

# Package ‘klustR’

May 8, 2026

**Title** D3 Dynamic Cluster Visualizations

**Description** Used to create dynamic, interactive 'D3.js'  
based parallel coordinates and principal component plots in 'R'.  
The plots make visualizing k-means or other clusters simple and informative.

**Version** 0.1.0

**URL** <https://mckaymdavis.github.io/klustR/>,  
<https://github.com/McKayMDavis/klustR>

**BugReports** <https://github.com/McKayMDavis/klustR/issues>

**License** GPL (>= 3)

**Depends** R (>= 3.6.0)

**Imports** htmlwidgets (>= 0.3.2), jsonlite

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** McKay Davis [aut, cre]

**Maintainer** McKay Davis <mckaymdavis@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-06-19 09:40:08 UTC

## Contents

klustR-shiny . . . . .	2
pacoplot . . . . .	2
pcplot . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

 klustR-shiny

*Shiny bindings for klustR widgets*


---

### Description

Output and render functions for using klustR widgets within Shiny applications and interactive Rmd documents.

### Usage

```
pacoplotOutput(outputId, width = "100%", height = "400px")
renderpacoplot(expr, env = parent.frame(), quoted = FALSE)
pcplotOutput(outputId, width = "100%", height = "400px")
renderpcplot(expr, env = parent.frame(), quoted = FALSE)
```

### Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended.
expr	An expression that generates a klustR graph.
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

---

 pacoplot

*Parallel Coordinates Plot for Clustering*


---

### Description

Creates an interactive parallel coordinates plot detailing each dimension and the cluster associated with each observation.

### Usage

```
pacoplot(data, clusters, colorScheme = "schemeCategory10",
  width = NULL, height = NULL, labelSizes = NULL, lineSize = NULL,
  measures = NULL)
```

**Arguments**

<code>data</code>	A dataframe of numeric columns.
<code>clusters</code>	A named integer matrix of clusters where names are the row names of the above dataframe and integers are the integer value of the row's associated cluster. This can be obtained from a function such as <code>stats::kmeans()\$cluster</code> .
<code>colorScheme</code>	The color scheme of the plot. May be a pre-configured D3 ordinal color scheme or a vector of html colors (hex or named) of the same length as the number of clusters.
<code>width</code>	The width of the plot window.
<code>height</code>	The height of the plot window.
<code>labelSizes</code>	A number or list of any combination of parameters shown that define the label sizes. <code>list(yaxis = 12, yticks = 10, tooltip = 15)</code>
<code>lineSize</code>	A number to adjust the size of the lines.
<code>measures</code>	A list of functions that is any combination of parameters shown that define the measurements for intervals and average lines displayed. Defaults to the options shown (median and 1st and 3rd quartile). <code>list(avg = median, upper = function(x){return(quantile(x, c(0.75)))}, lower = function(x){return(quantile(x, c(0.25)))})</code>

**Details**

- Hover over lines to display row label
- Click on a line to fade out all lines except the associated cluster
- Click on another line to bold this line as well
- Clicking a second time on a line will fade it out

**Examples**

```
# Barebones
df <- state.x77
clus <- kmeans(df, 5)$cluster
pacoplot(data = df, clusters = clus)

# With options
df <- state.x77
clus <- kmeans(df, 5)$cluster
pacoplot(data = df, clusters = clus,
          colorScheme = c("red", "green", "orange", "blue", "yellow"),
          labelSizes = list(yaxis = 16, yticks = 12),
          measures = list(avg = mean))
```

---

pcplot

*Principal Component Plot for K-Means Clustering*


---

### Description

Reduces dimensionality to 2D using principal component analysis (PCA) and displays a dynamic visualization of two principal components (PC).

### Usage

```
pcplot(data, clusters, barColor = "steelblue",
       colorScheme = "schemeCategory10", width = NULL, height = NULL,
       labelSizes = NULL, dotSize = NULL, pcGridlines = FALSE,
       barGridlines = FALSE)
```

### Arguments

data	A dataframe of numeric columns. Scaled data is preferred as PCA does not work the same with non-scaled data.
clusters	A named integer matrix of clusters where names are the row names of the above dataframe and integers are the integer value of the row's associated cluster. This can be obtained from a function such as <code>stats::kmeans()\$cluster</code> .
barColor	The color to use for the bar-chart fill. May be any html color (hex or named).
colorScheme	The color scheme of the PCA plot. May be a pre-configured D3 ordinal color scheme or a vector of html colors (hex or named) of the same length as the number of clusters.
width	The width of the plot window.
height	The height of the plot window.
labelSizes	A number or list of any combination of parameters shown that define the label sizes. <code>list(yaxis = 12, yticks = 10, tooltip = 15)</code>
dotSize	A number to adjust the size of the dots.
pcGridlines	TRUE FALSE Show grid-lines on the PC plots?
barGridlines	TRUE FALSE Show grid-lines on the bar-charts?

### Details

- Clicking on axis labels will display a bar-chart of PC contribution
- Clicking on legend colors will fade out all points but the points in the cluster selected
- Hover over points to see the label and point coordinates

**Examples**

```
# Barebones
scaled_df <- scale(state.x77)
clus <- kmeans(scaled_df, 5)$cluster
pcplot(data = scaled_df, clusters = clus)

# With Options
scaled_df <- scale(state.x77)
clus <- kmeans(scaled_df, 5)$cluster
pcplot(data = scaled_df, clusters = clus,
        barColor = "red",
        colorScheme = c("red", "green", "orange", "blue", "yellow"),
        labelSizes = list(yaxis = 20, yticks = 15, tooltip = 25),
        pcGridlines = TRUE, barGridlines = TRUE)
```

# Index

[klustR-shiny](#), 2

[pacoplot](#), 2

[pacoplotOutput \(klustR-shiny\)](#), 2

[pcplot](#), 4

[pcplotOutput \(klustR-shiny\)](#), 2

[renderpacoplot \(klustR-shiny\)](#), 2

[renderpcplot \(klustR-shiny\)](#), 2