Package 'TwoSampleCalImpute'

November 8, 2026

Type Package

Title Two-Sample Calibrated Imputation for Survey Data

Version 0.1.0

Description Provides utilities for preparing, validating, summarising, and documenting survey data from primary and auxiliary samples, and for specifying and checking survey-design information. The package supports data-management and design-specification steps in two-sample calibrated imputation workflows. For background on survey calibration, see Deville and Särndal (1992) <doi:10.1080/01621459.1992.10475217>.

Depends R (>= 4.3.0)

License GPL (>= 3)

Encoding UTF-8

Config/roxygen2/version 8.1.0

Suggests covr, devtools, pkgdown, rcmdcheck, roxygen2, testthat, usethis

URL https://github.com/Kamun2026/TwoSampleCalImpute

BugReports https://github.com/Kamun2026/TwoSampleCalImpute/issues

NeedsCompilation no

Author Samuel Joel Kamun [aut, cre]

Maintainer Samuel Joel Kamun <samuelkamun@gmail.com>

Repository CRAN

Date/Publication 2026-11-08 16:50:27 UTC

Contents

print.tsci_data 2
print.tsci_design 3
tsci_clean_data 4
tsci_clusters 5
tsci_create_design 6

tsci_data_dictionary 7
tsci_design_check 8
tsci_design_plot 9
tsci_design_summary 10
tsci_export_data 11
tsci_fpc 12
tsci_import_data 13
tsci_merge_data 14
tsci_missing_summary 15
tsci_prepare_data 16
tsci_recode_variables 17
tsci_strata 18
tsci_subset_data 19
tsci_transform_data 20
tsci_update_design 21
tsci_validate_data 22
tsci_variable_summary 23
tsci_weights 24

Index **26**

`print.tsci_data` *Print a tsci_data object*

Description

Displays a summary of a standardized TSCI data object, including its sample type, number of observations, and number of variables.

Usage

```
## S3 method for class 'tsci_data'
print(x, ...)
```

Arguments

`x`	A `tsci_data` object.
`...`	Additional arguments (currently unused).

Value

Invisibly returns `x`, the original `tsci_data` object. The object retains its `tsci_data` and `list` classes.

Examples

```
survey_data <- data.frame(
  id = 1:3,
  outcome = c(1, 0, 1)
)

obj <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

print(obj)
```

---

print.tsci_design *Print a tsci_design object*

Description

Displays the sample type, number of observations, and whether sampling weights have been specified for a survey-design object.

Usage

```
## S3 method for class 'tsci_design'
print(x, ...)
```

Arguments

x	A tsci_design object.
...	Additional arguments (currently unused).

Value

Invisibly returns x, the original tsci_design object. The object retains its tsci_design and list classes.

Examples

```
survey_data <- data.frame(
  id = 1:3,
  outcome = c(1, 0, 1),
  weight = c(1, 1, 1)
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
```

```
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight"
)

print(design_object)
```

`tsci_clean_data` *Clean survey data*

Description

Performs controlled data-cleaning operations on a `tsci_data` object without silently changing observed values. Completely empty rows or columns are removed only when the corresponding option is set to `TRUE`.

Usage

```
tsci_clean_data(data, remove_empty_rows = FALSE, remove_empty_columns = FALSE)
```

Arguments

`data`	A `tsci_data` object.
`remove_empty_rows`	Logical; whether completely empty rows should be removed. Defaults to `FALSE`.
`remove_empty_columns`	Logical; whether completely empty columns should be removed. Defaults to `FALSE`.

Value

A cleaned `tsci_data` object containing the resulting data, with its sample type, identifier, and metadata preserved. The variable list is updated to include only variables still present.

Examples

```
survey_data <- data.frame(
  id = c(1, 2, 3),
  outcome = c(1, NA, 0),
  auxiliary = c(NA, NA, NA)
)

imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
```

```
  id = "id"
)

cleaned <- tsci_clean_data(
  imported,
  remove_empty_columns = TRUE
)
names(cleaned$data)
```

`tsci_clusters` *Extract the Cluster Variable*

Description

Extracts the name of the cluster variable stored in a `tsci_design` object. If no cluster variable was specified, the function returns `NULL`. The stored specification is validated against the variables in the design data before the name is returned.

Usage

```
tsci_clusters(design)
```

Arguments

`design`	A `tsci_design` object containing the survey design specification and its associated data.

