The **wheelchart** package

Diagrams with circular or other shapes using Ti\textit{k}Z and \LaTeX3

Matthias Floré

Version 3.0 (2024/03/08)

**Abstract**

This package is based on the package \texttt{tikz} (see [5]) and can be used to draw various kinds of diagrams such as a bar chart, doughnut chart, infographic, pie chart, ring chart, square chart, sunburst chart, waffle chart and wheel chart with Ti\textit{k}Z. It provides several options to customize the diagrams. It is also possible to specify a plot for the shape of the chart. Furthermore a legend can be added and the table of contents can be displayed as one of these diagrams. Other tools for creating wheel charts or pie charts can be found in [2], [1], [4], [6] and [3].

**Contents**

4 Keys

5 Additional examples

6 Version history

References

Index

1 Usage

2 The main macro

3 Additional macros

The **wheelchart** package
1 Usage

The package \texttt{wheelchart} can be used by putting the following in the preamble.

```latex
\usepackage{wheelchart}
```

The package \texttt{wheelchart} loads the package \texttt{tikz} and the TikZ library \texttt{calc}.

Many examples in this manual use colors which can be defined by giving \texttt{dvipsnames} as an option to `\texttt{documentclass}`.
2 The main macro

\texttt{\wheelchart[\{options\}\{\wheelchart\ \data\}]}

This command can be placed inside a \texttt{tikzpicture} environment. It draws a wheelchart with \texttt{\wheelchart\ data}. With the initial settings, the \texttt{\wheelchart\ data} is a comma separated list in which each item corresponds to one slice of the wheelchart and consists of data separated by a /. The precise syntax of the \texttt{\wheelchart\ data} will be explained below. The \texttt{\options} can be given with the keys described in Section 4.

\texttt{\exampleforthismanual}

To simplify the creation of examples in this manual, we define the \texttt{\wheelchart\ data} below.

\begin{verbatim}
\gdef\exampleforthismanual{%
14/Apricot/\{A, B, C, E, K\}/north east lines/0/0/Gray, 
40/LimeGreen/\{B, C\}/grid/0/15/Black, 
20/Melon/\{A, C\}/0.5/0/none, 
16/OliveGreen/\{A, B, E, K\}/dots/0/0/none, 
28/Peach/\{A, B, C, E, K\}/fivepointed stars/0/0/Lavender, 
32/Plum/\{A, B, C, E, K\}/bricks/0/-15/none, 
50/WildStrawberry/\{B, C, E, K\}/1/0/DarkOrchid%
}
\end{verbatim}

The default wheelchart with these data is shown below.

\begin{tikzpicture}
\wheelchart{\exampleforthismanual}
\end{tikzpicture}
3 Additional macros

\texttt{\textbackslash WAngle\{\textbackslash number\}\{\textbackslash angle\ pos\}\{\textbackslash angle\ shift\}\{\textbackslash pos\}\{\textbackslash sep\}\}

This command gives the angle in degrees of the point which is constructed as follows.

1. Consider the inner start angle and the inner end angle of slice \textbackslash \textbackslash number\. This \textbackslash \textbackslash number is computed modulo the total number of slices. Form the convex combination of these two angles with parameter \textbackslash \textbackslash angle\ pos\. Then add \textbackslash \textbackslash angle\ shift\. Then consider the point with this angle and as radius the inner radius.

2. Consider the similar point constructed with the outer start angle, the outer end angle and the outer radius of slice \textbackslash \textbackslash number\. Then construct the straight line between those two points.

3. Consider the radius given by the command \texttt{\textbackslash WCRadius} with arguments \textbackslash \textbackslash number, \textbackslash \textbackslash pos\} and \textbackslash \textbackslash sep\}. 

4. Consider the intersection of the previous line and the arc with the previous radius. The command \texttt{\textbackslash WAngle} gives the angle in degrees of this point.

The default value for \textbackslash \textbackslash number is \texttt{\textbackslash WCount}. 

The command \texttt{\textbackslash WAngle} can be used in the \textbackslash \textbackslash options of the command \texttt{\wheelchart}. It can also be used after the command \texttt{\wheelchart}. In that case, the computed angles will correspond to the last \texttt{\wheelchart}.

The command \texttt{\textbackslash WAngle} should not be used with a plot.

These can be referenced after the command \texttt{\wheelchart}.

The outer end angle of slice 2 is 236.18°.

The outer radius of slice 3 is 3.2.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{chart}
\caption{Wheelchart with \texttt{\textbackslash WAngle} command used.}
\end{figure}
\usepackage{siunitx}
\begin{tikzpicture}
\sisetup{round-mode=places,round-precision=2}
\wheelchart[
  counterclockwise,
  data=slice \WCcount\,
  inner start angle: \ang{\WCangle{0}{0}{0}{0}}\,
  inner end angle: \ang{\WCangle{1}{0}{0}{0}}\,
  outer start angle: \ang{\WCangle{0}{0}{1}{0}}\,
  outer end angle: \ang{\WCangle{1}{0}{1}{0}}\,
  inner radius: \WCradius{0}{0}\,
  outer radius: \WCradius{1}{0},
  gap=0.2,
  inner radius{list}=(2.2,2,1.8),
  legend entry=
  \fill[Dandelion] (\WCcoordinate{\WCcount +1}{inner start}) circle[radius=4pt];
  \fill[RubineRed] (\WCcoordinate{inner end}) circle[radius=4pt];
  \fill[YellowGreen] (\WCcoordinate{\WCcount +1}{outer start}) circle[radius=4pt];
  \fill[RoyalPurple] (\WCcoordinate{outer end}) circle[radius=4pt];
  \draw[-,dashed] (\WCpoint{\WCcount -1}{0.6}{0}{0.5}{0})--(\WCpoint{0.4}{0}{0.5}{0});
  \draw[dashed] (\WCcoordinate{inner end})--(\WCcoordinate{inner end})
    --(\WCpoint{1}{0}{1}{0}\WClistsep) coordinate (A);
  \draw[dashed] (\WCcoordinate{\WCcount +1}{inner start})
    --(\WCpoint{\WCcount +1}{0}{0}{0}{1}) coordinate (B);
  \draw[<->] (A)--(B) node[\WClistpos,midway] {gap};
],
  outer radius{list}=(2.8,3,3.2),
  slices style{list}={Gray!25,Gray!50,Gray!75},
  start angle=0,
  total count=3,
  \WClistpos={above left,below left,right},
  \WClistsep={1.2,1.2,0.6}
]{}
\node[align=left] (N) at (-1.5,-6.5) {%
  The outer end angle of slice 2 is \ang{\WCangle[2]{1}{0}{1}{0}}.\%
  The outer radius of slice 3 is \WCradius[3]{1}{0}.\%
};
\draw[->] (\WCcoordinate[2]{outer end})--(N) node[left, pos=0.7, align=left]
  These can be referenced\after the command \textbackslash wheelchart;\%
\draw[->] (\WCpoint{3}{0.2}{0}{1}{0})--(N);
\end{tikzpicture}

\textbf{\WCcoordinate\{\textit{number}\}\{\textit{name}\}}
\begin{itemize}
  \item If the key \textit{discrete} is false then this command gives the coordinate positioned at \langle \textit{name} \rangle of slice \langle \textit{number} \rangle. The \langle \textit{name} \rangle can be \textit{inner end}, \textit{inner start}, \textit{outer end} or \textit{outer start}.
  \item If the key \textit{discrete} is true then this command gives the coordinate positioned at point \langle \textit{name} \rangle of slice \langle \textit{number} \rangle. The \langle \textit{name} \rangle can be an integer from 1 till the number of points of slice \langle \textit{number} \rangle.
\end{itemize}

The \langle \textit{number} \rangle is computed modulo the total number of slices.
The default value for \langle \textit{number} \rangle is \WCcount.
The command \texttt{\WCcoordinate} can be used in the \langle \textit{options} \rangle of the command \texttt{\wheelchart}. It can also be used after the command \texttt{\wheelchart}. In that case, the coordinate will correspond to the last \texttt{\wheelchart}.

\textbf{\WCcount}
This macro gives the current number of the slice in the \textit{\wheelchart data}.

