

# Package: knfi (via r-universe)

March 3, 2025

**Title** Analysis of Korean National Forest Inventory Database

**Version** 1.0.1.9

**Maintainer** Sinyoung Park <youngsin0306@kookmin.ac.kr>

**Description** Understanding the current status of forest resources is essential for monitoring changes in forest ecosystems and generating related statistics. In South Korea, the National Forest Inventory (NFI) surveys over 4,500 sample plots nationwide every five years and records 70 items, including forest stand, forest resource, and forest vegetation surveys. Many researchers use NFI as the primary data for research, such as biomass estimation or analyzing the importance value of each species over time and space, depending on the research purpose. However, the large volume of accumulated forest survey data from across the country can make it challenging to manage and utilize such a vast dataset. To address this issue, we developed an R package that efficiently handles large-scale NFI data across time and space. The package offers a comprehensive workflow for NFI data analysis. It starts with data processing, where `read_nfi()` function reconstructs NFI data according to the researcher's needs while performing basic integrity checks for data quality. Following this, the package provides analytical tools that operate on the verified data. These include functions like `summary_nfi()` for summary statistics, `diversity_nfi()` for biodiversity analysis, `iv_nfi()` for calculating species importance value, and `biomass_nfi()` and `cwd_biomass_nfi()` for biomass estimation. Finally, for visualization, the `tvis_nfi()` function generates graphs and maps, allowing users to visualize forest ecosystem changes across various spatial and temporal scales. This integrated approach and its specialized functions can enhance the efficiency of processing and analyzing NFI data, providing researchers with insights into forest ecosystems. The NFI Excel files (.xlsx) are not included in the R package and must be downloaded separately. Users can access these NFI Excel files by visiting the Korea Forest Service Forestry Statistics

Platform

<<https://kfss.forest.go.kr/stat/ptl/article/articleList.do?curMenu=11694&bbsId=microdataboard>>

to download the annual NFI Excel files, which are bundled in .zip archives. Please note that this website is only available in Korean, and direct download links can be found in the notes section of the `read_nfi()` function.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Additional\_repositories** <https://SYOUNG9836.github.io/drat>

**URL** <https://github.com/SYOUNG9836/knfi>,

<https://syoun9836.github.io/knfi/>

**BugReports** <https://github.com/SYOUNG9836/knfi/issues>

**Imports** dplyr, magrittr, readxl, stringr, vegan, tidyr, BiodiversityR, data.table, sf, plotrix, rlang, cellranger, stats, broom, ggplot2, sp, cowplot, ggpubr, drat, purrr, scales

**Depends** R (>= 3.6)

**Suggests** kadmin, knitr, rmarkdown

**VignetteBuilder** knitr

**Language** en-US

**Config/pak/sysreqs** cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libssl-dev libproj-dev libsqlite3-dev tcl tk tk-dev tk-table libudunits2-dev libx11-dev zlib1g-dev

**Repository** <https://syoun9836.r-universe.dev>

**RemoteUrl** <https://github.com/syoun9836/knfi>

**RemoteRef** HEAD

**RemoteSha** 017b20a0af540203a5d6c4104b0afa3a19d1c578

## Contents

biomass_nfi . . . . .	3
cwd_biomass_nfi . . . . .	5
diversity_nfi . . . . .	7
filter_nfi . . . . .	9
iv_nfi . . . . .	11
nfi_col . . . . .	13
nfi_donghae . . . . .	13
read_nfi . . . . .	14

*biomass\_nfi* 3

summary_nfi . . . . .	16
switchcol_nfi . . . . .	18
tsvis_nfi . . . . .	19

**Index** 22

---

biomass\_nfi                      *Calculate Tree Biomass for National Forest Inventory Data*

---

**Description**

biomass\_nfi() function estimates volume, aboveground biomass, biomass, carbon storage and carbon dioxide storage. It can provide summaries for individual plots, the entire study area, or specific groups within the study area using parameters byplot, plotgrp and treegrp. When calculating biomass at the individual trees and plots level, users have flexibility in specifying data inclusion criteria and analysis levels using parameters clusterplot, largetreear, stockedland, and talltree. These parameters determine whether to treat cluster plots as single plots, to include large tree survey plots, or to focus only on Stocked land and tall trees. Users can also choose the criteria for post-stratification using the strat parameter.

**Usage**

```
biomass_nfi(  
  data,  
  byplot = FALSE,  
  plotgrp = NULL,  
  treegrp = NULL,  
  continuousplot = FALSE,  
  strat = "FORTYP_SUB",  
  clusterplot = FALSE,  
  largetreear = TRUE,  
  stockedland = TRUE,  
  talltree = TRUE  
)
```

**Arguments**

- data : A list generated by read\_nfi that contains 'plot' and 'tree' data frames.
- byplot : A logical flag (default FALSE); if TRUE, calculates statistics for each plot separately. If FALSE, calculates for the entire dataset or groups specified by plotgrp and treegrp.
- plotgrp : A character vector; variables from 'plot' tables for grouping. Use c() to combine multiple variables.
- treegrp : A character vector; variables from 'tree' tables for grouping. Use c() to combine multiple variables.
- continuousplot : A logical flag (default TRUE); if TRUE, includes only plots that have been continuously measured in all NFI cycles (5th, 6th, etc.). If FALSE, includes plots regardless of missing cycle measurements.

