Entity | stump |
| Purpose | Above ground biomass (tons) |
| Code | <pre># average wood density (600 kg/m3) for solid wood, stumps assumed to be solid stump\$ag_biomass stump <- with(stump, volume stump * 600 /1000)</pre> |

| | |
|----------------|---|
| # | 4.6 |
| Caption | Stump - Height before felling |
| Type | R Script |
| Entity | stump |
| Purpose | Tree height before felling (m) |
| Code | <pre>stump\$removal_height <- 1.3 + stump\$est_dbh^2/(1.5642090 + 0.2300615 * stump\$est_dbh)^2</pre> |

| | |
|----------------|---|
| # | 4.7 |
| Caption | Stump - Tree AG biomass |
| Type | R Script |
| Entity | stump |
| Purpose | Tree above-ground biomass before felling (tons) |
| Code | <pre>stump\$removal_biomass <- 269.63396 * (((stump\$est_dbh/100)^2 * stump\$removal_height)^0.95193) /1000 stump\$removal_biomass <- stump\$removal_biomass * stump\$count stump</pre> |

| | |
|----------------|--|
| # | 4.8 |
| Caption | Stump - BG biomass |
| Type | R Script |
| Entity | stump |
| Purpose | Stump below-ground biomass (tons) |
| Code | <pre>stump\$bg_biomass_stump <- stump\$removal_biomass * 0.265;</pre> |

| | |
|----------------|---|
| # | 4.9 |
| Caption | Stump - Total biomass |
| Type | R Script |
| Entity | stump |
| Purpose | Stump total biomass (tons) |
| Code | <pre>stump\$biomass_stump <- stump\$ag_biomass_stump + stump\$bg_biomass_stump</pre> |

| | |
|----------------|--|
| # | 4.10 |
| Caption | Stump - Removal volume |
| Type | R Script |
| Entity | stump |
| Purpose | Tree volume, removed part (m ³) |
| Code | <pre>stump\$removal_volume <- with(stump, 0.50 *pi * (0.01 * est_dbh / 2)^2 * removal_height); stump\$removal_volume <- stump\$removal_volume * stump\$count_stump # full stem minus what was left in the forest (i.e. stump volume) stump\$removal volume <- stump\$removal volume - stump\$volume stump</pre> |

| | |
|----------------|--|
| # | 4.11 |
| Caption | Stump – Carbon |
| Type | R Script |
| Entity | stump |
| Purpose | Stump total carbon (tons) |
| Code | <code>stump\$carbon_stump <- stump\$biomass_stump * 0.50 ;</code> |

| | |
|----------------|---|
| # | 4.12 |
| Caption | Stump - Removal biomass |
| Type | R Script |
| Entity | stump |
| Purpose | Tree removal biomass (tons) |
| Code | <code>stump\$removal_tree_biomass <- stump\$removal_biomass - stump\$ag_biomass_stump</code> |

2.3.4. Bamboo

| | |
|----------------|---|
| # | 5.1 |
| Caption | Bamboo - Count |
| Type | R Script |
| Entity | bamboo |
| Purpose | Number of bamboo stems |
| Code | <pre># if missing count, set to 1 bamboo\$stem_count[is.na(bamboo\$stem_count)] <- 1 bamboo <- subset(bamboo, !is.na(dbq)) bamboo\$count_bamboo <- bamboo\$stem_count</pre> |

| | |
|----------------|--|
| # | 5.2 |
| Caption | Bamboo - AG Biomass |
| Type | R Script |
| Entity | bamboo |
| Purpose | Bamboo above-ground biomass (tons) |
| Code | <pre>bamboo\$ag_biomass_bamboo <- 61.08613*(((bamboo\$dbq / 100)^2 * bamboo\$hbq)^0.7126) # convert to tons bamboo\$ag_biomass_bamboo <- bamboo\$ag_biomass_bamboo /1000 bamboo\$ag_biomass_bamboo <- bamboo\$ag_biomass_bamboo * bamboo\$count_bamboo</pre> |

| | |
|----------------|--|
| # | 5.3 |
| Caption | Bamboo - BG Biomass |
| Type | R Script |
| Entity | bamboo |
| Purpose | Bamboo below-ground biomass (tons) |
| Code | <pre>bamboo\$bg_biomass_bamboo <- 0.265 * bamboo\$ag_biomass_bamboo ;</pre> |