\textbf{\WCcountdiscrete}
If the key \textit{discrete} is true then this macro gives the current number of the Ti\kZ pic from the key \textit{discrete pic}.

\textbf{\WDataangle}
This macro is similar to \texttt{\WDataangle} but also takes into account the keys \texttt{data angle pos}, \texttt{data angle shift}, \texttt{data pos} and \texttt{data sep} (with respect to the key \texttt{counterclockwise}).

\textbf{\WCetocthelinkedname}
\textbf{\WCetocthelinkednumber}
These macros are defined when the key \texttt{etoc level} is used.

\textbf{\texttt{\textbackslash WLegend}}

If the key \texttt{legend row} is used then the resulting legend is stored in the macro \texttt{\textbackslash WLegend}.

\textbf{\texttt{\textbackslash WList}(name)}

This macro is defined when the key \texttt{\textbackslash WList}(name) is used and gives the element in the \langle list \rangle given to the key \texttt{\textbackslash WList}(name) with an index \texttt{\textbackslash WCount} modulo the length of this \langle list \rangle. The \langle name \rangle is the one given to the key \texttt{\textbackslash WList}(name).

\textbf{\texttt{\textbackslash WMidangle}}

This macro gives the angle in degrees modulo 360 of the middle of the current slice.

\begin{tikzpicture}
\wheelchart[
  data angle shift=422.45,
  data style={
    rotate=\WCdataangle,
    draw=Magenta,
    fill=GreenYellow,
    anchor=west,
    text=Gray
  },
  inner data={\ang{\WCmidangle}},
  inner data style={
    rotate=\WCmidangle,
    font=\ttfamily
  }
]{\exampleforthismanual}
\end{tikzpicture}

\textbf{\texttt{\textbackslash WPerct}}

This macro displays \texttt{\textbackslash WPercentagerounded} followed by a \% symbol.

If the package \texttt{siunitx} is loaded then the following code is used outside the key \texttt{arc data}. The package \texttt{siunitx} can be loaded before or after the package \texttt{wheelchart}.

\begin{verbatim}
\qty{\WPercentagerounded}{\percent}
\end{verbatim}

If the package \texttt{siunitx} is not loaded then the following code is used outside the key \texttt{arc data}.

\begin{verbatim}
\WPercentagerounded,\%
\end{verbatim}

Inside the key \texttt{arc data}, the following code is used.

\begin{verbatim}
\WPercentagerounded(\,)(\,\%)
\end{verbatim}
\WCpercentage
This macro gives the percentage of the current slice where the total is computed with the values of the key \texttt{value}. Note that rounding errors can occur.

\begin{tikzpicture}
\wheelchart[
data=\WCvarC||\WCperc,\slices style={\WCvarB!*\fpeval{4*\WCpercentage}}]
\exampleforthismanual\end{tikzpicture}

\WCpercentagerounded
This macro displays \WCpercentage rounded up to the number of decimals determined by the key \texttt{perc precision}.

\WCpoint\{(\textit{number})\}\\\{(\textit{angle pos})\}\\\{(\textit{shift})\}\\\{(\textit{pos})\}\\\{(\textit{sep})\}
This command gives the point where the angle is determined by \WCangle and the radius by \WCradius computed with the given arguments.
The \texttt{(number)} is computed modulo the total number of slices.
The default value for \texttt{(number)} is \WCcount.
The command \WCpoint can be used in the \texttt{(options)} of the command \wheelchart. It can also be used after the command \wheelchart. In that case, the point will correspond to the last \wheelchart.
The command \WCpoint should not be used with a plot.

\WCradius\{(\textit{number})\}\\\{(\textit{pos})\}\\\{(\textit{sep})\}
This command gives the convex combination with parameter \texttt{(pos)} of the inner radius of slice \texttt{(number)} minus \texttt{(sep)} and the outer radius of slice \texttt{(number)} plus \texttt{(sep)}.
The \texttt{(number)} is computed modulo the total number of slices.
The default value for \texttt{(number)} is \WCcount.
The command \WCradius can be used in the \texttt{(options)} of the command \wheelchart. It can also be used after the command \wheelchart. In that case, the computed radius will correspond to the last \wheelchart.
The command \WCradius should not be used with a plot.

\WCtotalcount
This macro gives the total number of slices.

\WCtotalnum
This macro gives the sum of all values of the key \texttt{value}. 

\usepackage{siunitx}
\begin{tikzpicture}
\wheelchart[
data=\WCvarC||\WCperc,\slices style={\WCvarB!*\fpeval{4*\WCpercentage}}]
\exampleforthismanual\end{tikzpicture}
\begin{tikzpicture}
\wheelchart[
  data=\\WCvarC: \WCvarA,
  middle={%
    \textbf{\Huge Fruit}\%\n    \WCtotalcount{} species\%
    \WCtotalnum{} pieces%
  }\]
\end{tikzpicture}

The \texttt{\wheelchart data} in the command \texttt{\wheelchart} is a list in which the items are separated by the value of the key \texttt{separator rows}. Each item in this list corresponds to one slice of the wheelchart and consists of data separated by the value of the key \texttt{separator columns}. With the initial settings, these individual data are interpreted as the macros \texttt{\WCvarA}, \texttt{\WCvarB}, \texttt{\WCvarC}, ..., \texttt{\WCvarZ}, \texttt{\WCvarAA} and so on and can be accessed within the \texttt{\langle options\rangle} of the command \texttt{\wheelchart} if applicable.

The names of these macros can be specified with \texttt{\langle prefix\rangle} and \texttt{\langle name\rangle} which are determined by respectively the keys \texttt{header prefix} and \texttt{header}.

Initially, only \texttt{\WCvarA}, \texttt{\WCvarB} and \texttt{\WCvarC} are used for \texttt{value}=\texttt{\WCvarA}, \texttt{slices style}=\texttt{\WCvarB} and \texttt{data}=\texttt{\WCvarC}.

Other ways to specify data are by using for example a list such as an array with the package \texttt{tikz}, a list with the package \texttt{listofitems} or with the key \texttt{\WClist{name}}.
4 Keys

The keys in this Section can be given as \textit{(options)} to the command \texttt{\textbackslash wheelchart}. If applicable, an optional non-empty \texttt{(range)} between braces can be given to a key after the \textit{(key name)} except for the key \texttt{slices} where the \texttt{(range)} is mandatory. This \texttt{(range)} is processed with \texttt{\textbackslash foreach} with the option \texttt{parse=true}. Hereafter the elements are processed with \texttt{\textbackslash fp\_eval:n}. If such a \texttt{(range)} is given to a key then the options given to this key will only be applied to a slice if the number of the slice is in the \texttt{(range)}. The \texttt{(range)} only makes sense for a key which is processed for each slice. For example, the \texttt{(range)} does not make sense for the key \texttt{middle}.

Furthermore, it is possible to add \texttt{\{list\}} after the \textit{(key name)}. Then a list can be given to the key. This list is processed analogously as how the key \texttt{WCList\{name\}} works. Then the result is given to the key.
Below are some examples for the options \texttt{\{range\}} and \texttt{\{list\}}.

