

# Package ‘tempR’

January 8, 2020

**Type** Package

**Title** Temporal Sensory Data Analysis

**Version** 0.9.9.16

**Date** 2020-1-7

**Author** J.C. Castura [aut, cre, ctb] (<<https://orcid.org/0000-0002-1640-833X>>)

**Maintainer** J.C. Castura <[jcastura@compusense.com](mailto:jcastura@compusense.com)>

## Description

Analysis and visualization of data from temporal sensory methods, including for temporal check-all-that-apply (TCATA) and temporal dominance of sensations (TDS).

**Depends** R (>= 3.2.2)

**Imports** grDevices, stats, graphics

**License** GPL (>= 2)

**LazyData** TRUE

**Encoding** UTF-8

**RoxygenNote** 7.0.2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-01-08 22:30:15 UTC

## R topics documented:

adjust.brightness . . . . .	2
bootstrap.band . . . . .	3
convert.tcata . . . . .	3
convert.tcategory . . . . .	4
count.selections . . . . .	5
dist.city.block . . . . .	5
fill.gaps . . . . .	6
get.chance . . . . .	7
get.decluttered . . . . .	7
get.differences . . . . .	8

get.dominance.rates . . . . .	9
get.mat.diff.sign . . . . .	10
get.significance . . . . .	11
get.significance.diff . . . . .	12
get.smooth . . . . .	13
get.times . . . . .	14
lengthwhichis.na . . . . .	15
make.palettes . . . . .	15
plot_pca.trajectories . . . . .	16
pretty_palette . . . . .	18
similarity.tcata.repeatability . . . . .	19
similarity.tcata.replication . . . . .	20
std.time . . . . .	21
TCATA data set: orange juice . . . . .	22
TCATA data set: Syrah wines . . . . .	23
tcata.diff.plot . . . . .	24
tcata.line.plot . . . . .	26
TDS data set: orange juice . . . . .	28
TDS data set: snack bars . . . . .	29
tds.diff.plot . . . . .	29
tds.plot . . . . .	31
tempR . . . . .	32

## Index 33

---

adjust.brightness	<i>Adjust color brightness</i>
-------------------	--------------------------------

---

### Description

Select suitable colors for highlighting plots.

### Usage

```
adjust.brightness(rgb.in, percent = 10)
```

### Arguments

rgb.in	rgb of input color
percent	the degree to which input color will be modified/brightened

### Value

hex hex code for new color

### Examples

```
(rgb.in <- c(col2rgb("red")))
adjust.brightness(rgb.in, percent = 10)
```

---

bootstrap.band	<i>Get bootstrap confidence bands for attribute selections</i>
----------------	--

---

**Description**

Get bootstrap confidence bands for TCATA attribute citation rates or TDS attribute dominance rates.

**Usage**

```
bootstrap.band(X, boot = 999, alpha = 0.05, return.bias = FALSE)
```

**Arguments**

X	data frame of indicator data (with possible values 0 or 1)
boot	number of virtual panels
alpha	alpha level for bootstrap confidence bands
return.bias	indicates whether to return bias associated with bootstrap mean value

**Details**

Get bootstrap confidence bands for TCATA attribute citation rates or TDS attribute dominance rates.

**Value**

lcl lower  $100(\alpha/2)\%$  bootstrap confidence limit  
 ccl upper  $100(1-\alpha/2)\%$  bootstrap confidence limit  
 bias provided if `output.bias = TRUE`

**Examples**

```
x <- ojtcata[ojtcata$samp == 1 & ojtcata$attribute == "Sweetness", -c(1:4)]
x.boot.ci <- bootstrap.band(x, boot = 99) # 99 is only for illustrative purposes
x.boot.ci
```

---

convert.tcata	<i>Convert TCATA data</i>
---------------	---------------------------

---

**Description**

Converts TCATA data from a set of onset-offset times to an indicator vector (0s and 1s). Also works for TDS data.

**Usage**

```
convert.tcata(X, times, decimal.places = 2)
```

**Arguments**

`X` matrix with onset (start) times in first column and offset (stop) times in second column

`times` time slices for output indicator vector

`decimal.places` decimal places used in times; used for naming of the indices of `out.vec`

**Value**

`out.vec` indicator vector(0s and 1s)

**Examples**

```
X <- rbind(c(3.18, 6.83), c(8.46, 11.09), c(18.61, 21.80))
times <- seq(0, 25, by = 0.01)
Xnew <- convert.tcata(X, times)
Xnew
```

---

convert.tcategory      *Convert Temporal Category data*

---

**Description**

Converts Temporal Category data from a set of onset-offset times and ratings to an vector of ratings.

**Usage**

```
convert.tcategory(X, in.scores, times, decimal.places = 2)
```

**Arguments**

`X` matrix with onset (start) times in first column and offset (stop) times in second column

`in.scores` vector of category values corresponding to rows of `X`

`times` time slices for output vector

`decimal.places` decimal places used in times; used for naming of the indices of `out.vec`

**Value**

`out.vec` indicator vector(0s and 1s)

**Examples**

```
X <- rbind(c(3.18, 6.83), c(8.46, 11.09), c(18.61, 21.80))
in.scores <- c(7, 6, 5)
times <- seq(0, 25, by = 0.01)
Xnew <- convert.tcategory(X, in.scores, times)
Xnew
```

---

count.selections	<i>Count attribute selections</i>
------------------	-----------------------------------

---

**Description**

Count the number of times that the attribute was selected (or optionally: deselected) in a single TCATA or TDS evaluation.

**Usage**

```
count.selections(x, deselections = FALSE)
```

**Arguments**

x                    vector of binary data (with possible values 0 or 1)  
deselections        set to TRUE if purpose is to count the number of deselections

**Details**

Count the number of times that the attribute was selected (or, optionally, deselected) in a single TCATA or TDS evaluation.

**Value**

count of selections (or deselections if deselections = TRUE)

**Examples**

```
data(bars)
paste0(bars[1, -c(1:4)], collapse = "")
# this attribute was checked 3 times and unchecked 2 times
count.selections(bars[1, -c(1:4)])
count.selections(bars[1, -c(1:4)], deselections = TRUE)
```

---

dist.city.block	<i>Calculate city block distance between two matrices</i>
-----------------	---

---

**Description**

Calculates the city block distance between two matrices.

**Usage**

```
dist.city.block(x, y)
```

**Arguments**

x                    first matrix  
y                    second matrix

**Value**

cbdist city block distance between x and y

**Examples**

```
x <- matrix(0, nrow = 5, ncol = 7)
y <- matrix(1, nrow = 5, ncol = 7)
dist.city.block(x, y)

y <- matrix(c(rep(0, 15), rep(1, 20)), nrow = 5, ncol = 7)
dist.city.block(x, y)
```

---

fill.gaps

*Fills gaps*

---

**Description**

Replace gaps in TDS and TCATA data with replacement responses.

**Usage**

```
fill.gaps(y, subst = 0, repl = 1)
```

**Arguments**

y                    vector (or data frame) of Bernoulli data which may contain gaps  
subst                value occurring in a gap (which represents real data outside a gap). Default is 0.  
repl                 value occurring for a response (used to replace gap values). Default is 1.

