

# Package ‘Wats’

December 21, 2013

**Title** Wrap Around Time Series graphics

**Description** Wrap-around Time Series (WATS) Plots for Interrupted Time Series Designs

**Version** 0.1-17

**Date** 2013-12-19

**Author** Will Beasley [aut, cre], Joe Rodgers [aut], Matthew Schuelke [ctb]

**Maintainer** Will Beasley <wibeasley@hotmail.com>

**URL** <https://github.com/wibeasley/Wats>, <https://r-forge.r-project.org/projects/wats/>

**Depends** R (>= 3.0.0), stats

**Imports** colorspace, ggplot2, grid, lubridate, plyr, RColorBrewer, scales, testit, zoo

**Suggests** devtools, knitr, testthat

**License** GPL (>= 2)

**LazyData** TRUE

**VignetteBuilder** knitr

## R topics documented:

AnnotateData . . . . .	2
AugmentCycleData . . . . .	3
CartesianPeriodic . . . . .	3
CartesianRolling . . . . .	5
CountyMonthBirthRate . . . . .	7
PolarizeCartesian . . . . .	8
PolarPeriodic . . . . .	9
Wats . . . . .	11
<b>Index</b>	<b>12</b>

---

AnnotateData

*Finds midpoints and bands for the within and between cycles.*


---

## Description

Finds midpoints and bands for the within and between cycles.

## Usage

```
AnnotateData(dsLinear, dvName, centerFunction, spreadFunction,
  cycleTallyName = "CycleTally", stageIDName = "StageID",
  stageProgressName = "StageProgress",
  proportionThroughCycleName = "ProportionThroughCycle",
  proportionIDName = "ProportionID",
  terminalPointInCycleName = "TerminalPointInCycle")
```

## Arguments

<code>dsLinear</code>	The data.frame to containing the detailed data.
<code>dvName</code>	The name of the dependent/criterion variable.
<code>centerFunction</code>	A function to calculate the center of a subsample.
<code>spreadFunction</code>	A function to calculate the bands of a subsample.
<code>cycleTallyName</code>	The variable name indicating how many cycles have been completed.
<code>stageIDName</code>	The variable name indicating the stage. In a typical interrupted time series, these values are 1 before the interruption and 2 after.
<code>stageProgressName</code>	The variable name indicating the stage in a decimal form. This is mostly for internal uses.
<code>proportionThroughCycleName</code>	The variable name indicating how far the point is through a cycle. For example, 0 degrees would be 0, 180 degrees would be 0.5, 359 degrees would be 0.9972, and 360 degrees would be 0.
<code>proportionIDName</code>	The variable name indicating the ordinal position through a cycle.
<code>terminalPointInCycleName</code>	The variable name indicating the last point within a given cycle.

## Value

Returns a data.frame with additional variables «Say what they are».

## Examples

```
a <- 32+323
```

---

AugmentCycleData	<i>Calculates variables necessary for WATS Plots</i>
------------------	--

---

**Description**

Calculates variables necessary for WATS Plots

**Usage**

```
AugmentYearDataWithMonthResolution(dsLinear, dateName, stageIDName)
```

**Arguments**

dsLinear	The data.frame to containing the detailed data.
dateName	The variable name in dsLinear containing the date or datetime value.
stageIDName	The variable name indicating the stage. In a typical interrupted time series, these values are 1 before the interruption and 2 after.

**Value**

Returns a data.frame with additional variables: CycleTally, ProportionThroughCycle, ProportionID, and TerminalPointInCycle.

**Examples**

```
a <- 32+323
```

---

CartesianPeriodic	<i>Linear Plot with Periodic Elements</i>
-------------------	---

---

**Description**

Shows the interrupted time series in Cartesian coordinates and its a periodic/cyclic components.

