

# Package ‘radarchart’

May 9, 2026

**Title** Radar Chart from 'Chart.js'

**Version** 0.3.1

**Description** Create interactive radar charts using the 'Chart.js' 'JavaScript' library and the 'htmlwidgets' package. 'Chart.js' <<http://www.chartjs.org/>> is a lightweight library that supports several types of simple chart using the 'HTML5' canvas element. This package provides an R interface specifically to the radar chart, sometimes called a spider chart, for visualising multivariate data.

**Depends** R (>= 3.1.2)

**License** MIT + file LICENSE

**LazyData** true

**URL** <https://github.com/mangothecat/radarchart>

**BugReports** <https://github.com/mangothecat/radarchart/issues>

**Imports** htmlwidgets, htmltools, grDevices

**RoxygenNote** 5.0.1

**Suggests** testthat, knitr, rmarkdown, tidy, shiny

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Doug Ashton [aut, cre],  
Shane Porter [aut],  
Nick Downie [ctb] (chart.js library),  
Tanner Linsley [ctb] (chart.js library),  
William Entriken [ctb] (chart.js library)

**Maintainer** Doug Ashton <[dashton@mango-solutions.com](mailto:dashton@mango-solutions.com)>

**Repository** CRAN

**Date/Publication** 2016-12-20 11:47:12

## Contents

chartJSRadar . . . . .	2
chartJSRadarOutput . . . . .	3

chartJSRadar_html . . . . .	4
colourMatrix . . . . .	4
renderChartJSRadar . . . . .	5
runExampleApp . . . . .	5
setRadarScale . . . . .	6
skills . . . . .	6
skillsByName . . . . .	7

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

---

chartJSRadar	<i>Make a ChartJS Radar Plot</i>
--------------	----------------------------------

---

## Description

R bindings to the radar plot in the chartJS library

## Usage

```
chartJSRadar(scores, labs, width = NULL, height = NULL, main = NULL,
  maxScale = NULL, scaleStepWidth = NULL, scaleStartValue = 0,
  responsive = TRUE, labelSize = 18, showLegend = TRUE, addDots = TRUE,
  colMatrix = NULL, polyAlpha = 0.2, lineAlpha = 0.8,
  showToolTipLabel = TRUE, ...)
```

## Arguments

scores	Data frame or named list of scores for each axis. If labs is not specified then labels are taken from the first column (or element).
labs	Labels for each axis. If left unspecified labels are taken from the scores data set. If set to NA then labels are left blank.
width	Width of output plot
height	Height of output plot
main	Character: Title to be displayed
maxScale	Max value on each axis
scaleStepWidth	Spacing between rings on radar
scaleStartValue	Value at the centre of the radar
responsive	Logical. whether or not the chart should be responsive and resize when the browser does
labelSize	Numeric. Point label font size in pixels
showLegend	Logical whether to show the legend
addDots	Logical. Whether to show a dot for each point
colMatrix	Numeric matrix of rgb colour values. If NULL defaults are used

polyAlpha	Alpha value for the fill of polygons
lineAlpha	Alpha value for the outlines
showToolTipLabel	Logical. If TRUE then data set labels are shown in the tooltip hover over
...	Extra options passed straight to chart.js. Names must match existing options <a href="http://www.chartjs.org/docs/#getting-started-global-chart-configuration">http://www.chartjs.org/docs/#getting-started-global-chart-configuration</a>

## Examples

```
# Using the data frame interface
chartJSRadar(scores=skills)

# Or using a list interface
labs <- c("Communicator", "Data Wangler", "Programmer", "Technologist", "Modeller", "Visualizer")

scores <- list("Rich" = c(9, 7, 4, 5, 3, 7),
              "Andy" = c(7, 6, 6, 2, 6, 9),
              "Aimee" = c(6, 5, 8, 4, 7, 6))

# Default settings
chartJSRadar(scores=scores, labs=labs)

# Fix the max score
chartJSRadar(scores=scores, labs=labs, maxScale=10)

# Fix max and spacing
chartJSRadar(scores=scores, labs=labs, maxScale=12, scaleStepWidth = 2)

# Change title and remove legend
chartJSRadar(scores=scores, labs=labs, main = "Data Science Radar", showLegend = FALSE)

# Add pass through settings for extra options
chartJSRadar(scores=scores, labs=labs, maxScale =10, scaleLineWidth=5)
```

