

# Package ‘VizOR’

July 2, 2014

**Type** Package

**Title** Graphical visualization tools for complex observational data  
with focus on health sciences (VizOR)

**Version** 0.7-9

**Date** 2013-11-14

**Maintainer** David C. Norris <david@UnusualSolutionsThatWork.com>

**Copyright** 2010,2011,2012,2013 Genentech, Inc.

**Description** Provides individual- and aggregate-level graphical depictions of  
patterns of treatment and response in patient registries, and a graphical  
tool for examining potential for confounding in analyses of observational data.

**License** GPL (>= 2)

**LazyLoad** yes

**Depends** lattice, grid,

**Imports** rms

**Suggests** rpanel

**Author** Drew Griffin Levy [aut],David C. Norris [aut, cre]

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-11-15 00:41:22

## R topics documented:

VizOR-package . . . . .	2
alphanize . . . . .	3
color . . . . .	4
color.colored . . . . .	4
colored . . . . .	5
controller . . . . .	6
droplevels.colored . . . . .	6
fastback . . . . .	7
key . . . . .	8
key.colored . . . . .	8
plotbraids . . . . .	9
predReg . . . . .	10
prepanel.prevalence . . . . .	12
radarplot . . . . .	12
relevel.colored . . . . .	14
remvar . . . . .	15
representNA . . . . .	16
representNA.data.frame . . . . .	17
timeline . . . . .	17
vlbw . . . . .	19
[.colored . . . . .	20
<b>Index</b>	<b>21</b>

---

VizOR-package	<i>Graphical visualization tools for complex observational data (VizOR). Package functionality includes individual- and aggregate-level graphics depicting patterns of treatment and response in patient registries, and a graphical tool for identifying potential for confounding in analyses of observational data.</i>
---------------	--

---

## Description

```

Package:  VizOR
Type:    Package
Version:  0.7-9
Date:    2013-11-14
License:  GPL (>= 2)
LazyLoad: yes

```

TODO: Provide a general introduction to the package, including an orientation to its most important functions.

**Author(s)**

Drew Griffin Levy, PhD  
Senior Medical Science Director  
Genentech, Inc.  
Evidence Science & Innovation | US Medical Affairs  
<levy.drew@gene.com>

David C. Norris, MD  
David Norris Consulting, LLC  
<david@UnusualSolutionsThatWork.com>

**See Also**

[Hmisc rms](#)

---

alpatize

*Apply alpha transparency to a color vector*

---

**Description**

Apply alpha transparency to a color vector

**Usage**

```
alpatize(color, alpha)
```

**Arguments**

color	A vector of colors
alpha	A vector of alpha values, recycled as necessary

**Value**

A vector of colors, with alpha applied

**Author(s)**

David C. Norris

---

color	<i>Generic function color</i>
-------	-------------------------------

---

**Description**

This generic function supports extraction of colors from colored objects

**Usage**

```
color(x, ...)
```

**Arguments**

x	A colored object
...	Additional arguments

**Value**

A vector of colors

---

color.colored	<i>Returns the vector of colors associated with the factor levels.</i>
---------------	--

---

**Description**

Returns the vector of colors associated with the factor levels.

**Usage**

```
## S3 method for class 'colored'  
color(x, ...)
```

**Arguments**

x	A colored factor
...	Additional arguments (unused)

**Value**

A (character) vector of colors, having the same length as x

**Author(s)**

David C. Norris

**See Also**

[colored](#)

---

colored	<i>Create a colored factor, an object of class <code>c('colored', 'factor')</code> or <code>c('colored', 'ordered')</code>, from a character vector or factor <code>x</code>.</i>
---------	---

---

## Description

Colored factors permit persistent association of color to a factor, supporting consistent graphical treatment across multiple plots.