| | |
|----------------|---|
| # | 5.4 |
| Caption | Bamboo - Biomass |
| Type | R Script |
| Entity | bamboo |
| Purpose | Bamboo total biomass (tons) |
| Code | <pre>bamboo\$biomass_bamboo <- bamboo\$ag_biomass_bamboo + bamboo\$bg_biomass_bamboo ;</pre> |

| | |
|----------------|--|
| # | 5.5 |
| Caption | Bamboo – Carbon |
| Type | R Script |
| Entity | bamboo |
| Purpose | Bamboo total carbon (tons) |
| Code | <pre>bamboo\$carbon_bamboo <- bamboo\$biomass_bamboo * 0.50 ;</pre> |

2.3.5. Liana (climbers)

| | |
|----------------|---|
| # | 6.1 |
| Caption | Liana - Count |
| Type | R Script |
| Entity | liana |
| Purpose | Number of climbers |
| Code | <pre># select only record when diameter is given liana <- subset(liana, !is.na(liana_diameter)) # if missing count, set to 1 liana\$count[is.na(liana\$count)] <- 1 liana\$count liana <- liana\$count</pre> |

| | |
|----------------|--|
| # | 6.2 |
| Caption | Liana – Volume |
| Type | R Script |
| Entity | liana |
| Purpose | Volume of climbers (m ³) |
| Code | <pre>liana\$vol liana <- pi* (liana\$liana diameter / 2)^2 * liana\$liana length /10000 ; liana\$vol_liana <- liana\$vol_liana * liana\$count_liana</pre> |

| | |
|----------------|---|
| # | 6.3 |
| Caption | Liana – Biomass |
| Type | R Script |
| Entity | liana |
| Purpose | Biomass of climbers (tons) |
| Code | <pre># model for total above-ground biomass, Malaysia. Table 5 & 6, model 14 in article: # http://www.hindawi.com/journals/ijecol/2013/658140/ liana\$biomass_liana <- 10^(0.275 + 0.470*log(liana\$liana_diameter, 10) + 0.452 * log(liana\$liana length, 10)) * 1.011 ; liana\$biomass liana <- liana\$biomass liana /1000 ; liana\$biomass_liana <- liana\$biomass_liana * liana\$count_liana ;</pre> |

| | |
|----------------|---|
| # | 6.4 |
| Caption | Liana – Carbon |
| Type | R Script |
| Entity | liana |
| Purpose | Carbon of climbers (tons) |
| Code | <pre>liana\$carbon_liana <- 0.5 * liana\$biomass_liana ;</pre> |

2.3.6. Regeneration

| | |
|----------------|---|
| # | 7.1 |
| Caption | Regen - Count total |
| Type | R Script |
| Entity | regen |
| Purpose | Number of small trees |
| Code | <pre> regen\$h_1 <- with(regen, count1 + count2) regen\$h_2 <- with(regen, count3 + count4) regen\$h_3 <- with(regen, count5 + count6) regen\$h_4 <- with(regen, count7 + count8) regen\$h_5 <- with(regen, count9 + count10) regen\$h_6 <- with(regen, count11+ count12) regen\$count_regen <- with(regen, h_1 + h_2 + h_3 + h_3 + h_5 + h_6) </pre> |

| | |
|----------------|--|
| # | 7.2 |
| Caption | Regen - Basal area |
| Type | R Script |
| Entity | regen |
| Purpose | Basal area estimate of small trees (m ²) |
| Code | <pre> regen\$h_1 <- with(regen, count1 + count2) regen\$h_2 <- with(regen, count3 + count4) regen\$h_3 <- with(regen, count5 + count6) regen\$h_4 <- with(regen, count7 + count8) regen\$h_5 <- with(regen, count9 + count10) regen\$h_6 <- with(regen, count11+ count12) # ***** dbh_regen <- c(0, 0, 0, 2.5, 3.5, 4.5) # mean dbh in height classes h_regen <- c(0.25, 0.8, 1.3, 1.8, 2.5, 3.5) # mean h in height classes ba <- 0 ba_est <- 0 for (i in 1:length(h_regen)) { ba[i] <- pi * dbh_regen[i]^2 /40000 } ba_est <- ba_est + regen\$h_1 * ba[1] ba_est <- ba_est + regen\$h_2 * ba[2] ba_est <- ba_est + regen\$h_3 * ba[3] ba_est <- ba_est + regen\$h_4 * ba[4] ba_est <- ba_est + regen\$h_5 * ba[5] ba_est <- ba_est + regen\$h_6 * ba[6] regen\$basal_area_regen <- ba_est </pre> |

