# Package 'ggperiodic'

August 31, 2018

**Title** Easy Plotting of Periodic Data with 'ggplot2'

**Version** 0.1.0

**Description** Implements methods to plot periodic data in any arbitrary range on the fly.

**License** GPL-3

**URL** https://github.com/eliocamp/ggperiodic

**BugReports** https://github.com/eliocamp/ggperiodic/issues

**Depends** R (>= 2.10)

**Imports** dplyr, ggplot2, sticky, tidyselect

**Suggests** covr, knitr, maps, rmarkdown, testthat, vdiffr

**VignetteBuilder** knitr

**ByteCompile** true

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.0

**NeedsCompilation** no

**Author** Elio Campitelli [cre, aut]

**Maintainer** Elio Campitelli <elio.campitelli@cima.fcen.uba.ar>

**Repository** CRAN

**Date/Publication** 2018-08-31 20:00:05 UTC

## R topics documented:

## get_period *Get period information from an object*

### Description

Get period information from an object

### Usage

```
get_period(object)
```

### Arguments

| | |
|---|---|
| object | a periodic object |

## ggperiodic *ggperiodic: Easy Plotting of Periodic Data with 'ggplot2'*

### Description

Implements methods to plot periodic data in any arbitrary range on the fly.

### Overview

The only thing you need to do is add the periodic information to a data frame with `periodic()`. You then can manually wrap your data around any domain with `wrap()` or just let ggplot2 do it automatically for you

### Author(s)

**Maintainer**: Elio Campitelli <elio.campitelli@cima.fcen.uba.ar>

### See Also

Useful links:

- https://github.com/eliocamp/ggperiodic
- Report bugs at https://github.com/eliocamp/ggperiodic/issues

## is.periodic *Check if an object is periodic*

### Description

Check if an object is periodic

### Usage

```
is.periodic(object)
```

### Arguments

| | |
|---|---|
| object | an object |

## periodic *Add or remove periodic variables*

### Description

Creates a periodic object by specifying the periodic variables and their periods.

### Usage

```
periodic(object, ...)

## Default S3 method:
periodic(object, period, ...)

## S3 method for class 'data.frame'
periodic(object, ...)
```

### Arguments

| | |
|---|---|
| object | the object to coerce to periodic |
| ... | name-value pairs of expressions defining the period |
| period | a numeric vector whose range defines the period |

### Details

This is a generic function and ggperiodic provides methods for vectors, data frames and ggplot2 Layers (geoms and stats).

## Examples

```
library(ggplot2)

x <- seq(0, 360 - 20, by = 20)
df <- data.frame(x = x, y = cos(x*pi/180))
df_p <- periodic(df, x = c(0, 360))

ggplot(df_p, aes(x, y)) +
   geom_line() +            # periodic data
   geom_point(data = df)    # non periodic data

# Extend domain
ggplot(df_p, aes(x, y), x = c(-180, 540)) +
   geom_line() +
   geom_point(data = df)

# with non regular intervals
x <- runif(30, 0, 360)
df <- periodic(data.frame(x = x, y = cos(x*pi/180)),
               x = c(0, 360))
ggplot(df, aes(x, y), x = c(-180, 540)) +
   geom_point()
```

---

`unperiodic` *Remove periodic specifications*

---

## Description

Remove periodic specifications

## Usage

```
unperiodic(object, ...)
```

## Arguments

| | |
|---|---|
| `object` | the object to remove periodicities |
| `...` | arguments to methods |

wrap *Wrap periodic data to an arbitrary range*

## Description

Wrap periodic data to an arbitrary range

## Usage

```
wrap(object, ...)

## S3 method for class 'periodic_df'
wrap(object, ..., .group = NULL)
```

## Arguments

| | |
|---|---|
| object | a periodic data frame |
| ... | name-value pairs of expressions defining range specifications |
| .group | optional group column (see examples) |

## Examples

```
x <- seq(0, 360 - 20, by = 20)
df <- data.frame(x = x, y = cos(x*pi/180))
df_p <- periodic(df, x = c(0, 360))

# wrap in default rante
df_wrapped <- wrap(df_p)
range(df_wrapped$x)
range(df$x)

# specify range
df_wrapped <- wrap(df_p, x = c(-145, 365))
range(df_wrapped$x)

# with non regular intervals
x <- runif(30, 0, 360)
df <- periodic(data.frame(x = x, y = cos(x*pi/180)),
               x = c(0, 360))
df_wrapped <- wrap(df, x = c(-180, 540))
range(df_wrapped$x)
range(df$x)
## Not run:
# This example illustrates the use of the .group parameter
library(ggplot2)
map <- periodic(map_data("world"), long = long)

# If wrapped without .group, the repated parts of the map
```

```
# have the same group and so polygons are not correctly defined.
map_wrapped <- wrap(map, long = c(-180, 360))
ggplot(map_wrapped, aes(long, lat, group = group)) +
    geom_path()

# Using groups, you get the correct grouping.
map_wrapped <- wrap(map, long = c(-180, 360), .group = group)
ggplot(map_wrapped, aes(long, lat, group = group)) +
    geom_path()

## End(Not run)
```

# Index