## Usage

```
colored(x, color.key, ordered = is.ordered(x),
        default = NA)
```

## Arguments

<code>x</code>	A character vector or factor
<code>color.key</code>	A named vector or list mapping factor levels (the names) to colors (the values). If missing (as is likely in initial exploratory analyses), this is generated automatically as a convenience to the user. In case an ordered factor is generated, the order of the levels is determined by the ordering in <code>color.key</code> .
<code>ordered</code>	A logical value stating whether the factor should be ordered
<code>default</code>	An optional argument; when provided, it gives a string in <code>names(color.key)</code> to which all non-matching values of <code>x</code> should be mapped. The effect will be to collect possibly many factor levels under a single key entry. A common choice might be, for example, <code>default='OTHER'</code> .

## Value

An object of class `c('colored', ['ordered', ]'factor')`

## Author(s)

David C. Norris

## Examples

```
# This is an example of de novo construction of a colored factor
weekdays <- colored(c("Mon", "Tue", "Wed", "Thu", "Fri"),
                    color.key=c(Mon="blue", Tue="red", Wed="yellow",
                                Thu="purple", Fri="green"))

# This demonstrates how one might use the 'colored' constructor
# to expand the level set of an existing factor.
week <- colored(weekdays,
                color.key=c(Sun="white", key(weekdays), Sat="gray"),
                ordered=TRUE)
```

```
# Note that 'droplevels.factor' works fine on colored factors
levels(week)
levels(droplevels(week))
```

---

controller	<i>Provides a GUI panel offering dynamic control of the adjust-to settings of the partial effects plot of an rms fit object.</i>
------------	--

---

### Description

The parameters are identified automatically by inspection of the fit object, and widgets appropriate to their data types are chosen automatically.

### Usage

```
controller(fit, datadist = NULL)
```

### Arguments

fit	The rms fit object to be visualized
datadist	Optionally, a datadist for the fitted object may be provided, conveniently enabling use of this function with models in which logical covariates are not typed as such, but represented in 0,1, or in which covariates more properly coded as ordered factors are represented instead as a finite number of integer values.

### Value

A controller panel is popped up on the screen, along with a partial effects plot for the fit object. No value is returned.

### Author(s)

David C. Norris

---

droplevels.colored	<i>Drop unused levels from a colored factor</i>
--------------------	---

---

### Description

Drop unused levels from a colored factor and its color key

### Usage

```
## S3 method for class 'colored'
droplevels(x, ...)
```

**Arguments**

x                    a colored factor  
...                  further arguments passed to methods

**Details**

This subclass method extends `droplevels.factor` to handle updating of the special color key attribute of colored factors.

**Value**

a colored factor with no unused levels

**Author(s)**

David C. Norris

---

fastback                    *fastback*

---

**Description**

This function serves to ‘complete’ `rms:fastbw`, such that it converts one fitted model to another, the restricted model.

**Usage**

```
fastback(fit, data, ...)
```

**Arguments**

fit                    A model fit of class `rms`  
data                   The data against which `fit` was estimated  
...                    Other parameters to be passed to `fastbw`

**Value**

A fitted model of the same type as `fit`, with regressors chosen by stepwise backward regression

**Author(s)**

David C. Norris

---

key	<i>Generic function 'key'</i>
-----	-------------------------------

---

**Description**

This generic function supports extraction of color keys from colored factors, via the specific method `key.colored`

**Usage**

```
key(x, ...)
```

**Arguments**

x	A colored object
...	Additional arguments

**Value**

A vector of colors

---

key.colored	<i>Extract the color key from a colored factor</i>
-------------	--

---

**Description**

Return a colored factor's key, mapping its levels to their colors. Returns a named vector, mapping the factor levels (names) to their associated colors (values). Useful for constructing plot keys.

