

# Package ‘pca3d’

July 1, 2015

**Type** Package

**Title** Three Dimensional PCA Plots

**Version** 0.8

**Date** 2015-06-30

**Author** January Weiner

**URL** <http://logfc.wordpress.com>

**Maintainer** January Weiner <january.weiner@gmail.com>

**Description**

Functions simplifying presentation of PCA models in a 3D interactive representation using 'rgl'.

**License** GPL-2

**Imports** rgl,ellipse

**Suggests** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-07-01 22:59:10

## R topics documented:

defaultPalettePCA3D . . . . .	2
listShapes . . . . .	2
makeMoviePCA . . . . .	3
metabo . . . . .	3
pca2d . . . . .	4
snapshotPCA3d . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

defaultPalettePCA3D *Default palette*

---

### Description

Default set of colors for the pca3d package. This is a colorblind-friendly palette, following the R cookbook.

### Usage

```
defaultPalettePCA3D(n = NULL, transparent = NULL, d3 = FALSE)
```

### Arguments

n	Number of colors to return
transparent	character string which will be pasted to each color
d3	if true, no transparency information will be added to the colors

### Details

The default palette contains 21 colors.

### Value

A character vector with the color palette

---

listShapes *Show all permissible shapes for pca3d*

---

### Description

Show all permissible shapes for pca3d

### Usage

```
listShapes()
```

### Details

Show all permissible shapes for the functions pca3d and pca2d. The shapes may be abbreviated using (matching is done with [pmatch](#)).

### Value

A data frame with permissible 3d shapes for plotting and their pch counterparts is returned invisibly.

---

makeMoviePCA	<i>Create a PCA movie</i>
--------------	---------------------------

---

**Description**

Make a basic movie using movie3d()

**Usage**

```
makeMoviePCA(...)
```

**Arguments**

... Any parameters will be passed to movie3d()

**Details**

A wrapper around movie3d with some default settings. Used to create a movie on the fly.

**Value**

The value returned by movie3d()

**Examples**

```
## Not run:  
data( metabo )  
pca <- prcomp( metabo[,-1], scale.= TRUE )  
pca3d(pca, group=metabo[,1])  
makeMoviePCA()  
  
## End(Not run)
```

---

metabo	<i>Metabolic profiles in tuberculosis.</i>
--------	--

---

**Description**

Relative abundances of metabolites from serum samples of three groups of individuals

**Details**

A data frame with 136 observations on 425 metabolic variables.

Serum samples from three groups of individuals were compared: tuberculin skin test negative (NEG), positive (POS) and clinical tuberculosis (TB).

## References

Weiner J 3rd, Parida SK, Maertzdorf J, Black GF, Repsilber D, et al. (2012) Biomarkers of Inflammation, Immunosuppression and Stress Are Revealed by Metabolomic Profiling of Tuberculosis Patients. PLoS ONE 7(7): e40221. doi:10.1371/journal.pone.0040221

## Examples

```
data(metabo)
# maybe str(metabo) ; plot(metabo) ...
pca <- prcomp( metabo[,-1] )
```

---

pca2d

*Show a three- or two-dimensional plot of a prcomp object*

---

## Description

Show a three- two-dimensional plot of a prcomp object or a matrix, using different symbols and colors for groups of data

## Usage

```
pca2d(pca, components = 1:2, col = NULL, title = NULL, new = FALSE,
      axes.color = "black", bg = "white", radius = 1, group = NULL,
      shape = NULL, palette = NULL, fancy = FALSE, biplot = FALSE,
      biplot.vars = 5, legend = NULL, show.scale = FALSE,
      show.labels = FALSE, labels.col = "black", show.axes = TRUE,
      show.axe.titles = TRUE, axe.titles = NULL, show.plane = TRUE,
      show.shadows = FALSE, show.centroids = FALSE, show.group.labels = FALSE,
      show.ellipses = FALSE, ellipse.ci = 0.95, ...)
```

```
pca3d(pca, components = 1:3, col = NULL, title = NULL, new = FALSE,
      axes.color = "grey", bg = "white", radius = 1, group = NULL,
      shape = NULL, palette = NULL, fancy = FALSE, biplot = FALSE,
      biplot.vars = 5, legend = NULL, show.scale = FALSE,
      show.labels = FALSE, labels.col = "black", show.axes = TRUE,
      show.axe.titles = TRUE, axe.titles = NULL, show.plane = TRUE,
      show.shadows = FALSE, show.centroids = FALSE, show.group.labels = FALSE,
      show.shapes = TRUE, show.ellipses = FALSE, ellipse.ci = 0.95)
```