<code>strat</code>	: A character vector; the variable used for post-stratification. In the National Forest Inventory of Korea, it is typically used by forest type.
<code>clusterplot</code>	: A logical flag (default FALSE); if TRUE, treats each cluster plot as a single unit. If FALSE, calculates for each subplot separately.
<code>largetreearea</code>	: A logical flag (default TRUE); if TRUE, includes large tree survey plots in the analysis. If FALSE, only uses standard tree plots.
<code>stockedland</code>	: A logical flag (default TRUE); if TRUE, includes only stocked land. If FALSE, includes all land types.
<code>talltree</code>	: A logical flag (default TRUE); if TRUE, includes only tall trees. If FALSE, includes both trees and shrubs.

### Details

This function calculates biomass using methodologies employed for national statistics (mean, variance, standard error, relative standard error):

- Applies national carbon emission factors to calculate biomass at the individual tree level.
- Estimates biomass per hectare at the cluster or subplot level, with options to include only basic survey trees or both basic and large tree survey trees.
- Uses the Double Sampling for Post-stratification (DSS) method to derive annual statistics.
- Applies the Weighted Moving Average (WMA) method to integrate annual statistics from the 20% of plots surveyed each year into a single time point.

### Value

A data frame that includes biomass estimates. The structure depends on the input parameters:

- If `byplot = TRUE`, each row represents a plot.
- If `byplot = FALSE`, each row represents the entire dataset or a group specified by `plotgrp` and `treegrp`

### Note

Biomass calculation involves dividing the data into groups based on `plotgrp` and then applying post-stratification to each group. As a result, if the data for each group is not sufficiently large, the relative standard error (RSE) may be high. It is important to check the RSE and other statistical measures in the biomass results.

### References

Son, Y., Kim, R., Lee, K., Pyo, J., Kim, S., Hwang, J., Lee, S., & Park, H. (2014). Carbon emission factors and biomass allometric equations by species in Korea. Korea Forest Research Institute. Yim, J., Moon, G., Lee, M., Kang, J., Won, M., Ahn, E., & Jeon, J. (2021). 2020 Forest inventory of Korea. Korea Forest Research Institute.

**Examples**

```

data("nfi_donghae")

# Basic usage
biomass <- biomass_nfi(nfi_donghae, continuousplot = TRUE)

# Calculate biomass by administrative district
district_biomass <- biomass_nfi(nfi_donghae, plotgrp = "SGG", continuousplot=TRUE)

# Calculate biomass for each plot
plot_biomass <- biomass_nfi(nfi_donghae, byplot = TRUE)

```

---

cwg_biomass_nfi	<i>Calculate biomass of Coarse Woody Debris</i>
-----------------	---

---

**Description**

cwg\_biomass\_nfi() function estimates volume, carbon storage and carbon dioxide storage of Coarse Woody Debris (CWD). It can estimate individual plots, the entire study area, or specific groups within the study area using parameters byplot, plotgrp and treegrp. Users can choose the criteria for post-stratification using the strat parameter. Users can specify whether to focus only on Stocked land using the stockedland parameter.

**Usage**

```

cwg_biomass_nfi(
  data,
  byplot = FALSE,
  plotgrp = NULL,
  treegrp = NULL,
  continuousplot = FALSE,
  strat = "FORTYP_SUB",
  stockedland = TRUE
)

```

**Arguments**

data	: A list generated by <a href="#">read_nfi</a> that contains 'plot' and 'cwg' data frames.
byplot	: A logical flag (default FALSE); if TRUE, calculates statistics for each plot separately. If FALSE, calculates for the entire dataset.
plotgrp	: A character vector; variables from 'plot' tables for grouping. Use c() to combine multiple variables.
treegrp	: A character vector; variables from 'tree' tables for grouping. Use c() to combine multiple variables.

- `continuousplot` : A logical flag (default TRUE); if TRUE, includes only plots that have been continuously measured in all NFI cycles (5th, 6th, etc.). If FALSE, includes plots regardless of missing cycle measurements.
- `strat` : A character vector; the variable used for post-stratification. In the National Forest Inventory of Korea, it is typically used by forest type.
- `stockedland` : A logical flag (default TRUE); if TRUE, includes only stocked land. If FALSE, includes all land types.

### Details

This function calculates CWD biomass using methodologies employed for national statistics (mean, variance, standard error, relative standard error):

- Applies national carbon emission factors to calculate CWD biomass at the individual tree level.
- Estimates biomass per hectare at the plot level.
- Uses the Double Sampling for Post-stratification (DSS) method to derive annual statistics.
- Applies the Weighted Moving Average (WMA) method to integrate annual statistics from the 20% of plots surveyed each year into a single time point.