**Usage**

```
## S3 method for class 'colored'  
key(x, ...)
```

**Arguments**

x	A colored factor
...	Additional arguments (unused)

**Value**

A color key, in the form of a named vector

**Author(s)**

David C. Norris



**See Also**[colored](#)

---

`plotbraids`*plotbraids*

---

**Description**

Braided streams plot for cohort visualization

**Usage**

```
plotbraids(formula, data, idvar = "id", stratify = FALSE,
  steps = 1:3, outside = FALSE, xlab = NULL, ...,
  x.scales.labels = paste(formula[[2]], steps, sep = "."),
  x.scales = list(alternating = FALSE, relation = "same",
    labels = x.scales.labels,
    at = seq(length(steps)) - 0.5),
  scales = list(x = x.scales, y = list(draw = FALSE, relation = "free")),
  strip = TRUE,
  lattice.options = list(axis.padding = list(numeric = 0, factor = 0)))
```

**Arguments**

<code>formula</code>	A formula of the form <code>trt ~ seq</code> or <code>trt ~ seq   cond</code> , where <code>trt</code> is a treatment factor, <code>seq</code> is an integer sequence number, and the optional <code>cond</code> is a conditioning factor used to trellis the braided stream plot. Probably only two-valued conditioning factors will produce visually acceptable plots.
<code>data</code>	A data frame with columns named on the LHS and RHS of argument <code>formula</code>
<code>idvar</code>	A character vector naming columns of data that identify multiple records from the same individual, used to reshape data into wide form
<code>stratify</code>	A logical value, indicating whether whitespace should be introduced to stratify the braids by initial treatment
<code>steps</code>	Which values of the sequence number should be included in the plot. Presently, only vectors of the form <code>1:n</code> (for some integer <code>n</code> ) are supported.
<code>outside</code>	A logical value determining whether the state labels are to appear outside the panel.
<code>xlab</code>	x-axis label
<code>...</code>	Additional arguments passed to delegates
<code>x.scales.labels</code>	Labels for the treatment states
<code>x.scales</code>	Provides a hook for modifying the basically sensible default layout and labeling of the x-axis

scales	Provides a hook for modifying the basically sensible default layout of both x- and y-axes
strip	Allows user to provide a strip function if the default does not suffice
lattice.options	Allows specification of plot-specific lattice options

### Details

Displays 'paths' taken by individuals passing through a sequence of discrete states, such as a sequence of treatments.

### Value

A trellis plot object

### Author(s)

David C. Norris

### Examples

```
## We demonstrate a simple braided stream plot based on the built-in occupationalStatus data set.
## It should be noted that the semantics of these data are very slightly at odds with the intended
## application of the braided stream plot, since the index cases in this data set were the _sons_,
## rather than the fathers. Thus, although time goes left-to-right in this figure, the streamlines
## run right-to-left epidemiologically. Notwithstanding this minor technicality, the figure gives
## a lively, compelling and meaningful visualization of these data.
## 1. Build a 'wide-form' data set from the table 'occupationalStatus'
df.wide <- data.frame(status.1=rep(1:8, 8),
                     status.2=rep(1:8, each=8),
                     N=as.vector(occupationalStatus))
df.wide <- df.wide[rep(1:64, times=df.wide$N),-3]
## 2. Reshape this to the 'long-form' data set expected by 'plotbraids'
df.long <- reshape(df.wide, varying=paste("status", 1:2, sep="."), direction="long", timevar="gen")
df.long <- df.long[order(df.long$id),]
## TODO: Generate appropriate 'class' labels for status.
## TODO: Use this opportunity to demonstrate meaningful application of a colored factor.
## 3. Plot the braided stream plot
plotbraids(status ~ gen, df.long, stratify=TRUE, steps=1:2,
           outside=TRUE, xlab="Generation",
           x.scales.labels=c("Father", "Son"))
```

---

predReg

*Registry Ensemble Prediction*

---

### Description

Generate an `rms:Predict` object or data frame for an ensemble of simulated disease registries