**Usage**

```
CartesianPeriodic(dsLinear, dsPeriodic, xName, yName, stageIDName,
  periodicLowerName = "PositionLower", periodicUpperName = "PositionUpper",
  paletteDark = NULL, paletteLight = NULL, changePoints = NULL,
  changePointLabels = NULL, drawPeriodicBand = TRUE, jaggedPointSize = 2,
  jaggedLineSize = 0.5, bandAlphaDark = 0.4, bandAlphaLight = 0.15,
  changelineAlpha = 0.5, changelineSize = 3, title = NULL,
  xTitle = NULL, yTitle = NULL)
```

**Arguments**

<code>dsLinear</code>	The <code>data.frame</code> to containing the simple linear data. There should be one record per observation.
<code>dsPeriodic</code>	The <code>data.frame</code> to containing the reoccurring/periodic bands. There should be one record per observation per stage. If there are three stages, this <code>data.frame</code> should have three times as many rows as <code>dsLinear</code> .
<code>xName</code>	The variable name containing the date.
<code>yName</code>	The variable name containing the dependent/criterion variable.
<code>stageIDName</code>	The variable name indicating which stage the record belongs to. For example, before the first interruption, the <code>StageID</code> is 1, and is 2 afterwards.
<code>periodicLowerName</code>	The variable name showing the lower bound of a stage's periodic estimate.
<code>periodicUpperName</code>	The variable name showing the upper bound of a stage's periodic estimate.
<code>paletteDark</code>	A vector of colors used for the dark/heavy graphical elements. The vector should have one color for each <code>StageID</code> value. If no vector is specified, a default will be chosen, based on the number of stages.
<code>paletteLight</code>	A vector of colors used for the light graphical elements. The vector should have one color for each <code>StageID</code> value. If no vector is specified, a default will be chosen, based on the number of stages.
<code>changePoints</code>	A vector of values indicate the interruptions between stages. It typically works best as a <code>Date</code> or a <code>POSIXct</code> class.
<code>changePointLabels</code>	The text plotted above each interruption.
<code>drawPeriodicBand</code>	A boolean value indicating if the bands should be plotted (whose values are take from the <code>periodicLowerName</code> and <code>periodicUpperName</code> ).
<code>jaggedPointSize</code>	The size of the observed data points.
<code>jaggedLineSize</code>	The size of the line connecting the observed data points.
<code>bandAlphaDark</code>	The amount of transparency of the band appropriate for a stage's $x$ values.
<code>bandAlphaLight</code>	The amount of transparency of the band comparison stages for a given $x$ value.
<code>changeLineAlpha</code>	The amount of transparency marking each interruption.
<code>changeLineSize</code>	The width of a line marking an interruption.
<code>title</code>	The string describing the plot.
<code>xTitle</code>	The string describing the $x$ -axis.
<code>yTitle</code>	The string describing the $y$ -axis.

**Value**

Returns a `ggplot2` graphing object

## Examples

```
require(Wats) #Load the package
filePathOutcomes <- file.path(devtools::inst(name="Wats"), "extdata", "BirthRatesOk.txt")
dsLinear <- read.table(filePathOutcomes, header=TRUE, sep="\t", stringsAsFactors=FALSE)
dsLinear$Date <- as.Date(dsLinear$Date)
dsLinear$MonthID <- NULL
changeMonth <- as.Date("1996-02-15")
dsLinear$StageID <- ifelse(dsLinear$Date < changeMonth, 1L, 2L)
dsLinear <- AugmentYearDataWithMonthResolution(dsLinear=dsLinear, dateName="Date")
hSpread <- function( scores ) { return( quantile(x=scores, probs=c(.25, .75)) ) }
portfolio <- AnnotateData(
  dsLinear,
  dvName = "BirthRate",
  centerFunction = median,
  spreadFunction = hSpread
)

CartesianPeriodic(
  portfolio$dsLinear,
  portfolio$dsPeriodic,
  xName = "Date",
  yName = "BirthRate",
  stageIDName = "StageID",
  changePoints = changeMonth,
  changePointLabels = "Bombing Effect"
)
```

---

CartesianRolling

---

*Linear Plot with Rolling Summaries*


---

## Description

Shows the interrupted time series in Cartesian coordinates without a periodic/cyclic components.

