Package `FIESTAutils`

October 12, 2022

**Type**  Package  
**Title**  Utility Functions for Forest Inventory Estimation and Analysis  
**Version**  1.1.3  
**Date**  2022-08-06  
**Description**  A set of tools for data wrangling, spatial data analysis, statistical modeling (including direct, model-assisted, photo-based, and small area tools), and USDA Forest Service data base tools. These tools are aimed to help Foresters, Analysts, and Scientists extract and perform analyses on USDA Forest Service data.

**Depends**  R (>= 3.5.0)  
**Imports**  data.table, DBI, graphics, hbsae, JoSAE, mase, methods, nlme, Rcpp, RColorBrewer, rgdal, R PostgreSQL, RSQLite, sae, sf, sp, sqldf, stats, units, utils, xml2, largeList  
**Suggests**  knitr  
**License**  GPL-3  
**Copyright**  See file COPYRIGHTS for details.  
**URL**  https://github.com/USDAForestService/FIESTAutils  
**BugReports**  https://github.com/USDAForestService/FIESTAutils/issues  
**Encoding**  UTF-8  
**LazyData** true  
**LinkingTo** Rcpp  
**RoxygenNote**  7.2.0  
**NeedsCompilation** yes  
**Author**  Tracey Frescino [aut], Chris Toney [aut], Grayson White [aut, cre]  
**Maintainer**  Grayson White <graysonwhite13@gmail.com>  
**Repository**  CRAN  
**Date/Publication**  2022-08-07 03:40:02 UTC
R topics documented:

- `datExportData`                           2
- `DBtestPostgreSQL`                       3
- `DBtestSQLite`                           4
- `DEFAULT_NODATA`                        5
- `GDT_NAMES`                              5
- `kindcd3old`                             5
- `multest_options`                        6
- `rasterToVRT`                            7
- `Rcpp_CmbTable-class`                    8
- `Rcpp_RunningStats-class`                9
- `ref_codes`                              9
- `ref_diacl2in`                           10
- `ref_domain`                             10
- `ref_estvar`                             10
- `ref_statecd`                            11
- `savedata_options`                      11
- `spMakeSpatial_options`                  13
- `strata_options`                        14
- `stunitco`                               15
- `table_options`                          16
- `title_options`                          18
- `unit_options`                           20

Index 21

---

datExportData  

*Spatial - Exports a data frame object.*

Description

Exports a data frame object to a specified output.

Usage

```r
(datExportData(  
  dfobj,  
  create_dsn = FALSE,  
  index.unique = NULL,  
  index = NULL,  
  savedata_opts = savedata_options()  
))
```
**DBtestPostgreSQL**

**Arguments**
- `dfobj` Data.frame class R object. Data frame object to export.
- `create_dsn` Boolean.
- `index.unique` String. Name of variable(s) in `dfobj` to make unique index.
- `index` String. Name of variable(s) in `dfobj` to make (non-unique) index.
- `savedata_opts` List. See `help(savedata_options())` for a list of options.

**Details**
Wrapper for `sf::st_write` function.

**Value**
An sf spatial object is written to the `out_dsn`.

**Note**
If `out_fmt` = "shp":
The ESRI shapefile driver truncates variable names to 10 characters or less. Variable names are changed before export using an internal function (`trunc10shp`). Name changes are output to the outfolder, 'outshpnni'_newnames.csv.
If sf object has more than 1 record, it cannot be exported to a shapefile.

**Author(s)**
Tracey S. Frescino

---

**DBtestPostgreSQL** Database - Test a SQLite database table.

**Description**
Checks a SQLite database.

**Usage**
```r
DBtestPostgreSQL(dbname, dbconnopen = TRUE)
```

**Arguments**
- `dbname` String. Name of PostgreSQL database.
- `dbconnopen` Logical. If TRUE, the dbconn connection is not closed.

**Value**
An S4 object that inherits from `DBIConnection` via the DBI package. For more information, see ‘help(DBI::dbConnect)’.
Author(s)
Tracey S. Frescino

DBtestSQLite  
*Database - Checks access to a SQLite database.*

Description
Checks a SQLite database.