| | |
|----------------|---|
| # | 7.3 |
| Caption | Regen - Volume |
| Type | R Script |
| Entity | regen |
| Purpose | Volume estimate of small trees (m ³) |
| Code | <pre> regen\$h <- NA vol <- 0 regen\$c_1 <- with(regen, count1 + count2) regen\$c_2 <- with(regen, count3 + count4) regen\$c_3 <- with(regen, count5 + count6) regen\$c_4 <- with(regen, count7 + count8) regen\$c_5 <- with(regen, count9 + count10) regen\$c_6 <- with(regen, count11+ count12) </pre> |


```

dbh_regen      <- c(0, 0, 0, 2.5, 3.5, 4.5) # mean dbh in height classes
h_regen        <- c(0.25, 0.8, 1.3, 1.8, 2.5, 3.5) # mean h in height classes
volume <- 0

# Basic form factor volume model
ff <- 0.515;

for (i in 1:length(h_regen)) {
  volume[i] <- (0.1291+1.5984 * ff) * pi * (0.01 * dbh_regen[i] / 2)^2 *
  h_regen[i]^0.764
}

vol <- vol + volume[1] * regen$c_1
vol <- vol + volume[2] * regen$c_2
vol <- vol + volume[3] * regen$c_3
vol <- vol + volume[4] * regen$c_4
vol <- vol + volume[5] * regen$c_5
vol <- vol + volume[6] * regen$c_6

regen$volume_regen <- vol

```

| | |
|---------|---|
| # | 7.4 |
| Caption | Regen - AG Biomass |
| Type | R Script |
| Entity | regen |
| Purpose | Above-ground biomass of small trees (tons) |
| Code | <pre> regen\$h_1 <- with(regen, count1 + count2) regen\$h_2 <- with(regen, count3 + count4) regen\$h_3 <- with(regen, count5 + count6) regen\$h_4 <- with(regen, count7 + count8) regen\$h_5 <- with(regen, count9 + count10) regen\$h_6 <- with(regen, count11+ count12) dbh_regen <- c(0, 0, 0, 2.5, 3.5, 4.5) # mean dbh in height classes h_regen <- c(0.25, 0.8, 1.3, 1.8, 2.5, 3.5) # mean h in height classes biomass <- 0 ag_biomass <- 0 for (i in 1:length(h_regen)) { biomass[i] <- 269.63396 * (((dbh_regen[i]/100)^2*h_regen[i])^0.95193) # Evergreen forest } ag_biomass <- ag_biomass + regen\$h_1 * biomass[1] ag_biomass <- ag_biomass + regen\$h_2 * biomass[2] ag_biomass <- ag_biomass + regen\$h_3 * biomass[3] ag_biomass <- ag_biomass + regen\$h_4 * biomass[4] ag_biomass <- ag_biomass + regen\$h_5 * biomass[5] ag_biomass <- ag_biomass + regen\$h_6 * biomass[6] regen\$ag_biomass_regen <- ag_biomass </pre> |

| | |
|---------|--|
| # | 7.5 |
| Caption | Regen - BG Biomass |
| Type | R Script |
| Entity | regen |
| Purpose | Below-ground biomass of small trees (tons) |
| Code | <pre> regen\$bg_biomass_regen <- regen\$ag_biomass_regen * 0.265 ; </pre> |

| | |
|---------|-----------------|
| # | 7.6 |
| Caption | Regen - Biomass |
| Type | R Script |

| | |
|----------------|---|
| Entity | regen |
| Purpose | Total biomass of small trees (tons) |
| Code | <code>regen\$biomass_regen <- regen\$ag_biomass_regen + regen\$bg_biomass_regen ;</code> |
| # | 7.7 |
| Caption | Regen - AG Carbon |
| Type | R Script |
| Entity | regen |
| Purpose | Above-ground carbon of small trees (tons) |
| Code | <code>regen\$ag_carbon_regen <- regen\$ag_biomass_regen * 0.50;</code> |
| # | 7.7 |
| Caption | Regen - BG Carbon |
| Type | R Script |
| Entity | regen |
| Purpose | Below-ground carbon of small trees (tons) |
| Code | <code>regen\$bg_carbon_regen <- regen\$bg_biomass_regen * 0.50 ;</code> |
| # | 7.8 |
| Caption | Regen - Carbon |
| Type | R Script |
| Entity | regen |
| Purpose | Total carbon of small trees (tons) |
| Code | <code>regen\$carbon_regen <- regen\$biomass_regen * 0.50 ;</code> |