### Value

A `data.frame` that includes CWD biomass for plot or study areas. The structure depends on the input parameters:

- If `byplot = TRUE`, each row represents a plot.
- If `byplot = FALSE`, each row represents the entire dataset or a group specified by `plotgrp` and `treegrp`

### Note

Biomass calculation involves dividing the data into groups based on `plotgrp` and then applying post-stratification to each group. As a result, if the data for each group is not sufficiently large, the relative standard error (RSE) may be high. It is important to check the RSE and other statistical measures in the biomass results.

### References

Son, Y., Kim, R., Lee, K., Pyo, J., Kim, S., Hwang, J., Lee, S., & Park, H. (2014). Carbon emission factors and biomass allometric equations by species in Korea. Korea Forest Research Institute. Yim, J., Moon, G., Lee, M., Kang, J., Won, M., Ahn, E., & Jeon, J. (2021). 2020 Forest inventory of Korea. Korea Forest Research Institute.

### Examples

```
data("nfi_donghae")

# Basic usage
cwd <- cwd_biomass_nfi(nfi_donghae, continuousplot = TRUE)
```

```
# Calculate CWD biomass grouped by administrative district and decay class
cwd_grp <- cwd_biomass_nfi(nfi_donghae, plotgrp = "SGG", treegrp = "DECAY", continuousplot = TRUE)

# Calculate CWD biomass for each plot
plot_biomass <- cwd_biomass_nfi(nfi_donghae, byplot = TRUE)
```

---

**diversity\_nfi***Calculate species diversity indices for National Forest Inventory Data*

---

## Description

`diversity_nfi()` function calculates species richness, evenness and the Shannon and Simpson diversity indices for National Forest Inventory (NFI) data. It can provide diversity measures for individual plots, the entire study area, or specific groups within the study area using parameters `byplot` or `plotgrp`. The function can calculate diversity at the species or genus level for different vegetation components (trees, herbs, vegetation, saplings). It uses the `diversity` function from the **vegan** package for core calculations. Users have flexibility in specifying data inclusion criteria and analysis levels using parameters `clusterplot`, `largetreeara`, `stockedland`, and `talltree`. These parameters determine whether to treat cluster plots as single plots, to include large tree survey plots, and to focus only on Stocked land and tall trees.

## Usage

```
diversity_nfi(
  data,
  sp = "SP",
  table = "tree",
  basal = FALSE,
  byplot = FALSE,
  plotgrp = NULL,
  continuousplot = FALSE,
  clusterplot = FALSE,
  largetreeara = FALSE,
  stockedland = TRUE,
  talltree = TRUE
)
```

## Arguments

<code>data</code>	: A list generated by <code>read_nfi</code> that contains 'plot' and one of ('tree', 'herb', 'veg', 'sapling') data frames.
<code>sp</code>	: A character vector; the column name of species information (e.g., "SP" for species, "GENUS" for genus-level analysis).
<code>table</code>	: A character vector; Specifies which vegetation table to use for diversity analysis. Must be one of 'tree', 'herb', 'veg', 'sapling'.

basal	: A logical flag (default FALSE); if TRUE, calculates tree diversity using basal area. If FALSE, uses number of individuals. Only applicable when table = "tree".
byplot	: A logical flag (default FALSE); if TRUE, calculates statistics for each plot separately. If FALSE, calculates for the entire dataset.
plotgrp	: A character vector; specifies variables from 'plot' table to use for grouping. Use c() to combine multiple variables.
continuousplot	: A logical flag (default TRUE); if TRUE, includes only plots that have been continuously measured in all NFI cycles (5th, 6th, etc.). If FALSE, includes plots regardless of missing cycle measurements.
clusterplot	: A logical flag (default FALSE); if TRUE, treats each cluster plot as a single unit. If FALSE, calculates for each subplot separately.
largetreearea	: A logical flag (default FALSE); if TRUE, includes large tree survey plots in the analysis. If FALSE, only uses standard tree plots.
stockedland	: A logical flag (default TRUE); if TRUE, includes only stocked land. If FALSE, includes all land types.
talltree	: A logical flag (default TRUE); if TRUE, includes only tall trees. If FALSE, includes both trees and shrubs.

### Details

The function calculates the following diversity indices (mean and standard error):

- Species richness: The total number of species surveyed.
- Shannon-Wiener index: Calculated as the sum of the proportions of individuals or basal area of each species relative to the total.
- Gini-Simpson index: Calculated as 1 minus Simpson's index. Ranges from 0 to 1, with higher values indicating greater diversity.
- Species evenness: Calculated by dividing Shannon diversity by the natural logarithm of species richness. Ranges from 0 to 1, with 1 indicating that all species are evenly distributed.

### Value

A data.frame that includes diversity indices. The structure depends on the input parameters:

- If byplot = TRUE, each row represents a plot.
- If byplot = FALSE, each row represents the entire dataset or a group specified by plotgrp

### Note

The 'herb', 'veg', and 'sapling' tables may contain a lot of errors, so use caution when interpreting results from these tables.

### References

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379–423. Simpson, E. H. (1949). Measurement of diversity. *Nature*, 163(4148), 688–688. Pielou, E. C. (1966). The measurement of diversity in different types of biological collections. *Journal of Theoretical Biology*, 13, 131–144.



**See Also**

[diversity](#) for calculating the Shannon and Simpson diversity indices.

