Package: newdata (via r-universe)

November 17, 2024

Title Generate New Data Frames for Prediction

Version 0.0.0.9020

Description Generates new data frames for predictive purposes. By
default, all specified variables vary across their range while
all other variables are held constant at the default reference
value. Types, classes, factor levels and time zones are always
preserved. The user can specify the length of each sequence,
require that only observed values and combinations are used and
add new variables.

License MIT + file LICENSE

URL `https://poissonconsulting.github.io/newdata/`,
`https://github.com/poissonconsulting/newdata/`

BugReports `https://github.com/poissonconsulting/newdata/issues`

Depends R (>= 4.0)

Imports chk, dplyr, hms, lifecycle, rlang, tibble, tidyr, vctrs

Suggests covr, datasets, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2.9000

Config/pak/sysreqs libicu-dev

Repository https://poissonconsulting.r-universe.dev

RemoteUrl https://github.com/poissonconsulting/newdata

RemoteRef HEAD

RemoteSha 089d893c862fabbbb8b660df5324331d33b231ea

Contents

`new_data` *Generate New Data*

Description

Generates a new data frame (in the form of a tibble) with each variable held constant or varying as a unique ordered sequence. All possible unique combinations are included and the columns are in the same order as those in `data`.

Usage

```
new_data(
  data,
  seq = character(0),
  ref = list(),
  obs_only = list(character(0)),
  length_out = 30
)
```

Arguments

`data`	The data frame to generate the new data from.
`seq`	A character vector of the variables in `data` to generate sequences for.
`ref`	A named list of reference values for variables that are not in seq.
`obs_only`	A list of character vectors indicating the sets of variables to only allow observed combinations for. If TRUE then obs_only is set to be seq.
`length_out`	A count indicating the maximum length of sequences for all types of variables except logical, character, factor and ordered factors.

Details

If an element of `ref` is a character vector and the corresponding column is a data frame, then the ref element is assigned the same factor levels as the column in the data. This is useful for choosing a factor level without having to set the correct levels.

Value

A tibble of the new data.

See Also

`new_value()` and `new_seq()`.

Examples

```
# an example data set
data <- tibble::tibble(
  vecint = c(1L, 3L),
  vecreal = c(1, 3),
  vecchar = c("b", "a"),
  vecdate = as.Date(c("2001-01-01", "2001-01-01"))
)

# vary count while holding other values constant
new_data(data, "vecint")
# vary continual
new_data(data, "vecreal")
new_data(data, c("vecchar", "vecint"))
```

`new_seq` *Generate New Sequence*

Description

Generate a new sequence of values. A sequence of values is used to predict the effect of a variable.

Usage

```
new_seq(x, length_out = NULL, ..., obs_only = NULL)

## S3 method for class 'logical'
new_seq(x, length_out = NULL, ..., obs_only = NULL)

## S3 method for class 'integer'
new_seq(x, length_out = NULL, ..., obs_only = NULL)

## S3 method for class 'double'
new_seq(x, length_out = NULL, ..., obs_only = NULL)

## S3 method for class 'character'
new_seq(x, length_out = NULL, ..., obs_only = NULL)

## S3 method for class 'factor'
new_seq(x, length_out = NULL, ..., obs_only = NULL)
```

```
## S3 method for class 'ordered'
new_seq(x, length_out = NULL, ..., obs_only = NULL)

## S3 method for class 'Date'
new_seq(x, length_out = NULL, ..., obs_only = NULL)

## S3 method for class 'POSIXct'
new_seq(x, length_out = NULL, ..., obs_only = NULL)

## S3 method for class 'hms'
new_seq(x, length_out = NULL, ..., obs_only = NULL)
```

Arguments

`x`	The object to generate the sequence from.
`length_out`	The maximum length of the sequence.
`...`	These dots are for future extensions and must be empty.
`obs_only`	A flag specifying whether to only use observed values.

Details

By default the sequence of values for objects of class numeric is 30 evenly space values across the range of the data. Missing values are always removed unless it's the only value or the object is zero length. The length of the sequence can be varied using the `length_out` argument which gives the reference value when 1 and can even be 0. For integer objects the sequence is the unique integers. For character objects it's the actual values sorted by how common they are followed by their actual value. For factors it's the factor levels in order with the trailing levels dropped first. For ordered factors the intermediate levels are dropped first. For Date vectors it's the unique dates; same for hms vectors. For POSIXct vectors the time zone is preserved. For logical objects the longest possible sequence is `c(TRUE, FALSE)`.

Value

A vector of the same class as the object.

