# Package 'ggchord2'

July 11, 2026

**Title** Chord Diagrams with 'ggplot2'

**Version** 0.0.1

**Description** A 'ggplot2' extension that provides functions for drawing chord diagrams for visualising flows between categories. The package extends 'ggplot2' by adding geoms and stats for drawing chord sectors, arcs, and labels.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** geomtextpath, ggplot2 (>= 3.4.0), rlang

**Suggests** dplyr, knitr, rmarkdown

**RoxygenNote** 7.3.3

**VignetteBuilder** knitr

**URL** https://nrennie.gitlab.io/ggchord2/

**NeedsCompilation** no

**Author** Nicola Rennie [aut, cre, cph]

**Maintainer** Nicola Rennie <nrennie.research@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-07-11 09:20:07 UTC

## Contents

geom_chord_arcs *Chord diagram arc ribbons Draws the ribbon (arc) layer of a chord diagram.*

## Description

Chord diagram arc ribbons Draws the ribbon (arc) layer of a chord diagram.

## Usage

```
geom_chord_arcs(
  mapping = NULL,
  data = NULL,
  stat = "chord",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  gap_degree = 2,
  start_degree = 90,
  direction = 1,
  inner_radius = 0.6,
  outer_radius = 0.75,
  n_bezier = 100,
  ...
)
```

## Arguments

| | |
|---|---|
| mapping | Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | The data to be displayed in this layer. There are three options:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.<br>A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)). |
| stat | The statistical transformation to use on the data for this layer. When using a geom_*() function to construct a layer, the stat argument can be used to override the default coupling between geoms and stats. The stat argument accepts the following:<br>• A Stat ggproto subclass, for example StatCount. |

- A string naming the stat. To give the stat as a string, strip the function name of the `stat_` prefix. For example, to use `stat_count()`, give the stat as `"count"`.
- For more information and other ways to specify the stat, see the layer stat documentation.

`position` A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The `position` argument accepts the following:

- The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position.
- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.

`gap_degree` Degrees of blank space between adjacent sectors. Default 2.

`start_degree` Angle (degrees) of the first sector's start edge. Default 90.

`direction` 1 = counter-clockwise sectors (default), -1 = clockwise.

`inner_radius` Radius of the inner circle (0–1 coordinate space). Default 0.6.

`outer_radius` Radius of the outer circle (0–1 coordinate space). Default 0.75.

`n_bezier` Number of points used to approximate each Bézier / arc curve. Default 100.

`...` Other arguments passed on to `layer()`'s `params` argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the `position` argument, or aesthetics that are required can *not* be passed through `....` Unknown arguments that are not part of the 4 categories below are ignored.

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, `colour = "red"` or `linewidth = 3`. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the `params`. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.
- When constructing a layer using a `stat_*()` function, the `...` argument can be used to pass on parameters to the geom part of the layer. An example

of this is `stat_density(geom = "area", outline.type = "both")`. The geom's documentation lists which parameters it can accept.

- Inversely, when constructing a layer using a `geom_*()` function, the `...` argument can be used to pass on parameters to the `stat` part of the layer. An example of this is `geom_area(stat = "density", adjust = 0.5)`. The stat's documentation lists which parameters it can accept.
- The `key_glyph` argument of `layer()` may also be passed on through `....` This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

## Value

A `ggplot2` layer.

## Examples

```
library(ggplot2)
flows <- data.frame(
  source = c("A", "A", "B", "B", "C", "C"),
  target = c("B", "C", "A", "C", "A", "B"),
  freq   = c(50, 30, 60, 25, 35, 20)
)
ggplot(
  data = flows,
  mapping = aes(
    source = source,
    target = target,
    freq = freq
  )
) +
  geom_chord_arcs()
```

---

`geom_chord_diagram` *Chord Diagram*

---

## Description

Chord Diagram

## Usage

```
geom_chord_diagram(
  mapping = NULL,
  data = NULL,
  gap_degree = 2,
  start_degree = 90,
  direction = 1,
  inner_radius = 0.6,
```

```
  outer_radius = 0.75,
  n_bezier = 100,
  show_labels = TRUE,
  arc_params = list(),
  sector_params = list(),
  label_params = list()
)
```