**Value**

out vector (or data frame) of Bernoulli data with filled gaps

**Examples**

```
# vector with gaps: x with NA gaps (e.g. due to attribute cuing)
(x <- rep(c(rep(NA, 4), rep(1, 4)), 2))
fill.gaps(x, subst = NA)

# array with gaps: y with an gap of 0s (e.g. due to attribute fading)
(y <- structure(c(0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0,
                 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0),
               .Dim = c(3L, 10L),
```

```
fill.gaps(y) .Dimnames = list(1:3, 1:10))
```

---

get.chance	<i>TDS chance proportion</i>
------------	------------------------------

---

### Description

Obtains the TDS chance proportion based on the number of attributes, as proposed by Pineau et al. (2009; Eq. 1).

### Usage

```
get.chance(attributes = c(), include.stop = FALSE)
```

### Arguments

attributes	number of attributes used in the TDS ballot.
include.stop	default is FALSE. Default should be kept if time standardization is applied. Optionally, set to TRUE if analyzing data on the raw timeline.

### References

Pineau, N., Schlich, P., Cordelle, S., Mathonnière, C., Issanchou, S., Imbert, A., Rogeaux, M., Etiévant, P., & Köster, E. (2009). Temporal dominance of sensations: Construction of the TDS curves and comparison with time–intensity. *Food Quality and Preference*, 20, 450–455. <http://dx.doi.org/10.1016/j.foodqual.2009.04.005>

### Examples

```
# example using 'bars' data set
attributes <- unique(bars$attribute)
chance <- get.chance(attributes)
chance
```

---

get.decluttered	<i>Get decluttering matrix indicating where to show/hide reference lines</i>
-----------------	--

---

### Description

Declutter TCATA curves by hiding reference lines from plots showing TCATA curves.

### Usage

```
get.decluttered(x = x, n.x = n.x, y = y, n.y = n.y, alpha = 0.05)
```

**Arguments**

x	selections for sample of interest (can be a vector if several samples of interest)
n.x	evaluations of x (can be a vector if several samples of interest)
y	selections for comparison (can be a vector if several comparisons will be made)
n.y	evaluations of y (can be a vector if several comparisons of interest)
alpha	significance level

**Value**

declutter vector in which 1 indicates "show" and NA indicates "hide"

**References**

Castura, J.C., Antúnez, L., Giménez, A., Ares, G. (2016). Temporal check-all-that-apply (TCATA): A novel temporal sensory method for characterizing products. *Food Quality and Preference*, 47, 79-90. <http://dx.doi.org/10.1016/j.foodqual.2015.06.017>

**See Also**

[fisher.test](#)

**Examples**

```
# example using 'ojtcata' data set
data(ojtcata)
x <- aggregate(ojtcata[, -c(1:4)], list(samp = ojtcata$samp, attribute = ojtcata$attribute), sum)
p.1.checked <- x[x$samp == 1, -c(1:2)]
p.1.eval <- length(unique(ojtcata$cons))
p.not1.checked <- aggregate(x[x$samp != 1, -c(1:2)],
  list(attribute = x$attribute[x$samp != 1]), sum)[, -1]
p.not1.eval <- length(unique(ojtcata$cons)) * (length(unique(ojtcata$samp)) - 1)

# reference lines for contrast products
p.1.refline <- p.not1.checked / p.not1.eval
# decluttering matrix corresponds to the dimensions of p.1.refline
p.1.declutter <- get.decluttered(x = unlist(p.1.checked), n.x = p.1.eval,
  y = unlist(p.not1.checked), n.y = p.not1.eval)
(p.1.declutter <- matrix(p.1.declutter, nrow = nrow(p.1.checked)))
```

---

get.differences

*Get vector of difference in dominance rates*

---

**Description**

Get vector of difference in dominance rates



**Usage**

```
get.differences(x, y)
```

**Arguments**

**x** matrix of dominance indicators for a single productxattribute (rows = evaluations, columns = times)

**y** matrix of dominance indicators for a different product (same attribute)

**Value**

out vector of differences in dominance rates

**References**

Pineau, N., Schlich, P., Cordelle, S., Mathonnière, C., Issanchou, S., Imbert, A., Rogeaux, M., Etiévant, P., & Köster, E. (2009). Temporal dominance of sensations: Construction of the TDS curves and comparison with time–intensity. *Food Quality and Preference*, 20, 450–455. <http://dx.doi.org/10.1016/j.foodqual.2009.04.005>

**Examples**

```
# example using 'bars' data set
bars.m <- aggregate(bars[, -c(1:4)], list(samples = bars$sample, attribute = bars$attribute), mean)
bars.m <- bars.m[order(bars.m$sample, bars.m$attribute), ]
attributes <- unique(bars$attribute)
times <- get.times(colnames(bars.m)[-c(1:2)])
bar1 <- bars.m[bars.m$sample == 1 & bars.m$attribute == "Caramelized Flavour", -c(1:2)]
bar2 <- bars.m[bars.m$sample == 2 & bars.m$attribute == "Caramelized Flavour", -c(1:2)]
b.diff <- get.differences(bar1, bar2)
round(b.diff, 3)

# toy example
x <- data.frame(t10 = c( NA,  0,  0,  0,  1,  1,  0,  0,  1,  0, NA),
               t15 = c(  1,  0,  0,  1,  1,  1,  0,  1,  0,  1,  0),
               t20 = c(  1,  1,  1,  1,  1,  1,  1,  0,  1, NA,  0))
y <- data.frame(t10 = c( NA, NA,  0,  0,  1,  1,  0,  0,  0,  0, NA),
               t15 = c(  0,  0,  0,  0,  1,  0,  1,  1,  0,  1,  1),
               t20 = c(  1,  0,  1,  1,  0,  0,  1, NA,  1, NA,  0))
get.differences(x, y)
```

---

```
get.dominance.rates    Get TDS dominance rates
```

---

**Description**

Get TDS dominance rates.

**Usage**

```
get.dominance.rates(citations, n)
```

**Arguments**

citations	matrix of dominance counts
n	number of observations (evaluations) per cell

**References**

Pineau, N., Schlich, P., Cordelle, S., Mathonnière, C., Issanchou, S., Imbert, A., Rogeaux, M., Etiévant, P., & Köster, E. (2009). Temporal dominance of sensations: Construction of the TDS curves and comparison with time–intensity. *Food Quality and Preference*, 20, 450–455. <http://dx.doi.org/10.1016/j.foodqual.2009.04.005>

**Examples**

```
x <- rbind(c( 6,  6,  8, 14, 16, 22, 22, 21, 13, 11, 14, 7,  7,  6,  5,  3),
           c(14, 24, 31, 36, 37, 39, 44, 48, 51, 55, 48, 40, 30, 20, 10,  5),
           c( 7,  8,  9, 15, 17, 21, 21, 20, 21, 22, 18, 17, 17, 20, 20, 20),
           c(44, 23, 23, 26,  1,  2,  2,  2,  2,  3,  4,  7, 15, 14, 18, 22),
           c(20, 30, 20,  0, 20,  7,  2,  0,  4,  0,  7, 20, 22, 31, 38, 41))
colnames(x) <- 0:15
get.dominance.rates(x, n = 91)
```

---

get.mat.diff.sign      *Pairwise comparisons*

---

**Description**

p-value for pairwise comparisons.