**Examples**

```
data("nfi_donghae")

# Calculate tree diversity indices using basal area
tree_ba_diversity <- diversity_nfi(nfi_donghae, sp = "SP", table = "tree",
                                  basal = TRUE, continuousplot = TRUE)

# Calculate tree diversity indices using number of individuals
tree_indi_diversity <- diversity_nfi(nfi_donghae, sp = "SP", table = "tree",
                                     basal = FALSE, continuousplot = TRUE)
```

---

 filter\_nfi

---

*Filter National Forest Inventory Data*


---

**Description**

The `filter_nfi()` function provides hierarchical and non-hierarchical filtering approaches for the complex structure of National Forest Inventory data based on user-provided condition expressions (`expr_texts`). This function enables effective filtering while maintaining the relationship between plot data (parent data) and other data (child data).

**Usage**

```
filter_nfi(data, expr_texts, hier = TRUE)
```

**Arguments**

<code>data</code>	: A list generated by <a href="#">read_nfi</a> . Each dataframe should have a 'SUB_PLOT' column.
<code>expr_texts</code>	: @param <code>expr_texts</code> : A character vector; expressions specifying filtering conditions. Each expression should combine the dataframe name, dollar sign, and condition, with separate expressions for each data frame. (e.g., <code>c("plot\$OWN_CD == '5'", "tree\$FAMILY == 'Pinaceae'")</code> ). Conditions must be valid R expressions.
<code>hier</code>	: A logical flag (default TRUE); indicates whether to apply hierarchical filtering (TRUE) or non-hierarchical filtering (FALSE). Hierarchical filtering ensures that connected dataframes are filtered based on the results of filters applied to the parent frame.

## Details

This function parses expressions targeting specific columns in the dataframes within the provided list.

Expression requirements:

- Each expression in `expr_texts` must start with a valid dataframe name in the list (e.g., "plot", "tree", "cwd") and combine the dataframe name, dollar sign, and condition (e.g. `c("plot$OWN_CD == '5'")`).
- Separate expressions must be provided for each dataframe being filtered (e.g. `c("plot$OWN_CD == '5'", "tree$FAMILY == 'Pinaceae' | tree$WDY_PLNTS_TYP_CD == '1'")`).

Hierarchical filtering (`hier = TRUE`):

- Filters applied to plot table affect all connected child data (tree, CWD, stump, etc.).
- Filters applied to child data only operate within that dataframe and do not affect other dataframes.
- Example: If only coniferous forest subplots are selected in the plot table, the child data will retain only the tree, CWD, stump, etc., associated with those subplots.

Non-hierarchical filtering (`hier = FALSE`):

- Filters applied to the parent dataframe (plot table) do not affect the child data.
- Filtering results from child data affect all other parent and child data.
- Example: If only certain species are selected in the tree table, the plot table, CWD table, stump table, etc., will be filtered based on the remaining subplots from this selection.

## Value

A list of dataframes.

## Examples

```
data("nfi_donghae")

# Applying hierarchical filtering to select only privately owned forest subplots.
# Ensures all child tables' subplots match the filtered plot table's subplots.
# Expected results after filtering:
# all(nfi_donghae$tree$SUB_PLOT %in% nfi_donghae$plot$SUB_PLOT) result: TRUE
nfi_donghae <- filter_nfi(nfi_donghae, c("plot$OWN_CD == '5'"), hier = TRUE)

# Non-hierarchical filtering to select only privately owned forest subplots.
# Child tables remain unfiltered and may not correspond to the plot table subplots.
# Expected results after filtering:
# all(nfi_donghae$tree$SUB_PLOT %in% nfi_donghae$plot$SUB_PLOT) result: FALSE
nfi_donghae <- filter_nfi(nfi_donghae, c("plot$OWN_CD == '5'"), hier = FALSE)

# Non-Hierarchical Filtering with only woody plants.
# Other tables remain filtered and correspond to the tree table.
# Expected results after filtering:
# all(nfi_donghae$plot$SUB_PLOT %in% nfi_donghae$tree$SUB_PLOT) result: TRUE
```

```
nfi_donghae <- filter_nfi(nfi_donghae, c("tree$WDY_PLNTS_TYP_CD == '1'"), hier = FALSE)

# Combining multiple filters across different dataframes
nfi_donghae <- filter_nfi(nfi_donghae,
  c("plot$OWN_CD == '5'",
    "tree$FAMILY == 'Pinaceae' | tree$WDY_PLNTS_TYP_CD == '1'"))
```

---

iv\_nfi

*Calculates the importance values for National Forest Inventory Data*


---

## Description

`iv_nfi()` function calculates the importance values of tree species based on frequency, density and coverage. It can estimate the entire study area or specific groups within it using the `plotgrp` parameter. It uses the `importancevalue` function from the **BiodiversityR** package for core calculations. Users have flexibility in specifying data inclusion criteria and analysis levels using parameters `frequency`, `clusterplot`, `largetreearea`, `stockedland`, and `talltree`. These parameters determine whether to include frequency in importance calculations, to treat cluster plots as single plots, to include large tree survey plots, and to focus only on Stocked land and tall trees.