**Usage**

```

predReg(genReg, N, M = 100, fit = NULL,
  adjust.to = fit$Design$limits["Adjust to", ],
  do.pred = function(df) { fit.call <- fit$call
    fit.call$data <- quote(df)
    eval(parse(text = "predReg.fit <- eval(fit.call)"))
    eval(parse(text =
      "predReg.fit$Design$limits['Adjust to',names(adjust.to)]<<-adjust.to")
    )
    if (eval(parse(text = "is(predReg.fit,'lrm')")))
      eval(parse(text = "Predict(predReg.fit, fun=plogis)"))
    else eval(parse(text = "Predict(predReg.fit)")) },
  ...)

```

**Arguments**

genReg	A function that returns a simulated registry dataset, taking as its first parameter the desired size of the simulated registry, and possibly other parameters passed through via the ... arg
N	Size of generated registries
M	Size of the ensemble
fit	A fitted model usually intended to serve as a template for a model to be fitted to the simulated registries. This may be NULL when do.pred is provided explicitly in the call
adjust.to	A list of adjust-to values for the fitted models, defaulting to the adjust-to parameters of fit
do.pred	A function to be run on the data generated during each iteration of the simulation, generating either an rms:Predict object, or a (usually, named) atomic vector
...	Additional parameters passed to genReg

**Details**

Given a function for generating a simulated disease registry, this function generates an ensemble of such registries. It then returns an rms:Predict object that contains ensemble-averaged predictions and confidence bounds.

**Value**

Depending on the return type of do.pred, either an rms:Predict object containing ensemble-averaged predictions with confidence bounds reflecting their estimated ensemble variance, or else a data frame collecting the vector returned by do.pred

**Author(s)**

David C. Norris

---

```
prepanel.prevalence
```

*Adapts y-axis limits to prevalence range and any nonzero box width*

---

### Description

When generating a ‘Prevalence Plot’ via `bwplot(panel=panel.prevalence)`, specifying `prepanel=prepanel.prevalence` spares one from calculating the y-axis bounds.

### Usage

```
prepanel.prevalence(x, y, ...)
```

### Arguments

<code>x</code>	Vector of (numeric or difftime) treatment durations
<code>y</code>	The treatment factor
<code>...</code>	Used to pass prevalence and <code>box.width</code> parameters

### Author(s)

David C. Norris

### See Also

[panel.prevalence](#)

---

```
radarplot
```

*A multidimensional ‘Table 1 at-a-glance’*

---

### Description

Plots a specialized radar (aka, ‘star’) plot to compare summary statistics on a handful of covariates, across a multi-dimensional set of comparison groups, as in the standard ‘Table 1’ of experimental studies, where the groups correspond to treatment assignments.

### Usage

```
radarplot(x, data, datadist = getOption("datadist"),
  xlim = c(-1.4, 1.4), ylim = c(-1.2, 1.2),
  rescale = c("IQR", "range"), treatment = NULL,
  stratify = NULL, strength = NULL, include.na = FALSE,
  overall = FALSE, ...)
```

