Package 'loadshaper'

May 17, 2022

Type Package

Title Producing Load Shape with Target Peak and Load Factor

Version 1.1.1

Date 2022-05-08

Language en-US

Description Modifying a load shape to match specific peak and
load factor is a fundamental component for various power system
planning and operation studies. This package is an efficient tool
to modify a reference load shape while matching the desired peak
and load factor. The package offers both linear and non-linear method,
described in <https://rpubs.com/riazakhan94/load_shape_match_peak_energy>.
The user can control the shape of the final load shape by regulating
certain parameters. The package provides validation metrics for
assessing the derived load shape in terms of preserving time series
properties. It also offers powerful graphics, that allows the user to
visually assess the derived load shape.

Imports stats, graphics

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation no

Author Md Riaz Ahmed Khan [aut, cre]

Maintainer Md Riaz Ahmed Khan <mdriazahmed.khan@jacks.sdstate.edu>

Repository CRAN

Date/Publication 2022-05-17 16:20:02 UTC

R topics documented:

`ercot` *ERCOT Hourly Load Data*

Description

Hourly load data of different operational areas of Electric Reliability Council of Texas (ERCOT) for year 2019 - 2021. The extra day (2/29/2020) from 2020 was intentionally omitted to have 8760 data for each year.

Usage

```
ercot
```

Format

A data frame with 26280 rows and 15 variables.

Source

https://www.ercot.com/gridinfo/load/load_hist/

Examples

```
loads <- ercot[ercot$Year == 2019, ]$COAST
plot(loads, type = "l")
linear_loadshape <- lslin(loads, target_lf = 0.50)
summary(linear_loadshape)
#------------------------------------
loads2 <- ercot[ercot$Year == 2020, ]$ERCOT
plot(loads2, type = "l")
linear_loadshape2 <- lslin(loads2, target_lf = 0.7)
summary(linear_loadshape2)
```

lscore *Load Shape Score*

Description

`lscore` provides a diagnostic score for evaluating the derived load shape in retaining time series properties.

Usage

```
lscore(ls, type = "acf", output = 2, lag = NULL)
```

Arguments

`ls`	An object of class `lslin` or `lslog`, created using function `lslin` or `lslog`
`type`	Type of correlation to be evaluate, either `"acf"` or `"pacf"`
`output`	Type of output to be used, either 1 or 2; uses `ls$y` if 1 and `ls$y2` if 2
`lag`	Maximum lag at which to calculate the acf or pacf. Same as `lag.max` in `acf`. If `Null`, then default is used.

Details

The diagnostic measure is calculated as a weighted mean absolute percent error (MAPE) of auto correlation or partial auto correlation values of the derived series with respect to the original. The values are calculated for given lag. Lag = 0 is omitted from calculation for auto correlation as it would be always 1. If o_i and d_i are the correlation values of original and derived load shape at lag i, then weighted MAPE is calculated as

$$wmape = \sum_{i=1}^{lag} w_i * |(o_i - d_i)/o_i|$$

where $w_i = \frac{|o_i|}{\sum_{i=1}^{lag} |o_i|}$

Since `wmape` is a measure of error, lower value indicates better preservation of time series property.

Value

A list of the followings:

- `wmape`: Weighted MAPE.
- `lag`: Lags at which ACF or PACF values were evaluated and used in calculating `wmape`.
- `type`: Type of Correlation (ACF or PACF)
- `cor_x`: ACF/PACF values of the original load.
- `cor_y`: ACF/PACF values of the derived load.
- `weight`: Weights at different lags used to calculate `wmape`.

Examples

```
loads <- ercot[ercot$Year == 2019, ]$COAST
linear_loadshape <- lslin(loads, target_lf = 0.4)
# --------------
scores_1 <- lscore(linear_loadshape, type = "acf", lag = 20)
print(scores_1)
# --------------
scores_2 <- lscore(linear_loadshape, type = "pacf")
print(scores_2)
```

`lslin` *Linear Method for Matching Peak and Load Factor*

Description

`lslin` applies linear method to a reference load shape to match the peak and load factor to target values. See "Details" for the algorithm.

Usage

```
lslin(x, target_max = 10000, target_lf = 0.7)
```

Arguments

`x`	A numeric array, representing reference load shape. All values must be strictly positive containing no `NA`(s). The length of `x` must be greater > 167.
`target_max`	Target peak value of resultant load shape, must be > 0.
`target_lf`	Target load factor of resultant load shape, must be numeric in between 0 and 1 (exclusive).