## Usage

```
iv_nfi(
  data,
  sp = "SP",
  frequency = TRUE,
  plotgrp = NULL,
  continuousplot = FALSE,
  clusterplot = FALSE,
  largetreearea = FALSE,
  stockedland = TRUE,
  talltree = TRUE
)
```

## Arguments

<code>data</code>	: A list generated by <code>read_nfi</code> that contains 'plot' and 'tree' data frames.
<code>sp</code>	: A character vector; the column name of species information (e.g., "SP" for species, "GENUS" for genus-level analysis).
<code>frequency</code>	: A logical flag (default TRUE); if TRUE, includes frequency in importance value calculations.
<code>plotgrp</code>	: A character vector; specifies variables from 'plot' table to use for grouping. Use <code>c()</code> to combine multiple variables.

- `continuousplot` : A logical flag (default TRUE); if TRUE, includes only plots that have been continuously measured in all NFI cycles (5th, 6th, etc.). If FALSE, includes plots regardless of missing cycle measurements.
- `clusterplot` : A logical flag (default FALSE); if TRUE, treats each cluster plot as a single unit. If FALSE, calculates for each subplot separately.
- `largetreearea` : A logical flag (default FALSE); if TRUE, includes large tree survey plots in the analysis. If FALSE, only uses standard tree plots.
- `stockedland` : A logical flag (default TRUE); if TRUE, includes only stocked land. If FALSE, includes all land types.
- `talltree` : A logical flag (default TRUE); if TRUE, includes only tall trees. If FALSE, includes both trees and shrubs.

### Details

The importance value (ranging from 0 to 100) is calculated as the mean of:

1. Relative frequency:  $(\text{number of plots where a species is observed} / \text{total survey plots}) * 100$
2. Relative density:  $(\text{total number of individuals of a species} / \text{sum of all species' densities}) * 100$
3. Relative coverage:  $(\text{total basal area of a species} / \text{sum of all species' basal area}) * 100$

### Value

A data frame that includes importance value for tree species. Each row represents a combination of a tree species and groups specified by `plotgrp` and `treegrp`.

### Note

- Consider calculating importance by genus rather than species due to potential incompleteness in species classification.
- Since the frequencies of each species may be identical across the nation, it may be desirable to exclude frequency from the importance calculation.

### References

Curtis, J. T. & McIntosh, R. P. (1951). An upland forest continuum in the prairie-forest border region of Wisconsin. *Ecology*, 32(3), 476–496.

### See Also

[importancevalue](#) for calculating the importance values.

### Examples

```
data("nfi_donghae")
# Calculate importance values without frequency
importance <- iv_nfi(nfi_donghae, sp = "SP", frequency = FALSE, continuousplot = TRUE)
```

```
# Calculate importance values using genus
genus_importance <- iv_nfi(nfi_donghae, sp = "GENUS", continuousplot = TRUE)
```

---

nfi_col	<i>The Korean and English names of the column names</i>
---------	---

---

**Description**

The Korean and English names of the column names

**Usage**

```
nfi_col
```

**Format**

An object of class `data.frame` with 174 rows and 3 columns.

---

nfi_donghae	<i>National Forest Inventory data for Donghae-si, Gangwon-do, Republic of Korea for testing the function</i>
-------------	--

---

**Description**

National Forest Inventory data for Donghae-si, Gangwon-do, Republic of Korea for testing the function

**Usage**

```
nfi_donghae
```

**Format**

An object of class `list` of length 3.

---

read\_nfi *Read Korean National Forest Inventory*

---

### Description

read\_nfi() function reads and processes the Korean National Forest Inventory (NFI). It loads annual NFI files from a local computer, transforms the data into an analysis-friendly format, and performs data integrity verification. Users can specify districts and tables to load. When loading data from the original NFI Excel files, the function will automatically handle the translation of Korean column names to English - no manual translation is required. NFI data can be downloaded from <https://kfss.forest.go.kr/stat/>.

### Usage

```
read_nfi(
  dir,
  district = NULL,
  tables = c("tree", "cwd"),
  pattern = "xlsx",
  ...
)
```

### Arguments

dir	: A character vector; The directory containing NFI files.
district	: A character vector; The district names in Korean (sido, sigungu, or eupmyon-dong levels). If NULL, the entire dataset is loaded. Combine multiple districts using c().
tables	: A character vector; tables to import. Options: 'tree', 'cwd', 'stump', 'sapling', 'veg', 'herb', 'soil'. Combine multiple tables using c(). e.g., c('tree', 'cwd', 'stump', 'sapling', 'veg', 'herb', 'soil').
pattern	: A character vector; (default "xlsx"); file pattern to match when loading NFI files. Use regular expressions to filter specific files (e.g., "NFI5.*xlsx" for 5th NFI files only)
...	: Additional arguments to be passed to list.files()

### Details

The function can load the following tables:

plot Base table containing subplot data including site, stand and non-forest area, among other details (automatically included).

tree Tree survey table including species, DBH, and height, among others. Data is collected from trees and large trees survey plot of subplot.

cwd Coarse woody debris table including species, tree decay level, and cause of death, among other details. Data is collected only at the center subplot of the cluster plot.

stump Stumps table including species and diameter at 20 cm above the ground, among other details. Data is collected only at the center subplot of the cluster.

sapling Saplings table including species, diameter at 20 cm above the ground, and the number of individuals, among other details. Data is collected only at the sapling survey plot of the subplot.

veg Vegetation table (both woody and herbaceous plants). It records species, number of individuals, and dominance, among others. Data is collected from three vegetation survey plots located within each selected center subplot. The selection includes 25% of the total number of center subplots.

herb Herbaceous table focused on the herbaceous list. Data is collected only at the sapling survey plot of the subplot.

soil Soil table including the thickness of the organic layer and soil depth, among others. Data is collected from three soil survey plots located within each selected center subplot. The selection includes 25% of the total number of center subplots.

