# Package ‘pxmake’

January 22, 2025

**Title** Make PX-Files in R

**Version** 0.14.2

**Description** Create PX-files from scratch or read and modify existing ones. Includes a function for every PX keyword, making metadata manipulation simple and human-readable.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** arrow, dplyr, furrr, magrittr, methods, openxlsx, purrr, readxl, rlang, stats, stringi, stringr, tibble, tidyr, tools, utils, vctrs

**Suggests** knitr, rmarkdown, withr, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Config/testthat/start-first** 60-metamake-bexltall

**VignetteBuilder** knitr

**Depends** R (>= 3.5.0)

**LazyData** true

**URL** https://github.com/StatisticsGreenland/pxmake, https://statisticsgreenland.github.io/pxmake/

**BugReports** https://github.com/StatisticsGreenland/pxmake/issues

**NeedsCompilation** no

**Author** Johan Ejstrud [cre, aut],
Lars Pedersen [aut],
Statistics Greenland [cph] (https://stat.gl/)

**Maintainer** Johan Ejstrud <johan@ejstrud.com>

**Repository** CRAN

**Date/Publication** 2025-01-22 16:50:02 UTC

# Contents

age_classification . . . . 3
greenlanders . . . . 4
population_gl . . . . 4
px . . . . 5
px_add_totals . . . . 6
px_aggregallowed . . . . 7
px_autopen . . . . 8
px_axis_version . . . . 9
px_baseperiod . . . . 10
px_cellnote . . . . 11
px_cellnotex . . . . 13
px_cfprices . . . . 14
px_charset . . . . 15
px_classification . . . . 16
px_codepage . . . . 18
px_confidential . . . . 19
px_contact . . . . 20
px_contents . . . . 21
px_contvariable . . . . 22
px_copyright . . . . 23
px_creation_date . . . . 24
px_data . . . . 25
px_decimals . . . . 27
px_description . . . . 28
px_descriptiondefault . . . . 29
px_domain . . . . 30
px_elimination . . . . 31
px_figures . . . . 33
px_heading . . . . 34
px_infofile . . . . 35
px_keywords . . . . 36
px_language . . . . 37
px_languages . . . . 38
px_last_updated . . . . 39
px_link . . . . 40
px_map . . . . 41
px_matrix . . . . 43
px_micro . . . . 44
px_next_update . . . . 45
px_note . . . . 46
px_notex . . . . 47
px_order . . . . 49
px_precision . . . . 50
px_save . . . . 51
px_save_classification . . . . 52
px_showdecimals . . . . 53

px_source . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
px_stockfa . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
px_stub . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
px_subject_area . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
px_subject_code . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
px_tableid . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
px_timeval . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
px_title . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61
px_units . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
px_update_frequency . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
px_validate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
px_valuenote . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
px_valuenotex . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
px_values . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
px_variable_label . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
px_variable_type . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

**Index** **72**

## `age_classification` *Age classification*

### Description

Example data set to create age classification with aggreations form 10 and 25 years classes.

### Usage

```
age_classification
```

### Format

A data frame:

**valuecode** Value code

**valuetext** Value text

**10-years_classes** Aggregtation into 10 years classes

**25-years_classes** Aggregtation into 25 years classes

## `greenlanders` *Greenlanders*

### Description

A fictive data set with demographic data for Greenlanders split in two cohorts.

### Usage

```
greenlanders
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 100 rows and 4 columns.

## `population_gl` *Population Greenland*

### Description

A subset of the population count data available in Statistic Greenland's BEESTA table.

### Usage

```
population_gl
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 30 rows and 4 columns.

### Source

`https://bank.stat.gl/pxweb/en/Greenland/Greenland__BE__BE01__BE0120/BEXSTA.px/`

px *Create a px object*

## Description

Create a px object from a PX-file, an Excel metadata workbook, or a data frame.

## Usage

```
px(input = NULL, data = NULL, validate = TRUE)
```

## Arguments

| | |
|---|---|
| `input` | Path to PX-file, path to an Excel metadata workbook, a data frame or path to an `.rds` or `.parquet` file with a data frame. If input is a data frame or NULL, a px object with minimal metadata is created. |
| `data` | Either a data frame or a path to an `.rds` or `.parquet` file with a data frame. This can only be used if `input` is an Excel metadata workbook. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

## Value

A px object

## Examples

```
# Create px object from dataset
x1 <- px(population_gl)

# Create px object from PX-file
px_path <- tempfile(fileext = ".px")
url <- "https://bank.stat.gl:443/sq/0cf06962-19f1-4d5c-8d43-b7ed0009617d"
download.file(url, px_path)

x2 <- px(px_path)
```

## px_add_totals *Add total levels to variables*

### Description

Adds a total level, which is the sum of the figures for all other levels of the variable. NA values are ignored when calculating the sum.

The default name of the total level is 'Total', unless px_elimination is set, in which case the elimination value becomes the name of the total level.

### Usage

```
px_add_totals(x, value, na.rm = TRUE, validate)

## S3 method for class 'px'
px_add_totals(x, value, na.rm = TRUE, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | A character vector of variables to add total levels to. |
| `na.rm` | Optional. Logical. If TRUE, NAs are removed before summing. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object

### See Also

px_elimination

### Examples

```
# Create small px object example
x0 <- px(subset(population_gl, age == "65+"))
x0$data

# Add total level to one variable
x1 <- px_add_totals(x0, "gender")
x1$data

# Add total level to multiple variables
x2 <- px_add_totals(x0, c("gender", "age"))
```

```
x2$data

# The name of the total level can be changed with px_elimination()
x3 <-
  x0 |>
  px_elimination("T") |>
  px_add_totals("gender")

x3$data
```

---

`px_aggregallowed` *AGGREGALLOWED*

---

## Description

Inspect or change AGGREGALLOWED.

## Usage

```
px_aggregallowed(x, value, validate)

## S3 method for class 'px'
px_aggregallowed(x, value, validate = TRUE)
```

## Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string. If missing, the current AGGREGALLOWED is returned. If NULL, AGGREGALLOWED is removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

## Value

A px object or a character string.

## See Also

Statistics Sweden's documentation

### Examples

```
# Set AGGREGALLOWED
x1 <-
   px(population_gl) |>
   px_aggregallowed('NO')

# Print AGGREGALLOWED
px_aggregallowed(x1)

# Remove AGGREGALLOWED
x2 <- px_aggregallowed(x1, NULL)
px_aggregallowed(x2)
```

## px_autopen *AUTOPEN*

### Description

Inspect or change AUTOPEN.

### Usage

```
px_autopen(x, value, validate)

## S3 method for class 'px'
px_autopen(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string. If missing, the current AUTOPEN is returned. If NULL, AUTOPEN is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set AUTOPEN
x1 <-
   px(population_gl) |>
   px_autopen('YES')

# Print AUTOPEN
px_autopen(x1)

# Remove AUTOPEN
x2 <- px_autopen(x1, NULL)
px_autopen(x2)
```

## `px_axis_version` *AXIS-VERSION*

### Description

Inspect or change AXIS-VERSION.

### Usage

```
px_axis_version(x, value, validate)

## S3 method for class 'px'
px_axis_version(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string. If missing, the current AXIS-VERSION is returned. If NULL, AXIS-VERSION is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

## Examples

```
# Set AXIS-VERSION
x1 <-
   px(population_gl) |>
   px_axis_version('2010')

# Print AXIS-VERSION
px_axis_version(x1)

# Remove AXIS-VERSION
x2 <- px_axis_version(x1, NULL)
px_axis_version(x2)
```

# `px_baseperiod` *BASEPERIOD*

## Description

Inspect or change BASEPERIOD.