## Usage

```
CartesianRolling(dsLinear, xName, yName, stageIDName,
  rollingLowerName = "RollingLower", rollingCenterName = "RollingCenter",
  rollingUpperName = "RollingUpper", paletteDark = NULL,
  paletteLight = NULL, colorSparse = grDevices::adjustcolor("tan1", 0.5),
  changePoints = NULL, changePointLabels = NULL, drawJaggedLine = TRUE,
  drawRollingLine = TRUE, drawRollingBand = TRUE,
  drawSparseLineAndPoints = TRUE, jaggedPointSize = 2,
  jaggedLineSize = 0.5, rollingLineSize = 1, sparsePointSize = 4,
  sparseLineSize = 0.5, bandAlpha = 0.4, changeLineAlpha = 0.5,
  changeLineSize = 3, title = NULL, xTitle = NULL, yTitle = NULL)
```

## Arguments

dsLinear	The data.frame to containing the data.
xName	The variable name containing the date.
yName	The variable name containing the dependent/criterion variable.

stageIDName	The variable name indicating which stage the record belongs to. For example, before the first interruption, the StageID is 1, and is 2 afterwards.
rollingLowerName	The variable name showing the lower bound of the rolling estimate.
rollingCenterName	The variable name showing the rolling estimate.
rollingUpperName	The variable name showing the upper bound of the rolling estimate.
paletteDark	A vector of colors used for the dark/heavy graphical elements. The vector should have one color for each StageID value. If no vector is specified, a default will be chosen, based on the number of stages.
paletteLight	A vector of colors used for the light graphical elements. The vector should have one color for each StageID value. If no vector is specified, a default will be chosen, based on the number of stages.
colorSparse	The color of the 'slowest' trend line, which plots only one value per cycle.
changePoints	A vector of values indicate the interruptions between stages. It typically works best as a Date or a POSIXct class.
changePointLabels	The text plotted above each interruption.
drawJaggedLine	A boolean value indicating if a line should be plotted that connects the observed data points.
drawRollingLine	A boolean value indicating if a line should be plotted that connects the rolling estimates specified by rollingCenterName.
drawRollingBand	A boolean value indicating if a band should be plotted that envelopes the rolling estimates (whose values are take from the rollingLowerName and rollingUpperName).
drawSparseLineAndPoints	A boolean value indicating if the sparse line and points should be plotted.
jaggedPointSize	The size of the observed data points.
jaggedLineSize	The size of the line connecting the observed data points.
rollingLineSize	The size of the line connecting the rolling estimates.
sparsePointSize	The size of the sparse estimates.
sparseLineSize	The size of the line connecting the sparse estimates.
bandAlpha	The amount of transparency of the rolling estimate band.
changeLineAlpha	The amount of transparency marking each interruption.
changeLineSize	The width of a line marking an interruption.
title	The string describing the plot.
xTitle	The string describing the x-axis.
yTitle	The string describing the y-axis.

**Value**

Returns a ggplot2 graphing object

## Examples

```
require(Wats) #Load the package
filePathOutcomes <- file.path(devtools::inst(name="Wats"), "extdata", "BirthRatesOk.txt")
dsLinear <- read.table(filePathOutcomes, header=TRUE, sep="\t", stringsAsFactors=FALSE)
dsLinear$Date <- as.Date(dsLinear$Date)
dsLinear$MonthID <- NULL
changeMonth <- as.Date("1996-02-15")
dsLinear$StageID <- ifelse(dsLinear$Date < changeMonth, 1L, 2L)
dsLinear <- AugmentYearDataWithMonthResolution(dsLinear=dsLinear, dateName="Date")
hSpread <- function( scores ) { return( quantile(x=scores, probs=c(.25, .75)) ) }
portfolio <- AnnotateData(
  dsLinear,
  dvName = "BirthRate",
  centerFunction = median,
  spreadFunction = hSpread
)

CartesianRolling(
  portfolio$dsLinear,
  xName = "Date",
  yName = "BirthRate",
  stageIDName = "StageID",
  changePoints = changeMonth,
  changePointLabels = "Bombing Effect"
)
```

---

CountyMonthBirthRate	<i>Monthly Growth Fertility Rates (GFR) for 12 urban Oklahoma counties</i>
----------------------	--

---

## Description

Monthly Growth Fertility Rates (GFR) for 12 urban counties in Oklahoma between January 1990 and December 1999. The GFR is defined as the number of births divided by the number of females (ages 15-44), multiplied by 1,000.

## Format

A data frame with 1,440 observations on the following 8 variables.