2.3.7. Plot

| | |
|----------------|---|
| # | 8.1 |
| Caption | Plot- Count |
| Type | R Script |
| Entity | plot |
| Purpose | Number of plots |
| Code | <code>plot\$plot_count <- 1 ;</code> |

| | |
|----------------|---|
| # | 8.2 |
| Caption | Plot- Count (Accessible) |
| Type | R Script |
| Entity | Plot |
| Purpose | Number of accessible plots |
| Code | <code>plot\$plot_count_accessible <- ifelse(plot\$accessibility==0, 1, 0) ;</code> |

| | |
|----------------|---|
| # | 8.3 |
| Caption | Plot- Canopy closure |
| Type | R Script |
| Entity | plot |
| Purpose | Plot canopy closure (%) |
| Code | <pre>x <- cbind(plot\$coverage_center, plot\$coverage_north, plot\$coverage_east, plot\$coverage_south, plot\$coverage_west) # average, without NAs plot\$plot_canopy_closure <- as.integer(rowMeans(x, na.rm = TRUE)) rm(x)</pre> |

| | |
|----------------|---|
| # | 8.4 |
| Caption | Plot - Basal area |
| Type | R Script |
| Entity | plot |
| Purpose | Plot basal area (m ²) |
| Code | <pre>treePlot <- sqldf("select plot_id_ , sum (basal_area / plot_area) as basal_area_tree from tree group by plot_id_"); regenPlot <- sqldf("select plot id , sum (basal area regen / plot area) as basal_area_regen from regen group by plot_id_"); plotResults <- merge(x=treePlot , y=regenPlot , by="plot id " , all=TRUE); plotResults[is.na(plotResults)] <- 0; plotResults\$plot_basal_area = plotResults\$basal_area_tree + plotResults\$basal area regen; plotResults <- plotResults[, c('plot id ' , 'plot basal area')]; # first, remove column plot_basal_area from data frame plot, # otherwise during the above merge, there will be 2 columns(plot_basal_area.x and # plot_basal_area.y) plot\$plot_basal_area <- NULL; plot <- merge(x=plot , y=plotResults , by="plot_id_ " , all=TRUE);</pre> |

| | |
|----------------|-------------------------------|
| # | 8.5 |
| Caption | Plot – Volume |
| Type | R Script |
| Entity | plot |
| Purpose | Plot volume (m ³) |

| | |
|-------------|--|
| Code | <pre> treePlot <- sqldf("select plot id , sum (volume / plot area) as volume tree from tree group by plot_id_ "); regenPlot <- sqldf("select plot id , sum (volume regen / plot area) as volume regen from regen group by plot id "); plotResults <- merge(x=treePlot , y=regenPlot , by="plot_id_ " , all=TRUE); plotResults[is.na(plotResults)] <- 0; plotResults\$plot volume = plotResults\$volume tree + plotResults\$volume regen; plotResults <- plotResults[, c('plot_id_' , 'plot_volume')]; # first, remove column plot_volume from data frame plot, # otherwise during the above merge, there will be 2 columns (plot volume.x and # plot_volume.y) plot\$plot_volume <- NULL; plot <- merge(x=plot , y=plotResults , by="plot_id_ " , all=TRUE); </pre> |
|-------------|--|