## Usage

```
px_baseperiod(x, value, validate)

## S3 method for class 'px'
px_baseperiod(x, value, validate = TRUE)
```

## Arguments

`x` A px object

`value` Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current BASEPERIOD is returned. If NULL, BASEPERIOD is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

## Value

A px object, a character string, or a data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set BASEPERIOD for all languages
x1 <-
  px(population_gl) |>
  px_baseperiod('year')

# Print BASEPERIOD
px_baseperiod(x1)

# Set BASEPERIOD for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_baseperiod(tribble(~language, ~value,
                        'en', 'year',
                        'kl', 'ukioq'))
px_baseperiod(x2)

# Remove BASEPERIOD
x3 <- px_baseperiod(x2, NULL)
px_baseperiod(x3)
```

---

# px_cellnote *CELLNOTE*

---

## Description

Inspect or change CELLNOTE.

## Usage

```
px_cellnote(x, value, na_to_star, validate)

## S3 method for class 'px'
px_cellnote(x, value, na_to_star = TRUE, validate = TRUE)
```

## Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A data frame with columns 'cellnote' and one or more columns with the names of the STUB and HEADING variables. The 'cellnote' column is the cellnote text, and the STUB/HEADING columns control which cells the note applies to. Use star (*) if a note applies to all cells in a variable. Use column 'language' to set CELLNOTE for specific languages. If 'value' is missing, the current CELLNOTE is returned. If value is NULL, CELLNOTE is removed. |
| na_to_star | Optional. Convert all NAs to '*'. |

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

## Value

A px object or data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set CELLNOTE for a value
library(tibble)
x1 <-
  population_gl |>
  px() |>
  px_cellnote(
    tribble(~gender,  ~age,  ~year, ~cellnote,
             'male', '0-6', '2004', 'Approximation'))

x2 <-
  x1 |>
  px_cellnote(
    tribble(~gender,   ~age,  ~year, ~cellnote,
           'female',    '*', '2014', 'Uncertainty in ages'))

# Print CELLNOTE
px_cellnote(x2)

# Set CELLNOTE in multiple languagese
x3 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_cellnote(
    tribble(~age, ~year, ~language, ~cellnote,
             '*', '2003',  'en', 'Some of the figures are from 2003',
             '*', '2003', 'kl', 'Kisitsisit ilaat 2003-imeersuupput'))
px_cellnote(x3)

# Remove CELLNOTE
x4 <- px_cellnote(x3, NULL)
px_cellnote(x4)
```

# px_cellnotex *CELLNOTEX*

## Description

Inspect or change CELLNOTEX.

## Usage

```
px_cellnotex(x, value, na_to_star, validate)

## S3 method for class 'px'
px_cellnotex(x, value, na_to_star = TRUE, validate = TRUE)
```

## Arguments

`x` A px object

`value` Optional. A data frame with columns 'cellnotex' and one or more columns with the names of the STUB and HEADING variables. The 'cellnotex' column is the cellnotex text, and the STUB/HEADING columns control which cells the note applies to. Use star (*) if a note applies to all cells in a variable. Use column 'language' to set CELLNOTEX for specific languages. If 'value' is missing, the current CELLNOTEX is returned. If value is NULL, CELLNOTEX is removed.

`na_to_star` Optional. Convert all NAs to '*'.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

## Value

A px object or data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set CELLNOTEX for a value
library(tibble)
x1 <-
  population_gl |>
  px() |>
  px_cellnotex(
    tribble(~gender,  ~age,  ~year, ~cellnote,
             'male', '0-6', '2004', 'Approximation'))
```

```
x2 <-
  x1 |>
  px_cellnote(
    tribble(~gender,   ~age,  ~year, ~cellnote,
           'female',    '*', '2014', 'Uncertainty in ages'))

# Print CELLNOTEX
px_cellnotex(x2)

# Set CELLNOTEX in multiple languagese
x3 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_cellnotex(
    tribble(~age, ~year, ~language, ~cellnote,
             '*', '2003',  'en', 'Some of the figures are from 2003',
             '*', '2003', 'kl', 'Kisitsisit ilaat 2003-imeersuupput'))
px_cellnotex(x3)

# Remove CELLNOTEX
x4 <- px_cellnotex(x3, NULL)
px_cellnotex(x4)
```

## px_cfprices *CFPRICES*

### Description

Inspect or change CFPRICES.

### Usage

```
px_cfprices(x, value, validate)

## S3 method for class 'px'
px_cfprices(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current CFPRICES is returned. If NULL, CFPRICES is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set CFPRICES for all languages
x1 <-
  px(population_gl) |>
  px_cfprices('C')

# Print CFPRICES
px_cfprices(x1)

# Set CFPRICES for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_cfprices(tribble(~language, ~value,
                      'en', 'C',
                      'kl', 'F'))
px_cfprices(x2)

# Remove CFPRICES
x3 <- px_cfprices(x2, NULL)
px_cfprices(x3)
```

## px_charset *CHARSET*

### Description

Inspect or change CHARSET.

### Usage

```
px_charset(x, value, validate)

## S3 method for class 'px'
px_charset(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string. If missing, the current CHARSET is returned. If NULL, CHARSET is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set CHARSET
x1 <-
   px(population_gl) |>
   px_charset('ANSI')

# Print CHARSET
px_charset(x1)

# Remove CHARSET
x2 <- px_charset(x1, NULL)
px_charset(x2)
```

## `px_classification` *Create a classification object*

### Description

Create a classification object from a data frame or .vs and .agg files.

### Usage

```
px_classification(name, prestext, domain, df, vs_path, agg_paths)
```

## Arguments

name Optional. Name of the classification.

prestext Optional. Presentation text.

domain Optional. Character vector with domain names. Used to link to PX-file.

df Optional. A data frame with required column 'valuecode' and optional column 'valuetext', if the codes have texts. Each additional column represents an aggregation. The column name is the name of the aggregation.

vs_path Optional. Path to a values set (.vs) file.

agg_paths Optional.

- If NULL, aggregation paths are automatically taken from the [Aggreg] field in the .vs file.
- Use a vector of paths to one or more aggregation files (.agg) to manually choose aggregations.
- Use character(0) if aggregations from the .vs files should not be added automatically.

## Details

A classification is a combination of a value set and zero, one, or more aggregations. The classification can be saved as .vs and .agg files (see `px_save_classification()`).

If a classification is created from a data frame, the arguments `name` and `prestext` and `domain` are required. If a classification is created from .vs and .agg files, all other arguments should be empty.

Only type 'V' value sets are supported. Type 'H' and 'N' value sets are not supported.

## Value

A classification object

## Examples