**Fips** The county's value according to the Federal Information Processing Standards. integer

**CountyName** The lower case name of the county. character

**Year** The year of the record, ranging from 1990 to 1999. integer

**Month** The month of the record, ranging from 1 to 12. integer

**FecundPopluation** The number of females in the county, ages of 15 to 44. integer

**BirthCount** The number of birth in a county for the given month. numeric - double precision float

**Date** The year and month of the record, with a date of the 15th. Centering the date within the month makes the value a little more representative and the graphs a little easier.date

**Gfr** The Growth Fertility Rate. numeric - double precision float

## Details

«Joe, can you please finish this sentence?» The monthly birth counts were copied from county records by ZZZ during the summer of 200?. It was collected for the purposes of [Rodgers, St. John, & Coleman \(2005\)](#).

The US Census' intercensal estimates are used for the January values of FecundPopluation. Values for February-December are interpolated using [approx](#).

## Author(s)

Will Beasley

## References

Rodgers, J. L., St. John, C. A. & Coleman R. (2005). [Did Fertility Go Up after the Oklahoma City Bombing?](#) An Analysis of Births in Metropolitan Counties in Oklahoma, 1990-1999. *Demography*, 42, 675-692.

[Intercensal estimates for 199x.](#)

[Intercensal estimates for 200x.](#)

## Examples

```
require(ggplot2)
ggplot(dsCountyMonth, aes(x=Date, y=Gfr, color=factor(Fips))) +
  geom_line() +
  labs(title="County Fertility - Longitudinal")

ggplot(dsCountyMonth, aes(x=Gfr, color=factor(Fips))) +
  geom_density() +
  labs(title="Distributions of County Fertility")
```

---

PolarizeCartesian

*Manipulate Cartesian data to use in the WATS polar plot*

---

## Description

Three operations are performed. First, within each stage, the first row is repeated at the end, to close the loop. Second, multiple points are interpolated (still in a Cartesian coordinates) so that the polar graph doesn't have sharp edges. These sharp edges would be artifacts of the conversion, and not reflect the observed data. Third, the Cartesian points are converted to polar coordinates.

## Usage

```
PolarizeCartesian(dsLinear, dsStageCycle, yName, stageIDName,
  cycleTallyName = "CycleTally",
  proportionThroughCycleName = "ProportionThroughCycle",
  periodicLowerName = "PositionLower",
  periodicCenterName = "PositionCenter",
  periodicUpperName = "PositionUpper", plottedPointCountPerCycle = 120,
  graphFloor = min(base::pretty(x = dsLinear[, yName])))
```



**Arguments**

<code>dsLinear</code>	The <code>data.frame</code> to containing the simple linear data. There should be one record per observation.
<code>dsStageCycle</code>	The <code>data.frame</code> to containing the reoccurring/periodic bands. There should be one record per observation per stage. If there are three stages, this <code>data.frame</code> should have three times as many rows as <code>dsLinear</code> .
<code>yName</code>	The variable name containing the dependent/criterion variable.
<code>stageIDName</code>	The variable name indicating which stage the record belongs to. For example, before the first interruption, the <code>StageID</code> is 1, and is 2 afterwards.
<code>cycleTallyName</code>	The variable name indicating how many <i>complete</i> cycles have occurred at that observation.
<code>proportionThroughCycleName</code>	The variable name showing how far through a cycle the observation (or summarized observations) occurred.
<code>periodicLowerName</code>	The variable name showing the lower bound of a stage's periodic estimate.
<code>periodicCenterName</code>	The variable name showing the center estimate of a stage's periodic estimate.
<code>periodicUpperName</code>	The variable name showing the upper bound of a stage's periodic estimate.
<code>plottedPointCountPerCycle</code>	The number of points that are plotted per cycle. If the polar graph has 'sharp corners', then increase this value.
<code>graphFloor</code>	The value of the criterion/dependent variable at the center of the polar plot.

**Value**

Returns a `data.frame`.

**Examples**

```
532 + 9/78
```

---

PolarPeriodic	<i>Polar Plot with Periodic Elements</i>
---------------	--

---

**Description**

Shows the interrupted time series in Cartesian coordinates and its a periodic/cyclic components.