**Arguments**

<code>x</code>	Typically, a formula of the form $S \sim x$ , $S \sim x   t$ or $S \sim x   t * u$ , where $S$ is the (suitably normalized) summary matrix of a <code>summary.formula.cross</code> object having factors $\{x[, t[, u]]\}$ as covariates. In the resulting radar plot, the $x$ factor will correspond to colored polygons overlaid within each panel of a trellis of 0, 1 or 2 dimensions, defined by the (optional) terms $t$ and $u$ . Alternatively, $x$ may be a fitted model in which the LHS is the experimental outcome, and the RHS includes the spoke variables.
<code>data</code>	Either a <code>summary.formula.cross</code> object (when $x$ is a formula), or a <code>data.frame</code> (when $x$ is a fitted model).
<code>datadist</code>	An <code>rms:datadist</code> object describing the underlying dataset. By suitably modifying <code>datadist\$limits</code> , greater control may be achieved over the automatic calculation of strength-of-association measures.
<code>xlim</code>	A positioning parameter that really should be automated!
<code>ylim</code>	A positioning parameter that really should be automated!
<code>rescale</code>	How to rescale continuous covariates for plotting on the spokes
<code>treatment</code>	When $x$ is an outcomes model, identifying the treatment variable (and any variables to stratify on, see below) permits the formula embedded in the model to be used to generate a <code>summary.formula.cross</code> object
<code>stratify</code>	When $x$ is an outcomes model, identifying the treatment variable (see above) and any variables to stratify on, permits the formula embedded in the model to be used to generate a <code>summary.formula.cross</code> object
<code>strength</code>	When the default choice of strength-of-association measure is not suitable, it may be specified explicitly as a named vector with elements corresponding to the spoke variables, in the order they appear in the model formula. The names will be printed at the outer endpoints of the spokes, and will typically be numbers (e.g., odds ratios) rounded to 2-3 significant figures.
<code>include.na</code>	Should the summary include NA values? Passed to the <code>summary.formula</code> call that generates the radarplot data.
<code>overall</code>	Should the summary include an 'ALL' row? Passed to the <code>summary.formula</code> call that generates the radarplot data.
<code>...</code>	Other parameters to be passed to <code>panel.radarplot</code>

**Details**

The covariates appear on the radii, or 'spokes' of the radarplot, while the several (up to 3) dimensions defining the comparison groups appear as a color-coded overlay, plus up to two trellis dimensions. In generic applications, the client code employs `Hmisc::summary.formula(method='cross')` to create the multi-dimensional array for plotting, and then rescales the array elements by a linear or affine transformation to the interval  $[0,1]$ . Ordered factors may be treated as quasi-continuous variables according to their internal integer representation; logical covariates (and also, conveniently, 2-valued factors) appear as proportions; and unordered categorical variables with  $n$  levels may be transformed to  $n-1$  dummy variables, which may then be plotted as proportions. An optional `strength` parameter may be specified, providing a strength-of-association measure connecting the spoke variables to the experimental outcome. This is reflected graphically by fanning the spokes

out into sectors subtending angles proportional to the strength of associations. Used in this way, the radar plot draws attention to those spoke variables with greatest potential to confound the outcome, and so becomes a useful exploratory tool. To provide convenient support for this specialized usage, `radarplot` optionally accepts a regression model in which the outcome of interest is the regressand, and the spoke variables are regressors. From this model, strength-of-association measures are automatically derived, appropriately to the class of the regression model. Invoked in this way, `radarplot` is also able to free client code from the burden of generating and rescaling a `summary.formula.cross` object. At this time, models of class `lrm` and `cph` are supported. See the example code below.

### Author(s)

David C. Norris

### Examples

```
library(rms)
df <- upData(mtcars,
  cyl=factor(cyl,levels=2*(2:4),labels=paste(2*(2:4),"cyl", sep="-")),
  am=factor(am,levels=0:1,labels=c("automatic","manual")),
  gear=factor(gear,levels=3:5,labels=paste(3:5,"speed", sep="-")),
  labels=c(
    mpg="Miles per gallon"
    ,cyl="Number of cylinders"
    ,disp="Displacement"
    ,hp="Gross horsepower"
    ,drat="Rear axle ratio"
    ,wt="Weight"
    ,qsec="1/4 mile time"
    ,am="Transmission type"
    ,gear="Number of forward gears"
    ,carb="Number of carburetors"
  ),
  units=c(
    wt="lb/1000"
    ,disp="in^3"
    ,qsec="sec"
  ),
  drop='vs'
)
s <- summary(cbind(mpg, disp, hp, drat, wt) ~ cyl + gear + am,
  method='cross', overall=TRUE, data=df)
dd <- datadist(df)
radarplot(S ~ cyl | gear*am, data=s, datadist=dd, rescale="range")

## TODO: Provide example of convenient usage with 'lrm' and 'cph' models.
```

**Description**

Reorder levels of a colored factor.