```
# Create classification from data frame
library(tibble)

c1 <- px_classification(name = "Age5",
                        prestext = "Ages 0-9 - 60+",
                        domain = "age",
                        df = tribble(
                           ~valuecode,    ~valuetext,   ~`25 years classes`,
                                "0-4",     "0-4 years",              "0-24",
                                "5-9",     "5-9 years",              "0-24",
                              "10-14",   "10-14 years",              "0-24",
                              "15-19",   "15-19 years",              "0-24",
                              "20-24",   "20-24 years",              "0-24",
                              "25-29",   "25-29 years",             "25-49",
                              "30-34",   "30-34 years",             "25-49",
                              "35-39",   "35-39 years",             "25-49",
                              "40-44",   "40-44 years",             "25-49",
                              "45-49",   "45-49 years",             "25-49",
                              "50-54",   "50-54 years",             "50-74",
                              "55-59",   "55-59 years",             "50-74",
                              "60-64",   "60-64 years",             "50-74",
                              "65-69",   "65-69 years",             "50-74",
                              "70-74",   "70-74 years",             "50-74",
                                "75+",     "75+ years",               "75+"
                                )
                        )

## Not run:
# Create classification from .vs file and use aggregations mentioned in .vs
c2 <- px_classification(vs_path = "path/to/value_set.vs")

# Create classification from .vs file and manually specify aggregation files
c3 <- px_classification(vs_path = "path/to/value_set.vs",
                        agg_paths = c("path/to/aggregation1.agg",
                                      "path/to/aggregation2.agg"
                                      )
                       )



## End(Not run)
```

## px_codepage *CODEPAGE*

### Description

Inspect or change CODEPAGE.

### Usage

```
px_codepage(x, value, validate)

## S3 method for class 'px'
px_codepage(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string. If missing, the current CODEPAGE is returned. If NULL, CODEPAGE is removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Details

CODEPAGE controls which encoding PX-files are read and stored in. Use `iconvlist()` to see available encodings on your system.

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

`px_save()`

### Examples

```
# Set CODEPAGE
x1 <-
   px(population_gl) |>
   px_codepage('utf-8')

# Print CODEPAGE
px_codepage(x1)

# Remove CODEPAGE
x2 <- px_codepage(x1, NULL)
px_codepage(x2)
```

## `px_confidential` *CONFIDENTIAL*

### Description

Inspect or change CONFIDENTIAL.

### Usage

```
px_confidential(x, value, validate)

## S3 method for class 'px'
px_confidential(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string. If missing, the current CONFIDENTIAL is returned. If NULL, CONFIDENTIAL is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

## Value

A px object or a character string.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set CONFIDENTIAL
x1 <-
   px(population_gl) |>
   px_confidential('1')

# Print CONFIDENTIAL
px_confidential(x1)

# Remove CONFIDENTIAL
x2 <- px_confidential(x1, NULL)
px_confidential(x2)
```

---

# px_contact *CONTACT*

---

## Description

Inspect or change CONTACT.

## Usage

```
px_contact(x, value, validate)

## S3 method for class 'px'
px_contact(x, value, validate = TRUE)
```

## Arguments

`x` A px object

`value` Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current CONTACT is returned. If NULL, CONTACT is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

## Value

A px object, a character string, or a data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set CONTACT for all languages
x1 <-
  px(population_gl) |>
  px_contact('Johan Ejstrud')

# Print CONTACT
px_contact(x1)

# Set CONTACT for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_contact(tribble(~language, ~value,
                      'en', 'Johan Ejstrud',
                      'kl', 'Lars Pedersen'))
px_contact(x2)

# Remove CONTACT
x3 <- px_contact(x2, NULL)
px_contact(x3)
```

---

# `px_contents` *CONTENTS*

---

## Description

Inspect or change CONTENTS.

## Usage

```
px_contents(x, value, validate)

## S3 method for class 'px'
px_contents(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string to set the value for all languages or a data frame with columns ’language’ and ’value’ to set it for specific languages. If ’value’ is missing, the current CONTENTS is returned. If NULL, an error is thrown because CONTENTS cannot be removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden’s documentation

### Examples

```
# Set CONTENTS for all languages
x1 <-
  px(population_gl) |>
  px_contents('Population')

# Print CONTENTS
px_contents(x1)

# Set CONTENTS for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_contents(tribble(~language, ~value,
                      'en', 'Population',
                      'kl', 'Innuttaasut'))
px_contents(x2)
```

## `px_contvariable` *CONTVARIABLE*

### Description

Inspect or change CONTVARIABLE.

Setting CONTVARIABLE indexes several variables in table2. Removing CONTVARIABLE removes the indexing from table2.

### Usage

```
px_contvariable(x, value, validate)

## S3 method for class 'px'
px_contvariable(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string. If missing, the current CONTVARIABLE is returned. If NULL, CONTVARIABLE is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set CONTVARIABLE
x1 <-
  px(population_gl) |>
  px_contvariable('gender')

# After setting CONTVARIABLE some variables are index by it, e.g. UNITS
px_units(x1)

# Remove CONTVARIABLE
x2 <- px_contvariable(x1, NULL)
px_contvariable(x2)

# Removing CONTVARIABLE also removes the index from UNITS
px_units(x2)
```

## `px_copyright` *COPYRIGHT*

### Description

Inspect or change COPYRIGHT.

### Usage

```
px_copyright(x, value, validate)

## S3 method for class 'px'
px_copyright(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string. If missing, the current COPYRIGHT is returned. If NULL, COPYRIGHT is removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set COPYRIGHT
x1 <-
   px(population_gl) |>
   px_copyright('YES')

# Print COPYRIGHT
px_copyright(x1)

# Remove COPYRIGHT
x2 <- px_copyright(x1, NULL)
px_copyright(x2)
```

## `px_creation_date` *CREATION-DATE*

### Description

Inspect or change CREATION-DATE.

### Usage

```
px_creation_date(x, value, validate)

## S3 method for class 'px'
px_creation_date(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string. If missing, the current CREATION-DATE is returned. If NULL, CREATION-DATE is removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set CREATION-DATE
x1 <-
   px(population_gl) |>
   px_creation_date('19960612 14:20')

# Print CREATION-DATE
px_creation_date(x1)

# Remove CREATION-DATE
x2 <- px_creation_date(x1, NULL)
px_creation_date(x2)
```

## `px_data` *DATA*

### Description

Inspect or change DATA.

## Usage

```
px_data(x, value, validate)

## S3 method for class 'px'
px_data(x, value, validate = TRUE)
```

## Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A data frame. If missing, the current DATA is returned. If NULL, all data rows are removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

## Details

It is not recommended to change the data table of a px object with this function, since it does update any metadata.

## Value

A px object or data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
x1 <- px(population_gl)

# Print data table
px_data(x1)

# Change data table (risky business)
population_gl_2024 <- subset(population_gl, year == 2024)

x2 <- px_data(x1, population_gl_2024)
```

## px_decimals *DECIMALS*

### Description

Inspect or change DECIMALS.

### Usage

```
px_decimals(x, value, validate)

## S3 method for class 'px'
px_decimals(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string. If missing, the current DECIMALS is returned. If NULL, an error is thrown because DECIMALS cannot be removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set DECIMALS
x1 <-
   px(population_gl) |>
   px_decimals('3')

# Print DECIMALS
px_decimals(x1)
```

## px_description DESCRIPTION

### Description

Inspect or change DESCRIPTION.

### Usage

```
px_description(x, value, validate)

## S3 method for class 'px'
px_description(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current DESCRIPTION is returned. If NULL, DESCRIPTION is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use px_validate() to manually preform the check. |

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden's documentation

px_title()

### Examples

```
# Set DESCRIPTION for all languages
x1 <-
  px(population_gl) |>
  px_description('Population')

# Print DESCRIPTION
px_description(x1)

# Set DESCRIPTION for individual languages
library(tibble)
```

```
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_description(tribble(~language, ~value,
                     'en', 'Population',
                     'kl', 'Innuttaasut'))
px_description(x2)

# Remove DESCRIPTION
x3 <- px_description(x2, NULL)
px_description(x3)
```

## px_descriptiondefault *DESCRIPTIONDEFAULT*

### Description

Inspect or change DESCRIPTIONDEFAULT.