**Usage**

```
PolarPeriodic(dsLinear, dsStageCyclePolar, xName, yName, stageIDName,
  periodicLowerName = "PositionLower", periodicUpperName = "PositionUpper",
  paletteDark = NULL, paletteLight = NULL, changePoints = NULL,
  changePointLabels = NULL, drawObservedLine = TRUE,
  drawPeriodicBand = TRUE, jaggedPointSize = 2, jaggedLineSize = 0.5,
  bandAlphaDark = 0.4, bandAlphaLight = 0.15, colorLabels = "gray50",
  colorGridlines = "gray80", changeLineAlpha = 0.5, changeLineSize = 3,
```

```

tickLocations = base::pretty(x = dsLinear[, yName],
graphFloor = min(tickLocations), graphCeiling = max(tickLocations),
cardinalLabels = NULL,
originLabel = paste("A point at the origin represents a value of",
graphFloor))

```

## Arguments

<code>dsLinear</code>	The data.frame to containing the simple linear data. There should be one record per observation.
<code>dsStageCyclePolar</code>	The data.frame to containing the bands for a single period. There should be one record per theta per stage. If there are three stages, this data.frame should have three times as many rows as <code>dsLinear</code> .
<code>xName</code>	The variable name containing the date.
<code>yName</code>	The variable name containing the dependent/criterion variable.
<code>stageIDName</code>	The variable name indicating which stage the record belongs to. For example, before the first interruption, the StageID is 1, and is 2 afterwards. #
<code>periodicLowerName</code>	The variable name showing the lower bound of a stage's periodic estimate. #
<code>periodicUpperName</code>	The variable name showing the upper bound of a stage's periodic estimate.
<code>paletteDark</code>	A vector of colors used for the dark/heavy graphical elements. The vector should have one color for each StageID value. If no vector is specified, a default will be chosen, based on the number of stages.
<code>palettelight</code>	A vector of colors used for the light graphical elements. The vector should have one color for each StageID value. If no vector is specified, a default will be chosen, based on the number of stages.
<code>changePoints</code>	A vector of values indicate the interruptions between stages. It typically works best as a Date or a POSIXct class.
<code>changePointLabels</code>	The text plotted above each interruption.
<code>drawObservedLine</code>	A boolean value indicating if the longitudinal observed line should be plotted (whose values are take from <code>dsLinear</code> ).
<code>drawPeriodicBand</code>	A boolean value indicating if the bands should be plotted (whose values are take from the <code>periodicLowerName</code> and <code>periodicUpperName</code> fields).
<code>jaggedPointSize</code>	The size of the observed data points.
<code>jaggedLineSize</code>	The size of the line connecting the observed data points.
<code>bandAlphaDark</code>	The amount of transparency of the band appropriate for a stage's $x$ values.
<code>bandAlphaLight</code>	The amount of transparency of the band comparison stages for a given $x$ value.
<code>changeLineAlpha</code>	The amount of transparency marking each interruption.
<code>colorLabels</code>	The color for <code>cardinalLabels</code> and <code>originLabel</code> .
<code>colorGridlines</code>	The color for the gridlines.
<code>changeLineSize</code>	The width of a line marking an interruption.

tickLocations	The desired locations for ticks showing the value of the criterion/dependent variable.
graphFloor	The value of the criterion/dependent variable at the center of the polar plot.
graphCeiling	The value of the criterion/dependent variable at the outside of the polar plot.
cardinalLabels	The four labels placed where ‘North’, ‘East’, ‘South’, and ‘West’ typically are.
originLabel	Explains what the criterion variable’s value is at the origin. Use NULL if no explanation is desired.

Value

Returns a grid graphical object (ie, a **grob**.)

Examples

32+7854

---

Wats	Wats
------	------

---

Description

Wats

# Index

## \*Topic **Cartesian**

CartesianPeriodic, [3](#)

CartesianRolling, [5](#)

## \*Topic **datasets**

CountyMonthBirthRate, [7](#)

## \*Topic **polar**

PolarizeCartesian, [8](#)

PolarPeriodic, [9](#)

AnnotateData, [2](#)

AugmentCycleData, [3](#)

AugmentYearDataWithMonthResolution  
(AugmentCycleData), [3](#)

AugmentYearDataWithSecondResolution  
(AugmentCycleData), [3](#)

CartesianPeriodic, [3](#)

CartesianRolling, [5](#)

CountyMonthBirthRate, [7](#)

PolarizeCartesian, [8](#)

PolarPeriodic, [9](#)

Wats, [11](#)

Wats-package (Wats), [11](#)