Methods (by class)

- `new_seq(logical)`: Generate new sequence of values for logical objects
- `new_seq(integer)`: Generate new sequence of values for integer objects
- `new_seq(double)`: Generate new sequence of values for double objects
- `new_seq(character)`: Generate new sequence of values for character objects
- `new_seq(factor)`: Generate new sequence of values for factors
- `new_seq(ordered)`: Generate new sequence of values for ordered factors
- `new_seq(Date)`: Generate new sequence of values for Date vectors
- `new_seq(POSIXct)`: Generate new sequence of values for POSIXct vectors
- `new_seq(hms)`: Generate new sequence of values for hms vectors

See Also

new_value() and new_data().

Examples

```
# by default the sequence of values for objects of class numeric
# is 30 evenly space values across the range of the data
new_seq(c(1, 4))
# missing values are always removed
new_seq(c(1, 4, NA))
# unless it's the only value
new_seq(NA_real_)
# or the object is zero length
new_seq(numeric())
# the length of the sequence can be varied using the length_out argument
new_seq(c(1, 4), length_out = 3)
new_seq(c(1, 4), length_out = 2)
# which gives the reference value when 1
new_seq(c(1, 4), length_out = 1)
# and can even be 0
new_seq(c(1, 4), length_out = 0)
# for integer objects the sequence is the unique integers
new_seq(c(1L, 4L))
new_seq(c(1L, 100L))
# for character objects it's the actual values sorted by
# how common they are followed by their actual value
new_seq(c("a", "c", "c", "b", "b"))
new_seq(c("a", "c", "c", "b", "b"), length_out = 2)
# for factors its the factor levels in order
new_seq(factor(c("a", "b", "c", "c"), levels = c("b", "a", "g")))
# with the trailing levels dropped first
new_seq(factor(c("a", "b", "c", "c"), levels = c("b", "a", "g")),
  length_out = 2
)
# for ordered factors the intermediate levels are dropped first
new_seq(ordered(c("a", "b", "c", "c"), levels = c("b", "a", "g")),
  length_out = 2
)
# for Date vectors it's the unique dates
new_seq(as.Date(c("2000-01-01", "2000-01-04")))
# same for hms vectors
new_seq(hms::as_hms(c("00:00:01", "00:00:04")))
# for POSIXct vectors the time zone is preserved
new_seq(as.POSIXct(c("2000-01-01 00:00:01", "2000-01-01 00:00:04"),
  tz = "PST8PDT"
))
# for logical objects the longest possible sequence is `c(TRUE, FALSE)`
new_seq(c(TRUE, TRUE, FALSE), length_out = 3)
```

new_value — *Generate New Reference Value*

Description

Generate a new reference value for a vector.

Usage

```
new_value(x, ..., obs_only = NULL)
```

Arguments

`x`	The object to generate the reference value from.
`...`	These dots are for future extensions and must be empty.
`obs_only`	A flag specifying whether to only use observed values.

Details

By default the reference value for double vectors is the mean, unless obs_only = TRUE, in which case its the median of the unique values. For integer vectors it's the floored mean unless obs_only = TRUE, in which case it's also the median of the unique values. For character vectors it's the minimum of the most common values while for factors it's the first level. Ordered factors, Dates, times (hms), POSIXct and logical vectors are treated like integers. The factor levels and time zone are preserved.

Value

A scalar of the same class as the object.

See Also

`xnew_value()` and `new_seq()`.

Examples

```
# the reference value for objects of class numeric is the mean
new_value(c(1, 4))
# unless obs_only = TRUE, in which case its the median of the unique values
new_value(c(1, 4), obs_only = TRUE)

# for integer objects it's the floored mean
new_value(c(1L, 4L))

# for character objects it's the minimum of the most common values
new_value(c("a", "b", "c", "c", "b"))

# for factors its the first level and the factor levels are preserved
```

```
new_value(factor(c("a", "b", "c", "c"), levels = c("b", "a", "g")))

# other classes are treated like integers
new_value(ordered(c("a", "b", "c", "c"), levels = c("b", "a", "g")))
new_value(as.Date(c("2000-01-01", "2000-01-04")))
new_value(hms::as_hms(c("00:00:01", "00:00:04")))
new_value(as.POSIXct(c("2000-01-01 00:00:01", "2000-01-01 00:00:04")),
  tzone = "PST8PDT"
)
new_value(c(TRUE, FALSE, TRUE))
```

`obs_data` *Observed Data*

Description

An example tibble of observed data.