### Usage

```
px_descriptiondefault(x, value, validate)

## S3 method for class 'px'
px_descriptiondefault(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string. If missing, the current DESCRIPTIONDEFAULT is returned. If NULL, DESCRIPTIONDEFAULT is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use px_validate() to manually preform the check. |

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set DESCRIPTIONDEFAULT
x1 <-
   px(population_gl) |>
   px_descriptiondefault('YES')

# Print DESCRIPTIONDEFAULT
px_descriptiondefault(x1)

# Remove DESCRIPTIONDEFAULT
x2 <- px_descriptiondefault(x1, NULL)
px_descriptiondefault(x2)
```

## px_domain *DOMAIN*

### Description

Inspect or change DOMAIN.

### Usage

```
px_domain(x, value, validate)

## S3 method for class 'px'
px_domain(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string or data frame.

- Use character to set DOMAIN for all languages and STUB/HEADING variables.
- Use data frame with columns 'variable-code', 'language' and 'domain' to set DOMAIN for specific variables.
- If missing, the current DOMAIN is returned.
- If NULL, DOMAIN is removed for all variables.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden’s documentation

### Examples

```
# Set DOMAIN for all languages
x1 <-
  px(population_gl) |>
  px_domain('aggregation1')

# Print DOMAIN
px_domain(x1)

# Set DOMAIN for individual variables
library(tibble)
x2 <-
  x1 |>
  px_domain(tribble(~`variable-code`, ~domain,
                    'gender', 'aggregation2',
                    'age',    'aggregation3'))
px_domain(x2)

# Set DOMAIN for individual languages
x3 <-
  x2 %>%
  px_languages(c('en', 'kl')) |>
  px_domain(tribble(~`variable-code`, ~language, ~domain,
                    'gender',    'en',      'aggregation2_en',
                    'gender',    'kl',      'aggregation2_kl',
                    'age',       'en',      'aggregation3_en'))
px_domain(x3)

# Remove DOMAIN
x4 <- px_domain(x3, NULL)
px_domain(x4)
```

## px_elimination *ELIMINATION*

### Description

Inspect or change ELIMINATION.

### Usage

```
px_elimination(x, value, validate)

## S3 method for class 'px'
px_elimination(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string or data frame.

- Use character to set ELIMINATION for all languages and STUB/HEADING variables.
- Use data frame with columns 'variable-code', 'language' and 'elimination' to set ELIMINATION for specific variables.
- If missing, the current ELIMINATION is returned.
- If NULL, ELIMINATION is removed for all variables.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set ELIMINATION for all languages
x1 <-
  px(population_gl) |>
  px_elimination('YES')

# Print ELIMINATION
px_elimination(x1)

# Set ELIMINATION for individual variables
library(tibble)
x2 <-
  x1 |>
  px_elimination(tribble(~`variable-code`, ~elimination,
                    'gender', 'All',
                    'age',    'Total'))
px_elimination(x2)

# Set ELIMINATION for individual languages
x3 <-
  x2 %>%
  px_languages(c('en', 'kl')) |>
  px_elimination(tribble(~`variable-code`, ~language, ~elimination,
                    'gender',    'en',       'All_en',
                    'gender',    'kl',       'All_kl',
                    'age',       'en',       'Total_en'))
```

```
px_elimination(x3)

# Remove ELIMINATION
x4 <- px_elimination(x3, NULL)
px_elimination(x4)
```

## px_figures *Change figures variable*

### Description

Inspect or change which variable is used as figures. The previous figures variable is changed to STUB. There can only be one figures variable.

### Usage

```
px_figures(x, value, validate)

## S3 method for class 'px'
px_figures(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. Name of variable to use as FIGRUES. If missing, the current PX_FIGURES variable is returned. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use px_validate() to manually preform the check. |

### Value

A px object or a character string

### See Also

px_stub px_heading

### Examples

```
x1 <- px(population_gl)

# Print FIGURES
px_figures(x1)

# Change 'age' to FIGURES variable, 'n' i changed to STUB
x2 <- px_figures(x1, 'age')
px_figures(x2)
px_stub(x2)
```

## px_heading HEADING

### Description

Inspect or change HEADING.

### Usage

```
px_heading(x, value, validate)

## S3 method for class 'px'
px_heading(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character vector of variable names to change to STUB. This also changes the HEADING order. With names in variables becoming 1, 2, ... If missing, the current HEADING variables are returned. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use px_validate() to manually preform the check. |

### Value

A px object or a character vector.

### See Also

Statistics Sweden's documentation

px_stub px_figures

### Examples

```
x1 <- px(population_gl)

# Print HEADING
px_heading(x1)

# Add 'gender' to HEADING
x2 <- px_heading(x1, 'gender')
px_heading(x2)

# Change order of HEADING
x3 <- px_heading(x2, 'year')
px_heading(x3)
```

## px_infofile *INFOFILE*

### Description

Inspect or change INFOFILE.

### Usage

```
px_infofile(x, value, validate)

## S3 method for class 'px'
px_infofile(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current INFOFILE is returned. If NULL, INFOFILE is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use px_validate() to manually preform the check. |

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set INFOFILE for all languages
x1 <-
  px(population_gl) |>
  px_infofile('infofile_en')

# Print INFOFILE
px_infofile(x1)

# Set INFOFILE for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_infofile(tribble(~language, ~value,
                      'en', 'infofile_en',
                      'kl', 'infofile_kl'))
px_infofile(x2)

# Remove INFOFILE
x3 <- px_infofile(x2, NULL)
px_infofile(x3)
```

## px_keywords *px keywords*

### Description

Properties of all px keywords. Used internally by the package.

### Usage

```
px_keywords
```

### Format

A data frame:

**keyword** Name

**mandatory** Is required in a PX-file

**table_meta** Is metadata for entire table; not individual variables or cells

**language_dependent** Is language dependent

**indexed_by_contvariable** Is indexed if CONTVARIABLE is set

**quote_value** Value should be quoted in PX-file

**default_value** Default value for mandatory keywords

**documentation** URL to Statistic Sweden's documentation

**order** Recommended order

### Source

https://www.scb.se/globalassets/vara-tjanster/px-programmen/PX-file_format_specification_2013.pdf

## px_language *LANGUAGE*

### Description

Inspect or change LANGUAGE.

### Usage

```
px_language(x, value, validate)

## S3 method for class 'px'
px_language(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string. If missing, the current LANGUAGE is returned. If NULL, LANGUAGE is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Details

If LANGUAGES is defined, changing LANGUAGE will also add is to LANGUAGES.

### Value

A px object

### See Also

`px_languages`

### Examples

```
# Set LANGUAGE to 'en'
x1 <-
  population_gl |>
  px() |>
  px_language('en')

# Print LANGUAGE
px_language(x1)

# Remove LANGUAGE
```

```
x2 <- px_language(x1, NULL)
px_language(x2)
```

## px_languages *LANGUAGES*

### Description

Inspect or change LANGUAGES.