**Usage**

```
get.mat.diff.sign(x = x, y = y, n.x = n.x, n.y = n.x, test.type = "f")
```

**Arguments**

x	citations for product x
y	citations for product y
n.x	total observations for x
n.y	total observations for y
test.type	So far only Fisher's exact test is implemented ("f")

## References

Castura, J.C., Antúnez, L., Giménez, A., Ares, G. (2016). Temporal check-all-that-apply (TCATA): A novel temporal sensory method for characterizing products. *Food Quality and Preference*, 47, 79-90. <http://dx.doi.org/10.1016/j.foodqual.2015.06.017>

## See Also

[fisher.test](#)

## Examples

```
# Toy TCATA citations data for two samples: s1, s2
s1 <- t(data.frame(sweet = c(10, 23, 25, 26, 26, 43, 44),
                  bitter = c( 4, 18, 19, 27, 36, 43, 54),
                  sour = c(40, 53, 85, 70, 46, 33, 24)))
s2 <- t(data.frame(sweet = c(11, 33, 45, 46, 56, 43, 44),
                  bitter = c( 0, 11, 11, 14, 25, 35, 34),
                  sour = c(30, 33, 35, 20, 26, 23, 24)))
colnames(s1) <- colnames(s2) <- paste0("time_", seq(5, 35, by = 5), "s")
n <- 90
signif <- get.mat.diff.sign(s1, s2, n, n)
signif
```

---

get.significance	<i>TDS significance proportion</i>
------------------	------------------------------------

---

## Description

Obtains the TDS significance proportion based on the number of observations and chance, as proposed by Pineau et al. (2009; Eq. 1).

## Usage

```
get.significance(chance, n, alpha = 0.05)
```

## Arguments

chance	chance proportion; see <a href="#">get.chance</a> .
n	number of observations.
alpha	significance level for binomial test of 2 independent proportions (based on normal approximation; see: Pineau et al., 2009, Eq. 1)

## Details

The TDS significance level proposed by Pineau et al. (2009, Eq. 1) provides a simple and widely used heuristic approach for contextualizing observed dominance rates, but should not be used for statistical inference.

## References

Pineau, N., Schlich, P., Cordelle, S., Mathonnière, C., Issanchou, S., Imbert, A., Rogeaux, M., Etiévant, P., & Köster, E. (2009). Temporal dominance of sensations: Construction of the TDS curves and comparison with time–intensity. *Food Quality and Preference*, 20, 450–455. <http://dx.doi.org/10.1016/j.foodqual.2009.04.005>

## Examples

```
# example using 'bars' data set
attributes <- unique(bars$attribute)
chance <- get.chance(attributes)
signif <- get.significance(chance, nrow(unique(bars[, 1:2])))
signif
```

---

get.significance.diff *Get least significant differences for pairwise comparisons*

---

## Description

Get least significant differences for pairwise comparisons (see Pineau et al., 2009, Eq. 2).

## Usage

```
get.significance.diff(x, y, alpha = 0.05)
```

## Arguments

x	matrix of dominance data (0/1) related to one entity
y	matrix of dominance data (0/1) related to another entity
alpha	significance for one-sided test (default 0.05)

## Details

Calculation of least significant differences for TDS difference curves based on Pineau et al. (2009, Eq. 2). The absolute value of the observed dominance rate for a give attribute\*time must exceed the corresponding least significant difference calculated here to be considered significant.

## Value

out least significant difference (at level alpha) for dominance differences in matrix

## References

Pineau, N., Schlich, P., Cordelle, S., Mathonnière, C., Issanchou, S., Imbert, A., Rogeaux, M., Etiévant, P., & Köster, E. (2009). Temporal dominance of sensations: Construction of the TDS curves and comparison with time–intensity. *Food Quality and Preference*, 20, 450–455. <http://dx.doi.org/10.1016/j.foodqual.2009.04.005>

**Examples**

```

# toy data example
x <- data.frame(t10 = c(rep(NA, 15), rep(0, 50), rep(1, 20)),
               t15 = c(rep(NA, 4), rep(0, 61), rep(1, 20)),
               t20 = c(rep(0, 55), rep(1, 30)))
y <- data.frame(t10 = c(rep(NA, 15), rep(0, 50), rep(1, 20)),
               t15 = c(rep(NA, 0), rep(0, 21), rep(1, 64)),
               t20 = c(rep(0, 35), rep(1, 50)))

signif.xy <- get.significance.diff(x, y)
#compare with observed differences
diff.xy <- get.differences(x, y)
abs(diff.xy) > signif.xy

# real data example - differences between Bar 1 and Bar 2 on the attribute "Grain Flavour"
attributes <- unique(bars$attribute)
times <- get.times(colnames(bars)[-c(1:4)])
bar1 <- bars[bars$sample == 1 & bars$attribute == "Grain Flavour", -c(1:4)]
bar2 <- bars[bars$sample == 2 & bars$attribute == "Grain Flavour", -c(1:4)]
signif.1vs2 <- get.significance.diff(bar1, bar2)
# review observed difference in dominance rates vs. least significant differences
diff.1vs2 <- get.differences(bar1, bar2)
abs(diff.1vs2) > signif.1vs2
# differences between samples start at 1.1s and occur throughout the 45.0 evaluation period

```

get.smooth

*Convenience function for curve smoothing***Description**

Smooth TCATA curves, constraining smooth within low.bound and up.bound.

**Usage**

```
get.smooth(y, w = NULL, spar = 0.5, low.bound = 0, up.bound = 1)
```

**Arguments**

y	the vector of proportions (or counts) to be smoothed. If a data frame is provided then smoothing is conducted on each row.
w	an optional vector of weights; see <a href="#">smooth.spline</a>
spar	smoothing parameter; see <a href="#">smooth.spline</a>
low.bound	lower bound for smoothed proportions
up.bound	upper bound for smoothed proportions

**Value**

out smoothed vector (or data frame with smoothed rows)

**References**

Castura, J.C., Antúnez, L., Giménez, A., Ares, G. (2016). Temporal check-all-that-apply (TCATA): A novel temporal sensory method for characterizing products. *Food Quality and Preference*, 47, 79-90. <http://dx.doi.org/10.1016/j.foodqual.2015.06.017>

**See Also**

[smooth.spline](#), [predict](#)

**Examples**

```
# example using 'syrah' data set
low1 <- t(syrah[seq(3, 1026, by = 6), -c(1:4)])
colnames(low1) <- 10:180
x <- get.smooth(low1)
round(x, 3)
```

---

get.times

*Get times*

---

**Description**

Convenience function to convert exported time labels, e.g. from character format c('time\_0.1s', 'time\_0.2s', ...) or related format to numeric format c(0.1, 0.2, ...).

**Usage**

```
get.times(time.char, trim.left = "time_", trim.right = "s")
```

**Arguments**

time.char	vector of characters containing the time
trim.left	string to be trimmed from left
trim.right	string to be trimmed from right

**Details**

Convenience function for getting times from column headers from common data export formats.

**Value**

times vector of times in numeric format

**Examples**

```
get.times(colnames(bars)[-c(1:4)])

(sample.colnames <- paste0("X", 0:30))
get.times(sample.colnames, trim.left = "X", trim.right = "")
```

---

lengthwhichis.na	<i>Count observations with missing data</i>
------------------	---

---

**Description**

Count observations with missing data.