Details

The algorithm first evaluates the load factor of the reference load shape `x`, which is defined by the ratio of average to peak value. If the target load factor is greater than reference level, then all base values are multiplied by a number > `1`. If the target load factor is less than reference level, then all base values are multiplied by a number < `1`. The multipliers increase/decrease linearly and are applied to the based values after ordered.

If x' is the ordered version of x, then x'_i will be multiplied by $1 - (i - 1) * \beta$, where β is a constant calculated as:

$$\beta = \frac{\sum_{i=1}^{n} x'_i - target\,load\,factor}{\sum_{i=1}^{n} x'_i(i-1)}$$

The load factor of the derived series matches the target. For $target < base$, β is positive and vice-versa.

The algorithm attempts hard to match the load factor of the derived load shape to the base load factor. β becomes large in magnitude for large difference of base and target load factor. In case $\beta > 1$, it is possible to get negative multipliers which force the values to be negative. This particular situation can occur when target load factor is significantly smaller than the base load factor.

If the target load factor is much bigger than the base load factor, one/both of the followings can occur:

- As a linearly increasing function is multiplied by a decreasing function (x'), it is possible that the maximum of the product can exceed the maximum value of the base (x'), resulting in a different load factor.
- As a linearly increasing function is multiplied by a decreasing function (x'), it is possible that the product is not strictly decreasing. The product array is re-ordered to produce the final values.

The return object contains a data frame `df`, having the following columns:

- `x_index`: An index given to the original load shape `x`, starting from 1 to `length(x)`.
- `x`: The original array `x`, unaltered.
- `x_rank`: The rank of the data points of the given array `x`, from 1 for the peak to `length(x)` for the lowest value.
- `x_ordered`: Sorted `x` (largest to smallest).
- `x_pu`: Per unit `x`, derived by diving `x` by `max(x)`.
- `x_ordered_pu`: Per unit `x`, sorted from largest to smallest.
- `mult`: Derived multipliers, would be applied to sorted per unit `x`.
- `y_ordered_pu`: Product of per unit sorted `x` and `mult`.
- `y_ordered_pu2`: `y_ordered_pu`, sorted again, in case `y_ordered_pu` does not become decreasing.
- `y_pu`: Resultant load shape in per unit. This is derived by re-ordering `y_ordered_pu2` with respect to their original rank.
- `y`: Resultant load shape. This is derived by multiplying `y_pu` by `taget_max` / `base_max`

Value

A list of class `"lslin"`, having following elements:

- `df`: A data frame. See "Details".
- `beta`: Slope of the linearly increasing/decreasing multipliers. See "Details".
- `max_mult`: Maximum of the multipliers.

- `min_mult`: Minimum of the multipliers.
- `base_load_factor`: Load factor of the reference load shape `x`.
- `target_load_factor`: Target load factor.
- `derived_load_factor`: Load factor of the derived load shape (`object$df$y`).
- `base_max`: Peak value of the base load shape, `x`
- `target_max`: Target peak value of the new load shape.
- `derived_max`: Peak value of the derived load shape (`object$df$y`)
- `base_min`: Minimum value of the base load shape, `x`
- `derived_min`: Minimum value of the derived load shape (`object$df$y`)
- `dec_flag`: A logical flag stating whether the multipliers resulted in strictly decreasing values. `TRUE` indicates the order was not preserved. Only applicable for `target_max` > `base_max`. See "Details".
- `lf_flag`: A logical flag indicating if the load factor of the derived shape differs from the target by more than 1%.
- `min_pu_flag`: A logical flag indicating existence of negative values in the derived load shape. `TRUE` indicates the existence of negative values. Only applicable for `target_max` < `base_max`. See "Details".

See Also

`lslog`, `print.lslin`, `summary.lslin`, `plot.lslin`, `lscore`

Examples

```
loads <- ercot[ercot$Year == 2019, ]$COAST
plot(loads, type = "l")
linear_loadshape <- lslin(loads, target_lf = 0.50)
summary(linear_loadshape)
#------------------------------------
loads2 <- ercot[ercot$Year == 2020, ]$ERCOT
plot(loads2, type = "l")
linear_loadshape2 <- lslin(loads2, target_lf = 0.7)
summary(linear_loadshape2)#'
#------------------------------------
loads3 <- ercot[ercot$Year == 2020, ]$ERCOT
plot(loads3, type = "l")
linear_loadshape3 <- lslin(loads3, target_lf = 0.95)
summary(linear_loadshape3)
#------------------------------------
loads4 <- ercot[ercot$Year == 2020, ]$SCENT
plot(loads3, type = "l")
linear_loadshape4 <- lslin(loads4, target_lf = 0.3)
summary(linear_loadshape4)
```

lslog *Logistic Method for Matching Peak and Load Factor*

Description

Logistic Method for Matching Peak and Load Factor

Usage

```
lslog(
  x,
  target_max = 10000,
  target_lf = 0.7,
  k = 1,
  inf_pos = 0.5,
  iter = 500,
  def_l = 1
)
```