### Usage

```
px_languages(x, value, validate)

## S3 method for class 'px'
px_languages(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character vector. If missing, the current LANGUAGES are returned. If NULL, LANGUAGES are removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Details

If LANGUAGE is defined it should be one of the values in LANGUAGES.

If LANGUAGE is set, it is considered the main language. If LANGUAGE is not set, the first language in LANGUAGES is considered the main language.

### Value

A px object

### See Also

`px_language`

### Examples

```
# Set LANGUAGES to 'en' and 'kl', with 'en' as main language
x1 <-
  population_gl |>
  px() |>
  px_languages(c('en', 'kl'))

# Print LANGUAGES
px_languages(x1)

# Remove LANGUAGES
x2 <- px_languages(x1, NULL)
px_languages(x2)
```

## `px_last_updated` *LAST-UPDATED*

### Description

Inspect or change LAST-UPDATED.

### Usage

```
px_last_updated(x, value, validate)

## S3 method for class 'px'
px_last_updated(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current LAST-UPDATED is returned. If NULL, LAST-UPDATED is removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden's documentation

## Examples

```
# Set LAST-UPDATED for all languages
x1 <-
  px(population_gl) |>
  px_last_updated('17070501 15:55')

# Print LAST-UPDATED
px_last_updated(x1)

# Set LAST-UPDATED for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_last_updated(tribble(~language, ~value,
                      'en', '17070501 15:55',
                      'kl', '20080621 15:55'))
px_last_updated(x2)

# Remove LAST-UPDATED
x3 <- px_last_updated(x2, NULL)
px_last_updated(x3)
```

# px_link *LINK*

## Description

Inspect or change LINK.

## Usage

```
px_link(x, value, validate)

## S3 method for class 'px'
px_link(x, value, validate = TRUE)
```

## Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current LINK is returned. If NULL, LINK is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use px_validate() to manually preform the check. |

## Value

A px object, a character string, or a data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set LINK for all languages
x1 <-
  px(population_gl) |>
  px_link('https://stat.gl/?lang=en')

# Print LINK
px_link(x1)

# Set LINK for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_link(tribble(~language, ~value,
                      'en', 'https://stat.gl/?lang=en',
                      'kl', 'https://stat.gl/'))
px_link(x2)

# Remove LINK
x3 <- px_link(x2, NULL)
px_link(x3)
```

# px_map *MAP*

## Description

Inspect or change MAP.

## Usage

```
px_map(x, value, validate)

## S3 method for class 'px'
px_map(x, value, validate = TRUE)
```

## Arguments

x A px object

value Optional. A character string or data frame.

- Use character to set MAP for all languages and STUB/HEADING variables.
- Use data frame with columns 'variable-code', 'language' and 'map' to set MAP for specific variables.
- If missing, the current MAP is returned.
- If NULL, MAP is removed for all variables.

validate Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

## Value

A px object, a character string, or a data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set MAP for all languages
x1 <-
  px(population_gl) |>
  px_map('greenland')

# Print MAP
px_map(x1)

# Set MAP for individual variables
library(tibble)
x2 <-
  x1 |>
  px_map(tribble(~`variable-code`, ~map,
                  'gender', 'cities',
                  'age',    'municipalities'))
px_map(x2)

# Set MAP for individual languages
x3 <-
  x2 %>%
  px_languages(c('en', 'kl')) |>
  px_map(tribble(~`variable-code`, ~language, ~map,
                  'gender',    'en',      'cities_en',
                  'gender',    'kl',      'cities_kl',
                  'age',       'en',      'municipalities_en'))
```

```
px_map(x3)

# Remove MAP
x4 <- px_map(x3, NULL)
px_map(x4)
```

## px_matrix *MATRIX*

### Description

Inspect or change MATRIX.

### Usage

```
px_matrix(x, value, validate)

## S3 method for class 'px'
px_matrix(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string. If missing, the current MATRIX is returned. If NULL, an error is thrown because MATRIX cannot be removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set MATRIX
x1 <-
   px(population_gl) |>
   px_matrix('POPULATION')

# Print MATRIX
px_matrix(x1)
```

## px_micro *Create micro PX-files*

### Description

Split one px object into many small PX-files (micro files), with count of the variables in it.

### Usage

```
px_micro(x, out_dir = NULL, keyword_values = NULL)
```

### Arguments

`x` A px object.

`out_dir` Directory to save PX-files in.

`keyword_values` Optional. A data frame with column 'variable' and one or more of: 'px_contents', 'px_title', 'px_description', and 'px_matrix'. The columns will be added as keywords to the table for each non-HEADING variable that match the 'variable' column. It probably work for other keywords as well.

Use the column 'filename' to control the filename of each micro file. The filename path is relative to 'out_dir'.

Use the column 'language' if the PX-file has multiple languages.

### Details

The HEADING variables are use in all the micro files, and a file is created for each non-HEADING variable. The new PX-files are saved in a directory specified by `out_dir`.

The main loop uses the furrr package for parallelisation. Use future::plan() to choose how to parallelise.

### Value

Nothing

### Examples

```
# Create px object with cohort as HEADING
x <-
  greenlanders |>
  px() |>
  px_stub(names(greenlanders)) |>
  px_heading("cohort")

# Create micro files, one for each of the non-HEADING variables (gender, age,
# municipality)
px_micro(x)
```

## px_next_update *NEXT-UPDATE*

### Description

Inspect or change NEXT-UPDATE.

### Usage

```
px_next_update(x, value, validate)

## S3 method for class 'px'
px_next_update(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string. If missing, the current NEXT-UPDATE is returned. If NULL, NEXT-UPDATE is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set NEXT-UPDATE
x1 <-
   px(population_gl) |>
   px_next_update('20240621 15:55')

# Print NEXT-UPDATE
px_next_update(x1)

# Remove NEXT-UPDATE
x2 <- px_next_update(x1, NULL)
px_next_update(x2)
```

px_note *NOTE*

## Description

Inspect or change NOTE.

## Usage

```
px_note(x, value, validate)

## S3 method for class 'px'
px_note(x, value, validate = TRUE)
```

## Arguments

x A px object

value Optional. A character string, a data frame, or a list.

- Use character, to set NOTE for the entire table across all languages.
- Use a data frame with columns 'language' and 'value' to set NOTE for the entire table in a specific language.
- Use a data frame with the columns 'variable-code' and 'note', to set NOTE for a specific variable across all languages. Add the column 'language' to set NOTE for specific language.
- Use a list of the above elements to set NOTE in muliple ways. This is the same as calling NOTE multiple times with different values.
- If missing, the current NOTE is returned.
- If NULL, NOTE is removed for the table and all variables.

validate Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

## Details

NOTE has a lot of possible ways to specify `value`, because it can be set both for the entire PX-file and for individual variables.

## Value

A px object, a character string, a data frame, or a list of character strings and/or data frames.