Value

A character string of length one containing the name of the cluster variable, or `NULL` if no cluster variable was specified. An error is raised if `design` is not a `tsci_design` object, if the stored cluster specification is invalid, or if the specified variable is absent from the design data.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 1, 1, 1),
  cluster = c("C1", "C1", "C2", "C2")
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)
```

```
design_object <- tsci_create_design(
  data_object,
  weights = "weight",
  clusters = "cluster"
)

tsci_clusters(design_object)

design_without_clusters <- tsci_create_design(
  data_object,
  weights = "weight"
)
tsci_clusters(design_without_clusters)
```

`tsci_create_design` *Create a survey design object*

Description

Constructs a `tsci_design` object from a `tsci_data` object and specified survey-design variables. The weight variable must contain numeric, finite, strictly positive values. If supplied, the finite-population-correction (FPC) variable must also contain numeric, finite, strictly positive values.

Usage

```
tsci_create_design(data, weights, strata = NULL, clusters = NULL, fpc = NULL)
```

Arguments

`data`	A `tsci_data` object containing the survey observations.
`weights`	Character string naming the survey-weight variable in `data$data`.
`strata`	Optional character string naming the stratification variable in `data$data`. Defaults to `NULL`.
`clusters`	Optional character string naming the cluster variable in `data$data`. Defaults to `NULL`.
`fpc`	Optional character string naming the finite-population- correction variable in `data$data`. Defaults to `NULL`.

Value

An object of class `tsci_design`, represented as a list containing the survey-design information constructed from the input data object. The object includes the survey weights and the specified design variables, with the underlying data-object information retained according to the package's design-object constructor.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 2, 1, 2),
  stratum = c("A", "A", "B", "B")
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight",
  strata = "stratum"
)

print(design_object)
```

---

tsci_data_dictionary *Create a data dictionary for a TSCI data object*

Description

Produces a structured data dictionary describing the variables in a `tsci_data` object.

Usage

```
tsci_data_dictionary(data)
```

Arguments

`data`	A `tsci_data` object.

Value

A data frame with one row per variable and five columns: `variable` (variable name), `class` (R class or classes), `n` (number of observations), `missing` (number of missing values), and `description` (variable description from the object's metadata, or `NA` if no matching description is available).

Examples

```
survey_data <- data.frame(
  id = 1:3,
  outcome = c(1, NA, 0),
  age = c(35, 42, 29)
)

imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id",
  metadata = list(
    description = c(
      outcome = "Recorded outcome",
      age = "Age in years"
    )
  )
)

tsci_data_dictionary(imported)
```

---

tsci_design_check *Validate survey-design information*

Description

Checks the structural and numerical validity of a `tsci_design` object, including its underlying data object, number of observations, survey weights, stratification variable, and finite-population correction values.

Usage

```
tsci_design_check(design)
```

Arguments

`design`	A `tsci_design` object to validate.

Value

An object of class `c("tsci_design_check", "list")` containing:

valid Logical value indicating whether all checks pass.

checks Named list of logical results for the individual structural and numerical checks.

sample_type Sample type recorded in the underlying `tsci_data` object.

observations Number of rows in the underlying data.

id Identifier information stored in the design object.

strata Stratification variable information stored in the design object.

has_weights Logical value indicating whether survey weights are stored.

has_fpc Logical value indicating whether finite-population correction values are stored.

The `valid` field is `TRUE` only when every individual check passes. An input that is not a `tsci_design` object causes an error.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 2, 1, 2),
  stratum = c("A", "A", "B", "B")
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight",
  strata = "stratum"
)

check_result <- tsci_design_check(design_object)
check_result$valid
check_result$checks
```

---

tsci_design_plot *Plot the Survey-Design Structure*

Description

Produces a base-R plot indicating whether weights, strata, clusters, and finite-population correction (FPC) values are specified in a `tsci_design` object. A component is displayed as specified when its corresponding stored value is not `NULL`.

Usage

```
tsci_design_plot(design)
```

Arguments

`design`	A `tsci_design` object containing the survey design specification and its associated data.

Value

The input `tsci_design` object, returned invisibly. The function also produces a plot with one indicator for each of the four design components: weights, strata, clusters, and FPC. An error is raised if `design` is not a `tsci_design` object.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 2, 1, 2),
  stratum = c("A", "A", "B", "B")
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight",
  strata = "stratum"
)

returned_design <- tsci_design_plot(design_object)
stopifnot(inherits(returned_design, "tsci_design"))
```

`tsci_design_summary` *Summarise a survey design*

Description

Produces a concise summary of the main components of a `tsci_design` object.