For more details, refer to the National Forest Inventory guidelines.

This function performs several data integrity validation.

1. Corrects administrative region information for subplots. (col: SIDO, SIDO\_CD, SGG, SGG\_CD, EMD, EMD\_CD)
2. Adds ecoregion and catchment for subplots. (col: ECOREGION, CATCHMENT)
3. Verifies and corrects coniferous/deciduous classification of tree species. (col: CONDEC\_CLASS, CONDEC\_CLASS\_CD, WDY\_PLNTS\_TYP, WDY\_PLNTS\_TYP\_CD)
4. Adds scientific names for species. (col: SCIENTIFIC\_NAME)
5. Adds Korean and English names for plant families and genera. (col: FAMILY, FAMILY\_KOREAN, GENUS, GENUS\_KOREAN)
6. Adds whether a plant is native or cultivated, and identifies if it is a food, medicinal, fiber, or ornamental resource. (col: NATIVE\_CULTIVATED, FOOD, MEDICINAL, FIBER, ORNAMENTAL)
7. Calculates basal area for individual tree (col: BASAL\_AREA)
8. Calculates forest type, dominant species, and dominant species percentage for each subplot and cluster plot. (col: FORTYP\_SUB, DOMIN\_PERCNT\_SUB, DOMIN\_SP\_SUB, FORTYP\_CLST, DOMIN\_PERCNT\_CLST, DOMIN\_SP\_CLST) Species classification and taxonomy follow the standards set by the Korean Plant Names Index Committee of the Korea National Arboretum <http://www.nature.go.kr/kpni/index.do>.

### Value

A data.frame; the processed NFI data, structured for easy analysis.

### Note

To manually download subsets of the annual NFI file, visit the Korea Forest Service Forestry Statistics Platform (<https://kfss.forest.go.kr/stat/>), download .zip files, and extract them.

-The 5th National Forest Inventory file: <https://kfss.forest.go.kr/stat/ptl/article/articleFileDownload.do?fileSeq=2995&workSeq=2203> -The 6th National Forest Inventory file: <https://kfss.forest.go.kr/stat/ptl/article/articleFileDownload.do?fileSeq=2996&workSeq=2204> -The 7th National

Forest Inventory file: <https://kfss.forest.go.kr/stat/ptl/article/articleFileDown.do?fileSeq=2997&workSeq=2205>

Use `data("nfi_col")` to view the Korean and English names of the column names.

While the National Forest Inventory undergoes rigorous quality control, including internal reviews and field inspections, errors may still exist due to the extensive nature of the survey (approximately 4,000 plots and over 70 items in the 7th phase). Please use the data cautiously and report any anomalies to help improve our algorithms.

If you want to save the results to your computer, you can save them in Excel format. For example, you can use the following code: `writexl::write_xlsx(data, "data.xlsx")`

If you want to read the saved data back, use the code below: `path <- "../nfi_donghae.xlsx"`  
`sheet_names <- readxl::excel_sheets(path)` for `(sheet_name in sheet_names) {nfi[[sheet_name]]`  
`<- readxl::read_excel(path, sheet = sheet_name) }`

## Examples

```
# Load tree and CWD data for all districts
nfi5_data <- read_nfi("D:/NFI/", district = NULL, tables = c("tree", "cwd"), recursive = TRUE)
```

---

summary\_nfi

*Calculate descriptive statistics for National Forest Inventory Data*

---

## Description

`summary_nfi()` is a function that calculates comprehensive descriptive statistics for National Forest Inventory (NFI) data. It can provide summaries for individual plots, the entire study area, or specific groups within the study area using parameters `byplot` or `plotgrp`. Users have flexibility in specifying data inclusion criteria and analysis levels using parameters `clusterplot`, `largetreearea`, `stockedland`, and `talltree`. These parameters determine whether to treat cluster plots as single plots, to include large tree survey plots, and to focus only on Stocked land and tall trees.