## Arguments

| | |
|---|---|
| mapping | A ggplot2::aes() mapping. Must supply at minimum source, target, and freq. |
| data | Data frame. If NULL, inherits from the plot. |
| gap_degree | Degrees of gap between sectors (default 2). |
| start_degree | Starting angle in degrees for the first sector (default 90). |
| direction | 1 = counter-clockwise (default), -1 = clockwise. |
| inner_radius | Normalised inner radius of the sector ring (default 0.6). |
| outer_radius | Normalised outer radius of the sector ring (default 0.75). |
| n_bezier | Smoothness: number of points per curve (default 100). |
| show_labels | Logical. Draw node labels? Default TRUE. |
| arc_params | Named list of additional parameters passed to geom_chord_arcs. |
| sector_params | Named list of additional parameters passed to geom_chord_sectors. |
| label_params | Named list of additional parameters passed to geom_chord_labels. |

## Value

A list of ggplot2 layers

## Aesthetics

geom_chord_diagram() understands the following aesthetics (required in **bold**):

- source – name of the originating node
- target – name of the receiving node
- freq – numeric flow / weight
- fill – fill colour of ribbons / sectors
- colour – border colour
- alpha – transparency (default 0.6 for ribbons, 1 for sectors)
- linewidth – border line width

## Examples

```
library(ggplot2)
flows <- data.frame(
  source = c("A", "A", "B", "B", "C", "C"),
  target = c("B", "C", "A", "C", "A", "B"),
  freq   = c(50, 30, 60, 25, 35, 20)
)
ggplot(
  data = flows,
  mapping = aes(
    source = source,
    target = target,
    freq = freq
  )
) +
  geom_chord_diagram()
```

---

`geom_chord_labels` *Chord diagram horizontal node labels Draws horizontal text labels outside the sector bands.*

---

## Description

Chord diagram horizontal node labels Draws horizontal text labels outside the sector bands.

## Usage

```
geom_chord_labels(
  mapping = NULL,
  data = NULL,
  stat = "chord",
  position = "identity",
  na.rm = FALSE,
  show.legend = FALSE,
  inherit.aes = TRUE,
  gap_degree = 2,
  start_degree = 90,
  direction = 1,
  inner_radius = 0.6,
  outer_radius = 0.75,
  n_bezier = 100,
  ...
)
```

## Arguments

| | |
|---|---|
| `mapping` | Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping. |

`data` The data to be displayed in this layer. There are three options:

If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.

A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.

A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).

`stat` The statistical transformation to use on the data for this layer. When using a `geom_*()` function to construct a layer, the `stat` argument can be used to override the default coupling between geoms and stats. The `stat` argument accepts the following:

- A `Stat` ggproto subclass, for example `StatCount`.
- A string naming the stat. To give the stat as a string, strip the function name of the `stat_` prefix. For example, to use `stat_count()`, give the stat as `"count"`.
- For more information and other ways to specify the stat, see the layer stat documentation.

`position` A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The `position` argument accepts the following:

- The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position.
- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.

`gap_degree` Degrees of blank space between adjacent sectors. Default 2.

`start_degree` Angle (degrees) of the first sector's start edge. Default 90.

`direction` 1 = counter-clockwise sectors (default), -1 = clockwise.

`inner_radius` Radius of the inner circle (0–1 coordinate space). Default 0.6.

`outer_radius` Radius of the outer circle (0–1 coordinate space). Default 0.75.

`n_bezier` Number of points used to approximate each Bézier / arc curve. Default 100.

`...` Other arguments passed on to `layer()`'s `params` argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the `position` argument, or aesthetics that are required can *not* be passed through `...`. Unknown arguments that are not part of the 4 categories below are ignored.