Usage
DBtestSQLite(
    SQLitefn = NULL,
    gpkg = FALSE,
    dbconnopen = FALSE,
    outfolder = NULL,
    showlist = TRUE,
    returnpath = TRUE,
    createnew = TRUE,
    stopifnull = FALSE,
    overwrite = TRUE
)

Arguments
- `SQLitefn`: String. Name of SQLite database (*.sqlite).
- `gpkg`: Logical. If TRUE, Sqlite geopackage database.
- `dbconnopen`: Logical. If TRUE, the dbconn connection is not closed.
- `outfolder`: String. Optional. Name of output folder. If NULL, export to working directory.
- `showlist`: Logical. If TRUE, shows list of tables in database.
- `returnpath`: Logical. If TRUE, returns full path to SQLite file name. If FALSE, returns `SQLitefn`.
- `createnew`: If TRUE, creates new SQLite database.
- `stopifnull`: Logical. If TRUE, stops if SQLite database doesn’t exist.
- `overwrite`: Logical. If TRUE, overwrites data.

Value
Character string containing the path to the SQLite database of interest.

Author(s)
Tracey S. Frescino
Reference tables - NODATA Values.

**Description**
List of NODATA Values based on data type.

**Format**
A list of 6 components.

**Source**
gdal values.

Reference tables - gdal data types.

**Description**
Table with gdal data type names.

**Format**
A vector of 12 data type values.

**Source**
gdal values.

Reference table - List of RMRS plots that have fallen out of inventory because they were not found or they were in the wrong place.

**Description**
Table with variable codes (VALUE) and descriptions (MEANING).

**Format**
A dataframe
multest_options

Description

Returns a list of user-supplied parameters and parameter values for outputting multest with custom aesthetics.

Usage

```r
multest_options(
  multest_fmt = "csv",
  multest_outfolder = NULL,
  multest_dsn = NULL,
  multest_layer = NULL,
  multest.append = FALSE,
  multest.AOIonly = FALSE,
  ...
)
```

Arguments

- `multest_fmt` String. Format for multest output tables ("csv", "sqlite", "gpkg").
- `multest_outfolder` String. Outfolder for multest. If NULL, same as outfolder.
- `multest_dsn` String. Name of database if `multest_fmt = c(\'sqlite\', \'gpkg\')`.
- `multest_layer` String. Name of database layer if `multest_fmt = c(\'sqlite\', \'gpkg\')`.
- `multest.append` Logical. If TRUE, appends multest dataframe to output.
- `multest.AOIonly` Logical. If TRUE, appends multest dataframe (AOI=1) to output.
- `...` For extendibility.

Details

If no parameters, an empty list is returned.

Source

FIA query. SELECT bp.STATECD, bp.COUNTYCD, bp.PLOT_FIADB NEW_PLOT, bp.START_DATE NEW_START_DATE, bp_old.COUNTYCD OLD_COUNTYCD, bp_old.PLOT_FIADB OLD_PLOT, bp_old.END_DATE OLD_END_DATE, p.CN FROM fs_nims_rmrs.NIMS_BASE_PLOT bp JOIN fs_nims_rmrs.NIMS_BASE_PLOT bp_old on (bp.PREV_NBP_CN=bp_old.CN) JOIN fs_nims_rmrs.NIMS_PLOT_RMRS_VW p on(p.NBP_CN=bp_old.CN) WHERE p.KINDCD = 1 ORDER BY bp.STATECD, bp.COUNTYCD, bp_old.PLOT_FIADB
**Value**

A list of user-supplied parameters and parameter values for outputting multest.

**Author(s)**

Grayson W. White

**Examples**

```r
multest_options(multest.append = TRUE)
```

---

**rasterToVRT**

*Write a GDAL virtual raster file (VRT)*

**Description**

Write a GDAL VRT file for a source raster with options for repositioning and resampling the source data at a different pixel resolution.