Arguments

`x`	A numeric array, representing reference load shape. All values must be strictly positive containing no `NA`(s). The length of `x` must be > 167.
`target_max`	Target peak value of resultant load shape, must be > 0.
`target_lf`	Target load factor of resultant load shape, must be numeric in between 0 and 1 (exclusive).
`k`	Steepness parameter, must be a positive number. See "Details".
`inf_pos`	Inflection point parameter. See "Details".
`iter`	Number of iterations for solving certain parameter. Must be >= 30. See "Details".
`def_l`	Start parameter for solving `l`, must be a positive numeric.

Details

The algorithm first evaluates the load factor of the reference load shape `x`, which is defined by the ratio of average to peak value. If the target load factor is greater than reference level, then all base values are multiplied by a number > `1`. If the target load factor is less than reference level, then all base values are multiplied by a number < `1`. The multipliers increase or decrease with a sigmoid pattern.

The sigmoid function is a transformed version of

$$f(x) = \frac{L}{1 - exp(-k(x - x_0))}$$

Parameter k is shape parameter, shaping the "sigmoidness" of the function. Larger value of `k` indicates more steepness in the function and lower value results in changes in multipliers in more linear fashion.

Location parameter x_0 controls the inflection point of the function and derived from `inf_pos`. `inf_pos` = `0.5` indicates the inflection point of the sigmoid multipliers is halfway.

The L parameter in the sigmoid is numerically solved. The number of iterations is equal to the `iter` argument, optimized based on the minimum difference between the derived and target load factor.

The return object contains a data frame `df`, having the following columns:

- `x_index`: An index given to the original load shape `x`, starting from 1 to `length(x)`.
- `x`: The original array `x`, unaltered.
- `x_rank`: The rank of the data points of the given array `x`, from 1 for the peak to `length(x)` for the lowest value.
- `x_ordered`: Sorted `x` (largest to smallest).
- `x_pu`: Per unit `x`, derived by diving `x` by `max(x)`.
- `x_ordered_pu`: Per unit `x`, sorted from largest to smallest.
- `mult`: Derived multipliers, would be applied to sorted per unit `x`.
- `y_ordered_pu`: Product of per unit sorted `x` and `mult`.
- `y_ordered_pu2`: `y_ordered_pu`, sorted again, in case `y_ordered_pu` does not become decreasing.
- `y_pu`: Resultant load shape in per unit. This is derived by re-ordering `y_ordered_pu2` with respect to their original rank.
- `y`: Resultant load shape. This is derived by multiplying `y_pu` by `taget_max` / `base_max`

Value

A list of class `"lslog"`, having following elements:

- `df`: A data frame. See "Details".
- `k`: Steepness parameter. See "Details".
- `inf_pos`: Inflection point parameter. See "Details".
- `L`: Numerically solved optimized L parameter. See "Details".
- `max_mult`: Maximum of the multipliers.
- `min_mult`: Minimum of the multipliers.
- `base_load_factor`: Load factor of the reference load shape `x`.
- `target_load_factor`: Target load factor.
- `derived_load_factor`: Load factor of the derived load shape (`object$df$y`).
- `base_max`: Peak value of the base load shape, `x`
- `target_max`: Target peak value of the new load shape.
- `derived_max`: Peak value of the derived load shape (`object$df$y`)
- `base_min`: Minimum value of the base load shape, `x`
- `derived_min`: Minimum value of the derived load shape (`object$df$y`)
- `dec_flag`: A logical flag stating whether the multipliers resulted in strictly decreasing values. `TRUE` indicates the order was not preserved. Only applicable for `target_max` > `base_max`. See "Details".

- `lf_flag`: A logical flag indicating if the load factor of the derived shape differs from the target by more than 1%.
- `min_pu_flag`: A logical flag indicating existence of negative values in the derived load shape. `TRUE` indicates the existence of negative values. Only applicable for `target_max` < `base_max`. See "Details".