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, `colour = "red"` or `linewidth = 3`. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the `params`. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.
- When constructing a layer using a `stat_*()` function, the `...` argument can be used to pass on parameters to the `geom` part of the layer. An example of this is `stat_density(geom = "area", outline.type = "both")`. The geom's documentation lists which parameters it can accept.
- Inversely, when constructing a layer using a `geom_*()` function, the `...` argument can be used to pass on parameters to the `stat` part of the layer. An example of this is `geom_area(stat = "density", adjust = 0.5)`. The stat's documentation lists which parameters it can accept.
- The `key_glyph` argument of `layer()` may also be passed on through `...`. This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

## Value

A `ggplot2` layer.

## Examples

```
library(ggplot2)
flows <- data.frame(
  source = c("A", "A", "B", "B", "C", "C"),
  target = c("B", "C", "A", "C", "A", "B"),
  freq   = c(50, 30, 60, 25, 35, 20)
)
ggplot(
  data = flows,
  mapping = aes(
    source = source,
    target = target,
    freq = freq
  )
) +
  geom_chord_sectors() +
  geom_chord_arcs() +
  geom_chord_labels()
```

geom_chord_labels_curve *Chord diagram node labels with curved text Draws text labels outside the sector bands.*

## Description

Chord diagram node labels with curved text Draws text labels outside the sector bands.

## Usage

```
geom_chord_labels_curve(
  mapping = NULL,
  data = NULL,
  stat = "chord",
  position = "identity",
  na.rm = FALSE,
  show.legend = FALSE,
  inherit.aes = TRUE,
  gap_degree = 2,
  start_degree = 90,
  direction = 1,
  inner_radius = 0.6,
  outer_radius = 0.75,
  n_bezier = 100,
  ...
)
```

## Arguments

| | |
|---|---|
| mapping | Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | The data to be displayed in this layer. There are three options:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.<br>A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)). |
| stat | The statistical transformation to use on the data for this layer. When using a geom_*() function to construct a layer, the stat argument can be used to override the default coupling between geoms and stats. The stat argument accepts the following: |

- A `Stat` ggproto subclass, for example `StatCount`.
- A string naming the stat. To give the stat as a string, strip the function name of the `stat_` prefix. For example, to use `stat_count()`, give the stat as `"count"`.
- For more information and other ways to specify the stat, see the layer stat documentation.

`position` A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The `position` argument accepts the following:

- The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position.
- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.

`gap_degree` Degrees of blank space between adjacent sectors. Default 2.

`start_degree` Angle (degrees) of the first sector's start edge. Default 90.

`direction` 1 = counter-clockwise sectors (default), -1 = clockwise.

`inner_radius` Radius of the inner circle (0–1 coordinate space). Default 0.6.

`outer_radius` Radius of the outer circle (0–1 coordinate space). Default 0.75.

`n_bezier` Number of points used to approximate each Bézier / arc curve. Default 100.

`...` Other arguments passed on to `layer()`'s `params` argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the `position` argument, or aesthetics that are required can *not* be passed through `...`. Unknown arguments that are not part of the 4 categories below are ignored.

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, `colour = "red"` or `linewidth = 3`. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the `params`. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.

- When constructing a layer using a `stat_*()` function, the `...` argument can be used to pass on parameters to the geom part of the layer. An example of this is `stat_density(geom = "area", outline.type = "both")`. The geom's documentation lists which parameters it can accept.
- Inversely, when constructing a layer using a `geom_*()` function, the `...` argument can be used to pass on parameters to the `stat` part of the layer. An example of this is `geom_area(stat = "density", adjust = 0.5)`. The stat's documentation lists which parameters it can accept.
- The `key_glyph` argument of `layer()` may also be passed on through `....` This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

## Value

A `ggplot2` layer.

## Examples

```
library(ggplot2)
flows <- data.frame(
  source = c("A", "A", "B", "B", "C", "C"),
  target = c("B", "C", "A", "C", "A", "B"),
  freq   = c(50, 30, 60, 25, 35, 20)
)
ggplot(
  data = flows,
  mapping = aes(
    source = source,
    target = target,
    freq = freq
  )
) +
  geom_chord_sectors() +
  geom_chord_arcs() +
  geom_chord_labels_curve()
```

---

`geom_chord_labels_perp`

*Chord diagram perpendicular node labels Draws text labels radiating outward from the sector midpoint.*

---

## Description

Chord diagram perpendicular node labels Draws text labels radiating outward from the sector midpoint.