Usage

```
tsci_design_summary(design)
```

Arguments

`design`	A `tsci_design` object.

Value

A one-row data frame with seven columns: `sample_type` (type of sample represented by the underlying data), `observations` (number of observations), `variables` (number of variables in the data), `id` (identifier variable name, or `NA` if none is specified), `weights` (logical indicator of whether survey weights are stored), `strata` (stratification variable name, or `NA` if none is specified), and `fpc` (logical indicator of whether finite-population-correction values are stored).

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 2, 1, 2),
  stratum = c("A", "A", "B", "B")
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight",
  strata = "stratum"
)

tsci_design_summary(design_object)
```

`tsci_export_data` *Export survey data*

Description

Exports the data contained in a `tsci_data` object to a CSV file.

Usage

```
tsci_export_data(data, file, row_names = FALSE)
```

Arguments

`data`	A `tsci_data` object.
`file`	Character path for the output CSV file. The filename must have a `.csv` extension.
`row_names`	Logical; whether row names should be written.

Value

Invisibly returns the original `tsci_data` object supplied through `data`. The object retains its existing data and metadata.

Examples

```
survey_data <- data.frame(
  id = 1:3,
  outcome = c(1, 0, 1)
)
imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

output_file <- tempfile(fileext = ".csv")
returned <- tsci_export_data(imported, file = output_file)
file.exists(output_file)
identical(returned, imported)
unlink(output_file)
```

---

tsci_fpc *Extract the finite-population correction*

Description

Extracts and validates the finite-population-correction (FPC) values stored in a `tsci_design` object.

Usage

```
tsci_fpc(design)
```

Arguments

`design`	A `tsci_design` object containing the survey design specification and its associated data.

Value

A numeric vector containing one finite, strictly positive FPC value per observation, in the stored order, or `NULL` if no FPC was specified. An error is raised if `design` is not a `tsci_design` object, or if the stored FPC is non-numeric, has an incorrect length, contains missing or non-finite values, or contains non-positive values.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 2, 1, 2),
  fpc = c(100, 100, 200, 200)
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight",
  fpc = "fpc"
)

tsci_fpc(design_object)

design_without_fpc <- tsci_create_design(
  data_object,
  weights = "weight"
)
tsci_fpc(design_without_fpc)
```

---

tsci_import_data *Import survey data*

Description

Imports a data set from a supported file or an existing R object and returns it as a standardized `tsci_data` object.

Usage

```
tsci_import_data(
  data,
  sample_type = c("primary", "auxiliary"),
  id = NULL,
  variables = NULL,
  metadata = list()
)
```

Arguments

data	A data frame, or a character path to a CSV file.
sample_type	Either "primary" or "auxiliary".
id	Optional character identifier variable.
variables	Optional character vector of variables to retain.
metadata	Optional list of metadata.

Value

A tsci_data object, represented as a list containing the imported data, sample type, identifier, variable names, and metadata.

Examples

```
survey_data <- data.frame(
  id = 1:3,
  outcome = c(1, 0, 1),
  age = c(35, 42, 29)
)

imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

print(imported)
```

---

tsci_merge_data *Merge two TSCI data objects*

Description

Merge two TSCI data objects

Usage

```
tsci_merge_data(x, y, by, all = FALSE)
```

Arguments

x	A tsci_data object.
y	A second tsci_data object.
by	Character vector of variables used as merge keys.
all	Logical value indicating whether to retain unmatched observations from both objects. Defaults to FALSE.

Details

Merges two tsci_data objects using one or more key variables.

Value

A tsci_data object containing the merged data. The result retains the sample type and metadata from x; its identifier is retained only if that identifier is present in the merged data.

Examples

```
primary_data <- data.frame(
  id = c(1, 2, 3),
  outcome = c(1, 0, 1)
)

auxiliary_data <- data.frame(
  id = c(1, 2, 3),
  age_group = c("young", "middle", "older")
)

primary <- tsci_import_data(
  primary_data,
  sample_type = "primary",
  id = "id"
)

auxiliary <- tsci_import_data(
  auxiliary_data,
  sample_type = "auxiliary",
  id = "id"
)

merged <- tsci_merge_data(
  primary,
  auxiliary,
  by = "id"
)

names(merged$data)
```

---

tsci_missing_summary *Summarize missing values in a TSCI data object*

Description

Produces a variable-level summary of missing and observed values in a tsci_data object.

Usage

```
tsci_missing_summary(data, variables = NULL)
```

Arguments

data	A tsci_data object.
variables	Optional character vector of variables to summarize. If NULL, all variables are summarized.