**Usage**

```r
rasterToVRT(
  srcfile,
  relativeToVRT = 0,
  vrtfile = tempfile("tmprast", fileext = ".vrt"),
  resolution = NULL,
  subwindow = NULL,
  align = TRUE,
  resampling = "nearest"
)
```

**Arguments**

- `srcfile` Source raster file name.
- `relativeToVRT` Integer. Should `srcfile` be interpreted as relative to the .vrt file (value is 1) or not relative to the .vrt file (value is 0)? If value is 1, the .vrt file is assumed to be in the same directory as `srcfile` and basename(srcfile) is used in .vrt.
- `vrtfile` Output VRT file name.
- `resolution` A numeric vector of length two, with xres, yres. The pixel size must be expressed in georeferenced units. Both must be positive values. The source pixel size is used if resolution is not specified.
**subwindow**

A numeric vector of length four, with xmin, ymin, xmax and ymax values (e.g., sp::bbox or sf::st_bbox). Selects a subwindow of the source raster with corners given in georeferenced coordinates (in the source CRS). If not given, the upper left corner of the VRT will be the same as source, and the VRT extent will be the same or larger than source depending on resolution.

**align**

Logical scalar. If TRUE, the upper left corner of the VRT extent will be set to the upper left corner of the source pixel that contains subwindow xmin, ymax. The VRT will be pixel-aligned with source if the VRT resolution is the same as the source pixel size, otherwise VRT extent will be the minimum rectangle that contains subwindow for the given pixel size. If FALSE, the VRT upper left corner be exactly subwindow xmin, ymax, and the VRT extent will be the minimum rectangle that contains subwindow for the given pixel size. If subwindow is not given, the source window is the source raster extent in which case align=FALSE has no effect.

**resampling**

The resampling method to use if xsize, ysize of the VRT is different from the size of the underlying source rectangle (in number of pixels). The values allowed are nearest, bilinear, cubic, cubicspline, lanczos, average and mode.

### Details

rasterToVRT is useful for virtually clipping a raster to a subwindow, or virtually resampling at a different pixel resolution. The output VRT file will have the same coordinate system as the source raster.

---

### Description

C++ program to combine raster files.

### Extends

Class "C+Object", directly.

All reference classes extend and inherit methods from "envRefClass".

### Author(s)

Chris Toney
Rcpp_RunningStats-class

Class "Rcpp_RunningStats"

Description

C++ program to calculate mean and variance on a data stream.

Extends

Class "C++Object", directly.

All reference classes extend and inherit methods from "envRefClass".

Author(s)

Chris Toney

ref_codes

Reference tables - Code definitions.

Description

Table with variable codes (VALUE) and descriptions (MEANING).

Format

A dataframe with 7 columns, VARIABLE, VALUE, MEANING, COLORHEX, GROUP, GROUPNM, GROUPHEX.

Source

FIA look-up tables.

References

**ref_diacl2in**  
*Reference table - diameter 2-inch class codes (DIA).*

**Description**  
Table with min (MIN), max (MAX), and 2-inch class diameter codes (MEANING).

**Format**  
A dataframe with 3 columns, MIN, MAX, and MEANING.

**Source**  
Imported from comma-delimited file.

**References**  

---

**ref_domain**  
*Reference table - for generating tables.*

**Description**  
Table with row/column domain (VARNM) and their pretty names for table output (TABLENM).

**Format**  
A dataframe with 2 columns, VARNM and TABLENM.

**Source**  
FIA look-up table.

---

**ref_estvar**  
*Reference table - for generating estimates*

**Description**  
Data frame with variable names and descriptions

**Format**  
A data frame to use a reference for estimation variables and filters.
**ref_statecd**

---

**Reference table - state codes (STATECD).**

**Description**

Table with state codes (VALUE), name (MEANING), abbreviation (ABBR), and UNIT.

**Format**

A dataframe with 4 columns, VALUE, MEANING, ABBR, UNIT.

**Source**

Imported from comma-delimited file.

**References**


---

**savedata_options**

**Data saving options.**

**Description**

Returns a list of user-supplied parameters and parameter values for saving data.