**Usage**

```
lengthwhichis.na(x)
```

**Arguments**

x                      vector data which may contain missings

**Value**

count of observations where data are missing

**Examples**

```
x <- c(rep(NA,18), rep(1,18), rep(0,10), rep(NA, 10))
lengthwhichis.na(x)
```

---

make.palettes	<i>Convenience function for getting a pretty palette and highlight colours</i>
---------------	--

---

**Description**

Make a vector of n pretty colours, and n matching highlight colours.

**Usage**

```
make.palettes(n)
```

**Arguments**

n                      number of colours for each palette

**Value**

pal A character vector, cv, of colours that look pretty.

pal.light A character vector, cv, of matching highlight colours that look pretty.

**Examples**

```
make.palettes(8)
```

---

plot\_pca.trajectories *Plot trajectories based on Temporal Check-All-That-Apply (TCATA) data*

---

### Description

Plot trajectories following PCA on multiblock TCATA proportions, or same for Temporal Dominance of Sensations (TDS) proportions.

### Usage

```
plot_pca.trajectories(in.pca = in.pca, products.times = matrix(NA),
  attributes = c(), type = "smooth", span = 0.75, biplot = "distance",
  flip = c(FALSE, FALSE), dims = c(1, 2),
  att.offset.x = c(), att.offset.y = c(), inflate.factor = NA,
  xlab = "_auto_", ylab = "_auto_", xlim = NULL, ylim = NULL,
  attributes.col = "red", attributes.pch = 17,
  lwd = 1, traj.lab.loc = 0, traj.col = c(grDevices::grey(1/2)), traj.points = NA,
  traj.col.seg = NA, traj.cex = 1, traj.lab = c(), traj.lab.cex = 1,
  arrow.loc = NA, arrow.length = 0.1, arrow.col = NA, arrow.lwd = NA,
  contrails = list(), main = "", save.format = "eps", save.as = "")
```

### Arguments

in.pca	Any list object with components sdev, rotation, and x. Most often it is a prcomp object obtained from PCA on a matrix of proportions (or, if there is no missing data, on counts) with Product*Times in rows and Attributes in columns.
products.times	a 2-column matrix, with an ascending sort order on products (column 1) and a secondary ascending sort on times (column 2), corresponding to the rows of the matrix submitted to prcomp to obtain "in.pca".
attributes	a vector of attribute labels, corresponding to the attributes of the matrix submitted to prcomp to obtain "in.pca".
type	Determines how trajectories are drawn. Possible values are "smooth" (default) or "raw".
span	A tuning parameter used if smoothing trajectories using the loess function.
biplot	Controls the type of biplot displayed. Possible values are "distance" (plots trajectories based on scores, and attributes based on eigenvectors multiplied by the "inflation.factor"), or "correlation" (plots trajectories based on scores divided by the sqrt of their respective eigenvalues, and attributes based on eigenvectors multiplied by the sqrt of their respective eigenvalues).
flip	a vector of two logical values. Value indicates whether to mirror the coordinates in the x and y dimensions respectively. Default is c(FALSE, FALSE).
dims	a vector of two integers, specifying the principal compnts to display. Defaults is "c(1,2)", i.e. PC1 vs. PC2.



att.offset.x	A vector of numeric values corresponding to the labels in "attributes". Used to adjust the horizontal position of attribute labels to make the plot more readable.
att.offset.y	A vector of numeric values corresponding to the labels in "attributes". Used to adjust the vertical position of attribute labels to make the plot more readable.
inflate.factor	Scalar controlling the position of attribute labels. If "NA" (default), then this scalar is set to the largest absolute score divided by the largest absolute eigenvector based on the dimensions used. Use "1" for no inflation. Applies only when biplot = "distance".
xlab	Label for x axis.
ylab	Label for y axis.
xlim	Permits control of the x limit. Limits can be specified using a vector of 2 (ascending) numbers. If a single number is provided then values are selected such that the limits are 20% beyond the smallest and largest x coordinates, respectively. If unspecified then control over x axis limits is given to the plot function in R.
ylim	Permits control of the y limit using the same logic as is used for "xlim".
attributes.col	Color used to display attribute labels (see "attributes").
attributes.pch	Symbol for attribute coordinates.
lwd	Trajectory line width.
traj.lab.loc	Indicates where along the trajectory the trajectory label will be positioned. "1" indicates the start of the trajectory. The value "0" (default) is a special convention indicating the end of the trajectory.
traj.col	A vector of colors for trajectories. If not specified then all trajectories are shown in grey.
traj.points	Specifies the position of markers along smoothed trajectories, and used to indicate the progression of time.
traj.col.seg	A vector of colors for segments along trajectories. If NA (default) then no segments along the trajectories appear in a color other than those specified by "traj.col". This parameter applies to smoothed trajectories only.
traj.cex	Used with "traj.points" for smoothed trajectories. Controls the size of symbol displayed.
traj.lab	A vector of character labels that identify the trajectories. If unspecified, then products are identified by ascending natural numbers.
traj.lab.cex	Text size of traj.lab.
arrow.loc	Trajectory arrows locations for direction marker(s).
arrow.length	Trajectory arrows length. See length parameter in <a href="#">arrows</a> .
arrow.col	Trajectory arrows color. See col parameter in <a href="#">arrows</a> .
arrow.lwd	Trajectory arrows line width. See lwd parameter in <a href="#">arrows</a> .
contrails	list of data.frame objects with columns x, y, count, col; x and y are coordinates, count is the number of values at the coordinate, and col is the rgb colour.
main	plot title; see <a href="#">plot</a> .

save.format	If indicated, this will be the file type for the save image. Defaults to "eps" (eps format). Other possible values are "" (not saved) or "png" (png format).
save.as	The filename. Must be provided if the file will be saved.

## References

Castura, J.C., Antúnez, L., Giménez, A., Ares, G. (2016). Temporal check-all-that-apply (TCATA): A novel temporal sensory method for characterizing products. *Food Quality and Preference*, 47, 79-90. <http://dx.doi.org/10.1016/j.foodqual.2015.06.017>

Castura, J.C., Baker, A.K., & Ross, C.F. (2016). Using contrails and animated sequences to visualize uncertainty in dynamic sensory profiles obtained from temporal check-all-that-apply (TCATA) data. *Food Quality and Preference*, 54, 90-100. <http://dx.doi.org/10.1016/j.foodqual.2016.06.011>

## See Also

[prcomp](#), [par](#)

## Examples

```
# example using 'syrah' data set
syrah.pca <- prcomp(syrah[1:248, -c(1:4)], scale. = FALSE)
plot_pca.trajectories(syrah.pca, products.times = syrah[1:248, c(1, 4)],
                     attributes = colnames(syrah)[-c(1:4)], type = "raw")

# now with smoothing; may need to play with the span parameter to get appropriate smoothing
plot_pca.trajectories(syrah.pca, products.times = syrah[1:248, c(1, 4)],
                     attributes = colnames(syrah)[-c(1:4)], type = "smooth", span = 0.3)

# plots at each time point (trajectories join 2 points so start at timepoint 2, i.e., 11 s)
x <- 11:14 # for brevity show only the first 4 timeslices
# x <- 11:41 # uncomment this line to to run a longer demo
pca.list <- list()
for(i in seq_along(x)){
  pca.list[[x[i]-10]] <- syrah.pca
  pca.list[[x[i]-10]]$x <- pca.list[[x[i]-10]]$x[1:((x[i]-9)*6), ]
  plot_pca.trajectories(pca.list[[x[i]-10]], products.times = syrah[1:((x[i]-9)*6), c(1, 4)],
                       attributes = colnames(syrah)[-c(1:4)], type = "raw", inflate.factor = 1.5)
  Sys.sleep(3/4)
  # save plot if saving stills for a video; see Castura, Baker, & Ross (2016, Video 1)
}
```

---

```
pretty_palette
```

```
Get a pretty palette of colours
```

---

## Description

Create a vector of n pretty colours.