## Arguments

pca	Either a prcomp object or a matrix with at least three columns
components	Vector of length 3 (pca3d) or 2 (pca2d) containing the components to be shown
col	Either a single value or a vector of length equal to number of rows, containing color definitions for the plot points to be shown
title	Window title

<code>new</code>	Use TRUE to open a new window
<code>axes.color</code>	Axis color This option has no effect in <code>pca2d</code> .
<code>bg</code>	Background color
<code>radius</code>	Scaling item for the size of points to be shown. In <code>pca2d</code> , this corresponds to the <code>cex</code> parameter.
<code>group</code>	either NULL or a factor of length equal to number of rows. Factor levels can be used to automatically generate symbols and colors for the points shown
<code>shape</code>	Either a single value or a character vector describing the shapes to be used when drawing data points. Allowed shapes are: sphere, tetrahedron and cube, and may be abbreviated. In <code>pca2d</code> , the parameter is passed directly on to the <code>pch</code> option of the <code>points()</code> function.
<code>palette</code>	Specifies the color palette when colors are automatically assigned to the groups. See Details.
<code>fancy</code>	set <code>'show.labels'</code> , <code>'show.shadows'</code> , <code>'show.centroids'</code> and <code>'show.group.labels'</code> to TRUE.
<code>biplot</code>	Specify whether to show a biplot (see section 'biplots' below)
<code>biplot.vars</code>	Specify which loading to show on the biplot (see section 'biplots' below)
<code>legend</code>	If NULL, no legend will be drawn. Otherwise the value specifies the legend position in a form accepted by <a href="#">legend</a> and <a href="#">legend3d</a> .
<code>show.scale</code>	TRUE for showing a numeric scale at the edges of the plot. This option has no effect in <code>pca2d</code> .
<code>show.labels</code>	TRUE for showing labels (taken from the coordinate matrix or the <code>prcomp</code> object). Alternatively, a vector with labels to be shown next to the data points.
<code>labels.col</code>	Single value or vector describing the colors of the labels.
<code>show.axes</code>	TRUE to show the axes. This option has no effect in <code>pca2d</code> .
<code>show.axe.titles</code>	If TRUE, show axe titles (PC 1, PC 2 etc.) This option has no effect in <code>pca2d</code> .
<code>axe.titles</code>	A vector with two ( <code>pca2d</code> ) or three ( <code>pca3d</code> ) values containing the axe titles (corresponds to <code>xlab</code> and <code>ylab</code> in regular plot). If missing, but <code>show.axe.titles</code> is TRUE, axe titles will be generated automatically.
<code>show.plane</code>	If TRUE, show a grey horizontal plane at $y = 0$ . This option has no effect in <code>pca2d</code> .
<code>show.shadows</code>	If TRUE, show a "lollipop" representation of the points on the $y = 0$ plane: a vertical line joining the data point with the plane and a shadow. In <code>pca2d</code> , for each sample at $(x,y)$ , a grey line is drawn from $(x,y)$ to $(x,0)$ .
<code>show.centroids</code>	If TRUE and the group variable is defined, show cluster centroids (using appropriate group symbols) and lines from each data point to the corresponding centroid.
<code>show.group.labels</code>	Either TRUE/FALSE or a vector equal to the number of unique values in the group parameter. If set, labels for each of the defined group will be shown at the group's centroid. If the value of the parameter is TRUE, then the group names will be taken from the group parameter. Otherwise, the values from this parameter will be used.