Value

A data frame with one row per selected variable and five columns: variable (variable name), n (total number of observations), missing (number of missing values), observed (number of non-missing values), and missing_percent (percentage of observations with missing values, on a 0–100 scale).

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, NA, 0, NA),
  group = c("A", "B", "A", "B")
)

imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

tsci_missing_summary(imported, variables = "outcome")
```

---

tsci_prepare_data *Prepare survey data*

Description

Prepares a tsci_data object for subsequent validation and Two-Sample Calibrated Imputation analysis by applying the specified empty-row and empty-column cleaning operations.

Usage

```
tsci_prepare_data(
  data,
  remove_empty_rows = FALSE,
  remove_empty_columns = FALSE
)
```

Arguments

data	A tsci_data object.
remove_empty_rows	Logical; whether completely empty rows should be removed. Defaults to FALSE.
remove_empty_columns	Logical; whether completely empty columns should be removed. Defaults to FALSE.

Value

A prepared tsci_data object. Its data reflect the requested cleaning operations; its sample type, identifier, and metadata are preserved by the cleaning operation.

Examples

```
survey_data <- data.frame(
  id = c(1, 2, 3),
  outcome = c(1, NA, 0),
  empty_variable = c(NA, NA, NA)
)

imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

prepared <- tsci_prepare_data(
  imported,
  remove_empty_columns = TRUE
)
names(prepared$data)
```

---

tsci_recode_variables *Recode variables in a TSCI data object*

Description

Applies user-supplied recoding functions to one or more variables in a tsci_data object.

Usage

```
tsci_recode_variables(data, recodes)
```

Arguments

data	A tsci_data object.
recodes	A named list of functions. Each name must identify a variable in data.

Value

A `tsci_data` object containing the recoded variables. The sample type, identifier, variable list, and metadata are preserved from the input object.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  response = c("Yes", "No", "Yes", "No")
)

imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

recoded <- tsci_recode_variables(
  imported,
  recodes = list(response = function(x) {
    ifelse(x == "Yes", 1, 0)
  })
)

recoded$data$response
```

---

tsci_strata *Extract the Stratification Variable*

Description

Extracts the name of the stratification variable stored in a `tsci_design` object. If no stratification variable was specified, the function returns `NULL`. The stored specification is validated against the variables in the design data before the name is returned.

Usage

```
tsci_strata(design)
```

Arguments

`design`	A `tsci_design` object containing the survey design specification and its associated data.

Value

A character string of length one containing the name of the stratification variable, or NULL if no stratification variable was specified. An error is raised if design is not a tsci_design object, if the stored stratification specification is invalid, or if the specified variable is absent from the design data.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 1, 1, 1),
  stratum = c("A", "A", "B", "B")
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight",
  strata = "stratum"
)

tsci_strata(design_object)

design_without_strata <- tsci_create_design(
  data_object,
  weights = "weight"
)
tsci_strata(design_without_strata)
```

---

tsci_subset_data *Subset a TSCI data object*

Description

Subsets observations in a tsci_data object using a logical condition while preserving the TSCI data-object structure.

Usage

```
tsci_subset_data(data, subset)
```

Arguments

data	A tsci_data object.
subset	A logical vector identifying observations to retain.

Value

A tsci_data object containing the selected observations. The sample type, identifier, and metadata are preserved, and the variable list is restricted to variables present in the result.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0)
)

imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

selected <- tsci_subset_data(
  imported,
  subset = c(TRUE, FALSE, TRUE, FALSE)
)

nrow(selected$data)
selected$data$id
```

---

tsci_transform_data *Transform variables in a TSCI data object*

Description

Applies a user-supplied transformation function to one or more variables in a tsci_data object while preserving the TSCI data object structure.

Usage

```
tsci_transform_data(data, variables, fun, ...)
```

Arguments

data	A tsci_data object.
variables	Character vector of variable names to transform.
fun	A function used to transform each selected variable.
...	Additional arguments passed to fun.

Value

A tsci_data object containing the transformed variables.

Examples

```
x <- data.frame(id = 1:3, y = c(2, 4, 6))
obj <- tsci_import_data(x, sample_type = "primary", id = "id")
tsci_transform_data(obj, variables = "y", fun = log)
```

---

tsci_update_design *Update a survey design object*

Description

Updates selected components of an existing tsci_design object. Non-NULL values supplied for weights, strata, and fpc replace the corresponding stored components. The clusters argument is passed to the design-object constructor, so omitting it does not guarantee that a previously specified cluster setting is retained.