**Usage**

```
pretty_palette(n)
```

**Arguments**

n                    number of colours in the palette

**Value**

cv A character vector, cv, of colours that look pretty.

**Examples**

```
pretty_palette(8)
```

---

```
similarity.tcata.repeatability
```

*Quantify TCATA assessor repeatability*

---

**Description**

Quantify TCATA assessor repeatability using city block distance

**Usage**

```
similarity.tcata.repeatability(X)
```

**Arguments**

X                    list of matrices, where each matrix is a TCATA data (given as an indicator matrix) for assessor of interest for one rep

**Details**

Similarity between repeated evaluations given by a TCATA assessor is quantified. The repeatability index can take on values between 0 and 1, which indicate complete dissimilarity (non-repeatability) and complete similarity (repeatability), respectively.

**Value**

repeatability.index average city block distance between matrices from replicated evaluations

**References**

Castura, J.C., Antúnez, L., Giménez, A., Ares, G. (2016). Temporal check-all-that-apply (TCATA): A novel temporal sensory method for characterizing products. *Food Quality and Preference*, 47, 79-90. <http://dx.doi.org/10.1016/j.foodqual.2015.06.017>

**Examples**

```
# Toy data from one TCATA assessor on a product over three sessions: rep1, rep2, rep3
rep1 <- rbind(rep(0, 7),
             rep(0, 7),
             c(0, 0, 0, 1, 1, 1, 1),
             c(0, 0, 0, 1, 1, 1, 1),
             c(0, 0, 0, 1, 1, 1, 0))
rep2 <- rbind(c(0, 0, 0, 1, 1, 1, 0),
             rep(0, 7),
             c(0, 1, 1, 1, 1, 1, 0),
             rep(1, 7),
             c(0, 0, 0, 1, 1, 1, 1))
rep3 <- rbind(rep(0, 7),
             rep(0, 7),
             rep(1, 7),
             rep(1, 7),
             rep(1, 7))
rep.data <- list(rep1, rep2, rep3)

# Quantify similarity of assessor a1 to the other assessors
similarity.tcata.repeatability(rep.data)
```

---

similarity.tcata.replication

*Quantify TCATA assessor replication*

---

**Description**

Quantify TCATA assessor replication using city block distance

**Usage**

```
similarity.tcata.replication(this.assessor, other.assessors)
```

**Arguments**

`this.assessor` TCATA data (given as an indicator matrix) for assessor of interest

`other.assessors`

TCATA data (given as an indicator matrix) for other assessors

**Details**

Similarity between one TCATA assessor and other assessors on the panel is quantified. The replication index can take on values between 0 and 1, which indicate complete dissimilarity (disagreement) and complete similarity (agreement), respectively.

**Value**

replication.index city block distance between this assessor and other assessors

## References

Castura, J.C., Antúnez, L., Giménez, A., Ares, G. (2016). Temporal check-all-that-apply (TCATA): A novel temporal sensory method for characterizing products. *Food Quality and Preference*, 47, 79-90. <http://dx.doi.org/10.1016/j.foodqual.2015.06.017>

## Examples

```
# Toy TCATA data for three assessors: a1, a2, a3
a1 <- rbind(rep(0, 7),
            rep(0, 7),
            c(0, 0, 0, 1, 1, 1, 1),
            c(0, 0, 0, 1, 1, 1, 1),
            c(0, 0, 0, 1, 1, 1, 0))
a2 <- rbind(c(0, 0, 0, 1, 1, 1, 0),
            rep(0, 7),
            c(0, 1, 1, 1, 1, 1, 0),
            rep(1, 7),
            c(0, 0, 0, 1, 1, 1, 1))
a3 <- rbind(rep(0, 7),
            rep(0, 7),
            rep(1, 7),
            rep(1, 7),
            rep(1, 7))

# Quantify similarity of assessor a1 to the other assessors
similarity.tcata.replication(a1, rbind(a2, a3))
```

---

std.time

*Time standardize results*

---

## Description

Set results for a temporal evaluation to a timescale by trimming off time prior to the first onset and following the last offset time, and express the remaining times in terms of percentiles [0, 100].

## Usage

```
std.time(X, trim.left = TRUE, trim.right = TRUE, scale = TRUE, missing = 0)
```

## Arguments

X	vector (or data frame) of indicator data.
trim.left	Trim on the left? Default is TRUE.
trim.right	Trim on the right? Default is TRUE.
scale	Set to a [0, 1] scale? Default is TRUE.
missing	indicator for missing data; default is 0.

**Value**

out vector (or data frame) of trimmed and/or standardized indicator (0/1) data

**References**

Lenfant, F., Loret, C., Pineau, N., Hartmann, C., & Martin, N. (2009). Perception of oral food breakdown. The concept of sensory trajectory. *Appetite*, 52, 659-667.

**Examples**

```
# vector - toy data example
x <- rep(c(rep(0,18), rep(1,18)), 2)
names(x) <- 1:72
x          # raw time
std.time(x) # standardized time

# data frame - toy data example
y <- data.frame(rbind(c(c(rep(0,18),
                        rep(1,18)),
                      rep(0, 4)),
                    c(rep(c(rep(0,9),
                            rep(1,9)), 2),
                      1, rep(0, 3)),
                  rep(0, 40)))

colnames(y) <- 1:40
y          # raw time
std.time(y) # standardized time

# time standardization using 'bars' data set
# only sample 1 will be done (for illustrative purposes)
eval1 <- unique(bars[bars$sample == 1, (1:3)])
bar1.std <- data.frame(unique(bars[bars$sample == 1, (1:4)]), matrix(0, ncol = 101))

for (e in 1:nrow(eval1)){
  bar1.std[bar1.std$assessor == eval1$assessor[e] &
           bar1.std$session == eval1$session[e] &
           bar1.std$sample == eval1$sample[e],
           -c(1:4)] <- std.time(bars[bars$assessor == eval1$assessor[e] &
                                   bars$session == eval1$session[e] &
                                   bars$sample == eval1$sample[e],
                                   -c(1:4)])
}
colnames(bar1.std)[5:ncol(bar1.std)] <- 0:100
head(bar1.std)
```

---

TCATA data set: orange juice

*TCATA data set: orange juice*

---

**Description**

Raw results from 20-s TCATA evaluations of six orange juice samples by 50 consumers.