- The following wheelchart can be obtained with the 3 possibilities below.

\begin{tikzpicture}
\wheelchart[
  data\{list\}={
    An,example,where,some,of,the,
    keys,are,given,using,a,list
  },
  slices\{list\}={
    Thistle,Orchid,Fuchsia
  },
  total\ count=12
]
\end{tikzpicture}

\usepackage{listofitems}
\readlist\WCcolors{Thistle,Orchid,Fuchsia}
\setsepchar{ }
\readlist\WCdata{An example where some of the keys are given using a list}
data={\WCdata\{\WCcount\}},
slices\{list\}={
  /utils/exec={\pgfmathsetmacro\WCcolornumber{int(Mod(\WCcount-1,\WCcolorslen)+1)}},
  \WCcolors\{\WCcolornumber\}
},
\WCdata\{\WCdatalen\},
slices\{1,4,\ldots,\WCdatalen\}=Thistle,
slices\{2,5,\ldots,\WCdatalen\}=Orchid,
slices\{3,6,\ldots,\WCdatalen\}=Fuchsia,

- The following wheelchart can be obtained with the 3 possibilities below.
The \texttt{explode} key determines the distance of a slice from the center of the wheelchart. It can be used to create a pie chart with slices that are not evenly spaced. The \texttt{wheelchart/after slices} key is used to specify code that will be executed after each slice of the wheelchart. The \texttt{wheelchart/anchor xsep} and \texttt{wheelchart/anchor ysep} keys determine the default anchor of the key data in the case that \texttt{lines ext}=0. Note that rounding errors can occur in the computation of the angle which is used to determine the default anchor according to Table 1.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Angle (up to rounding errors) & Anchor of the key data in the case that \texttt{lines ext}=0 \\
\hline
0 & west \\
90 & south \\
180 & east \\
270 & north \\
For other angles not in \{0, 90, 180, 270\}: & \\
[0, anchor ysep] & west \\
[anchor ysep, 90 – anchor xsep] & south west \\
[90 – anchor xsep, 90 + anchor xsep] & south \\
[90 + anchor xsep, 180 – anchor ysep] & south east \\
[180 – anchor ysep, 180 + anchor xsep] & east \\
[180 + anchor ysep, 270 – anchor xsep] & north east \\
[270 – anchor xsep, 270 + anchor xsep] & north \\
[270 + anchor xsep, 360 – anchor ysep] & north west \\
[360 – anchor ysep, 360] & west \\
\hline
\end{tabular}
\caption{Anchor of the key data in the case that \texttt{lines ext}=0.}
\end{table}
The anchor of the key `data` can also be specified manually by using `data style={anchor=⟨anchor⟩}`.

```
/wheelchart/arc={⟨options⟩} (style, no default, initially empty)
```

If this key is set then an arc with the style determined by this key will be drawn following the arc or plot for a slice of the wheelchart.

```
/wheelchart/arc around line={⟨number⟩} (no default, initially 1)
```

The contents of the key `arc data` can consist of multiple lines separated by `\`. If the key `arc around text` is true then the corresponding line is determined by `⟨number⟩`.

```
/wheelchart/arc around text=⟨boolean⟩ (default true, initially false)
```

If true then the arc with the style determined by the key `arc` will be split in two parts such that the gap between these two parts leaves space for the contents of line `⟨number⟩` of the key `arc data` where `⟨number⟩` is determined by the key `arc around line`. The space between the arc and the contents of the key `arc data` can be increased with for example `- in arc data=-text-`.

```
/wheelchart/arc data={⟨text⟩} (no default, initially empty)
```

This key contains the `⟨text⟩` which will be placed following the arc or plot for a slice of the wheelchart using the decoration `text along path`. This requires the TikZ library `decorations.text`. The style of this decoration is given as follows. First, the option `raise=-0.5ex` is given. Then `text align` is determined by the key `arc data align`. Thereafter, the style of the key `arc data style` is added.

The `⟨text⟩` can consist of multiple lines separated by `\`. Braces or multiple pairs of braces are required around some macros.

```
/wheelchart/arc data align=⟨center|left|right⟩ (no default, initially center)
```

This key determines the alignment of the contents of the key `arc data`.

```
/wheelchart/arc data angle pos=⟨value⟩ (no default, initially 0.5)
```

```
/wheelchart/arc data angle shift=⟨angle⟩ (no default, initially 0)
```

These keys determine the position of the contents of the key `arc data` similar as the corresponding keys for the key `data`.

---

The anchor of the key `data` can also be specified manually by using `data style={anchor=⟨anchor⟩}`.

```
/wheelchart/arc={⟨options⟩} (style, no default, initially empty)
```

If this key is set then an arc with the style determined by this key will be drawn following the arc or plot for a slice of the wheelchart.

```
/wheelchart/arc around line={⟨number⟩} (no default, initially 1)
```

The contents of the key `arc data` can consist of multiple lines separated by `\`. If the key `arc around text` is true then the corresponding line is determined by `⟨number⟩`.

```
/wheelchart/arc around text=⟨boolean⟩ (default true, initially false)
```

If true then the arc with the style determined by the key `arc` will be split in two parts such that the gap between these two parts leaves space for the contents of line `⟨number⟩` of the key `arc data` where `⟨number⟩` is determined by the key `arc around line`. The space between the arc and the contents of the key `arc data` can be increased with for example `- in arc data=-text-`.

```
/wheelchart/arc data={⟨text⟩} (no default, initially empty)
```

This key contains the `⟨text⟩` which will be placed following the arc or plot for a slice of the wheelchart using the decoration `text along path`. This requires the TikZ library `decorations.text`. The style of this decoration is given as follows. First, the option `raise=-0.5ex` is given. Then `text align` is determined by the key `arc data align`. Thereafter, the style of the key `arc data style` is added.

The `⟨text⟩` can consist of multiple lines separated by `\`. Braces or multiple pairs of braces are required around some macros.

```
/wheelchart/arc data align=⟨center|left|right⟩ (no default, initially center)
```

This key determines the alignment of the contents of the key `arc data`.

```
/wheelchart/arc data angle pos=⟨value⟩ (no default, initially 0.5)
```

```
/wheelchart/arc data angle shift=⟨angle⟩ (no default, initially 0)
```

These keys determine the position of the contents of the key `arc data` similar as the corresponding keys for the key `data`.

---

The anchor of the key `data` can also be specified manually by using `data style={anchor=⟨anchor⟩}`.

```
/wheelchart/arc={⟨options⟩} (style, no default, initially empty)
```

If this key is set then an arc with the style determined by this key will be drawn following the arc or plot for a slice of the wheelchart.

```
/wheelchart/arc around line={⟨number⟩} (no default, initially 1)
```

The contents of the key `arc data` can consist of multiple lines separated by `\`. If the key `arc around text` is true then the corresponding line is determined by `⟨number⟩`.

```
/wheelchart/arc around text=⟨boolean⟩ (default true, initially false)
```

If true then the arc with the style determined by the key `arc` will be split in two parts such that the gap between these two parts leaves space for the contents of line `⟨number⟩` of the key `arc data` where `⟨number⟩` is determined by the key `arc around line`. The space between the arc and the contents of the key `arc data` can be increased with for example `- in arc data=-text-`.

```
/wheelchart/arc data={⟨text⟩} (no default, initially empty)
```

This key contains the `⟨text⟩` which will be placed following the arc or plot for a slice of the wheelchart using the decoration `text along path`. This requires the TikZ library `decorations.text`. The style of this decoration is given as follows. First, the option `raise=-0.5ex` is given. Then `text align` is determined by the key `arc data align`. Thereafter, the style of the key `arc data style` is added.

The `⟨text⟩` can consist of multiple lines separated by `\`. Braces or multiple pairs of braces are required around some macros.

```
/wheelchart/arc data align=⟨center|left|right⟩ (no default, initially center)
```

This key determines the alignment of the contents of the key `arc data`.

```
/wheelchart/arc data angle pos=⟨value⟩ (no default, initially 0.5)
```

```
/wheelchart/arc data angle shift=⟨angle⟩ (no default, initially 0)
```

These keys determine the position of the contents of the key `arc data` similar as the corresponding keys for the key `data`.
This key determines the direction of the contents of the key `arc data`. If the `⟨value⟩` is positive then the direction is the same as the direction of the slice. If the `⟨value⟩` is negative then the direction is reversed. The values 1 and -1 are recommended.

When the contents of the key `arc data` is placed, the corresponding domain for the arc or plot is estimated. A warning is given when the contents of the key `arc data` did (possibly) not fit. In this case, the absolute value of the key `arc data dir` should be increased.

If an error Dimension too large occurs then the absolute value of the key `arc data dir` should be increased or decreased depending on the situation.

The contents of the key `arc data` can consist of multiple lines separated by `\`. This splitting is done with \seq_set_split:Nnn or a variant thereof depending on the `⟨expansion type⟩` which determines the last letter in the signature. For most use cases, this `⟨expansion type⟩` is `n`, `e` or `f`.

In the example below, it is necessary to use `arc data expand=e` and to place `\noexpand` before `\bfseries`.

The contents of the key `arc data` can consist of multiple lines separated by `\`. The `⟨factor⟩` determines the spacing between these lines.

These keys determine the position of the contents of the key `arc data` similar as the corresponding keys for the key `data`.

This key accepts a list of keys which will be applied to the decoration for the key `arc data`.`

If `arc around text` is true then the arc with the style determined by the key `arc first half` will be appended to the first half of the arc.