## See Also

Statistics Sweden's documentation

## Examples

```
library(tibble)

# Set NOTE for entire PX-file
x1 <-
  px(population_gl) |>
  px_note('Note about PX-file')

# Print NOTE
px_note(x1)

# Set NOTE for entire PX-file in multiple languages
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_note(tribble(~language,            ~value,
                       'en',    'English note',
                       'kl', 'Kalaallisut note'
                 )
         )
px_note(x2)

# Set NOTE for variables
x3 <-
  x1 |>
  px_note(tribble(~`variable-code`, ~note,
                 'year', 'Some data collected in following year',
                 'age',  'Is rounded down'
                 )
        )
px_note(x3)

# Remove all NOTEs
x4 <- px_note(x3, NULL)
```

# px_notex *NOTEX*

## Description

Inspect or change NOTEX.

## Usage

```
px_notex(x, value, validate)

## S3 method for class 'px'
px_notex(x, value, validate = TRUE)
```

## Arguments

`x` A px object

`value` Optional. A character string, a data frame, or a list.

- Use character, to set NOTEX for the entire table across all languages.
- Use a data frame with columns 'language' and 'value' to set NOTEX for the entire table in a specific language.
- Use a data frame with the columns 'variable-code' and 'notex', to set NOTEX for a specific variable across all languages. Add the column 'language' to set NOTEX for specific language.
- Use a list of the above elements to set NOTEX in muliple ways. This is the same as calling NOTEX multiple times with different values.
- If missing, the current NOTEX is returned.
- If NULL, NOTEX is removed for the table and all variables.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

## Details

NOTEX has a lot of possible ways to specify `value`, because it can be set both for the entire PX-file and for individual variables.

## Value

A px object, a character string, a data frame, or a list of character strings and/or data frames.

## See Also

Statistics Sweden's documentation

## Examples

```
library(tibble)

# Set NOTEX for entire PX-file
x1 <-
  px(population_gl) |>
  px_notex('Note about PX-file')

# Print NOTEX
px_notex(x1)

# Set NOTEX for entire PX-file in multiple languages
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_notex(tribble(~language,            ~value,
                       'en',    'English notex',
                       'kl', 'Kalaallisut notex'
                 )
         )
px_notex(x2)

# Set NOTEX for variables
x3 <-
  x1 |>
  px_notex(tribble(~`variable-code`, ~notex,
                 'year', 'Some data collected in following year',
                 'age',  'Is rounded down'
                 )
        )
px_notex(x3)

# Remove all NOTEXs
x4 <- px_notex(x3, NULL)
```

## px_order *Change value order*

### Description

Inspect or change ORDER.

### Usage

```
px_order(x, value, validate)

## S3 method for class 'px'
px_order(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A data frame with the columns 'order' and one or more of the columns: 'variable-code', and 'code'. If 'value' is missing, the current ORDER is returned. If NULL, ORDER is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or data frame.

### Examples

```
# Set ORDER for a variable
library(tibble)
x1 <-
  population_gl |>
  px() |>
  px_order(tribble(~`variable-code`, ~order,
                   'gender', 8))

# Print ORDER
px_order(x1)

# Set ORDER for a value
x2 <-
  x1 |>
  px_order(tribble(~`variable-code`, ~code, ~order,
                   'age', '2004', 9))
px_order(x2)

# Remove ORDER
x3 <- px_order(x2, NULL)
px_order(x3)
```

## px_precision *PRECISION*

### Description

Inspect or change PRECISION.

### Usage

```
px_precision(x, value, validate)

## S3 method for class 'px'
px_precision(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A data frame with the columns 'precision' and one or more of the columns: 'variable-code', and 'code'. If 'value' is missing, the current PRECISION is returned. If NULL, PRECISION is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

## Value

A px object or data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set PRECISION for a variable
library(tibble)
x1 <-
  population_gl |>
  px() |>
  px_precision(tribble(~`variable-code`, ~precision,
                       'gender', 2))

# Print PRECISION
px_precision(x1)

# Set PRECISION for a value
x2 <-
  x1 |>
  px_precision(tribble(~`variable-code`, ~code, ~precision,
                       'age', '2004', 3))
px_precision(x2)

# Remove PRECISION
x3 <- px_precision(x2, NULL)
px_precision(x3)
```

# px_save *Save px object to file*

## Description

Save px object to file

## Usage

```
px_save(x, path, save_data = TRUE, data_path = NULL)
```

## Arguments

| | |
|---|---|
| x | A px object. |
| path | Path to file. The file extension determines the format. Can be:<br>• .px to save as a PX-file<br>• .xlsx to save as an Excel workbook |

`save_data` If FALSE, no 'Data' sheet is created in the Excel workbook. Can only be used if `path` is an `.xlsx` file.

`data_path` Path to an `.rds` or `.parquet` file to save data table at. This is usefull when saving an Excel workbook where the data has more rows than Excel can handle. Can only be used if `path` is an `.xlsx` file, and `save_data` is `TRUE`.

### Details

Use `px_codepage()` to change file encoding.

### Value

Nothing

### See Also

`px_codepage()`

### Examples

```
# Save px object to PX-file
tmp_dir <- tempdir()

x <- px(population_gl)

px_save(x, file.path(tmp_dir, "population.px"))

# Save px object to Excel workbook
px_save(x, file.path(tmp_dir, "population.xlsx"))
```

## px_save_classification

*Save classification as .vs and .agg files*

### Description

Save a classification object as .vs and .agg files. The .vs file contains the value set and the .agg files contain the aggregations.

### Usage

```
px_save_classification(c, path)
```

### Arguments

`c` A classification object

`path` Directory to save the files in

### Value

Nothing.

### Examples

```
# Save classification as .vs as .agg files
c <- px_classification(name = "Age5",
                       prestext = "Ages 0-9 - 60+",
                       domain = "age",
                       df = age_classification
                       )

px_save_classification(c, path = tempdir())
```

---

## px_showdecimals *SHOWDECIMALS*

---

### Description

Inspect or change SHOWDECIMALS.

### Usage

```
px_showdecimals(x, value, validate)

## S3 method for class 'px'
px_showdecimals(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string. If missing, the current SHOWDECIMALS is returned. If NULL, SHOWDECIMALS is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

## Examples

```
# Set SHOWDECIMALS
x1 <-
   px(population_gl) |>
   px_showdecimals('2')

# Print SHOWDECIMALS
px_showdecimals(x1)

# Remove SHOWDECIMALS
x2 <- px_showdecimals(x1, NULL)
px_showdecimals(x2)
```

---

# px_source *SOURCE*

---

## Description

Inspect or change SOURCE.

## Usage

```
px_source(x, value, validate)

## S3 method for class 'px'
px_source(x, value, validate = TRUE)
```

## Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current SOURCE is returned. If NULL, SOURCE is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

## Value

A px object, a character string, or a data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set SOURCE for all languages
x1 <-
  px(population_gl) |>
  px_source('Statistics Greenland')

# Print SOURCE
px_source(x1)

# Set SOURCE for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_source(tribble(~language, ~value,
                    'en', 'Statistics Greenland',
                    'kl', 'Naatsorsueqqissaartarfik'))
px_source(x2)

# Remove SOURCE
x3 <- px_source(x2, NULL)
px_source(x3)
```

---

# px_stockfa *STOCKFA*

---

## Description

Inspect or change STOCKFA.

## Usage

```
px_stockfa(x, value, validate)

## S3 method for class 'px'
px_stockfa(x, value, validate = TRUE)
```

## Arguments

`x` A px object

`value` Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current STOCKFA is returned. If NULL, STOCKFA is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set STOCKFA for all languages
x1 <-
  px(population_gl) |>
  px_stockfa('S')

# Print STOCKFA
px_stockfa(x1)

# Set STOCKFA for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_stockfa(tribble(~language, ~value,
                     'en', 'S',
                     'kl', 'F'))
px_stockfa(x2)

# Remove STOCKFA
x3 <- px_stockfa(x2, NULL)
px_stockfa(x3)
```

---

## px_stub *STUB*

---

### Description

Inspect or change STUB.