**Format**

A data frame with 1800 rows (50 consumers \* 6 samples \* 6 attributes) and 25 columns (4 headers + 21 time slices)

- [, 1] cons (int) consumer id
- [, 2] samp (chr) sample id
- [, 3] samp\_pos (int) position of sample in serving order
- [, 4] attribute (chr) sensory attribute
- [, 5:25] time\_99s (int) value is 1 if attribute is selected at time slice; otherwise value is 0

**References**

Ares, G., Jaeger, S. R., Antúnez, L., Vidal, L., Giménez, A., Coste, B., Picallo, A., & Castura, J. C. (2016). Comparison of TCATA and TDS for dynamic sensory characterization of food products. *Food Research International*, 78, 148-158. <http://dx.doi.org/10.1016/j.foodres.2015.10.023>

**Examples**

```
head(ojtcata) # review first 6 rows of 'ojtcata' data set
```

---

TCATA data set: Syrah wines

*TCATA data set: Syrah wines*

---

**Description**

TCATA citation proportions for three wine treatments evaluated using a two-sip evaluation protocol.

**Format**

A data frame with 1026 rows (3 treatments \* 2 sips \* 171 time slices) and 13 columns:

- [, 1] WineSip (chr) Code for wine and sip
- [, 2] Wine (chr) Code for wine (H=high, L=low, A=adjusted)
- [, 3] Sip (int) Sip number
- [, 4] Time (int) Time, in seconds
- [, 5] Astringency (num) citation proportions
- [, 6] Bitter (num) citation proportions
- [, 7] Dark Fruit (num) citation proportions

- [, 8] Earthy (num) citation proportions
- [, 9] Green (num) citation proportions
- [,10] Heat (num) citation proportions
- [,11] Red Fruit (num) citation proportions
- [,12] Spices (num) citation proportions
- [,13] Sour (num) citation proportions

## References

Baker, A. K., Castura, J. C., & Ross, C. F. (2016). Temporal check-all-that-apply characterization of Syrah wine finish. *Journal of Food Science*, 81, S1521-S1529. <http://dx.doi.org/10.1111/1750-3841.13328>.

## Examples

```
head(syrah, 3) # review first 3 rows of 'syrah' data set
```

---

tcata.diff.plot	<i>TCATA difference plot</i>
-----------------	------------------------------

---

## Description

Plots TCATA difference curves.

## Usage

```
tcata.diff.plot(x1 = x1, x2 = NA, n1 = 1, n2 = NA,
  attributes = c(), times = c(), lwd = 1,
  declutter = NA, get.decluttered = FALSE, emphasis = NA, alpha = 0.05, emphasis.lwd = 3,
  main = "", height = 8, width = 12,
  xlab = "Time", ylab = "Difference in citation proportion",
  axes.font = 1, axes.cex = 1, line.col = c(), x.increment = 5,
  legend.cex = 1, legend.font = 1, save.as = "")
```

## Arguments

x1	matrix of difference proportions, or of counts if n1 specified. If mat2 specified then proportions or counts apply to first sample. Attributes are in rows, times in columns.
x2	matrix of proportions for second sample, or of counts if n2 specified.
n1	number of observations for first sample
n2	number of observations for second sample
attributes	vector of attribute labels for row in x1 (and x2)
times	vector of times for columns in x1 (and x2)



lwd	Line width
declutter	indicator matrix with same dimensions of x1 to suppress output
get.decluttered	if TRUE then calculates the declutter matrix from get.mat.diff.sign
emphasis	set to 1 to emphasize significant differences
alpha	significance level to use for entrywise test of x1 and x2 (if counts)
emphasis.lwd	line weight for emphasizing significant differences
main	plot title; see <a href="#">plot</a>
height	plot height
width	plot width
xlab	label for x axis
ylab	label for y axis
axes.font	Font for axes labels; see <a href="#">par</a> .
axes.cex	Size for axes labels.
line.col	line color for attribute lines
x.increment	increment between time labels on x axis
legend.cex	symbol size for legend
legend.font	Font for the legend; see <a href="#">text</a> .
save.as	Filename to use if file will be saved.

## References

Castura, J.C., Antúnez, L., Giménez, A., Ares, G. (2016). Temporal check-all-that-apply (TCATA): A novel temporal sensory method for characterizing products. *Food Quality and Preference*, 47, 79-90. <http://dx.doi.org/10.1016/j.foodqual.2015.06.017>

## Examples

```
# difference between High and Low ethanol wines (sip 1)
x.diff.raw <- t(syrah[seq(1, 1026, by = 6), -c(1:4)]) -
  t(syrah[seq(3, 1026, by = 6), -c(1:4)])
x.diff.smooth <- get.smooth(x.diff.raw, low.bound = -1, up.bound = 1)
colnames(x.diff.smooth) <- colnames(x.diff.raw) <- times <- 10:180
tcata.diff.plot(x1 = x.diff.smooth, attributes = rownames(x.diff.smooth), times = times, lwd = 2,
  main = "Sip 1 differences: High-ethanol wine - Low-ethanol wine")

# an example based on the syrah data set (truncated for efficiency)
n <- 52
H1 <- t(syrah[seq(1, 126, by = 6), -c(1:4)] * n)
L1 <- t(syrah[seq(3, 126, by = 6), -c(1:4)] * n)
colnames(H1) <- colnames(L1) <- times <- 10:30
tcata.diff.plot(x1 = H1, x2 = L1, n1 = n, n2 = n,
  attributes = rownames(H1), get.decluttered = TRUE, lwd = 2)
```

---

tcata.line.plot      *Temporal Check-All-That-Apply (TCATA) curve*

---

### Description

Plots TCATA curves based on count or proportion data. Can also be used for plotting Temporal Dominance of Sensations (TDS) curves based on dominance counts or proportions.

### Usage

```
tcata.line.plot(X, n = 1, attributes = c(), times = c(), lwd = 1,
emphasis = NA, emphasis.lty = 1, emphasis.lwd = 3, declutter = NA,
chance.lty = 3, reference = NA, ref.lty = 2, ref.lwd = 1,
highlight = FALSE, highlight.lty = 1, highlight.lwd = 5, highlight.col = c(),
height = 8, width = 12, main = "",
xlab = "Time", ylab = "Citation proportion", axes.font = 1,
axes.cex = 1, xlim = c(), las = 0, line.col = c(),
x.increment = 5, legend.cex = 1, legend.font = 1,
legend.pos = "topleft", legend.ncol = 2, box = FALSE,
save.format = "", save.as = "" )
```

### Arguments

X	matrix of proportions (or, if there is no missing data, on counts), typically with Attributes in rows and times in columns.
n	The number of observations if X is a count matrix. Keep n = 1 if X is a matrix of proportions.
attributes	a vector of attribute labels, corresponding to the attributes in X.
times	a vector of time, corresponding to the times in X.
lwd	Line width used when plotting attribute curves in X.
emphasis	a matrix matching X in its dimensions, with a numeric value corresponding to points requiring emphasis, and NA for points without emphasis.
emphasis.lty	Controls the line type associated with the emphasis line.
emphasis.lwd	Controls the line weight associated with the emphasis line.
declutter	a matrix matching X in its dimensions, with a numeric value corresponding to points that will not be plotted.
chance.lty	line type for the chance line. (May be deprecated?)
reference	a matrix matching X in its dimensions, with a numeric value corresponding to points where a reference line will be displayed, and NA for points where no reference line will be shown.
ref.lty	reference line type.
ref.lwd	reference line width.