---

chartJSRadarOutput      *Widget output function for use in Shiny*

---

## Description

Widget output function for use in Shiny

## Usage

```
chartJSRadarOutput(outputId, width = "450", height = "300")
```

**Arguments**

outputId	output variable to read from
width	Must be valid CSS unit
height	Must be valid CSS unit

---

chartJSRadar_html	<i>Tell htmltools where to output the chart</i>
-------------------	---

---

**Description**

Tell htmltools where to output the chart

**Usage**

```
chartJSRadar_html(id, style, class, width, height, ...)
```

**Arguments**

id	The id of the target object
style	css stylings
class	class of the target
width	width of target
height	height of target
...	extra arguments currently unused

---

colourMatrix	<i>Check and prep the colour matrix</i>
--------------	---

---

**Description**

Check and prep the colour matrix

**Usage**

```
colourMatrix(colMatrix)
```

**Arguments**

colMatrix	A 3 x n matrix of integers between 0-255
-----------	--

**Value**

The checked and prepped matrix of the same size

**Examples**

```
radarchart:::colourMatrix(diag(255, nrow=3))
```

---

renderChartJSRadar	<i>Widget render function for use in Shiny</i>
--------------------	--

---

**Description**

Widget render function for use in Shiny

**Usage**

```
renderChartJSRadar(expr, env = parent.frame(), quoted = FALSE)
```

**Arguments**

expr	expression passed to <a href="#">shinyRenderWidget</a>
env	environment in which to evaluate expression
quoted	Logical. Is expression quoted?

---

runExampleApp	<i>Run an example Shiny app</i>
---------------	---------------------------------

---

**Description**

The radarchart package contains a number of demo Shiny apps to illustrate how to use the plots. The code is in `inst/shiny-examples/` and running this function will allow quick access to the apps.

**Usage**

```
runExampleApp(example)
```

**Arguments**

example	the name of the example. Choose from "basic" or "options".
---------	--

**Examples**

```
## Not run:
runExample("basic")

## End(Not run)
```

---

setRadarScale	<i>Autoscale the radar plot</i>
---------------	---------------------------------

---

**Description**

Autoscale the radar plot

**Usage**

```
setRadarScale(maxScale = NULL, scaleStepWidth = NULL, scaleStartValue = 0)
```

**Arguments**

maxScale	Numeric length 1. Desired max limit
scaleStepWidth	Numeric length 1. Spacing between rings
scaleStartValue	Numeric length 1. Value of the centre

**Value**

A list containing the scale options for chartjs

**Examples**

```
## Not run:  
setRadarScale(15, 3)  
setRadarScale(15, 5, 2)  
  
## End(Not run)
```

---

skills	<i>Skills in a team</i>
--------	-------------------------

---

**Description**

A dataset containing the skills vectors for three people

**Usage**

```
skills
```

**Format**

A data frame with 6 rows and 4 columns

**Label** The axis label for chartJSRadar

**Aimee** Vector of skills for Aimee

**Andy** Vector of skills for Andy

**Rich** Vector of skills for Rich

**Source**

Simulated

---

skillsByName	<i>Rotated version of skills data</i>
--------------	---------------------------------------

---

**Description**

A dataset containing the skills vectors for three people but by row rather than column. This data set is used to show how to rotate the data into a format accepted by [chartJSRadar](#).

**Usage**

skillsByName

**Format**

A data frame with 6 rows and 4 columns

**Name** Name of the team member

**Communicator** Their Communicator score: 0-10

**Data Wangler** Their Data Wangler score: 0-10

**Modeller** Their Modeller score: 0-10

**Programmer** Their Programmer score: 0-10

**Technologist** Their Technologist score: 0-10

**Visualizer** Their Visualizer score: 0-10

**Source**

Simulated

# Index

## \* datasets

skills, [6](#)

skillsByName, [7](#)

chartJSRadar, [2](#), [7](#)

chartJSRadar\_html, [4](#)

chartJSRadarOutput, [3](#)

colourMatrix, [4](#)

renderChartJSRadar, [5](#)

runExampleApp, [5](#)

setRadarScale, [6](#)

shinyRenderWidget, [5](#)

skills, [6](#)

skillsByName, [7](#)