This key determines the position of the arc similar as the corresponding key for the key `data`.

This key is similar to the key `arc first half` but will be appended to the second half of the arc.

This key determines the position of the arc similar as the corresponding key for the key `data`. Note that the actual distance is given by `0.5ex/1cm` plus `arc sep` to match the option `raise=-0.5ex` given to the decoration for the key `arc data`.
This key defines the center of the wheelchart.

This key contains the \texttt{(text)} which will be placed below left of the wheelchart. The \texttt{(text)} is placed in a node. This node is placed at a value determined by the key \texttt{caption \_left sep} below the \texttt{local bounding box} around the wheelchart. The style of this node is given as follows. First, the options \texttt{anchor=north west,align=left} are given. Thereafter, the style of the key \texttt{caption \_left style} is added.

The node where the contents of the key \texttt{caption left} is placed is at \texttt{value} below the south west of the \texttt{local bounding box} around the wheelchart.
\begin{tikzpicture}
\wheelchart[
  at=(5,2),
  caption=Caption,
  caption style={font=\scshape},
  caption left=Caption left,
  caption left style={font=\sffamily},
  middle=\texttt{name=WCname},
  name=WCname,
  start half,
  title=Title,
  title style={font=\bfseries},
  title left=Title left,
  title left style={font=\em}
]
\percent
1/Goldenrod/Text\ with\ multiple\ lines,
1/Mahogany/Text which is longer than the short text,
1/JungleGreen/Another text,
1/RoyalBlue/Short text
\draw[dashed] (WCname.south west) rectangle (WCname.north east);
\foreach\pos in {north,east,south,west}{
  \node at (WCname.\pos) {\pos};
}
\end{tikzpicture}

\wheelchart/contour={\langle\text\rangle} \hspace{1em} (style, no default, initially empty)

If this key is set then a contour with the style determined by this key will be drawn around the wheelchart. This requires a fixed inner and outer radius for all slices. This key does not apply if a plot is used.

\wheelchart/counterclockwise={\langle\text\rangle} \hspace{1em} (default true, initially false)

If true, the wheelchart will be drawn counterclockwise instead of clockwise.

\wheelchart/data={\langle\text\rangle} \hspace{1em} (no default, initially \WCvarC)

This key contains the \langle text \rangle which will be placed at each slice of the wheelchart. This can be suppressed
by using \texttt{data=\{\}}. The \texttt{text} is placed in a node. The style of this node is given as follows. First, the anchor is set following Table 1 and Table 3. Then the option \texttt{align=left} is added. Thereafter, the style of the key \texttt{data style} is added.

\texttt{/wheelchart/data angle pos=\{\texttt{value}\}\}} \quad \text{(no default, initially 0.5)}

\texttt{/wheelchart/data angle shift=\{\texttt{angle}\}} \quad \text{(no default, initially 0)}

\texttt{/wheelchart/data pos=\{\texttt{value}\}\}} \quad \text{(no default, initially 1)}

\texttt{/wheelchart/data sep=\{\texttt{value}\}\}} \quad \text{(no default, initially 0.2)}

If no plot is used then the position of the contents of the key \texttt{data} is determined as described for the commands \texttt{WCangle} and \texttt{WCradius}.

If a plot is used then the position of the contents of the key \texttt{data} is determined as follows.

1. The inner plot is evaluated in the point with as angle the convex combination with as parameter the key \texttt{data angle pos} of the inner start angle and the inner end angle, added with the key \texttt{data angle shift} in degrees (taking into account the key \texttt{counterclockwise}) and as radius the inner radius minus the key \texttt{data sep}.

2. The outer plot is evaluated in the similar point but using the outer start angle, the outer end angle and the outer radius plus the key \texttt{data sep}.

3. If \texttt{lines} \neq 0 then the values of the keys \texttt{lines sep} and \texttt{lines} are added to the radii above, in addition to the key \texttt{data sep}.

4. The contents of the key \texttt{data} is placed at the convex combination with as parameter the key \texttt{data pos} of the previous two points.

\begin{center}
\begin{tikzpicture}
\node at (0,0) {Lime \quad Olive \quad Plum} ;
\node at (0,-2) {Apricot \quad Melon \quad \ldots \quad Strawberry \quad Peach} ;
\node at (0,-4) {20\% \quad 10\% \quad 14\% \quad 16\% \quad 25\% \quad 8\%} ;
\end{tikzpicture}
\end{center}
\begin{tikzpicture}
\wheelchart{
  data angle pos(2)=0.3,
data angle pos(6)=0.8,
data angle shift(3)=-0.1,
data angle shift(5)=0.1,
data pos=\WClistB,
data sep=0,
lines\{1,2,4,6,7\}=0.5,
lines\{3,5\}=-1,
lines angle pos\{1\}=0.8,
lines angle shift\{7\}=-0.2,
lines ext=\WClistA,
lines ext dir\{1,...,3\}=left,
lines ext dir\{4,...,7\}=right,
lines ext fixed,
lines ext fixed left=-1,
lines ext fixed right=7,
lines pos=\WClistB,
lines sep=0.2*\WClistA,
xbar=(6\{1.5\},
\WClistA=(i,0),
\WClistB=(0,1),
wheel data=\WCperc,
wheel data pos=\WClistB,
wheel data pos\{1\}=1,
wheel data pos\{4\}=0,
wheel data sep=0.2
}\exampleforthismanual
\end{tikzpicture}

//wheelchart/data style={\langle options\rangle} (style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key data is placed.

//wheelchart/discrete={\langle boolean\rangle} (default true, initially false)

If true then TikZ pics are placed with the \langle code\rangle determined by the key discrete pic. The number of pics is determined by the key value. It is required to set the key discrete space at borders.

//wheelchart/discrete factor={\langle value\rangle} (no default, initially 1)

The algorithm to place the TikZ pics depends on the \langle value\rangle. The value 1 is recommended.

//wheelchart/discrete partitioning=angle|radius (no default, initially radius)

angle In this case, the TikZ pics are placed uniformly with respect to the angle.

radius In this case, the TikZ pics are placed uniformly with respect to the radius.

These options are illustrated in the examples below.
The \texttt{discrete pic} determines the TikZ pics.
/wheelchart/discrete sort=angle

angle In this case, the Ti\kern.8ptkZ pics are ordered with respect to the angle.

radius In this case, the Ti\kern.8ptkZ pics are ordered with respect to the radius.

These options are illustrated in the examples below.
This key determines whether space is left at the begin and end where the TikZ pics are placed. For example, suppose that 3 TikZ pics are placed at positions between 0 and 1. If `discrete space at borders` is true then these are placed at the positions \( \frac{1}{6}, \frac{3}{6} \) and \( \frac{5}{6} \). If `discrete space at borders` is false then these are placed at the positions 0, \( \frac{1}{2} \) and 1.

This key deliberately has no initial value in order to force awareness of the consequences of the settings of this key. In the example below, the cyan TikZ pics are aligned if `discrete space at borders` is false while this is not the case if `discrete space at borders` is true.

In the example below, the red and green TikZ pics overlap if `discrete space at borders` is false while
this is not the case if discrete space at borders is true.

\begin{tikzpicture}
\pgfkeys{
  /wheelchart,
  discrete,
  discrete pic={\fill (0,0) circle[radius=\WClistradius pt];},
  middle style={font=\ttfamily}
}
\wheelchart[
  discrete space at borders,
  middle={discrete space\at borders=true},
  WClistradius=4
]{2/Red/, 40/Gray/, 2/Green/}
\wheelchart[
  at={(7,0)},
  discrete space at borders=false,
  middle={discrete space\at borders=false},
  WClistradius={8,4,4}
]{2/Red/, 40/Gray/, 2/Green/}
\end{tikzpicture}

/wheelchart/domain={⟨start⟩};{⟨end⟩}  
(no default)

This key sets counterclockwise, start angle to ⟨start⟩ and total angle to ⟨end⟩ − ⟨start⟩.

/wheelchart/etoc code={⟨code⟩}  
(no default, initially \tableofcontents)

The ⟨code⟩ will be executed to build the ⟨wheelchart data⟩ if the key etoc level is used.