<code>show.ellipses</code>	A TRUTH/FALSE value indicating whether to show confidence interval ellipses or ellipsoids around each defined group
<code>ellipse.ci</code>	The confidence level of a pairwise confidence region for the CI. The default is 0.95, for a 95 the size of the ellipse being plotted.
<code>...</code>	For <code>pca2d</code> , any further argument will be passed on to the <code>points()</code> function.
<code>show.shapes</code>	A TRUTH/FALSE value indicating whether the different symbols (shapes) for the shown data points should be plotted (default TRUE).

### Details

The `pca3d` function shows a three dimensional representation of a PCA object or any other matrix. It uses the `rgl` package for rendering.

`pca2d` is the 2D counterpart. It creates a regular, two-dimensional plot on the standard graphic device. However, it takes exactly the same options as `pca3d`, such that it is easy to create 2D variants of the 3D graph.

Often, PCA visualisation requires using different symbols and colors for different groups of data. `pca3d()` and `pca2d()` aim at creating reasonable defaults, such that a simple call with two parameters – the `pca` object and the vector with group assignments of the samples – is sufficient for a basic diagnosis.

### Value

Both `pca2d` and `pca3d` return invisibly a data frame which can be used to generate a legend for the figure. The data frame has as many rows as there are groups, and column with the group name, assigned color and assigned shape.

### Biplots

If option `'biplot'` is TRUE, a biplot showing both the PCA results (samples) and variables is shown. This corresponds to the `biplot` function which works for the `prcomp` class objects. However, a biplot showing all variable loadings will be unreadable if the data is highly dimensional (for example, gene expression data). Therefore, the option `'biplot.vars'` specifies which variables are shown on the biplot.

If `'biplot.vars'` is a vector of length larger than one, it will be interpreted as a direct selection of the variables to be shown; for example, for a `prcomp` object `pca`, the variable selection will happen through `pca$rotation[biplot.vars,]`.

If `'biplot.vars'` is a single number, then for each of the components shown, a number of variables equal to `'biplot.vars'` with the highest absolute loadings will be shown on the biplot.

### Examples

```
data( metabo )
pca <- prcomp( metabo[, -1], scale.= TRUE )

pca3d( pca, group= metabo[, 1] )
pca2d( pca, group= metabo[, 1] )

## a bit more fancy:
```

```
## black background, white axes,  
## centroids  
pca3d( pca, group= metabo[,1],  
       fancy= TRUE, bg= "black",  
       axes.color= "white", new= TRUE )
```

---

snapshotPCA3d

*Save a 3D PCA snapshot*

---

### **Description**

Take a snapshot of the 3D PCA to a file.

### **Usage**

```
snapshotPCA3d(file)
```

### **Arguments**

file	Name of the file to save the snapshot to <code>data( metabo )</code> <code>pca &lt;- prcomp( metabo[,1], scale.= TRUE )</code> <code>pca3d(pca, group=metabo[,1])</code> <code>snapshotPCA3d("testfile.png")</code>
------	---

### **Details**

This is just a wrapper around `rgl.snapshot`.

# Index

\*Topic **PCA**

pca2d, [4](#)

\*Topic **biplot**

pca2d, [4](#)

\*Topic **datasets**

metabo, [3](#)

\*Topic **prcomp**

pca2d, [4](#)

\*Topic **princomp**

pca2d, [4](#)

biplot, [6](#)

defaultPalettePCA3D, [2](#)

legend, [5](#)

legend3d, [5](#)

listShapes, [2](#)

makeMoviePCA, [3](#)

metabo, [3](#)

pca2d, [4](#)

pca3d (pca2d), [4](#)

pca3d-package (pca2d), [4](#)

pmatch, [2](#)

prcomp, [6](#)

snapshotPCA3d, [7](#)