## Usage

```
geom_chord_labels_perp(
  mapping = NULL,
  data = NULL,
  stat = "chord",
  position = "identity",
  na.rm = FALSE,
  show.legend = FALSE,
  inherit.aes = TRUE,
  gap_degree = 2,
  start_degree = 90,
  direction = 1,
  inner_radius = 0.6,
  outer_radius = 0.75,
  n_bezier = 100,
  ...
)
```

## Arguments

| | |
|---|---|
| mapping | Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | The data to be displayed in this layer. There are three options:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.<br>A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)). |
| stat | The statistical transformation to use on the data for this layer. When using a geom_*() function to construct a layer, the stat argument can be used to override the default coupling between geoms and stats. The stat argument accepts the following:<br>• A Stat ggproto subclass, for example StatCount.<br>• A string naming the stat. To give the stat as a string, strip the function name of the stat_ prefix. For example, to use stat_count(), give the stat as "count".<br>• For more information and other ways to specify the stat, see the layer stat documentation. |
| position | A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following: |

- The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position.
- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.

`gap_degree` Degrees of blank space between adjacent sectors. Default 2.

`start_degree` Angle (degrees) of the first sector's start edge. Default 90.

`direction` 1 = counter-clockwise sectors (default), -1 = clockwise.

`inner_radius` Radius of the inner circle (0–1 coordinate space). Default 0.6.

`outer_radius` Radius of the outer circle (0–1 coordinate space). Default 0.75.

`n_bezier` Number of points used to approximate each Bézier / arc curve. Default 100.

`...` Other arguments passed on to `layer()`'s `params` argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the `position` argument, or aesthetics that are required can *not* be passed through `....` Unknown arguments that are not part of the 4 categories below are ignored.

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, `colour = "red"` or `linewidth = 3`. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the `params`. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.
- When constructing a layer using a `stat_*()` function, the `...` argument can be used to pass on parameters to the `geom` part of the layer. An example of this is `stat_density(geom = "area", outline.type = "both")`. The geom's documentation lists which parameters it can accept.
- Inversely, when constructing a layer using a `geom_*()` function, the `...` argument can be used to pass on parameters to the `stat` part of the layer. An example of this is `geom_area(stat = "density", adjust = 0.5)`. The stat's documentation lists which parameters it can accept.
- The `key_glyph` argument of `layer()` may also be passed on through `....` This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

## Value

A ggplot2 layer.

## Examples

```
library(ggplot2)
flows <- data.frame(
  source = c("A", "A", "B", "B", "C", "C"),
  target = c("B", "C", "A", "C", "A", "B"),
  freq   = c(50, 30, 60, 25, 35, 20)
)
ggplot(
  data = flows,
  mapping = aes(
    source = source,
    target = target,
    freq = freq
  )
) +
  geom_chord_sectors() +
  geom_chord_arcs() +
  geom_chord_labels_perp()
```

---

# geom_chord_sectors *Chord diagram sector bands Draws the outer sector band layer of a chord diagram.*

---

## Description

Chord diagram sector bands Draws the outer sector band layer of a chord diagram.

## Usage

```
geom_chord_sectors(
  mapping = NULL,
  data = NULL,
  stat = "chord",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  gap_degree = 2,
  start_degree = 90,
  direction = 1,
  inner_radius = 0.6,
  outer_radius = 0.75,
  n_bezier = 100,
  ...
)
```