**Usage**

```r
savedata_options(
  outfolder = NULL,
  out_fmt = "csv",
  outsp_fmt = "shp",
  outobj_fmt = "rds",
  out_dsn = NULL,
  out_layer = "outdat",
  outfn.pre = NULL,
  outfn.date = FALSE,
  addtitle = TRUE,
  raw_fmt = "csv",
  raw_dsn = NULL,
  overwrite_dsn = FALSE,
  overwrite_layer = TRUE,
  append_layer = FALSE,
)```

Arguments

outfolder String. The outfolder to write files to. If NULL, files are written to working directory, or if gui=TRUE, a window to browse.

out_fmt String. Format for output tables ("csv", "sqlite", "gpkg", "gdb").

outsp_fmt String. Format for output spatial ("shp", "sqlite", "gpkg", "gdb").

outobj_fmt String. Format for output spatial ("rda", "rds", "llo").

out_dsn String. Data source name for output. If extension is not included, out_fmt is used. Use full path if outfolder=NULL.

out_layer outlayer.

outfn.pre String. If savedata=TRUE, prefix for output files. If rawdata=TRUE, prefix for rawdata files (if raw_fmt = 'csv') or raw_dsn (if raw_fmt ! = 'csv').

outfn.date Logical. If TRUE, add current date to out_dsn.

addtitle Logical. If TRUE and savedata=TRUE, adds title to outfile.

raw_fmt String. Format for output rawdata tables ("sqlite", "gpkg", "csv", "gdb").

raw_dsn String. Data source name for rawdata output. If extension is not included, out_fmt is used. Use full path if outfolder=NULL.

overwrite_dsn Logical. If TRUE, overwrites raw_dsn, if exists.

overwrite_layer Logical. If TRUE, overwrites the output. If rawdata=TRUE, overwrites out_layer in rawdata folder (if raw_fmt = 'csv') or out_layers in raw_dsn (if raw_fmt != 'csv').

append_layer Logical. If TRUE, and appends data to existing *.csv files (if *_fmt = 'csv') or *_dsn layers (if *_fmt != 'csv').

add_layer Logical. If TRUE, adds layer to an existing out_dsn (if out_fmt != c('csv', 'shp')).

layer.pre Layer prefix.

Details

If no parameters, an empty list is returned.

Value

A list of user-supplied parameters and parameter values for saving data.

Author(s)

Grayson W. White
Examples

```r
savedata_options(outfolder = "path", overwrite_dsn = FALSE)
```

---

**spMakeSpatial_options  Make SpatialPoints options**

**Description**

Returns a list of user-supplied parameters and parameter values for making SpatialPoints.

**Usage**

```r
spMakeSpatial_options(
  xvar = NULL,
  yvar = NULL,
  xy.crs = 4269,
  prj = NULL,
  datum = NULL,
  zone = NULL,
  zoneS = FALSE,
  aea.param = "USGS",
  ...
)
```

**Arguments**

- `xvar`  String. Name of variable in xyplt defining x coordinate.
- `yvar`  String. Name of variable in xyplt defining y coordinate.
- `xy.crs`  PROJ.4 String or CRS object or Integer EPSG code defining Coordinate Reference System. (e.g., EPSG:4269-Geodetic coordinate system for North America, NAD83).
- `prj`  String. Projection, or coordinate system of the X/Y coordinates ("longlat", "utm", "aea"). If other, include PROJ.4 string in prj4str.
- `datum`  String. Datum of projection ("WGS84", "NAD83", "NAD27").
- `zone`  Integer. If prj="utm", the UTM zone.
- `zoneS`  Logical. If prj="utm", if the UTM zone is in the Southern hemisphere.
- `aea.param`  String. If prj="aea", the associated lat/lon parameters (USGS: " +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96 +x_0=0 +y_0=0"). If other, include PROJ.4 string in prj4str.
- `...`  For extendibility.
Details

If no parameters, an empty list is returned.

Value

A list of user-supplied parameters and parameter values for strata.

Author(s)

Grayson W. White

Examples

