

# Package: ggpolar (via r-universe)

August 20, 2024

**Type** Package

**Title** Dots and Their Connections in Polar Coordinate System

**Version** 0.2.2

**Maintainer** Shixiang Wang <w\_shixiang@163.com>

**Description** Provides basic graphing functions to fully demonstrate point-to-point connections in a polar coordinate space.

**License** GPL (>= 3)

**URL** <https://github.com/ShixiangWang/polar>

**BugReports** <https://github.com/ShixiangWang/polar/issues>

**Depends** ggplot2

**Imports** rlang

**Suggests** ezcox, ggnewscale, knitr, rmarkdown, survival, cli

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.1

**Repository** <https://shixiangwang.r-universe.dev>

**RemoteUrl** <https://github.com/ShixiangWang/polar>

**RemoteRef** HEAD

**RemoteSha** e6ee5025db8af0763d9647f4ac3f1021298fa613

## Contents

polar_connect . . . . .	2
polar_init . . . . .	2
<b>Index</b>	<b>4</b>

---

polar_connect	<i>Connects dots</i>
---------------	----------------------

---

**Description**

Check [polar\\_init\(\)](#) for examples.

**Usage**

```
polar_connect(data, x1, x2, ...)
```

**Arguments**

data	a data.frame contains connections of all events.
x1, x2	the column names (without quote) storing connected events.
...	parameters passing to <a href="#">ggplot2::geom_segment</a> , expect c(x, xend, y, yend) these 4 mapping parameters.

**Value**

a ggplot object.

---

polar_init	<i>Init a dot plot in polar system</i>
------------	----------------------------------------

---

**Description**

Init a dot plot in polar system

**Usage**

```
polar_init(data, x, ...)
```

**Arguments**

data	a data.frame contains all events, e.g., genes.
x	the column name (without quote) storing event list.
...	parameters passing to <a href="#">ggplot2::geom_point</a> .

**Value**

a ggplot object.

**Examples**

```

# -----
# Init a polar plot
# -----

data <- data.frame(x = LETTERS[1:7])

p1 <- polar_init(data, x = x)
p1

# Set aes value
p2 <- polar_init(data, x = x, size = 3, color = "red", alpha = 0.5)
p2

# Set aes mapping
set.seed(123L)
data1 <- data.frame(
  x = LETTERS[1:7],
  shape = c("r", "r", "r", "b", "b", "b", "b"),
  color = c("r", "r", "r", "b", "b", "b", "b"),
  size = abs(rnorm(7))
)
# Check https://ggplot2.tidyverse.org/reference/geom\_point.html
# for how to use both stroke and color
p3 <- polar_init(data1, x = x, aes(size = size, color = color, shape = shape), alpha = 0.5)
p3

# -----
# Connect polar dots
# -----
data2 <- data.frame(
  x1 = LETTERS[1:7],
  x2 = c("B", "C", "D", "E", "C", "A", "C"),
  color = c("r", "r", "r", "b", "b", "b", "b")
)
p4 <- p3 + polar_connect(data2, x1, x2)
p4

# Unlike polar_init, mappings don't need to be included in aes()
p5 <- p3 + polar_connect(data2, x1, x2, color = color, alpha = 0.8, linetype = 2)
p5

# Use two different color scales
if (requireNamespace("ggnewscale")) {
  library(ggnewscale)
  p6 = p3 +
    new_scale("color") +
    polar_connect(data2, x1, x2, color = color, alpha = 0.8, linetype = 2)
  p6 + scale_color_brewer()
  p6 + scale_color_manual(values = c("darkgreen", "magenta"))
}

```

# Index

ggplot2::geom\_point, [2](#)  
ggplot2::geom\_segment, [2](#)  
  
polar\_connect, [2](#)  
polar\_init, [2](#)  
polar\_init(), [2](#)