/wheelchart/etoc count total pages={⟨number⟩}  
(no default, initially 0)

If the key etoc level is used then the number of pages of the last section depends on ⟨number⟩ which can for example represent the total number of pages in the document or the number of pages before the start of the Appendix or the Index. For example, etoc count total pages=\totalpages can be used. To provide the command \totalpages, this requires \usepackage[page]{totalcount}, which should normally be loaded before the package wheelchart to give a correct result.

/wheelchart/etoc level={⟨level⟩}  
(no default)

If this key is used then the ⟨wheelchart data⟩ of the command \wheelchart can be left empty and is defined to match the sections of the level defined by ⟨level⟩. Here, \WCetocthelinkedname corresponds to \etocthelinkedname, \WCetocthelinkednumber to \etocthelinkednumber, \WCetocthelinkedpage to \etocthelinkedpage, \WCetocthenumber to \etocthenumber, \WCetocthenumber to \etocthenumber and \WCetocthepage to \etocthepage. The package etoc is required to provide these commands. Furthermore, \WCetocthenumberofpages corresponds to the number of pages of the current section. For the last section, this depends on the value of the key etoc count total pages.

/wheelchart/etoc name={⟨name⟩}  
(no default, initially empty)
The resulting ⟨wheelchart data⟩ from the key etoc level is stored globally and can be reused later with the key etoc use name.

/wheelchart/etoc use name={(name)} (no default)

If this key is used then the ⟨wheelchart data⟩ is reused from where etoc name has the same ⟨name⟩.

/wheelchart/expand list=false|once|true (no default, initially once)

false In this case, the ⟨wheelchart data⟩ of the command \wheelchart will not be expanded.

once In this case, the ⟨wheelchart data⟩ of the command \wheelchart will be expanded once.

true In this case, the ⟨wheelchart data⟩ of the command \wheelchart will be fully expanded.

The following example illustrates the difference between the possible values of the key expand list.

\begin{tikzpicture}
\def\WClistA{a,A}
\def\WClistB{b,B}
\def\WCdata{\WClistA,\WClistB}
\foreach\expandlist [count=\n] in {false,once,true}{
\wheelchart[at={(3.5*\n},0)},
data=\WCdata,expand list=\expandlist, radius=\n},
slices style(list)={Dandelion,CarnationPink, SpringGreen,ProcessBlue },
title={\expandlist}
\]{\WCdata}
}\end{tikzpicture}

The initial setting expand list=once works in most situations, even when commands such as \ref,\cite and \textbf are used such as in the example below.

\begin{tikzpicture}
\wheelchart{
\begin{tabular}{c}
%expand list=false,%false also works
%expand list=true,%true doesn't work
middle={expand list=\false %
\{normalfont or\ once\},
middle style={font=\large\ttfamily}
}\{TEXT\}
\end{tabular}
\}{\WCdata}
}\end{tikzpicture}

In the following example, the ⟨wheelchart data⟩ from the previous example is stored in a macro. In this case, we have to use the initial setting expand list=once.
In the example below, we have to use `expand list=true`.

This key is similar to the key `expand list` but applies to the items in the ⟨wheelchart data⟩ of the command `\wheelchart` which correspond to a slice of the wheelchart.
This key will shift the slices of the wheelchart with \textit{(value)} with respect to the center of the wheelchart.

The slices of the wheelchart, the wheel lines determined by the key \texttt{wheel lines} and the different kinds of data are placed in for loops. If the key \texttt{for loop end} is set then the \textit{(code)} given to this key will be executed at the end of the body of these for loops.

This key is similar to the key \texttt{for loop end} but the \textit{(code)} given to this key will be executed at the start of the body of the for loops.

The \textit{(value)} of this key defines half the distance between two slices of the wheelchart. This key does not apply if a plot is used.

If the value of the key \texttt{gap} is too large then a slice can partly disappear such as for example below when \texttt{gap max angle} is $155^\circ$. The \textit{(angle)} of the key \texttt{gap max angle} determines the inner arc of the slice as illustrated in the examples below.
/wheelchart/gap polar={⟨value⟩}  
(default 1, initially 0)

The ⟨value⟩ of this key defines half the polar gap in degrees between two slices of the wheelchart.

Note the difference between the keys explode, gap and gap polar. This is illustrated in the examples below.
\begin{tikzpicture}
  \wheelchart[
    gap polar=10, 
    legend entry\{1,2,3\}={
      \draw (\WCcoordinate{outer start}) --(\WCcoordinate{\WCcount +3} (outer start));
      \draw (\WCcoordinate{outer end}) --(\WCcoordinate{\WCcount +3} (outer end));
    },
    middle={\Large\emph{gap polar}},
    slices style={
      draw=Red, 
      fill=none, 
      ultra thick
    },
    total count=6
  ]{}
  \draw (0,0) circle[radius=2];
  \draw (0,0) circle[radius=3];
\end{tikzpicture}

/\wheelchart/gap radius={⟨value⟩}
  (default 0.05, initially 0)

The ⟨value⟩ of this key will be added to inner radius and subtracted from outer radius.

\begin{tikzpicture}
  \def\n{73}
  \wheelchart[
    data=, 
    gap radius=\WCvarC, 
    middle={\Huge\qty{\n}{\percent}},
   _slices style={
      draw=NavyBlue, 
      fill=none, 
      ultra thick
    },
    total count=6
  ]{
    \n/NavyBlue/0,
    \n-BurntOrange/0.2
  } \draw[Gray] (0,0) circle[radius=1.9];
\end{tikzpicture}

/\wheelchart/header={⟨list⟩}
  (no default)

The items in the ⟨list⟩ determine the names in the macros \⟨prefix⟩\langle name⟩.

/\wheelchart/header prefix={⟨prefix⟩}
  (no default, initially WC)

The ⟨prefix⟩ is used in the macros \⟨prefix⟩\langle name⟩.

/\wheelchart/inner data={⟨text⟩}
  (no default, initially empty)

This key contains the ⟨text⟩ which will be placed at each slice of the wheelchart. The ⟨text⟩ is placed in a node. The style of this node is given as follows. First, the option align=left is given. Thereafter, the style of the key inner data style is added.

/\wheelchart/inner data angle pos={⟨value⟩}
  (no default, initially 0.5)

/\wheelchart/inner data angle shift={⟨angle⟩}
  (no default, initially 0)

/\wheelchart/inner data pos={⟨value⟩}
  (no default, initially 0)

/\wheelchart/inner data sep={⟨value⟩}
  (no default, initially 0.2)

These keys determine the position of the contents of the key inner data similar as the corresponding keys for the key data. No lines are drawn for the inner data.

/\wheelchart/inner data style={⟨options⟩}
  (style, no default, initially empty)
This key accepts a list of keys which will be applied to the node where the contents of the key `inner data` is placed.

```
/\ wheelchart/inner plot={⟨code⟩}
```

(no default)

The ⟨code⟩ is a coordinate definition which will be used for the inner parts of the slices of the wheelchart. In the ⟨code⟩, #1 and #2 can be used where #1 corresponds to the angle and #2 corresponds to the radius. For example, a circle can be obtained with `inner plot={#1:{#2}}`.

```
/\ wheelchart/inner plot style={⟨options⟩}
```

(style, no default, initially empty)

This key accepts a list of keys which will be applied to the plot determined by the key `inner plot`.

```
/\ wheelchart/inner radius={⟨value⟩}
```

(no default, initially 2)

The ⟨value⟩ of this key defines the inner radius of the wheelchart.

```
/\ wheelchart/legend={⟨code⟩}
```

(no default, initially empty)

The ⟨code⟩ given to this key will be executed at the end of the command `<wheelchart>`.

```
/\ wheelchart/legend columns={⟨number⟩}
```

(no default, initially 1)

If the key `legend row` is used then the maximum number of times that the ⟨code⟩ given to the key `legend row` appears on one row is determined by ⟨number⟩. The environment (for example `tabular`, `tabularx` from the package `tabularx`, `tabulary` from the package `tabulary` or `tblr` from the package `tabularray`) which contains the macro \WClegend needs to have a suitable column specification according with ⟨number⟩ and the key `legend row`.

```
/\ wheelchart/legend entry={⟨code⟩}
```

(no default, initially empty)