Usage

```
tsci_update_design(
  design,
  weights = NULL,
  strata = NULL,
  clusters = NULL,
  fpc = NULL
)
```

Arguments

design	A tsci_design object to update.
weights	Optional character string naming the survey-weight variable in the underlying data. If NULL, the existing stored weights are used.
strata	Optional character string naming the stratification variable in the underlying data. If NULL, the existing stored strata setting is used.
clusters	Optional character string naming the cluster variable in the underlying data. Defaults to NULL.
fpc	Optional character string naming the finite-population- correction variable in the underlying data. If NULL, the existing stored FPC values are used.

Value

An object of class tsci_design containing the updated survey-design components and the underlying data-object information. The weights and FPC values are checked for numeric, finite, strictly positive values when their variable names are supplied. Invalid design inputs cause an error.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 2, 1, 2),
  updated_weight = c(2, 2, 3, 3),
  stratum = c("A", "A", "B", "B")
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight",
  strata = "stratum"
)

updated_design <- tsci_update_design(
  design_object,
  weights = "updated_weight"
)

print(updated_design)
```

---

tsci_validate_data *Validate a survey data set*

Description

Performs data-integrity checks before Two-Sample Calibrated Imputation analysis, including input validation, optional variable and weight checks, missingness summaries, and duplicate-key checks.

Usage

```
tsci_validate_data(
  data,
  data_name = "data",
  variables = NULL,
  weights = NULL,
  key = NULL
)
```

Arguments

data	A data frame to validate.
data_name	A character label used in validation messages. Defaults to "data".
variables	Optional character vector of variables to validate. Defaults to NULL.
weights	Optional survey weights to validate. Defaults to NULL.
key	Optional character vector identifying records for duplicate checks. Defaults to NULL.

Value

An object of class c("tsci_validation", "list") containing valid (validation status), data_name (the supplied data label), n (number of rows), p (number of columns), variables (column names), missing (missingness-check results), and duplicates (duplicate-key-check results). Invalid inputs cause an error.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, NA, 1),
  weight = c(1, 1, 1, 1)
)

validation <- tsci_validate_data(
  survey_data,
  variables = c("outcome", "weight"),
  weights = survey_data$weight,
  key = "id"
)

validation$valid
validation$n
validation$missing
```

---

tsci_variable_summary *Summarize variables in a TSCI data object*

Description

Produces descriptive summaries for selected variables in a tsci_data object.

Usage

```
tsci_variable_summary(data, variables = NULL)
```

Arguments

data	A tsci_data object.
variables	Optional character vector of variables to summarize. If NULL, all variables are summarized.

Value

A data frame with one row per selected variable and five columns: variable (variable name), class (R class), n (number of observations), missing (number of missing values), and unique (number of distinct non-missing values).

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, NA, 1),
  group = c("A", "B", "A", "B")
)

imported <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

tsci_variable_summary(imported, variables = "outcome")
```

---

tsci_weights *Extract survey weights*

Description

Extracts and validates the numeric survey weights stored in a tsci_design object.

Usage

```
tsci_weights(design)
```

Arguments

design	A tsci_design object containing survey weights.

Value

A numeric vector containing the survey weights in their stored order. The vector has one element per observation and contains only finite, strictly positive values. An error is raised if the design object is invalid, weights are absent, weights are non-numeric, or any weight is missing, non-finite, or non-positive.

Examples

```
survey_data <- data.frame(
  id = 1:4,
  outcome = c(1, 0, 1, 0),
  weight = c(1, 2, 1, 2)
)

data_object <- tsci_import_data(
  survey_data,
  sample_type = "primary",
  id = "id"
)

design_object <- tsci_create_design(
  data_object,
  weights = "weight"
)

tsci_weights(design_object)
```

Index

print.tsci_data, 2
print.tsci_design, 3

tsci_clean_data, 4
tsci_clusters, 5
tsci_create_design, 6
tsci_data_dictionary, 7
tsci_design_check, 8
tsci_design_plot, 9
tsci_design_summary, 10
tsci_export_data, 11
tsci_fpc, 12
tsci_import_data, 13
tsci_merge_data, 14
tsci_missing_summary, 15
tsci_prepare_data, 16
tsci_recode_variables, 17
tsci_strata, 18
tsci_subset_data, 19
tsci_transform_data, 20
tsci_update_design, 21
tsci_validate_data, 22
tsci_variable_summary, 23
tsci_weights, 24