highlight	a matrix matching X in its dimensions, with a numeric value corresponding to points requiring highlighting, and NA for points without highlighting
highlight.lty	Controls the line type associated with the highlighting.
highlight.lwd	Controls the line weight associated with the highlighting line.
highlight.col	a vector of colours for attributes corresponding to rows of X.
height	Window height.
width	Window width.
main	plot title; see <a href="#">plot</a> .
xlab	Label for the x axis.
ylab	Label for the y axis.
axes.font	Font for axes labels; see <a href="#">par</a> .
axes.cex	Size for axes labels.
xlim	Permits control of the x limit. Limits can be specified using a vector of 2 (ascending) numbers.
las	numeric in 0,1,2,3; the style of the axis labels. See <a href="#">par</a> .
line.col	A vector of colors for lines corresponding to rows of X.
x.increment	Specifies the interval between times when labelling the x axis.
legend.cex	Size of markers shown in the legend.
legend.font	Font for the legend; see <a href="#">text</a> .
legend.pos	Location of the legend in the plot; defaults to "topleft".
legend.ncol	Number of columns in legend.
box	draw box around plot area; see: <a href="#">box</a>
save.format	If indicated, this will be the file type for the save image. Defaults to "eps" (eps format). Other possible values are "" (not saved) or "png" (png format).
save.as	Filename if the file will be saved.

## References

Castura, J.C., Antúnez, L., Giménez, A., Ares, G. (2016). Temporal check-all-that-apply (TCATA): A novel temporal sensory method for characterizing products. *Food Quality and Preference*, 47, 79-90. <http://dx.doi.org/10.1016/j.foodqual.2015.06.017>

## Examples

```
# example using 'syrah' data set
low1 <- t(syrah[seq(3, 1026, by = 6), -c(1:4)])
colnames(low1) <- 10:180
tcata.line.plot(get.smooth(low1), lwd = 2, main = "Low-ethanol wine (Sip 1)")

# example using 'ojtcata' data set
data(ojtcata)
x <- aggregate(ojtcata[, -c(1:4)], list(samp = ojtcata$samp, attribute = ojtcata$attribute), sum)
p.1.checked <- x[x$samp == 1, -c(1:2)]
```

```

p.1.eval <- length(unique(ojtcata$cons))
p.not1.checked <- aggregate(x[, -c(1:2)], list(attribute = x$attribute), sum)[, -1]
p.not1.eval <- length(unique(ojtcata$cons)) * (length(unique(ojtcata$samp)) - 1)

# reference lines for contrast products
p.1.refline <- p.not1.checked / p.not1.eval
# decluttering matrix corresponds to the dimensions of p.1.refline
p.1.declutter <- matrix(get.decluttered(x = unlist(p.1.checked), n.x = p.1.eval,
                                     y = unlist(p.not1.checked), n.y = p.not1.eval),
                       nrow = nrow(p.1.checked))
times <- get.times(colnames(x)[-c(1:2)])
attributes <- unique(x$attribute)
palettes <- make.palettes(length(attributes))
tcata.line.plot(p.1.checked, n = p.1.eval, attributes = attributes, times = times,
               reference = p.1.refline, ref.lty = 3, declutter = p.1.declutter,
               highlight = TRUE, highlight.lwd = 4,
               line.col = palettes$pal, highlight.col = palettes$pal.light,
               main = "Sample 1", height = 7, width = 11, legend.cex = 0.7)

```

---

TDS data set: orange juice

*TDS data set: orange juice*

---

## Description

Raw results from 20-s TDS evaluations of six orange juice samples by 50 consumers.

## Format

A data frame with 1800 rows (50 consumers \* 6 samples \* 6 attributes) and 25 columns (4 headers + 21 time slices)

- [, 1] cons (int) consumer id
- [, 2] samp (chr) sample id
- [, 3] samp\_pos (int) position of sample in serving order
- [, 4] attribute (chr) sensory attribute
- [, 5:25] time\_99s (int) value is 1 if attribute is selected at time slice; otherwise value is 0

## References

Ares, G., Jaeger, S. R., Antúnez, L., Vidal, L., Giménez, A., Coste, B., Picallo, A., & Castura, J. C. (2016). Comparison of TCATA and TDS for dynamic sensory characterization of food products. *Food Research International*, 78, 148-158. <http://dx.doi.org/10.1016/j.foodres.2015.10.023>

## Examples

```
head(ojtds) # review first 6 rows of 'ojtds' data set
```

---

TDS data set: snack bars

*TDS data set: snack bars*

---

### Description

Raw TDS results from 24 assessors who evaluated four snack bars in triplicate.

### Format

A data frame with 1440 rows (24 assessors \* 3 sessions \* 4 samples \* 5 attributes) and 455 columns (4 header rows + 451 time slices)

- [,1] assessor (chr) assessor id
- [,2] session (chr) session id
- [,3] sample (chr) sample id
- [,4] attribute (chr) sensory attribute
- [,5:455] time\_99.9s (chr) value is 1 if attribute is dominant at time slice; otherwise value is 0

### References

Findlay, C. J., Castura, J. C., & Valeriote, E. (2014). Temporal methods: A comparative study of four different techniques. In *17th IUFoST Congress*. 17-21 August. Montréal, Québec, Canada.

### Examples

```
head(bars, 2) # review first 2 rows of 'bars' data set
```

---

```
tds.diff.plot
```

*Plot TDS difference curves*

---

### Description

Plots TDS difference curves based on differences in dominance counts or dominance rates.

### Usage

```
tds.diff.plot(  
  X,  
  times = NULL,  
  attributes = NULL,  
  xlab = "Time (seconds)",  
  ylab = "Dominance rate",  
  line.col = 1,  
  lty = 1,  
  lwd = 1,  
  main = ""  
)
```

## Arguments

X	matrix of differences in dominance rates (Attributes in rows, Times in columns).
times	a vector of times, corresponding to the times in X.
attributes	a vector of attribute labels, corresponding to the attributes in X.
xlab, ylab	Labels for the x and y axes; see <code>plot</code>
line.col	A vector of colors for lines corresponding to attributes; see <code>par</code>
lty, lwd	line type and weight for attributes; see <code>par</code>
main	plot title; see <code>plot</code>

## Details

Currently the differences in dominance rates are always displayed. Suppression of differences in dominances rates within a threshold range is not yet implemented.