See Also

`lslin`, `print.lslog`, `summary.lslog`, `plot.lslog`, `lscore`

Examples

```
loads <- ercot[ercot$Year == 2019, ]$COAST
plot(loads, type = "l")
logistic_loadshape <- lslog(loads, target_lf = 0.50, k = 0.5)
summary(logistic_loadshape)
#---------------------------------------------------
loads2 <- ercot[ercot$Year == 2020, ]$ERCOT
plot(loads2, type = "l")
logistic_loadshape2 <- lslog(loads2, target_lf = 0.6,
                            k = 0.5, inf_pos = 0.4)
summary(logistic_loadshape2)
#---------------------------------------------------
loads3 <- ercot[ercot$Year == 2020, ]$ERCOT
plot(loads3, type = "l")
logistic_loadshape3 <- lslog(loads3, target_lf = 0.9)
summary(logistic_loadshape3)
```

`plot.lslin` *Plot Linear Load Shape*

Description

Plot method of `lslin` object

Usage

```
## S3 method for class 'lslin'
plot(
  x,
  case = 1,
  col = c(1, 2),
  scatter = FALSE,
  legend = TRUE,
  leg_pos = "topright",
  ... = NULL
)
```

Arguments

`x`	An object of class `lslin`, created by `lslin` function.
`case`	A numeric value from {1, 2, 3} to select the type of plot. See "Details".
`col`	Color of the plots. Can be numeric or text or mixed as in `col`. For `length(col)` < 2, a default second color is used.
`scatter`	Logical. Scatter plot if `TRUE`, line plot if `FALSE`.
`legend`	Logical indicating if legend to be displayed.
`leg_pos`	A text value for position/location of the legend. Default is `topright`. See `legend` for full list of keywords.
`...`	`NULL`. Used for S3 generic/method consistency.

Details

If `scatter = FALSE` then per unit load duration curve for `case = 1`, per unit load for `case = 2`, actual load (in original unit) for `case = 3`. If `scatter = TRUE` then per unit scatter plot for `case = 1 or 2`, actual load scatter plot for `case = 3`.

Value

`NULL`.

Examples

```
loads <- ercot[ercot$Year == 2019, ]$COAST
linear_loadshape <- lslin(loads, target_lf = 0.5)
# --------------
plot(linear_loadshape, col = c(2, 4))
plot(linear_loadshape, case = 2, col = c(2, 4))
plot(linear_loadshape, case = 3,
     col = c("salmon", "deepskyblue"),
     leg_pos = "topleft")
```

`plot.lslog` *Plot Logistic Load Shape*

Description

Plot method of `lslog` object

Usage

```
## S3 method for class 'lslog'
plot(
  x,
  case = 1,
  col = c(1, 2),
  scatter = FALSE,
  legend = TRUE,
  leg_pos = "topright",
  ... = NULL
)
```

Arguments

`x`	An object of class `lslog`, created by `lslog` function.
`case`	A numeric value from {1, 2, 3} to select the type of plot. See "Details".
`col`	Color of the plots. Can be numeric or text or mixed as in `col`. For `length(col)` < 2, a default second color is used.
`scatter`	Logical. Scatter plot if `TRUE`, line plot if `FALSE`.
`legend`	Logical indicating if legend to be displayed.
`leg_pos`	A text value for position/location of the legend. Default is `topright`. See `legend` for full list of keywords.
`...`	`NULL`. Used for S3 generic/method consistency.

Details

If `scatter` = `FALSE` then per unit load duration curve for `case` = 1, per unit load for `case` = 2, actual load (in original unit) for `case` = 3. If `scatter` = `TRUE` then per unit scatter plot for `case` = `1 or 2`, actual load scatter plot for `case` = 3.

Value

`NULL`.

Examples

```
loads <- ercot[ercot$Year == 2019, ]$COAST
loads <- ercot[ercot$Year == 2019, ]$COAST
log_loadshape <- lslog(loads, target_lf = 0.5,
                       inf_pos = 0.3, k = 0.8)
# --------------
plot(log_loadshape, col = c(2, 4))
plot(log_loadshape, case = 2, col = c(2, 4))
plot(log_loadshape, case = 3,
     col = c("salmon", "deepskyblue"),
     leg_pos = "topleft")
```

`print.lscore` *Print Summary of Load Shape Score*

Description

Print Summary of Load Shape Score

Usage

```
## S3 method for class 'lscore'
print(x, ... = NULL)
```

Arguments

`x`	An object of class `lscore`, created by the function `lscore`.
`...`	`NULL`. Used for S3 generic/method consistency.

Value

`NULL`. Prints the summary of the load shape score.

Note

Same as `summary.lscore`

See Also

`lslin`, `lscore`, `lscore`

Examples

```
loads <- ercot[ercot$Year == 2019, ]$SOUTH
# --------------
log_loadshape <- lslog(loads, target_lf = 0.5)
print(lscore(log_loadshape, type = "acf"))
print(lscore(log_loadshape, type = "pacf"))
# --------------
lin_loadshape <- lslin(loads, target_lf = 0.5)
print(lscore(lin_loadshape, type = "acf"))
print(lscore(lin_loadshape, type = "pacf"))
```

`print.lslin` *Print Linear Load Shape*

Description

Print method for summarizing `lslin` object

Usage

```
## S3 method for class 'lslin'
print(x, ... = NULL)
```

Arguments

`x`	An object of class `lslin`, created by the function `lslin`.
`...`	`NULL`. Used for S3 generic/method consistency.