| | |
|----------------|--|
| # | 8.6 |
| Caption | Plot - Biodiv. Sp. Count (Biodiversity – Species count) |
| Type | R Script |
| Entity | plot |
| Purpose | Number of unique tree species |
| Code | <pre> # Number of unique species per plot, only trees counted. Code is ready also to add # regen data into the analysis, if needed. treeSpecies <- sqldf("SELECT plot id , species code FROM tree "); # regenSpecies <- sqldf("SELECT plot_id_ , regen_species_code AS species_code # FROM regen "); # plotData <- rbind(treeSpecies, regenSpecies) ; plotData <- treeSpecies ; SpeciesCount <- sqldf("SELECT plot id , COUNT(DISTINCT species code) AS plot_species_count FROM plotData GROUP BY plot_id_ "); # SpeciesCount <- SpeciesCount[, c('plot_id_' , 'plot_species_count')]; plot\$plot_species_count <- NULL; plot <- merge(x=plot , y=SpeciesCount , by="plot id " , all=TRUE); </pre> |

| | |
|----------------|--|
| # | 8.7 |
| Caption | Plot - Biodiv. Shannon (Biodiversity – Shannon index) |
| Type | R Script |
| Entity | plot |
| Purpose | Shannon diversity index by plot, only trees |
| Code | <pre> treeSpecies <- sqldf("SELECT plot id , 1/plot area AS no trees, species code FROM tree "); CountPerPlot <- sqldf("SELECT plot_id_ , SUM(no_trees) AS total_count FROM treeSpecies GROUP BY plot id "); SpeciesCountPerPlot <- sqldf("SELECT plot_id_ , SUM(no_trees) AS species_count FROM treeSpecies GROUP BY plot id , species code"); SpeciesCountPerPlot <- sqldf("SELECT * FROM SpeciesCountPerPlot LEFT JOIN CountPerPlot USING (plot_id_) "); SpeciesShannon <- sqldf("SELECT plot_id_ , -1 * SUM(species_count/total_count * log(species count/total count)) AS plot species shannon FROM SpeciesCountPerPlot GROUP BY plot_id_ "); plot\$plot_species_shannon <- NULL; plot <- merge(x=plot , y=SpeciesShannon , by="plot id " , all=TRUE); </pre> |

| | |
|----------------|---|
| # | 8.8 |
| Caption | Plot - Biodiv. Simpson (Biodiversity – Simpson index) |
| Type | R Script |
| Entity | plot |
| Purpose | Simpson diversity index by plot, only trees |
| Code | <pre> # Simpson diversity index by plot, only trees treeSpecies <- sqldf("SELECT plot_id_ ,1/plot_area AS no_trees, species_code FROM tree "); CountPerPlot <- sqldf("SELECT plot_id_ , SUM(no_trees) AS total_count FROM treeSpecies GROUP BY plot_id_ "); SpeciesCountPerPlot <- sqldf("SELECT plot_id_ , SUM(no_trees) AS species_count FROM treeSpecies GROUP BY plot_id_ , species_code"); SpeciesCountPerPlot <- sqldf("SELECT * FROM SpeciesCountPerPlot LEFT JOIN CountPerPlot USING (plot_id_) "); SpeciesSimpson <- sqldf("SELECT plot_id_ , SUM(species_count * (species_count - 1)/(total_count * (total_count - 1))) AS plot_species_simpson FROM SpeciesCountPerPlot GROUP BY plot id "); plot\$plot_species_simpson <- NULL; plot <- merge(x=plot , y=SpeciesSimpson , by="plot_id_ " , all=TRUE); </pre> |

