

# Package: ggBubbles (via r-universe)

November 4, 2024

**Type** Package

**Title** Mini Bubble Plots for Comparison of Discrete Data with 'ggplot2'

**Version** 0.1.4

**VignetteBuilder** knitr

**Depends** R (>= 3.5.0)

**Imports** dplyr, ggplot2

**Suggests** BiocStyle, knitr, rmarkdown, tibble

**Description** When comparing discrete data mini bubble plots allow displaying more information than traditional bubble plots via colour, shape or labels. Exact overlapping coordinates will be transformed so they surround the original point circularly without overlapping. This is implemented as a `position_surround()` function for 'ggplot2'.

**License** LGPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Thomas Schwarzl [aut, cre]  
(<https://orcid.org/0000-0001-7697-7000>)

**Maintainer** Thomas Schwarzl <[schwarzl@embl.de](mailto:schwarzl@embl.de)>

**Repository** CRAN

**Date/Publication** 2019-09-04 08:20:06 UTC

## Contents

<code>calc_offset</code> . . . . .	2
<code>get_offset_table</code> . . . . .	2
<code>MusicianInterests</code> . . . . .	3
<code>MusicianInterestsSmall</code> . . . . .	3
<code>PositionSurround</code> . . . . .	4
<code>position_surround</code> . . . . .	4

**Index****5**


---

calc_offset	<i>Calculate offsets for a specific point, in a layer, position</i>
-------------	---

---

**Description**

each side has several layers, with a number of positions in the layer

**Usage**

```
calc_offset(position, layer, side, offset_x = 0.1, offset_y = 0.1)
```

**Arguments**

position	number for position at the particular side on the layer
layer	number of layer
side	side for offset 1 - top 2 - right 3 - bottom 4 - left
offset_x	offset for x axis
offset_y	offset for y axis

**Value**

integer vector of length 2 position 1 is new x value, position y is new y value

---

get_offset_table	<i>Calculates offset table for number of maximum overlapping positions</i>
------------------	--

---

**Description**

Calculates offset table for number of maximum overlapping positions

**Usage**

```
get_offset_table(max_positions, offset_x, offset_y)
```

**Arguments**

max_positions	number of maximal exact overlaps
offset_x	offset for position distance
offset_y	offset for in-between layer distance

**Value**

data frame with position, offsets\_x and offsets\_y

---

MusicianInterests      *Survey about genre interests of some hobby musicians*

---

**Description**

Tibble of what genre they are interested in, what instrument they play and what level they play their instrument at (1 = beginner, 2 = intermediate, 3 = experienced, 4 = very experienced, 5 = pro). Also there is an ID for the musician.

**Usage**

```
data(MusicianInterests)
```

**Format**

An object of class "data.frame";

**Examples**

```
library(ggBubbles)
data(MusicianInterests)
head(MusicianInterests)
```

---

MusicianInterestsSmall  
*Small test data of musician, interest and experience study*

---

**Description**

Data.frame of what genre they are interested in, what instrument they play and what level they play their instrument at.

**Usage**

```
data(MusicianInterestsSmall)
```

**Format**

An object of class "data.frame";

**Examples**

```
library(ggBubbles)
data(MusicianInterestsSmall)
head(MusicianInterestsSmall)
```

---

PositionSurround	<i>ggproto for position_surround()</i>
------------------	--

---

### Description

ggproto for position\_surround()

---

position_surround	<i>Surrounds exact overlapping points around the center</i>
-------------------	---

---

### Description

Bubble plots sometimes can be hard to interpret, especially if you want to overlay an additional feature. Instead of having to colour one blob with this function you can plot the individuals contributing to the bubble and colour them accordingly.

### Usage

```
position_surround(offset = 0.1)
```

### Arguments

offset	setting offset for x and y axis added to the points surrounding the exact position. Default is 0.1
--------	---

### Value

ggproto

### Examples

```
library(ggplot2)
library(ggBubbles)
data(MusicianInterestsSmall)

ggplot(data = MusicianInterestsSmall, aes(x = Instrument, y = Genre, col = Level)) +
  geom_point(position = position_surround(), size = 4) +
  scale_colour_manual(values = c("#333333", "#666666", "#999999", "#CCCCCC")) + theme_bw()
```

# Index

## \* datasets

MusicianInterests, 3

MusicianInterestsSmall, 3

PositionSurround, 4

## \* position adjustments

position\_surround, 4

calc\_offset, 2

get\_offset\_table, 2

MusicianInterests, 3

MusicianInterestsSmall, 3

position\_surround, 4

PositionSurround, 4