Value

`NULL`. Prints the summary of the derived load shape using linear method.

Note

Same as `summary.lslin`

See Also

`lslin`

Examples

```
# ---------------------
loads <- ercot[ercot$Year == 2019, ]$COAST
linear_loadshape <- lslog(loads, target_lf = 0.5, k = 0.5)
print(linear_loadshape)
# ---------------------
loads2 <- ercot[ercot$Year == 2019, ]$ERCOT
linear_loadshape2 <- lslog(loads2, target_lf = 0.75, k = 1)
print(linear_loadshape2)
```

`print.lslog` *Print Logistic Load Shape*

Description

Print method of `lslog` object

Usage

```
## S3 method for class 'lslog'
print(x, ... = NULL)
```

Arguments

`x`	An object of class `lslog`, created by the function `lslog`.
`...`	`NULL`. Used for S3 generic/method consistency.

Value

`NULL`. Prints the summary of the derived load shape using linear method.

Note

Same as `summary.lslog`

See Also

`lslog`

Examples

```
# ---------------------
loads <- ercot[ercot$Year == 2019, ]$COAST
logistic_loadshape <- lslog(loads, target_lf = 0.5, k = 0.5)
print(logistic_loadshape)
# ---------------------
loads2 <- ercot[ercot$Year == 2019, ]$ERCOT
logistic_loadshape2 <- lslog(loads2, target_lf = 0.75, k = 1)
```

```
print(logistic_loadshape2)
```

summary.lscore *Summary of Load Shape Score*

Description

Summary method of `lscore` object

Usage

```
## S3 method for class 'lscore'
summary(object, ... = NULL)
```

Arguments

`object`	An object of class `lscore`, created by the function `lscore`.
`...`	`NULL`. Used for S3 generic/method consistency.

Value

`NULL`. Prints the summary of the load shape score.

Note

Same as `print.lscore`

See Also

`lslin`, `lslog`, `lscore`

Examples

```
loads <- ercot[ercot$Year == 2019, ]$SOUTH
# --------------
log_loadshape <- lslog(loads, target_lf = 0.5)
summary(lscore(log_loadshape, type = "acf"))
summary(lscore(log_loadshape, type = "pacf"))
# --------------
lin_loadshape <- lslin(loads, target_lf = 0.5)
summary(lscore(lin_loadshape, type = "acf"))
summary(lscore(lin_loadshape, type = "pacf"))
```

`summary.lslin` *Summary of Linear Load Shape*

Description

Summary method of `lslin` object

Usage

```
## S3 method for class 'lslin'
summary(object, ... = NULL)
```

Arguments

`object`	An object of class `lslin`, created by the function `lslin`.
`...`	`NULL`. Used for S3 generic/method consistency.

Value

`NULL`. Prints the summary of the derived load shape using linear method.

Note

Same as `print.lslin`

See Also

`lslin`

Examples

```
# ---------------------
loads <- ercot[ercot$Year == 2019, ]$COAST
linear_loadshape <- lslog(loads, target_lf = 0.5, k = 0.5)
summary(linear_loadshape)
# ---------------------
loads2 <- ercot[ercot$Year == 2019, ]$ERCOT
linear_loadshape2 <- lslog(loads2, target_lf = 0.75, k = 1)
summary(linear_loadshape2)
```

summary.lslog *Summary of Logistic Load Shape*

Description

Print method for summarizing `lslog` object

Usage

```
## S3 method for class 'lslog'
summary(object, ... = NULL)
```

Arguments

`object`	An object of class `lslog`, created by the function `lslog`.
`...`	`NULL`. Used for S3 generic/method consistency.

Value

`NULL`. Prints the summary of the derived load shape using linear method.

Note

Same as `print.lslog`

See Also

`lslog`

Examples

```
# ---------------------
loads <- ercot[ercot$Year == 2019, ]$COAST
logistic_loadshape <- lslog(loads, target_lf = 0.5, k = 0.5)
summary(logistic_loadshape)
# ---------------------
loads2 <- ercot[ercot$Year == 2019, ]$ERCOT
logistic_loadshape2 <- lslog(loads2, target_lf = 0.75, k = 1)
summary(logistic_loadshape2)
```

Index