The ⟨code⟩ given to this key will be executed for each slice of the wheelchart.

```
/\ wheelchart/legend only={⟨boolean⟩}
```

(default true, initially false)

If true then only the legend is constructed. This does not apply to the key `legend entry`. In this case it is not necessary to place the command `<wheelchart>` in a `tikzpicture` environment.

```latex
\usepackage{tikzlings}
\wheelchart[
    header={animal,accessory},
    legend columns=3,
    legend only, legend row=\begin{tabular}{*{3}{cl}}
        \multicolumn{6}{c}{\texttt{11 animals from the package \texttt{tikzlings}}}\
        \hline
        \WClegend\hline
    \end{tabular},
    separator columns={\{}\},
    separator rows={\},
    value=1
]{%
    bear basket;
    bee book;
    bug chef;
    cat crown;
    elephant football;
    koala handbag;
    owl hat;
    panda icecream;
    penguin milkshake;
    snowman santa;
    squirrel shovel
}27
```
/wheelchart/legend row={⟨code⟩} (no default)

If this key is set then a legend consisting of rows for an environment such as \texttt{tabular}, \texttt{tabularx} from the package \texttt{tabularx}, \texttt{tabulary} from the package \texttt{tabulary} or \texttt{tblr} from the package \texttt{tabularray} is constructed using the ⟨\texttt{code}⟩ for each slice of the wheelchart.

If a \texttt{tblr} environment from the package \texttt{tabularray} is used then the option \texttt{expand=\WClegend} needs to be given to this \texttt{tblr} environment and \texttt{\UseTblrLibrary{counter}} is required.

The maximum number of times that the ⟨\texttt{code}⟩ appears on one row is determined by the key \texttt{legend columns}.

The code automatically inserts \& and \\ after the ⟨\texttt{code}⟩ if necessary.

The result is stored in the macro \texttt{\WClegend}.

\begin{tabular}{llS[table-format=3.0]S[table-format=2.0{\,\unit{\%}}]l}
\hline
\textbf{Fruit} & \textbf{Value} & \textbf{Percentage} & \textbf{Vitamins} \\
\hline
Apricot & 14 & 7\% & A, B, C, E, K \\
Lime & 40 & 20\% & B, C \\
Melon & 20 & 10\% & A, C \\
Olive & 16 & 8\% & A, B, E, K \\
Peach & 28 & 14\% & A, B, C, E, K \\
Plum & 32 & 16\% & A, B, C, E, K \\
Strawberry & 50 & 25\% & B, C, E, K \\
\textbf{Total} & 200 & & \\
\hline
\end{tabular}
The \( \langle \text{value} \rangle \) is used in the positioning of the contents of the key \textbf{data}. The end point of the lines is determined similarly but without the key \textbf{data sep}.

If the \( \langle \text{value} \rangle \) of this key is nonzero and \textbf{lines ext} fixed is false then the lines between the wheelchart and the contents of the key \textbf{data} will be extended horizontally with a length defined by \( \langle \text{value} \rangle \).

The default direction in which the lines between the wheelchart and the contents of the key \textbf{data} will be extended horizontally if \textbf{lines ext} \( \neq 0 \) is determined by Table 2 and illustrated in the following example. This can be overruled by giving an explicit value to this key. Note that rounding errors can occur in the computation of the angle which is used to determine the default direction according to Table 2.

The direction in which the lines between the wheelchart and the contents of the key \textbf{data} will be extended horizontally if \textbf{lines ext} \( \neq 0 \) and if the key \textbf{lines ext dir} is not used.

<table>
<thead>
<tr>
<th>Angle (up to rounding errors)</th>
<th>right</th>
<th>value of the key lines ext top dir</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 90 – lines ext dir sep</td>
<td>right</td>
<td>value of the key lines ext bottom dir</td>
</tr>
<tr>
<td>90 + lines ext dir sep, 90 + lines ext dir sep</td>
<td>left</td>
<td>right</td>
</tr>
<tr>
<td>90 + lines ext dir sep, 270 – lines ext dir sep</td>
<td>left</td>
<td>right</td>
</tr>
<tr>
<td>270 + lines ext dir sep, 360</td>
<td>right</td>
<td>right</td>
</tr>
</tbody>
</table>

Table 2: The direction in which the lines between the wheelchart and the contents of the key \textbf{data} will be extended horizontally if \textbf{lines ext} \( \neq 0 \) and if the key \textbf{lines ext dir} is not used.
This key determines half the angle in degrees of the segment to which the keys `lines ext bottom dir` and `lines ext top dir` apply.

This key determines half the angle in degrees of the segment to which the keys `lines ext bottom dir` and `lines ext top dir` apply.

If true, the line between the wheelchart and the contents of the key `data` will be extended horizontally till the $x$ coordinate determined by the keys `lines ext fixed left` and `lines ext fixed right`.

If `lines ext fixed` is true, the lines are extended horizontally initially to the right till the $x$ coordinate `outer radius + lines sep + lines ext` and to the left till the opposite of this $x$ coordinate. This can be overruled by giving an explicit value to the key `lines ext fixed left` and/or `lines ext fixed right`.

The direction in which the lines between the wheelchart and the contents of the key `data` will be extended horizontally if `lines ext ≠ 0`.

<table>
<thead>
<tr>
<th>Anchor of the key <code>data</code></th>
<th>Value of the key <code>lines ext left anchor</code></th>
<th>Value of the key <code>lines ext right anchor</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>left</td>
<td></td>
<td></td>
</tr>
<tr>
<td>right</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Anchor of the key `data` in the case that `lines ext ≠ 0`. 
The keys similar to the corresponding keys for data but determine the start point of the lines.

This key accepts a list of keys which will be applied to the lines drawn by the key lines.
\usepackage{siunitx} \usetikzlibrary{decorations.markings}
\begin{tikzpicture}
\wheelchart[
data=\WCperc,\ndata angle pos=0.2,
data style={outer xsep=4pt},legend columns=2,legend row={\begin{tabular}{*{2}{l@{ }lr}}\WClegend\end{tabular}},legend={\begin{itemize} \item \node[anchor=north,draw,rounded corners,thick] at (0,-4.5) {\begin{tabular}{*{2}{l@{ }lr}}\WClegend\end{tabular}}; \end{itemize} },
\begin{tabular}{lr}
\exampleforthismanual
\end{tabular}]
\end{tikzpicture}

Apricot Lime Melon Olive Peach Plum Strawberry

\begin{tikzpicture}
\wheelchart[
data sep=0,\data style={inner sep=0pt,shift={(0,0.1)}},
\begin{tabular}{lr}
\exampleforthismanual
\end{tabular}]
\end{tikzpicture}

\begin{tikzpicture}
\wheelchart[
data=,\radius={1.5}(2),\slices style=\WCvarB!70,\start angle=331.2]
\exampleforthismanual
\end{tikzpicture}

\begin{tikzpicture}
\wheelchart[
data sep=0,\data style={inner sep=0pt,shift={(0,0.1)}},
\begin{tabular}{lr}
\exampleforthismanual
\end{tabular}]
\end{tikzpicture}

\begin{tikzpicture}
\wheelchart[
data sep=0,\data style={inner sep=0pt,shift={(0,0.1)}},
\begin{tabular}{lr}
\exampleforthismanual
\end{tabular}]
\end{tikzpicture}
\usetikzlibrary{decorations.markings}
\begin{tikzpicture}
\wheelchart[
  data angle shift=\WCvarG,
  data sep=0,
  data style={draw=\WCvarB,fill=\WCvarB!20},
  lines=1.5,
  lines ext=1,
  lines sep=-1,
  lines style={
    Black,
    postaction=decorate,
    decoration={
      markings,
      mark=at position 0 with {\fill[Black] (0,0) circle[radius=0.15];}
    }
  },
  pie,
  start angle=331.2
]{\exampleforthismanual}
\end{tikzpicture}

/wheelchart/middle={⟨text⟩}  
(no default, initially empty)

This key contains the ⟨text⟩ which will be placed at the center of the wheelchart. The ⟨text⟩ is placed in a node. The style of this node is given as follows. First, the option align=center is given. Thereafter, the style of the key middle style is added.