**Usage**

```
## S3 method for class 'colored'
relevel(x, ref, ...)
```

**Arguments**

x	A colored factor
ref	The reference level
...	Unused

**Value**

A colored factor

**Author(s)**

David C. Norris

**See Also**

[relevel](#)

**Examples**

```
# TODO: Provide an example
```

---

remvar

*remvar*

---

**Description**

Removes variables from a regression model formula

**Usage**

```
remvar(f, vars)
```

**Arguments**

f	A model formula
vars	A character vector giving names of variables to remove from f

**Details**

This is a utility function, intended to support fastbw

**Value**

A modified formula, omitting the specified variables

**Author(s)**

David C. Norris

---

representNA	<i>Convert "NA"s to NAs</i>
-------------	-----------------------------

---

**Description**

A basic utility function converting "NA" strings to R's missing type, NA.

**Usage**

```
representNA(x)
```

**Arguments**

x                    A vector of class 'character', which may contain the string "NA" to be replaced with NA.

**Value**

Returns the vector x, with all "NA" elements recoded as NA.

**Note**

It is quite possible that this function should be removed, and that the issue of NAs coded as "NA" ought to be resolved by standardized workflows or SOPs for upstream data management. Conversely, if we retain this function, it should be expanded through an optional argument allowing specification of a set of strings to be recoded as NA; this argument might default to `c("NA", ".")`, for example.

**Author(s)**

David C. Norris

**Examples**

```
## TODO: Provide an example
```



---

```
representNA.data.frame
```

*Apply the representNA function to all columns of a data frame*

---

### Description

Replace the string "NA" with R's missing value NA, wherever it appears in the given data frame.

### Usage

```
representNA.data.frame(df)
```

### Arguments

df                    The data frame to be 'cleaned up'.

### Value

Returns a data frame identical to df, with "NA" recoded as NA.

### Author(s)

David C. Norris

### Examples

```
# TODO: Provide an example
```

---

```
timeline
```

*A color-coded treatment time-line, with overlaid events*

---

### Description

This individual-level graphic depicts horizontal time intervals of an ongoing treatment course, color-coded by, e.g., agent. Categorical events which may occur during treatment, such as assessments of response, are annotated with color-coded arrows.

### Usage

```
timeline(figlabel, txs, resp, bsl, ptid = "patnum",
  condition = TRUE, formula = trt + resp ~ time | patnum,
  followed = list(from = as.difftime(0, units = "days"),
    to = as.difftime(Inf, units = "days")),
  tx.start = "start", tx.end = "end",
  treatment = as.character(formula[[2]][[2]]),
  response = as.character(formula[[2]][[3]]),
```

```

time = as.character(formula[[3]][[2]]),
timeunit = units(resp[[time]]), caption = NULL,
tx.key = key(txs[[treatment]]),
resp.key = key(resp[[response]]), cols.rows = c(2, 5),
prefix.string = "figs/plot",
xlim = c(0, as.double(min(followed$to, max(resp[[time]])), units = timeunit)),
xlab = paste("Time (", timeunit, ") ", sep = ""),
ylab = "",
filename = paste(figlabel, "followed", sub("[.]", "_", followed$from),
                 "-", sub("[.]", "_", followed$to), "y", sep = "")