### Usage

```
px_stub(x, value, validate)

## S3 method for class 'px'
px_stub(x, value, validate = TRUE)
```

## Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character vector of variable names to change to STUB. This also changes the STUB order. With names in `variables` becoming 1, 2, ... If missing, the current STUB variables are returned. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

## Value

A px object or a character vector.

## See Also

Statistics Sweden's documentation

`px_heading` `px_figures`

## Examples

```
x1 <- px(population_gl)
# Print STUB
px_stub(x1)
# Add 'year' to STUB
x2 <- px_stub(x1, 'year')
px_stub(x2)

# Change order of STUB
x3 <- px_stub(x2, c('age', 'gender'))
px_stub(x3)
```

# `px_subject_area` *SUBJECT-AREA*

## Description

Inspect or change SUBJECT-AREA.

## Usage

```
px_subject_area(x, value, validate)

## S3 method for class 'px'
px_subject_area(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current SUBJECT-AREA is returned. If NULL, an error is thrown because SUBJECT-AREA cannot be removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set SUBJECT-AREA for all languages
x1 <-
  px(population_gl) |>
  px_subject_area('Population')

# Print SUBJECT-AREA
px_subject_area(x1)

# Set SUBJECT-AREA for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_subject_area(tribble(~language, ~value,
                          'en', 'Population',
                          'kl', 'Innuttaasut'))
px_subject_area(x2)
```

## px_subject_code *SUBJECT-CODE*

### Description

Inspect or change SUBJECT-CODE.

### Usage

```
px_subject_code(x, value, validate)

## S3 method for class 'px'
px_subject_code(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string. If missing, the current SUBJECT-CODE is returned. If NULL, an error is thrown because SUBJECT-CODE cannot be removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set SUBJECT-CODE
x1 <-
   px(population_gl) |>
   px_subject_code('POP')

# Print SUBJECT-CODE
px_subject_code(x1)
```

## `px_tableid` *TABLEID*

### Description

Inspect or change TABLEID.

### Usage

```
px_tableid(x, value, validate)

## S3 method for class 'px'
px_tableid(x, value, validate = TRUE)
```

### Arguments

`x` A px object

`value` Optional. A character string. If missing, the current TABLEID is returned. If NULL, TABLEID is removed.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set TABLEID
x1 <-
   px(population_gl) |>
   px_tableid('POPGL')

# Print TABLEID
px_tableid(x1)

# Remove TABLEID
x2 <- px_tableid(x1, NULL)
px_tableid(x2)
```

---

## `px_timeval` *TIMEVAL*

---

### Description

Inspect or change TIMEVAL.

There can only be one time variable.

### Usage

```
px_timeval(x, value, validate)

## S3 method for class 'px'
px_timeval(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string. If missing, the current TIMEVAL is returned. If NULL, TIMEVAL is removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set TIMEVAL
x1 <-
   px(population_gl) |>
   px_timeval('year')

# Print TIMEVAL
px_timeval(x1)

# Remove TIMEVAL
x2 <- px_timeval(x1, NULL)
px_timeval(x2)
```

## `px_title` *TITLE*

### Description

Inspect or change TITLE.

TITLE can only be removed if DESCRIPTION is set.

### Usage

```
px_title(x, value, validate)

## S3 method for class 'px'
px_title(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current TITLE is returned. If NULL, TITLE is removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object, a character string, or a data frame.

### See Also

Statistics Sweden's documentation

`px_description()`

### Examples

```
# Set TITLE for all languages
x1 <-
  px(population_gl) |>
  px_title('Population GR')

# Print TITLE
px_title(x1)

# Set TITLE for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_title(tribble(~language, ~value,
                    'en', 'Population GR',
                    'kl', 'Innuttaasut KL'))
px_title(x2)
```

---

## `px_units` *UNITS*

---

### Description

Inspect or change UNITS.

## Usage

```
px_units(x, value, validate)

## S3 method for class 'px'
px_units(x, value, validate = TRUE)
```

## Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | Optional. A character string to set the value for all languages or a data frame with columns 'language' and 'value' to set it for specific languages. If 'value' is missing, the current UNITS is returned. If NULL, an error is thrown because UNITS cannot be removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

## Value

A px object, a character string, or a data frame.

## See Also

Statistics Sweden's documentation

## Examples

```
# Set UNITS for all languages
x1 <-
  px(population_gl) |>
  px_units('persons')

# Print UNITS
px_units(x1)

# Set UNITS for individual languages
library(tibble)
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_units(tribble(~language, ~value,
                    'en', 'persons',
                    'kl', 'inuit amerlassusaat'))
px_units(x2)
```

## px_update_frequency *UPDATE-FREQUENCY*

### Description

Inspect or change UPDATE-FREQUENCY.

### Usage

```
px_update_frequency(x, value, validate)

## S3 method for class 'px'
px_update_frequency(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string. If missing, the current UPDATE-FREQUENCY is returned. If NULL, UPDATE-FREQUENCY is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or a character string.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set UPDATE-FREQUENCY
x1 <-
   px(population_gl) |>
   px_update_frequency('Yearly')

# Print UPDATE-FREQUENCY
px_update_frequency(x1)

# Remove UPDATE-FREQUENCY
x2 <- px_update_frequency(x1, NULL)
px_update_frequency(x2)
```

## px_validate *Check px object*

### Description

Runs a number of checks on px object to see if it is valid.

### Usage

```
px_validate(x)
```

### Arguments

| | |
|---|---|
| x | A supposed px object. |

### Details

This check is run by default by all `px_*` functions, but can be skipped by using `validate = FALSE`. This can be useful on large px objects where the checks are time consuming. Instead of validating on every modifying function `px_validate()` can be run as the final step to validate the object.

### Value

A valid px object.

### Examples

```
# Turn off validation for modifying functions, and manually
# run validation as final step in creating px object.
x1 <-
  px(population_gl, validate = FALSE) |>
  px_title("Test", validate = FALSE) |>
  px_validate()
```

## px_valuenote *VALUENOTE*

### Description

Inspect or change VALUENOTE.

### Usage

```
px_valuenote(x, value, validate)

## S3 method for class 'px'
px_valuenote(x, value, validate = TRUE)
```

### Arguments

x — A px object

value — Optional. A data frame with the columns 'valuenote' and one or more of the columns: 'variable-code', 'code', and 'language'. If 'value' is missing, the current VALUENOTE is returned. If NULL, VALUENOTE is removed.

validate — Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use px_validate() to manually preform the check.