```r
spMakeSpatial_options()
```

---

**Description**

Returns a list of user-supplied parameters and parameter values for strata.

**Usage**

```r
strata_options(
  getwt = FALSE,
  getwtvar = "P1POINTCNT",
  strwtvar = "strwt",
  stratcombine = TRUE,
  minplotnum.strat = 2,
  pivot = FALSE,
  nonresp = FALSE,
  ...
)
```

**Arguments**

- **getwt**: Logical. If TRUE, calculates strata weights from stratalut getwtvar. If FALSE, strwtvar variable must be in stratalut.
- **getwtvar**: String. If getwt=TRUE, name of variable in stratalut to calculate weights (Default = 'P1POINTCNT').
- **strwtvar**: String. If getwt=FALSE, name of variable in stratalut with calculated weights (Default = 'strwt').
- **stratcombine**: Logical. If TRUE, and strata=TRUE, automatically combines strata categories if less than minplotnum.strat plots in any one stratum. See notes for more info.
stunitco 15

minplotnum.strat
   Integer. Minimum number of plots for a stratum within an estimation unit.

pivot    Logical. If TRUE, pivot stratalut.
nonresp  Logical. If TRUE, uses Westfall variance methods.
...

For extendibility.

Details
   If no parameters, an empty list is returned.

Value
   A list of user-supplied parameters and parameter values for strata.

Author(s)
   Grayson W. White

Examples

   strata_options(getwt = FALSE)

---

stunitco
   SpatialPolygonsDataFrame with FIA state, unit, county codes and names

Description
   Polygon feature class with state and county boundaries defined by Census Bureau, including Federal Information Processing Standards (FIPS) codes. The FIA Survey Unit code and name attributes (UNITCD, UNITNM) were appended to dataset, with joining columns of STATECD and COUNTYCD.

Format
   A SpatialPolygonsDataFrame with 3233 features and 8 attributes RS - FIA Research Station name RSCD - FIA Research Station code STATECD - FIPS state code STATENM - FIPS state name STATEAB - FIPS state abbreviation UNITCD - FIA survey unit code UNITNM - FIA survey unit name COUNTYCD - FIPS county code COUNTYNM - FIPS county name
Details

Derived from cb_2018_us_county_5m. STATEFP was converted to numeric and named STATECD. COUNTYFP was converted to numeric and named COUNTYCD. Lookup table for FIA Research Station (REF_RESEARCH_STATION) was downloaded from FIA DataMart on 20191105 (FDADB_1.6.1.00) and joined by STATECD. A lookup table for UNITCD was created from plot data using unique STATECD, COUNTYCD, UNITCD and joined to table.

Converted to simple feature
Transformed CRS from longlat(EPSG:4269) to Albers (EPSG:5070)
Saved to R object, with compression='xz'

Source


---

**table_options**

*Table aesthetics and output options.*

Description

Returns a list of user-supplied parameters and parameter values for outputting tables with custom aesthetics.

Usage