2.4. Error script

```
### =====
# Error calculation script based on
# "Formulas for estimators and their variances in NFI 28.2.2014 K.T. Korhonen & Olli Salmensuu
# (Metla, Joensuu/Finland), point estimators"
#
# @author Mino Togna, FAO
### =====
# **
# Calculate area error
# **
calculateAreaError <- function( plots , strata ){
  clusters <- getClusters( plots );
  strata <- addStratumCounts( strata , clusters , plots );
  # == (1)
  strata$propInClass <- strata$noCenterPlotsInClass / strata$noPlots;
  # == (2)
  strata$areaInClass <- strata$propInClass * strata$area;
  # == (3)
  clusters <- sqldf("select c.*, s.propInClass
                    from clusters c
                    join strata s
                    on s.stratum = c.stratum");
  clusters$x <- (clusters$noCenterPlotsInClass - clusters$propInClass * clusters$noPlots ) ^ 2;
  strata <- sqldf( "select
                  s.*,
                  sum(c.x) as x
                  from strata s
                  left outer join clusters c
                  on s.stratum = c.stratum
                  group by s.stratum");
  strata$var <- 1 / (strata$noPlots^2) * strata$noClusters / (strata$noClusters -1 ) * strata$x ;
  # == (4)
  strata$areaVariance <- strata$area^2 * strata$var;
  #absolute error
  strata$areaAbsoluteError <- sqrt( strata$areaVariance );
  # add se%(A(f)) - relative error
  strata$areaRelativeError <- 100 * strata$areaAbsoluteError / strata$areaInClass;
  return (strata);
};
# **
# Calculate quantity error
# **
calculateQuantityError <- function( data , plots, strata ) {
  # add plot weight to data
  data <- sqldf( "select d.*, p.weight from data d inner join plots p on d.plot_id = p.plot_id" );
};
results <- calculateAreaError( plots=plots , strata=strata );
strata <-sqldf( "select s.* ,
               r.areaInClass ,
               r.areaVariance,
               r.areaRelativeError
               from
               strata s
               join
               results as r
               on
               r.stratum = s.stratum" );
clusters <- getClusters( plots );
strata <- addStratumCounts( strata , clusters , plots );
#== add sum of quantity per ha to strata and clusters
strata <- sqldf(" select s.*, d.quantity from strata s join ( select stratum, sum(quantity *
class) as quantity from data group by stratum ) as d on s.stratum = d.stratum");

clusters <- sqldf(" select c.*, d.quantity from clusters c left outer join ( select cluster,
sum(quantity * class) as quantity from data group by cluster ) as d on c.cluster = d.cluster");
clusters[is.na(clusters)]<-0;
```

```

#== (5) = (7) / (6)
strata$meanQuantity <- strata$quantity / strata$noCenterPlotsInClass;
clusters <- sqldf("select c.*, s.meanQuantity
                  from clusters c
                  join strata s
                  on s.stratum = c.stratum");
clusters$x <- (clusters$quantity - clusters$meanQuantity * clusters$noCenterPlotsInClass ) ^ 2;
strata <- sqldf( "select
                s.*,
                sum(c.x) as x
                from strata s
                left outer join clusters c
                on s.stratum = c.stratum
                group by s.stratum" );

#== (8)
strata$meanQuantityVariance <- 1 / (strata$noCenterPlotsInClass^2) * strata$noClusters /
(strata$noClusters - 1 ) * strata$x;
# == (9)
strata$totalQuantity <- strata$areaInClass * strata$meanQuantity ;
# == (10)
strata$totalQuantityVariance <- strata$areaInClass^2 * strata$meanQuantityVariance +
strata$meanQuantity^2 * strata$areaVariance;
# add se for mean quantity se%(x(f))
strata$meanQuantityAbsolute <- sqrt( strata$meanQuantityVariance );
strata$meanQuantityRelative <- 100 * strata$meanQuantityAbsolute / strata$meanQuantity;
strata$totalQuantityAbsolute <- sqrt( strata$totalQuantityVariance );
strata$totalQuantityRelative <- sqrt( strata$meanQuantityRelative^2 +
strata$areaRelativeError^2 );
return (strata);
};
# ====
# extract a dataframe of unique clusters included in the data argument
# ====
getClusters <- function( data ) {
  # clusters
  clusters <- sqldf( "select distinct
                    stratum ,
                    cluster ,
                    sum(weight) as noPlots ,
                    sum(class * weight) as noCenterPlotsInClass ,
                    sum(class) as noPlotsInClass
                    from
                    data
                    group by
                    stratum ,
                    cluster" );

  return ( clusters );
};
# ====
# add no. plots and no. clusters to all strata
# ====
addStratumCounts <- function( strata , clusters , plots ) {
  # add no of plots to strata
  strata <- sqldf( "select
                  s.*,
                  p.noPlots,
                  p.noPlotsInClass,
                  p.noCenterPlotsInClass
                  from
                  strata s
                  left outer join
                  (select stratum,
                      sum(weight) as noPlots,
                      sum(class * weight) as noCenterPlotsInClass ,
                      sum(class) as noPlotsInClass
                      from plots group by stratum
                  ) as p
                  on
                  p.stratum = s.stratum");
  # add no of clusters to strata

```

```
strata <- sqldf( "select
                s.*,
                c.noClusters
            from
                strata s
            left outer join
                (select stratum, count(*) as noClusters from clusters group by stratum) as c
            on
                c.stratum = s.stratum");
return ( strata );
};
```