Usage

```
obs_data
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 3 rows and 9 columns.

Details

lgl A logical vector.

int An integer vector.

dbl A double vector.

chr A character vector.

fct A factor.

ord An ordered factor.

dte A Date vector.

dtt A POSIXct vector.

hms A hms vector.

Examples

```
obs_data
```

`xcast` *Cast New Values for* `xnew_data()`

Description

Casts a sequence of values to the same class as the original vector.

Usage

```
xcast(..., .data = xnew_data_env$data)
```

Arguments

`...`	TBD
`.data`	Normally defined by `xnew_data()`, users must pass a data frame or tibble if using this function directly.

Details

`xnew_seq()` is a wrapper function on `vctrs::vec_cast()` for use in `xnew_data()` to avoid having to repeating the column name.

See Also

`vctrs::vec_cast()` and `xnew_data()`

Examples

```
data <- tibble::tibble(
  period = factor(c("before", "before", "after", "after"),
    levels = c("before", "after")
  ),
  annual = factor(c(1, 3, 5, 8), levels = c(1, 3, 5, 8))
)

xnew_data(data, xcast(period = "before"))
xnew_data(data, xcast(period = "before", annual = c("1", "3")))
```

`xnew_data` *Generate New Data by Expansion*

Description

Generates a new data frame (in the form of a tibble)

Usage

```
xnew_data(.data, ..., .length_out = NULL)
```

Arguments

`.data`	The data frame to generate the new data from.
`...`	A list of variables to generate sequences for.
`.length_out`	NULL or a count specifying the maximum length of all sequences.

Details

By default, all specified variables vary across their range while all other variables are held constant at their reference value. Types, classes, factor levels and time zones are always preserved. The user can specify the length of each sequence, require that only observed values and combinations are used and add new variables.

See Also

`xnew_value()`, `xnew_seq()`, `xcast()` and `xobs_only()`

Examples

```
data <- tibble::tibble(
  period = factor(c("before", "before", "after", "after"),
    levels = c("before", "after")
  ),
  count = c(0L, 1L, 5L, 4L),
  annual = factor(c(2, 3, 5, 8), levels = c(1, 2, 3, 5, 8))
)

# By default all other variables are held constant at their reference value.
xnew_data(data)

# Specifying a variable causes it to vary across its range.
xnew_data(data, annual)

# The user can specify the length of a sequence.
xnew_data(data, xnew_seq(annual, length_out = 3))

# And only allow observed values.
```

```
xnew_data(data, xnew_seq(annual, length_out = 3, obs_only = TRUE))

# With multiple variables all combinations are produced
xnew_data(data, period, xnew_seq(annual, length_out = 3, obs_only = TRUE))

# To only preserve observed combinations use
xnew_data(data, xobs_only(period, annual))

# And to cast the values use
xnew_data(data, xcast(annual = "3"))
```

xnew_seq *Generate New Sequence for* `xnew_data()`

Description

Generate a new sequence of values for a vector.

Usage

```
xnew_seq(x, ...)
```

Arguments

`x`	The object to generate the sequence from.
`...`	Additional arguments passed to `new_seq()`.

Details

`xnew_seq()` is a wrapper function on `new_seq()` for use in `xnew_data()` to avoid having to repeating the column name.

See Also

`new_seq()` and `xnew_data()`.

Examples

```
data <- tibble::tibble(
  a = c(1L, 3L, 4L),
  b = c(4, 4.5, 6),
  d = c("a", "b", "c")
)

xnew_data(data, a, b = new_seq(b, length_out = 3), xnew_seq(d, length_out = 2))
```

xnew_value — *Generate New Reference Value for* `xnew_data()`

Description

Generate a new reference value for a vector.

Usage

```
xnew_value(x, ...)
```

Arguments

x	The object to generate the reference value from.
...	Additional arguments passed to `new_value()`.

Details

`xnew_value()` is a wrapper function on `new_value()` for use in `xnew_data()` to avoid having to repeating the column name.

See Also

`new_value()` and `xnew_data()`.

Examples

```
data <- tibble::tibble(
  a = c(1L, 3L, 4L),
  b = c(4, 4.5, 6),
  d = c("a", "b", "c")
)

xnew_data(data, a, b = new_value(b), xnew_value(d))
```

xobs_only — *Generate Observed Combinations Only*

Description

Generate Observed Combinations Only

Usage

```
xobs_only(..., .length_out = NULL, .data = xnew_data_env$data)
```

Arguments

`...`	One or more variables to generate combinations for.
`.length_out`	A count to override the default length of sequences.
`.data`	Normally defined by `xnew_data()`, users must pass a data frame or tibble if using this function directly.

See Also

`xnew_data()`

Examples

```
data <- tibble::tibble(
  period = factor(c("before", "before", "after", "after"),
    levels = c("before", "after")
  ),
  annual = factor(c(1, 3, 5, 8), levels = c(1, 3, 5, 8))
)
xnew_data(data, period, annual)
xnew_data(data, xobs_only(period, annual))
xnew_data(data, xobs_only(period, xnew_seq(annual, length_out = 3)))
```

Index