### Value

A px object or data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set VALUENOTE for a value
library(tibble)
x1 <-
  population_gl |>
  px() |>
  px_valuenote(
    tribble(~`variable-code`, ~code,  ~valuenote,
            'year', '2004', 'Counts are approximated'))

# Print VALUENOTE
px_valuenote(x1)

# Set VALUENOTE for a value in specific language
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_valuenote(
    tribble(~`variable-code`, ~code,  ~language, ~valuenote,
            'age', '0-6', 'en', 'Some of the figures are from 2003',
            'age', '0-6', 'kl', 'Kisitsisit ilaat 2003-imeersuupput'))
px_valuenote(x2)

# Remove VALUENOTE
x3 <- px_valuenote(x2, NULL)
px_valuenote(x3)
```

## px_valuenotex *VALUENOTEX*

### Description

Inspect or change VALUENOTEX.

### Usage

```
px_valuenotex(x, value, validate)

## S3 method for class 'px'
px_valuenotex(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A data frame with the columns 'valuenotex' and one or more of the columns: 'variable-code', 'code', and 'language'. If 'value' is missing, the current VALUENOTEX is returned. If NULL, VALUENOTEX is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set VALUENOTEX for a value
library(tibble)
x1 <-
  population_gl |>
  px() |>
  px_valuenotex(
    tribble(~`variable-code`, ~code,  ~valuenotex,
            'year', '2004', 'Counts are approximated'))

# Print VALUENOTEX
px_valuenotex(x1)

# Set VALUENOTEX for a value in specific language
```

```
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_valuenotex(
    tribble(~`variable-code`, ~code,  ~language, ~valuenotex,
            'age', '0-6', 'en', 'Some of the figures are from 2003',
            'age', '0-6', 'kl', 'Kisitsisit ilaat 2003-imeersuupput'))
px_valuenotex(x2)

# Remove VALUENOTEX
x3 <- px_valuenotex(x2, NULL)
px_valuenotex(x3)
```

## px_values *VALUES*

### Description

Inspect or change VALUES.

### Usage

```
px_values(x, value, validate)

## S3 method for class 'px'
px_values(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A data frame with the columns 'values' and one or more of the columns: 'variable-code', 'code', and 'language'. If 'value' is missing, the current VALUES is returned. If NULL, VALUES is removed. |
| validate | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use px_validate() to manually preform the check. |

### Value

A px object or data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
# Set VALUES for a value
library(tibble)
x1 <-
  population_gl |>
  px() |>
  px_values(
    tribble(~`variable-code`, ~code,  ~values,
            'year', '2004', 'Year 2024'))

# Print VALUES
px_values(x1)

# Set VALUES for a value in specific language
x2 <-
  x1 |>
  px_languages(c('en', 'kl')) |>
  px_values(
    tribble(~`variable-code`, ~code,  ~language, ~values,
            'age', '0-6', 'en', 'toddler',
            'age', '0-6', 'kl', 'meeraaqqap'))
px_values(x2)

# Remove VALUES
x3 <- px_values(x2, NULL)
px_values(x3)
```

## px_variable_label *Change VARIABLE-LABEL*

### Description

Inspect or change VARIABLE-LABEL.

The variable label is the name that is shown in the PX-file.

### Usage

```
px_variable_label(x, value, validate)

## S3 method for class 'px'
px_variable_label(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| x | A px object |
| value | Optional. A character string or data frame.<br>• Use character to set VARIABLE-LABEL for all languages and STUB/HEADING variables. |

- Use data frame with columns 'variable-code', 'language' and 'variable-label' to set VARIABLE-LABEL for specific variables.
- If missing, the current VARIABLE-LABEL is returned.
- If NULL, VARIABLE-LABEL is removed for all variables.

`validate` Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check.

### Value

A px object, a character string, or a data frame.

### Examples

```
# Set VARIABLE-LABEL for individual variables
library(tibble)
x1 <-
  px(population_gl) |>
  px_variable_label(tribble(~`variable-code`, ~`variable-label`,
                            'gender',         'Gender',
                            'age',            'Age'))
px_variable_label(x1)

# Set VARIABLE-LABEL for individual languages
x2 <-
  x1 %>%
  px_languages(c('en', 'kl')) |>
  px_variable_label(tribble(~`variable-code`, ~language, ~`variable-label`,
                            'gender',         'en',      'Gender',
                            'gender',         'kl',      'Suiaassuseq',
                            'age',            'en',      'Age',
                            'age',            'kl',      'Ukiut'))
px_variable_label(x2)

# Remove VARIABLE-LABEL
x3 <- px_variable_label(x2, NULL)
px_variable_label(x3)
```

## `px_variable_type` *VARIABLE-TYPE*

### Description

Inspect or change VARIABLE-TYPE.

### Usage

```
px_variable_type(x, value, validate)

## S3 method for class 'px'
px_variable_type(x, value, validate = TRUE)
```

### Arguments

| | |
|---|---|
| `x` | A px object |
| `value` | A data frame with columns 'variable-code' and 'variable-type'. If value is missing, the current VARIABLE-TYPE is returned. If NULL, all VARIABLE-TYPE is removed. |
| `validate` | Optional. If TRUE a number of validation checks are performed on the px object, and an error is thrown if the object is not valid. If FALSE, the checks are skipped, which can be usefull for large px objects where the check can be time consuming. Use `px_validate()` to manually preform the check. |

### Value

A px object or data frame.

### See Also

Statistics Sweden's documentation

### Examples

```
library(tibble)

# Set VARIABLE-TYPE
x1 <-
  px(population_gl) |>
  px_variable_type(tibble('variable-code' = 'year', 'variable-type' = 'time'))

# Print VARIABLE-TYPE
px_variable_type(x1)

# Remove VARIABLE-TYPE
x2 <- px_variable_type(x1, NULL)
px_variable_type(x2)
```

# Index

* **datasets**
    age_classification, 3
    greenlanders, 4
    population_gl, 4
    px_keywords, 36

age_classification, 3

greenlanders, 4

population_gl, 4
px, 5
px_add_totals, 6
px_aggregallowed, 7
px_autopen, 8
px_axis_version, 9
px_baseperiod, 10
px_cellnote, 11
px_cellnotex, 13
px_cfprices, 14
px_charset, 15
px_classification, 16
px_codepage, 18
px_codepage(), *52*
px_confidential, 19
px_contact, 20
px_contents, 21
px_contvariable, 22
px_copyright, 23
px_creation_date, 24
px_data, 25
px_decimals, 27
px_description, 28
px_description(), *62*
px_descriptiondefault, 29
px_domain, 30
px_elimination, *6*, 31
px_figures, 33, *34*, *57*
px_heading, *33*, 34, *57*
px_infofile, 35
px_keywords, 36
px_language, 37, *38*
px_languages, *37*, 38
px_last_updated, 39
px_link, 40
px_map, 41
px_matrix, 43
px_micro, 44
px_next_update, 45
px_note, 46
px_notex, 47
px_order, 49
px_precision, 50
px_save, 51
px_save(), *19*
px_save_classification, 52
px_save_classification(), *17*
px_showdecimals, 53
px_source, 54
px_stockfa, 55
px_stub, *33, 34*, 56
px_subject_area, 57
px_subject_code, 58
px_tableid, 59
px_timeval, 60
px_title, 61
px_title(), *28*
px_units, 62
px_update_frequency, 64
px_validate, 65
px_validate(), *5–10*, *12–14*, *16*, *18*, *20–30*, *32–35*, *37–40*, *42, 43*, *45, 46*, *48–50*, *53–55*, *57–64*, *66–68*, *70, 71*
px_valuenote, 65
px_valuenotex, 67
px_values, 68
px_variable_label, 69
px_variable_type, 70