## Arguments

| | |
|---|---|
| mapping | Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | The data to be displayed in this layer. There are three options:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.<br>A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)). |
| stat | The statistical transformation to use on the data for this layer. When using a geom_*() function to construct a layer, the stat argument can be used to override the default coupling between geoms and stats. The stat argument accepts the following:<br>• A Stat ggproto subclass, for example StatCount.<br>• A string naming the stat. To give the stat as a string, strip the function name of the stat_ prefix. For example, to use stat_count(), give the stat as "count".<br>• For more information and other ways to specify the stat, see the layer stat documentation. |
| position | A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:<br>• The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position.<br>• A string naming the position adjustment. To give the position as a string, strip the function name of the position_ prefix. For example, to use position_jitter(), give the position as "jitter".<br>• For more information and other ways to specify the position, see the layer position documentation. |
| na.rm | If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed. |
| show.legend | logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted. |
| inherit.aes | If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders(). |
| gap_degree | Degrees of blank space between adjacent sectors. Default 2. |

| | |
|---|---|
| start_degree | Angle (degrees) of the first sector's start edge. Default 90. |
| direction | 1 = counter-clockwise sectors (default), -1 = clockwise. |
| inner_radius | Radius of the inner circle (0–1 coordinate space). Default 0.6. |
| outer_radius | Radius of the outer circle (0–1 coordinate space). Default 0.75. |
| n_bezier | Number of points used to approximate each Bézier / arc curve. Default 100. |
| ... | Other arguments passed on to layer()'s params argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the position argument, or aesthetics that are required can *not* be passed through .... Unknown arguments that are not part of the 4 categories below are ignored. |

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, colour = "red" or linewidth = 3. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the params. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.
- When constructing a layer using a stat_*() function, the ... argument can be used to pass on parameters to the geom part of the layer. An example of this is stat_density(geom = "area", outline.type = "both"). The geom's documentation lists which parameters it can accept.
- Inversely, when constructing a layer using a geom_*() function, the ... argument can be used to pass on parameters to the stat part of the layer. An example of this is geom_area(stat = "density", adjust = 0.5). The stat's documentation lists which parameters it can accept.
- The key_glyph argument of layer() may also be passed on through .... This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

## Value

A ggplot2 layer.

## Examples

```
library(ggplot2)
flows <- data.frame(
  source = c("A", "A", "B", "B", "C", "C"),
  target = c("B", "C", "A", "C", "A", "B"),
  freq   = c(50, 30, 60, 25, 35, 20)
)
ggplot(
  data = flows,
  mapping = aes(
    source = source,
    target = target,
    freq = freq
  )
) +
  geom_chord_sectors()
```

stat_chord *stat_chord*

## Description

A stat layer that computes chord diagram geometry

## Usage

```
stat_chord(
  mapping = NULL,
  data = NULL,
  geom = "polygon",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  gap_degree = 2,
  start_degree = 90,
  direction = 1,
  inner_radius = 0.6,
  outer_radius = 0.75,
  n_bezier = 100,
  ...
)
```

## Arguments

| | |
|---|---|
| mapping | Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. |
| data | The data to be displayed in this layer. There are three options:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.<br>A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)). |
| geom | Use to override the default connection between geom and stat |
| position | A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:<br>• The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position. |

- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.

`gap_degree` Degrees of blank space between adjacent sectors. Default 2.

`start_degree` Angle (degrees) of the first sector's start edge. Default 90.

`direction` 1 = counter-clockwise sectors (default), -1 = clockwise.

`inner_radius` Radius of the inner circle (0–1 coordinate space). Default 0.6.

`outer_radius` Radius of the outer circle (0–1 coordinate space). Default 0.75.

`n_bezier` Number of points used to approximate each Bézier / arc curve. Default 100.

`...` Other arguments passed on to `layer()`'s `params` argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the `position` argument, or aesthetics that are required can *not* be passed through `....` Unknown arguments that are not part of the 4 categories below are ignored.

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, `colour = "red"` or `linewidth = 3`. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the `params`. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.
- When constructing a layer using a `stat_*()` function, the `...` argument can be used to pass on parameters to the `geom` part of the layer. An example of this is `stat_density(geom = "area", outline.type = "both")`. The geom's documentation lists which parameters it can accept.
- Inversely, when constructing a layer using a `geom_*()` function, the `...` argument can be used to pass on parameters to the `stat` part of the layer. An example of this is `geom_area(stat = "density", adjust = 0.5)`. The stat's documentation lists which parameters it can accept.
- The `key_glyph` argument of `layer()` may also be passed on through `....` This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

## Value

A ggplot2 stat layer.

# Index