## References

Pineau, N., Schlich, P., Cordelle, S., Mathonnière, C., Issanchou, S., Imbert, A., Rogeaux, M., Etiévant, P., & Köster, E. (2009). Temporal dominance of sensations: Construction of the TDS curves and comparison with time–intensity. *Food Quality and Preference*, 20, 450–455. <http://dx.doi.org/10.1016/j.foodqual.2009.04.005>

## Examples

```
# example using 'bars' data set
bars.m <- aggregate(bars[, -c(1:4)], list(samples = bars$sample, attribute = bars$attribute), mean)
bars.m <- bars.m[order(bars.m$sample, bars.m$attribute), ]
attributes <- unique(bars$attribute)
times <- get.times(colnames(bars.m)[-c(1:2)])
bar1 <- bars.m[bars.m$sample == 1, -c(1:2)]
bar2 <- bars.m[bars.m$sample == 2, -c(1:2)]
diff.1vs2 <- get.smooth(bar1 - bar2, low.bound = -1, up.bound = 1)
tds.diff.plot(diff.1vs2, times = times, attributes = attributes,
              lwd = 2, main = "TDS Differences (Bar 1 - Bar 2)")

# suppose we only want to show the curves where the difference in dominance rate
# is significantly different
# get samples sizes and dominance counts for each product
bars.s <- aggregate(bars[, -c(1:4)], list(samples = bars$sample, attribute = bars$attribute), sum)
bars.s <- bars.s[order(bars.s$sample, bars.s$attribute), ]
bar1.s <- bars.s[bars.s$sample == 1, -c(1:2)]
bar2.s <- bars.s[bars.s$sample == 2, -c(1:2)]
bar1.n <- nrow(unique(bars[bars$sample == 1, 1:2]))
bar2.n <- nrow(unique(bars[bars$sample == 2, 1:2]))

# prop.test2 is a wrapper for prop.test (with its default parameters)
# thus it will run chi-squared test with Yates continuity correction
prop.test2 <- function(x1, x2, n1, n2, alpha = 0.05){
  return((suppressWarnings(prop.test(c(x1,x2), c(n1, n2),
    alternative = "two.sided")))$p.value < alpha) %in% TRUE)
```

```

}
# find significant pairwise comparison, treating data as if independent
diff_1v2.signif <- mapply(prop.test2, unlist(bar1.s), unlist(bar2.s), bar1.n, bar2.n)
# update smoothed difference matrix with NA where non-significant pairs
diff_1v2.signif[!diff_1v2.signif] <- NA
diff.1vs2 <- diff.1vs2 + diff_1v2.signif - 1
# line segments that are non-significant are missing/NA so not plotted
tds.diff.plot(diff.1vs2, times = times, attributes = attributes,
              lwd = 2, main = "TDS Differences (Bar 1 - Bar 2)")

```

tds.plot

*Plot TDS curves***Description**

Plots TDS curves based on dominance rates, showing chance and significance lines.

**Usage**

```

tds.plot(X, attributes = NULL, times = NULL, chance = NULL, signif = NULL,
         line.col = 1, lty = 1, lwd = 1, las = 0, xlab = "Time (seconds)",
         ylab = "Dominance rate", main = "", height = 8, width = 12, box = FALSE, save.as = "")

```

**Arguments**

X	matrix of dominance rates (Attributes in rows, Times in columns).
attributes	a vector of attribute labels, corresponding to the attributes in X.
times	a vector of times, corresponding to the times in X.
chance	proportion indicating the chance level, usually 1/length(attributes) or 1/(1+length(attributes)).
signif	significance level associated with the number of observations and chance.
line.col	A vector of colors for lines corresponding to attributes; see <a href="#">par</a>
lty, lwd	line type and weight for attributes; see <a href="#">par</a>
las	numeric in 0,1,2,3; the style of the axis labels. See: <a href="#">par</a>
xlab, ylab	Labels for the x and y axes; see <a href="#">plot</a>
main	plot title; see <a href="#">plot</a>
height	Window height.
width	Window width.
box	draw box around plot area; see: <a href="#">box</a>
save.as	Filename if the file will be saved.

**References**

Pineau, N., Schlich, P., Cordelle, S., Mathonnière, C., Issanchou, S., Imbert, A., Rogeaux, M., Etiévant, P., & Köster, E. (2009). Temporal dominance of sensations: Construction of the TDS curves and comparison with time–intensity. *Food Quality and Preference*, 20, 450–455. <http://dx.doi.org/10.1016/j.foodqual.2009.04.005>

**Examples**

```

# example using 'bars' data set
bars.m <- aggregate(bars[, -c(1:4)], list(samples = bars$sample, attribute = bars$attribute), mean)
bars.m <- bars.m[order(bars.m$sample, bars.m$attribute), ]
attributes <- unique(bars$attribute)
times <- get.times(colnames(bars.m)[-c(1:2)])
chance <- get.chance(attributes)
signif <- get.significance(chance, nrow(unique(bars[, 1:2])))
tds.plot(get.smooth(bars.m[bars.m$sample == 1, -c(1:2)]), attributes = attributes,
         times = times, chance = chance, signif = signif,
         lwd = 2, main = "Bar 1")

# it is possible to hide the portion of the plot below the significance line:
rect(-2, -0.2, times[length(times)]+2, signif, col = "white", border = "transparent")
# re-add axes & significance line
axis(1, labels = seq(0, 45, by = 5), at = seq(0, 45, by = 5))
axis(2)
abline(h=signif, lty=3, col = "grey")

```

---

tempR

*tempR*


---

**Description**

Analysis and visualization of data from temporal sensory methods, including for temporal check-all-that-apply (TCATA) and temporal dominance of sensations (TDS).



# Index

## \*Topic **TCATA**

TCATA data set: orange juice, [22](#)

TCATA data set: Syrah wines, [23](#)

## \*Topic **TDS**

TDS data set: orange juice, [28](#)

TDS data set: snack bars, [29](#)

## \*Topic **datasets**

TCATA data set: orange juice, [22](#)

TCATA data set: Syrah wines, [23](#)

TDS data set: orange juice, [28](#)

TDS data set: snack bars, [29](#)

`adjust.brightness`, [2](#)

`arrows`, [17](#)

`bars` (TDS data set: snack bars), [29](#)

`bootstrap.band`, [3](#)

`box`, [27](#), [31](#)

`convert.tcata`, [3](#)

`convert.tcategory`, [4](#)

`count.selections`, [5](#)

`dist.city.block`, [5](#)

`fill.gaps`, [6](#)

`fisher.test`, [8](#), [11](#)

`get.chance`, [7](#), [11](#)

`get.decluttered`, [7](#)

`get.differences`, [8](#)

`get.dominance.rates`, [9](#)

`get.mat.diff.sign`, [10](#)

`get.significance`, [11](#)

`get.significance.diff`, [12](#)

`get.smooth`, [13](#)

`get.times`, [14](#)

`lengthwhichis.na`, [15](#)

`make.palettes`, [15](#)

`ojtcata` (TCATA data set: orange juice),  
[22](#)

`ojtds` (TDS data set: orange juice), [28](#)

`par`, [18](#), [25](#), [27](#), [30](#), [31](#)

`plot`, [17](#), [25](#), [27](#), [30](#), [31](#)

`plot_pca.trajectories`, [16](#)

`prcomp`, [18](#)

`predict`, [14](#)

`pretty_palette`, [18](#)

`similarity.tcata.repeatability`, [19](#)

`similarity.tcata.replication`, [20](#)

`smooth.spline`, [13](#), [14](#)

`std.time`, [21](#)

`syrah` (TCATA data set: Syrah wines), [23](#)

TCATA data set: orange juice, [22](#)

TCATA data set: Syrah wines, [23](#)

`tcata.diff.plot`, [24](#)

`tcata.line.plot`, [26](#)

TDS data set: orange juice, [28](#)

TDS data set: snack bars, [29](#)

`tds.diff.plot`, [29](#)

`tds.plot`, [31](#)

`tempR`, [32](#)

`text`, [25](#), [27](#)