```r
table_options("row.FIAnme = FALSE, 
col.FIAname = FALSE, 
row.orderby = NULL, 
col.orderby = NULL, 
row.add0 = FALSE, 
col.add0 = FALSE, 
rowlut = NULL, 
collut = NULL, 
rawonly = FALSE, 
raw.keep0 = FALSE, 
rowgrp = FALSE, 
rowgrpm = NULL, 
rowgrpord = NULL, 
totals = TRUE, 
allin1 = FALSE, 
metric = FALSE, 
estround = 1, 
pseround = 2, 
estnull = "--",
```
psenull = "--",
divideby = NULL,
...
)

Arguments

row.FIAname Logical. If TRUE, retrieves default FIA reference names for rowvar located in ref_codes data frame. Names are only available for certain variables (Check sort(unique(ref_codes$VARIABLE)) for available names. If row.FIAname = TRUE and rowvar is in ref_codes, the rowvar name is used for the output table, and the rowvar code is used to sort.

col.FIAname Logical. If TRUE, retrieves default FIA reference names for colvar located in ref_codes data frame. Names are only available for certain variables. Check: sort(unique(ref_codes$VARIABLE)) for available names. If col.FIAname = TRUE and rowvar is in ref_codes, the colvar name is used for the output table, and the colvar code is used to sort.

row.orderby String. Optional. Name of variable to sort table rows. Both the rowvar and row.orderby variables must be included in the same input data.frame. if NULL, and row.FIAname=FALSE or rowvar is not in ref_codes, the rows are ordered by rowvar.

col.orderby String. Optional. Name of variable to sort table columns. Both the colvar and col.orderby variables must be included in the same input data.frame. if NULL, and col.FIAname=FALSE or colvar is not in ref_codes, the columns are ordered by colvar.

row.add0 Logical. If TRUE, include rows with 0 values to the output table.

col.add0 Logical. If TRUE, include columns with 0 values to the output table.

rowlut Data frame. A lookup table with variable codes and code names to include as rows of output table (See notes for more information and format).

collut Data frame. A lookup table with variable codes and code names to include as columns of output table (See notes for more information and format).

rawonly Logical. If TRUE, only rawdata are output. If dataset includes many estimation units, and only raw data tables are desired, it is more efficient to output raw data only.

raw.keep0 Logical. If TRUE, keep 0 values in raw data tables.

rowgrp Logical. If TRUE, appends row groups to first column of table. Only available if group category exists in ref_codes table or defined in rowgrpnm (e.g., FORTYPGRPCD, OWNGRPCD).

rowgrpnm String. Name of variable for grouping rowvar. Variable must be included in same input table as rowvar.

rowgrpord String. Name of variable to sort row group variable. Variable must be included in same input table as rowgrpnm.

totals Logical. If TRUE, returns total estimate (mean * AREAUSED).

allin1 Logical. If TRUE, both estimates and percent sample error are output in one table as: estimates (percent sample error).
title_options

<table>
<thead>
<tr>
<th>metric</th>
<th>Logical. If TRUE, output if returned in metric units.</th>
</tr>
</thead>
<tbody>
<tr>
<td>estround</td>
<td>Integer. Number of decimal places for estimates.</td>
</tr>
<tr>
<td>pseround</td>
<td>Integer. Number of decimal places for percent sampling error.</td>
</tr>
<tr>
<td>estnull</td>
<td>Number or character. The number or symbol to use to indicate 'not sampled' for estimate.</td>
</tr>
<tr>
<td>psenull</td>
<td>Number or character. The number or symbol to use to indicate 'not sampled' for percent standard error.</td>
</tr>
<tr>
<td>divideby</td>
<td>String. Conversion number for output ('hundred', 'thousand', 'million').</td>
</tr>
<tr>
<td>...</td>
<td>For extendibility.</td>
</tr>
</tbody>
</table>

**Details**

If no parameters, an empty list is returned.

**Value**

A list of user-supplied parameters and parameter values for outputting tables with custom aesthetics.

**Author(s)**

Grayson W. White

**Examples**

```r
table_options(row.FIAname = TRUE, col.FIAname = TRUE)
```

---

**Description**

Returns a list of user-supplied parameters and parameter values for outputting title with custom aesthetics.

**Usage**

```r
title_options(
  title.main = NULL,
  title.ref = NULL,
  title.rowvar = NULL,
  title.colvar = NULL,
  title.unitvar = NULL,
  title.estvar = NULL,
  title.estvaren = NULL,
)```
title_options

```r
title.filter = NULL,
title.units = "acres",
...
```