## Usage

```
summary_nfi(
  data,
  byplot = FALSE,
  plotgrp = NULL,
  continuousplot = FALSE,
  clusterplot = FALSE,
  largetreearea = TRUE,
  stockedland = TRUE,
  talltree = TRUE
)
```



## Arguments

<code>data</code>	: A list containing 'plot' and 'tree' data frames, typically generated by <code>read_nfi</code> .
<code>byplot</code>	: A logical flag (default FALSE); if TRUE, calculates statistics for each plot separately. If FALSE, calculates for the entire dataset.
<code>plotgrp</code>	: A character vector; specifies variables from 'plot' table to use for grouping. Use <code>c()</code> to combine multiple variables.
<code>continuousplot</code>	: A logical flag (default TRUE); if TRUE, includes only plots that have been continuously measured in all NFI cycles (5th, 6th, etc.). If FALSE, includes plots regardless of missing cycle measurements.
<code>clusterplot</code>	: A logical flag (default FALSE); if TRUE, treats each cluster plot as a single unit. If FALSE, calculates for each subplot separately.
<code>largetreearea</code>	: A logical flag (default TRUE); if TRUE, includes large tree survey plots in the analysis. If FALSE, only uses standard tree plots.
<code>stockedland</code>	: A logical flag (default TRUE); if TRUE, includes only stocked land. If FALSE, includes all land types.
<code>talltree</code>	: A logical flag (default TRUE); if TRUE, includes only tall trees. If FALSE, includes both trees and shrubs.

## Details

The function calculates the following statistics: Plot-related statistics:

- Number of cluster plots
- Number of subplots
- Number of subplots with large trees ( $\geq 30$ cm) observed

Tree-related statistics:

- Number of individual trees
- Number of large trees
- Number of dominant trees
- Number of tree species

Tree measurements and stand statistics (mean and standard deviation):

- DBH (Diameter at Breast Height)
- Tree height
- Height of dominant trees
- Number of trees per hectare
- Basal area per hectare
- Volume per hectare

The `largetreearea` parameter affects calculations differently:

- For per-hectare statistics (trees per hectare, basal area per hectare, volume per hectare), setting `largetreearea = TRUE` includes data from large tree survey plots.
- For all other statistics, trees from large tree survey plots are always excluded, regardless of the `largetreearea` setting.

**Value**

A data.frame that includes summary statistics. The structure depends on the input parameters:

- If `byplot = TRUE`, each row represents a plot.
- If `byplot = FALSE`, each row represents the entire dataset or a group specified by `plotgrp`

**Examples**

```
data("nfi_donghae")

# Basic usage
summary_stats <- summary_nfi(nfi_donghae, continuousplot = TRUE)

# Summarize by the group, including all land types
grouped_stats <- summary_nfi(nfi_donghae, plotgrp = "OWN_CD",
                             stockedland = FALSE, continuousplot = TRUE)

# Summarize by individual plots, including both trees and shrubs
plot_summaries <- summary_nfi(nfi_donghae, byplot = TRUE, talltree = FALSE)
```

---

switchcol_nfi	<i>Switch column names</i>
---------------	----------------------------

---

**Description**

The `switchcol_nfi()` function allows switching between the original Korean column names and English column names. If the input data has English column names, it changes them to the original Korean names, and vice versa.

**Usage**

```
switchcol_nfi(data)
```

**Arguments**

`data` : A list generated by [read\\_nfi](#).

**Details**

Important: Data with original Korean column names cannot be used with any [read\\_nfi](#) functions except `switchcol_nfi()` itself. The option to revert to original Korean names is provided solely for users who wish to store or process the NFI data with original column names.

**Value**

A list of dataframes with switched column names.

## Examples

```
data("nfi_donghae")

#Switch column names from English to original Korean names
nfi_donghae_kor <- switchcol_nfi(nfi_donghae)

# Switch column names from original Korean to English names
nfi_donghae_eng <- switchcol_nfi(nfi_donghae_kor)
```

---

 tsvis\_nfi

*Visualize time series data for National Forest Inventory Data*


---

## Description

tsvis\_nfi() function analyzes and visualizes data in a time series format. It can visualize 'biomass', 'cwd', and 'iv' data as 'table', 'line', 'bar', or 'map'. Users need to select the specific biomass variable, such as volume or carbon to visualize biomass. For map visualization of biomass, users must choose the administrative unit level. It uses the [iv\\_nfi](#) and [biomass\\_nfi](#) and [cwd\\_biomass\\_nfi](#) function for core calculations. The function accepts various parameters depending on the analysis type: Common parameters: data, y, output, plotgrp, isannual, ncol, nrow Additional parameters by type: biomass: bm\_type(except for 'table'), admin(only 'map'), treegrp, strat, stockedland, talltree, clusterplot, largetreearea cwd: bm\_type(except for 'table'), admin(only 'map'), treegrp, strat, stockedland iv: sp, frequency, top When y is set to either 'biomass' or 'cwd': For map output, the function displays two side-by-side spatial visualizations: one showing the results and another showing the relative standard error (RSE). For line or bar outputs, results are displayed with error bars based on the standard error (SE) of the estimates.