/wheelchart/middle fill={⟨options⟩}  
(style, no default, initially empty)

If this key is set then the middle of the wheelchart will be filled with this style. This requires a fixed inner and outer radius for all slices. This key does not apply if a plot is used.

/wheelchart/middle style={⟨options⟩}  
(style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key middle is placed.

/wheelchart/name={⟨name⟩}  
(no default, initially wheelchart@name)

This key defines the ⟨name⟩ of the local bounding box around the wheelchart.

/wheelchart/outer plot={⟨code⟩}  
(no default)

This key is similar to the key inner plot but determines the outer parts of the slices of the wheelchart.
\begin{tikzpicture}
\wheelchart[
    inner plot={\WCcount*5/\WCtotalcount+(\WCtotalcount/2)*\WCvarC},
    outer plot={\WCcount*5/\WCtotalcount-(\WCtotalcount/2)*\WCvarC},
] \exampleforthismanual
\end{tikzpicture}

\begin{tikzpicture}
\usetikzlibrary{decorations.text}
\wheelchart[
    arc data=slice \WCcount/\WCtotalcount/\WCvarC, arc data expand=f,
    arc data pos=0.5, arc data line sep factor=1.5,
    data=, domain=0:450,
    \begin{align*}
    \text{inner plot} &= \{(\int((\#1)/180)*5+0.5-((-1)^\text{Mod}(\int((\#1)/180),2)))*2.5*\cos(\#1), \\
    &\quad (2.5-((-1)^\text{Mod}(\int((\#1)/180),2))*0.5)*\sin(\#1)\}\}
    \end{align*}
    , \begin{align*}
    \text{outer plot} &= \{(\int((\#1)/180)*5+0.5-((-1)^\text{Mod}(\int((\#1)/180),2)))*2.5*\cos(\#1), \\
    &\quad (2.5-((-1)^\text{Mod}(\int((\#1)/180),2))*0.5)*\sin(\#1)\}\}
    \end{align*}
    , value=width("\WCvarC")
] \exampleforthismanual
\end{tikzpicture}
/wheelchart/outer plot style={⟨options⟩}  
(style, no default, initially empty)
This key accepts a list of keys which will be applied to the plot determined by the key outer plot.

/wheelchart/outer radius={⟨value⟩}  
(no default, initially 3)
The ⟨value⟩ of this key defines the outer radius of the wheelchart.

/wheelchart/perc precision={⟨number⟩}  
(no default, initially 0)
This key defines the number of decimals up to which the percentage in the macros \WCperc and \WCpercentagerounded is rounded. The rounding is performed with l3fp. With the initial setting, for example 49.5 and 50.5 are both rounded to 50. With perc precision={0,1}, 49.5 is rounded to 50 and 50.5 to 51.

/wheelchart/pie={⟨boolean⟩}  
(default true, initially false)
If true, the inner radius of the wheelchart is set to 0.

/wheelchart/plot={⟨code⟩}  
(no default)
This key sets inner plot and outer plot.

Since the let operation from the TiKZ library calc is used, it is not possible to use the variable names \n, \p, \x and \y inside the ⟨code⟩.

Note that positions depend on the domain and not on the length of the plot. For example below, data angle pos=0.5. The corresponding value of the domain is 1 which gives the x coordinate 1 which is not in the middle of the plot. Whereas wheel data angle pos=sqrt(2)/2. The corresponding value of the domain is $\sqrt{2}$ which gives the x coordinate 2 which is in the middle of the plot.
This key sets inner plot style and outer plot style.

/\texttt{wheelchart/radius}={⟨inner radius⟩}{⟨outer radius⟩} (no default)

This key defines the inner and outer radius of the wheelchart.

/\texttt{wheelchart/samples}={⟨number⟩} (no default, initially 25)

This key determines the ⟨number⟩ of samples used in the plots.

/\texttt{wheelchart/separator columns}={⟨delimiter⟩} (no default, initially \slash)

/\texttt{wheelchart/separator rows}={⟨delimiter⟩} (no default, initially ,)

The ⟨wheelchart data⟩ in the command \texttt{\wheelchart} is a list in which the items are separated by the value of the key separator rows. Each item in this list corresponds to one slice of the wheelchart and consists of data separated by the value of the key separator columns.

/\texttt{wheelchart/slices}={⟨path⟩} (no default)

If this key is set then the shape of the slices of the wheelchart is defined by ⟨path⟩.

In the following example, a ; is placed at the beginning of the argument for the key slices because there is no path to be filled. Thereafter, a node is placed still within the argument for the key slices.
These keys determine the position of the slices if the key `slices` is used similar as the corresponding keys for the key `data`.

Below we list some keys to modify the shape of the slices. These keys only affect the shape of the slices and not the computation of the inner and outer sides. In particular, these keys do not affect the placement of `arc`, `arc data`, `data`, `inner data`, `lines`, `wheel data` and `wheel lines`. If this placement should be changed then the keys `inner plot` and `outer plot` can be used.

This key sets `slices end arc` and `slices start arc` but uses the opposite of `(value 1)` for `slices start arc`.

![Diagram of a wheel chart with colored slices labeled Apricot, Lime, Melon, Olive, Peach, and Plum.]

```latex
\begin{tikzpicture}
\wheelchart[
  slices arc={1}{0},
  wheel data=\WCcount,
  wheel data angle pos=1,
  wheel data pos=0.5,
  wheel data style={
    circle, 
    fill=\WCvarB!50 
  }
]{\exampleforthismanual}
\end{tikzpicture}
```

```latex
\begin{tikzpicture}
\wheelchart[
  data=, 
  radius={1}{4.5},
  slices arc={1}{0.66},
  slices style=\WCvarA, 
  start half, 
  value=1, 
  wheel data={%\textbf{Number \WCcount}\% \WCvarB% },
  wheel data pos=0.5, 
  wheel data style=White 
]{%\Yellow/Some text A, \Orange/Some text B, \Red/Some text C, \Green/Some text D, \Blue/Some text E% }
\end{tikzpicture}
```
/wheelchart/slices arc inner end={(boolean)}  (default true, initially false)
If true then the keys slices end arc, slices inner arc and slices start arc are set such that the inner part and the end of each of the slices of the wheelchart form one arc and such that the start has the opposite curvature as the end.

```
\begin{tikzpicture}
\foreach\a/\x in {0/0,45/4.5} {
\wheelchart[
at={(\x,0)},
data=,gap,
radius={1}{2.2},
slices arc inner end,
value=1,
wheel data=\WCvarC,
wheel data angle pos=0.6]{\exampleforthismanual}
}\end{tikzpicture}
```

/wheelchart/slices arc inner end start={(boolean)}  (default true, initially false)
If true then the keys slices end arc, slices inner arc and slices start arc are set such that the inner part and the end of each of the slices of the wheelchart form one arc and such that the start has the same curvature as the end.

```
\begin{tikzpicture}
\foreach\a/\x in {-60/0,0/4.5,60/10} {
\wheelchart[
at={(\x,0)},
data=,gap,
radius={0.66}{2},
slices arc inner end start,
slices inner angle shift=\a,
slices style={fill=none,draw=Turquoise,ultra thick},
total count=20]{}
}\end{tikzpicture}
```

/wheelchart/slices arc inner start={(boolean)}  (default true, initially false)
If true then the keys slices end arc, slices inner arc and slices start arc are set such that the inner part and the start of each of the slices of the wheelchart form one arc and such that the end has the opposite curvature as the start.

```
\begin{tikzpicture}
\foreach\a/\x in {-60/0,0/4.5,60/10} {
\wheelchart[
at={(\x,0)},
data=,gap,
radius={0.66}{2},
slices arc inner end start,
slices inner angle shift=\a,
slices style={fill=none,draw=Turquoise,ultra thick},
total count=20]{}
}\end{tikzpicture}
```
/wheelchart/slices arc inner start end = \texttt{(boolean)} \hspace{1cm} (default \texttt{true}, initially \texttt{false})

If true then the keys \texttt{slices end arc}, \texttt{slices inner arc} and \texttt{slices start arc} are set such that the inner part and the start of each of the slices of the wheelchart form one arc and such that the end has the same curvature as the start.