```

### Arguments

figlabel	A string to be used as a LaTeX figure label
txs	A data frame describing intervals of treatment
resp	A data frame describing treatment response assessments
bsl	A data frame of subject baseline characteristics
ptid	The name of the unique patient identifier (default is "patnum")
condition	An R expression giving a logical condition (a predicate on subject baseline characteristics) used for selecting the subjects to be plotted
formula	A formula of the form <code>trt+resp~time patnum</code> , interpreted as "plot treatment and response vs time, by patnum"
followed	A list with <code>difftime</code> components <code>from</code> and <code>to</code> , giving the minimum and maximum durations of follow-up for patients to be plotted. This is necessary to prevent the dwarfing of treatment courses for patients with short follow-up when plotted alongside those of patients with extended follow-up
tx.start	name of treatment start date column of <code>txs</code>
tx.end	name of treatment end date column of <code>txs</code>
treatment	name of treatment column of <code>txs</code>
response	name of response column of <code>resp</code>
time	name of time column of <code>resp</code>
timeunit	The time unit desired for the horizontal axis
caption	The figure caption may be provided explicitly or (if <code>NULL</code> ) constructed automatically
tx.key	The plot legend for treatments
resp.key	The plot legend for responses
cols.rows	Trellis layout as <code>c(ncols, nrows)</code>
prefix.string	Prefix string (including possibly a directory) for cached plot output
xlim	x-axis limits
xlab	x-axis label
ylab	y-axis label
filename	Filename pattern for cached plot output

**Note**

TODO: further notes

**Author(s)**

David C. Norris

**References**

TODO: Reference our white paper or pending publication

**See Also**

TODO: List objects to See Also as [help](#)

**Examples**

```
## TODO: Provide an example
## TODO: Document usage. If necessary, include sample data sets in package:VizOR.
```

---

vlbw

*Data on 671 very low birth weight infants*

---

**Description**

Documentation for included dataset vlbw

**Format**

A data frame containing 32 observations on 671 infants.

**Details**

Data on 671 infants with very low (<1600 grams) birth weight from 1981-87 were collected at Duke University Medical Center by Dr. Michael O'Shea.

**Source**

<http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets>

**References**

O'Shea M, Savitz DA, Hage ML, Feinstein KA: Prenatal events and the risk of subependymal / intraventricular haemorrhage in very low birth weight neonates. *Paediatric and Perinatal Epidemiology* 1992;6:352-362.

---

`[.colored`*Index a colored vector*

---

**Description**

Index a colored vector

**Usage**

```
## S3 method for class 'colored'  
x[...]
```

**Arguments**

<code>x</code>	A colored factor
<code>...</code>	Unused

**Value**

A colored factor

**Author(s)**

David C. Norris

**See Also**

[colored](#)

# Index

- \*Topic **category**
  - color.colored, 4
  - colored, 5
  - key.colored, 8
  - relevel.colored, 14
- \*Topic **classes**
  - colored, 5
- \*Topic **color**
  - alphanize, 3
  - color.colored, 4
  - key.colored, 8
  - relevel.colored, 14
- \*Topic **datagen**
  - predReg, 10
- \*Topic **datasets**
  - vlbw, 19
- \*Topic **dynamic**
  - controller, 6
- \*Topic **hplot**
  - plotbraids, 9
  - prepanel.prevalence, 12
  - radarplot, 12
  - timeline, 17
- \*Topic **htest**
  - radarplot, 12
- \*Topic **iplot**
  - controller, 6
- \*Topic **manip**
  - representNA, 16
  - representNA.data.frame, 17
- \*Topic **package**
  - VizOR-package, 2
- \*Topic **regression**
  - fastback, 7
- [.colored, 20
- alphanize, 3
- color, 4
- color.colored, 4
- colored, 4, 5, 9, 20
- controller, 6
- droplevels.colored, 6
- fastback, 7
- help, 19
- Hmisc, 3
- key, 8
- key.colored, 8
- panel.prevalence, 12
- plotbraids, 9
- predReg, 10
- prepanel.prevalence, 12
- radarplot, 12
- relevel, 15
- relevel (relevel.colored), 14
- relevel.colored, 14
- remvar, 15
- representNA, 16
- representNA.data.frame, 17
- rms, 3
- timeline, 17
- VizOR (VizOR-package), 2
- VizOR-package, 2
- vlbw, 19