## Usage

```
tsvis_nfi(
  data,
  y = "biomass",
  bm_type = NULL,
  output = "line",
  plotgrp = NULL,
  treegrp = NULL,
  isannual = TRUE,
  continuousplot = FALSE,
  admin = NULL,
  strat = "FORTYP_SUB",
  clusterplot = FALSE,
  largetreearea = TRUE,
  stockedland = TRUE,
  talltree = TRUE,
  sp = "SP",
  frequency = TRUE,
```

```

    ncol = NULL,
    nrow = NULL,
    top = 20
  )

```

### Arguments

<code>data</code>	: A list generated by <code>read_nfi</code> that contains 'plot' and 'tree' data frames.
<code>y</code>	: A character vector; the variable to visualize. Must be one of 'biomass', 'cwd', 'iv'.
<code>bm_type</code>	: A character vector; the specific variable to visualize for 'biomass' or 'cwd'. Must be one of 'volume', 'biomass', 'AG_biomass', 'carbon', 'co2'.
<code>output</code>	: A character vector; the desired type of visualization. Must be one of 'table', 'line', 'bar', 'map'.
<code>plotgrp</code>	: A character vector; specifies variables from the 'plot' table to use for grouping. Use <code>c()</code> to combine multiple variables, but if the output is a map, line or bar plot, only one variable can be used.
<code>treegrp</code>	: A character vector; variables from 'tree' tables for grouping. Use <code>c()</code> to combine multiple variables.
<code>isannual</code>	: A logical flag (default TRUE); if TRUE, the result is provided annually, if FALSE, it is provided in 5-year intervals.
<code>continuousplot</code>	: A logical flag (default TRUE); if TRUE, includes only plots that have been continuously measured in all NFI cycles (5th, 6th, etc.). If FALSE, includes plots regardless of missing cycle measurements.
<code>admin</code>	: A character vector; the administrative unit for visualizing 'biomass' or 'cwd' as a map. Must be one of 'sido', 'sgg', 'emg'.
<code>strat</code>	: A character vector; the variable used for post-stratification. In the National Forest Inventory of Korea, it is typically used by forest type.
<code>clusterplot</code>	: A logical flag (default FALSE); if TRUE, treats each cluster plot as a single unit. If FALSE, calculates for each subplot separately.
<code>largetreearea</code>	: A logical flag (default FALSE); if TRUE, includes large tree survey plots in the analysis. If FALSE, only uses standard tree plots.
<code>stockedland</code>	: A logical flag (default TRUE); if TRUE, includes only stocked land. If FALSE, includes all land types.
<code>talltree</code>	: A logical flag (default TRUE); if TRUE, includes only tall trees. If FALSE, includes both trees and shrubs.
<code>sp</code>	: A character vector; the column name of species information (e.g., "SP" for species, "GENUS" for genus-level analysis).
<code>frequency</code>	: A logical flag (default TRUE); if TRUE, includes frequency in importance value calculations.
<code>ncol</code>	: A numeric vector (default NULL); Number of columns in the grid layout when faceting by <code>plotgrp</code> .
<code>nrow</code>	: A numeric vector (default NULL); Number of columns in the grid layout when faceting by <code>plotgrp</code> .
<code>top</code>	: A numeric vector (default 20); Number of top species to display in 'iv' plot

**Value**

If the output is a map, line, or bar plot: an object of class **ggplot**. If the output is a table: a `data.frame`.

**Note**

To visualize data as a map, you need to agree to install the `kadmin` package during the function execution or install it in advance. The `kadmin` package loads shapefiles for Korea's Si, Do or Si, Gun, Gu or Eup, Myeon, Dong. Use `drat::addRepo("SYOUNG9836")` `install.packages("kadmin")` or `remotes::install_github("SYOUNG9836/kadmin")`

**See Also**

[iv\\_nfi](#) for calculating the importance values. [biomass\\_nfi](#) for calculating the biomass. [cwd\\_biomass\\_nfi](#) for calculating the biomass of Coarse Woody Debris.

**Examples**

```
data("nfi_donghae")

# Visualize importance values as a table
tsvis_iv <- tsvis_nfi(nfi_donghae, y = "iv", output = "table", continuousplot = TRUE)

# Create a bar plot of importance values at 5-year intervals
tsvis_iv_bar <- tsvis_nfi(nfi_donghae, y = "iv", output = "bar", isannual = FALSE,
                        continuousplot = TRUE)

# Generate a line plot of carbon biomass over time, grouped by age class
tsvis_bm_line <- tsvis_nfi(nfi_donghae, y = "biomass", top = 10, bm_type = "carbon",
                        output = "line", plotgrp = "AGE_CLS", continuousplot = TRUE)

# Create a map of volume at the sido level
tsvis_bm_map <- tsvis_nfi(nfi_donghae, admin = "sido",
                        y = "biomass", bm_type = "volume", output = "map", continuousplot = TRUE)
```

# Index

## \* datasets

nfi\_col, [13](#)

nfi\_donghae, [13](#)

biomass\_nfi, [3](#), [19](#), [21](#)

cwd\_biomass\_nfi, [5](#), [19](#), [21](#)

diversity, [7](#), [9](#)

diversity\_nfi, [7](#)

filter\_nfi, [9](#)

importancevalue, [11](#), [12](#)

iv\_nfi, [11](#), [19](#), [21](#)

nfi\_col, [13](#)

nfi\_donghae, [13](#)

read\_nfi, [3](#), [5](#), [7](#), [9](#), [11](#), [14](#), [17](#), [18](#), [20](#)

summary\_nfi, [16](#)

switchcol\_nfi, [18](#)

tsvis\_nfi, [19](#)