**Arguments**

- `title.main` String. TITLE, if `savedata=TRUE` and/or `returntitle=TRUE`: the complete title used for table. If `title.main=NULL`, the `title.*` parameters are used to generate title string. Note: if `title.ref` is not NULL, it is added to `title.main`.
- `title.ref` String. TITLE, if `savedata=TRUE` and/or `returntitle=TRUE`: the ending text of the table title (e.g., Nevada, 2004-2005). If NULL, = "".
- `title.rowvar` String. TITLE, if `savedata=TRUE` and/or `returntitle=TRUE`: pretty name for the row domain variable. If NULL, = `rowvar`.
- `title.colvar` String. TITLE, if `savedata=TRUE` and/or `returntitle=TRUE`: pretty name for the column domain variable. If NULL, = `colvar`.
- `title.unitvar` String. TITLE, if `savedata=TRUE` and/or `returntitle=TRUE`: pretty name for the estimation unit variable. If NULL, = `unitvar`.
- `title.estvar` String. TITLE: if `savedata=TRUE` and/or `returntitle=TRUE`: pretty name for the estimate variable. If NULL, `title.estvar = estvar.name`.
- `title.estvarn` String. TITLE: if `savedata=TRUE` and/or `returntitle=TRUE`: pretty name for the estimate variable. If NULL, `title.estvar = estvar.name`.
- `title.filter` String. TITLE, if `savedata=TRUE` and/or `returntitle=TRUE`: pretty name for filter(s). If `title.filter=NULL`, a default is generated from `cfilter`. If `title.filter=""`, no `title.filter` is used.
- `title.units` String.
  ... For extendibility.

**Details**

If no parameters, an empty list is returned.

**Value**

A list of user-supplied parameters and parameter values for outputting titles with custom aesthetics.

**Author(s)**

Grayson W. White

**Examples**

```r
title_options(title.main = "My fancy title", title.estvar = "Estimate title")
```
**unit_options**  

**Unit options.**

**Description**

Returns a list of user-supplied parameters and parameter values for unit.

**Usage**

```r
unit_options(
  unitvar2 = NULL,
  areaunits = "acres",
  minplotnum.unit = 10,
  unit.action = "keep",
  npixelvar = "npixels",
  ...
)
```

**Arguments**

- `unitvar2`  
  String. Name of a second level estimation unit variable in unitarea and cond or pltassgn with assignment for each plot (e.g., 'STATECD').

- `areaunits`  
  String. Units of areavar in unitarea ('acres', 'hectares').

- `minplotnum.unit`  
  Integer. Minimum number of plots for estimation unit.

- `unit.action`  
  String. What to do if number of plots in an estimation unit is less than minplotnum.unit ('keep', 'remove' 'combine'). If unit.action='combine', combines estimation unit to the following estimation unit, ordered in unitzonal or stratalut.

- `npixelvar`  
  String. Name of variable in unitlut defining number of pixels by estimation unit.

- `...`  
  For extendibility.

**Details**

If no parameters, an empty list is returned.

**Value**

A list of user-supplied parameters and parameter values for strata.

**Author(s)**

Grayson W. White

**Examples**

```r
unit_options()
```
Index

* classes
  Rcpp_CmbTable-class, 8
  Rcpp_RunningStats-class, 9
* datasets
  DEFAULT_NODATA, 5
  GDT_NAMES, 5
  kindcd3old, 5
  ref_codes, 9
  ref_diacl2in, 10
  ref_domain, 10
  ref_estvar, 10
  ref_statecd, 11
  stunitco, 15
* data
  datExportData, 2
  DBtestPostgreSQL, 3
  DBtestSQLite, 4
* options
  multest_options, 6
  savedata_options, 11
  spMakeSpatial_options, 13
  strata_options, 14
  table_options, 16
  title_options, 18
  unit_options, 20

C++Object, 8, 9

datExportData, 2
DBtestPostgreSQL, 3
DBtestSQLite, 4
DEFAULT_NODATA, 5

envRefClass, 8, 9

GDT_NAMES, 5
kindcd3old, 5

multest_options, 6

rasterToVRT, 7
Rcpp_CmbTable-class, 8
Rcpp_RunningStats-class, 9
ref_codes, 9
ref_diacl2in, 10
ref_domain, 10
ref_estvar, 10
ref_statecd, 11

savedata_options, 11
spMakeSpatial_options, 13
strata_options, 14
stunitco, 15

table_options, 16

title_options, 18

unit_options, 20