/wheelchart/slices arc match = \langle \texttt{arg 1} \rangle \langle \texttt{num 1} \rangle \langle \texttt{num 2} \rangle \langle \texttt{num 3} \rangle \langle \texttt{arg 2} \rangle \langle \texttt{arg 3} \rangle \langle \texttt{arg 4} \rangle \hspace{1cm} (no default)

This key modifies the shape of the slices according to the 7 arguments.

Here, \langle \texttt{arg 1} \rangle must be \texttt{end}, \texttt{inner}, \texttt{outer} or \texttt{start} and \langle \texttt{arg 2} \rangle, \langle \texttt{arg 3} \rangle and \langle \texttt{arg 4} \rangle must be \texttt{inner end}, \texttt{inner start}, \texttt{outer end} or \texttt{outer start}. For example, the key \texttt{slices arc inner end} sets \texttt{slices arc match} = \langle \texttt{inner} \rangle \langle \texttt{1} \rangle \langle \texttt{-1} \rangle \langle \texttt{1} \rangle \langle \texttt{inner end} \rangle \langle \texttt{inner start} \rangle \langle \texttt{outer end} \rangle.
/wheelchart/slices arc outer end=(boolean)  (default true, initially false)
If true then the keys slices end arc, slices outer arc and slices start arc are set such that the outer part and the end of each of the slices of the wheelchart form one arc and such that the start has the opposite curvature as the end.

/wheelchart/slices arc outer end start=(boolean)  (default true, initially false)
If true then the keys slices end arc, slices outer arc and slices start arc are set such that the outer part and the end of each of the slices of the wheelchart form one arc and such that the start has the same curvature as the end.

/wheelchart/slices arc outer start=(boolean)  (default true, initially false)
If true then the keys slices end arc, slices outer arc and slices start arc are set such that the outer part and the start of each of the slices of the wheelchart form one arc and such that the end has the opposite curvature as the start.
\begin{tikzpicture}
\wheelchart[
data=,
gap=0.1,
slices arc inner start,
slices arc outer start,
slices style={\WCvarB!50,\draw=\WCvarB,\ultra thick},
value=1,
wheels data=\WCcount,
wheels data pos=0.8]
\end{tikzpicture}

\begin{tikzpicture}
\foreach\a/\x in {0/0,45/5,90/10} {
\wheelchart[\at=\langle\x,0\rangle,\data=,\gap,\radius=\langle0.66\rangle\times2,\slices arc outer start,\slices outer angle shift=\a,\value=1]
\exampleforthismanual}
\end{tikzpicture}

/wheelchart/slices arc outer start end={(boolean)} (default true, initially false)

If true then the keys slices end arc, slices outer arc and slices start arc are set such that the outer part and the start of each of the slices of the wheelchart form one arc and such that the end has the same curvature as the start.

/wheelchart/slices Arrow=\{\angle\} 

(no default)

This key sets slices end to --\langle\WCpoint{1}{\langle\angle\rangle}{0.5}\rangle--\langle\WCpoint{1}{0}{0}\rangle-- and slices start to --\langle\WCpoint{0}{\langle\angle\rangle}{0.5}\rangle--cycle.

/wheelchart/slices arrow=\{\langle value 1\rangle\}\{\langle value 2\rangle\} 

(no default)

This key is similar to the key slices arc but draws an arrow.
The example below compares arrows constructed with the key `slices Arrow` and the key `slices arrow`. Using the key `slices Arrow`, the arrow tip lies on the circle but the line segments do not have the same length. Using the key `slices arrow`, the arrow tip does not lie on the circle but the line segments have the same length.
This key determines the end of the slice. Initially, this is a line segment from the outer end to the inner end of the slice.

\texttt{/wheelchart/slices end arc}\{\texttt{value 1}\}\{\texttt{value 2}\} \quad \text{(no default)}

This key determines the end of the slice.

The effect of \texttt{value 1} and \texttt{value 2} is shown in the figure and the table below.

If \texttt{value 1} > 0 then the arc points outwards the slice. If \texttt{value 1} < 0 then the arc points inwards the slice. Here, outwards and inwards are relative to the orientation of the four-sided polygon formed by the points whose coordinates are determined by the inner and outer radius of the first slice and the start angle and the angle at the inverse of the key \texttt{samples} between the start angle and the end angle of the first slice. If the start angle and the end angle of the first slice are equal then the end angle of the last slice is used instead. If this test is inconclusive then the orientation is set according to the key \texttt{counterclockwise}.

If \texttt{value 1} = 0 then a line segment is drawn.

If \texttt{value 1} and \texttt{value 2} are negative then an arc is drawn which behaves the same as an arc with \texttt{value 2} = 0 and such that its radius matches the radius of the arc corresponding to setting \texttt{value 1} to its opposite. This is illustrated in the table below.

\begin{matrix}
\begin{array}{c|ccc}
\langle \text{value 1} \rangle & \langle \text{value 2} \rangle = -0.5 & \langle \text{value 2} \rangle = 0 & \langle \text{value 2} \rangle = 0.5 \\
\hline
\langle \text{value 1} \rangle = 2 & \includegraphics[scale=0.5]{diagram1.png} & \includegraphics[scale=0.5]{diagram2.png} & \includegraphics[scale=0.5]{diagram3.png} \\
\langle \text{value 1} \rangle = 1 & \includegraphics[scale=0.5]{diagram4.png} & \includegraphics[scale=0.5]{diagram5.png} & \includegraphics[scale=0.5]{diagram6.png} \\
\langle \text{value 1} \rangle = 0 & \includegraphics[scale=0.5]{diagram7.png} & \includegraphics[scale=0.5]{diagram8.png} & \includegraphics[scale=0.5]{diagram9.png} \\
\langle \text{value 1} \rangle = -1 & \includegraphics[scale=0.5]{diagram10.png} & \includegraphics[scale=0.5]{diagram11.png} & \includegraphics[scale=0.5]{diagram12.png} \\
\langle \text{value 1} \rangle = -2 & \includegraphics[scale=0.5]{diagram13.png} & \includegraphics[scale=0.5]{diagram14.png} & \includegraphics[scale=0.5]{diagram15.png}
\end{array}
\end{matrix}

\[\langle \text{value 1} \rangle = \frac{\pi}{4}, \quad \langle \text{value 2} \rangle = \frac{\pi}{3}\]
\begin{tikzpicture}
\wheelchart[
  for loop start={
    \definecolor{WCcolor}{wave}{\fpeval{380+(\WCcount-1)*340/(\WCtotalcount-1)}}
  },
gap polar=180/\WCtotalcount,
  radius={(1.5)\WCtotalcount},
  slices end arc={-0.6\WCtotalcount},
  slices start arc={1.2\WCtotalcount},
  slices style=WCcolor,
  total count=20
}]
\end{tikzpicture}

/wheelchart/slices end arrow=\{(value 1)\{(value 2)\} (no default)

This key is similar to the key slices end arc but draws an arrow.

/wheelchart/slices end to=\{(value 1)\{(value 2)\} (no default)

This key sets the to path operation for the end of the slice. The angle at the inner side is determined by (value 1) and the angle at the outer side is determined by (value 2).

/wheelchart/slices inner=\{(path)\} (no default)

This key determines the inner part of the slice. Initially, this is an arc from the inner end to the inner start of the slice.
\begin{tikzpicture}
def{\a}{5}
def{\b}{12}
\wheelchart[
data sep=0.3,gap=0.1,inner data=\WCcount,inner data sep=2-2*cos(\b),
inner data style={
circle,fill=\WCvarB!50},slices inner=
{arc[
  start angle=\WCangle{1}{0}{0}{0},
  end angle=\WCangle{0.5}{\b}{0}{0},
  radius=\WCradius{0}{0}
]
  arcc[
  start angle=\WCmidangle-90,
  end angle=\WCmidangle+90,
  radius=2*sin(\b)
  ]
  arc[
  start angle=\WCangle{0.5}{-\b}{0}{0},
  end angle=\WCangle{0}{0}{0}{0},
  radius=\WCradius{0}{0}
  ]
},slices outer=
{arc[
  start angle=\WCangle{0}{0}{1}{0},
  end angle=\WCangle{0.5}{-\a}{1}{0},
  radius=\WCradius{1}{0}
  ]
  --(\WCpoint{0.5}{0}{1}{0.3})
  --(\WCpoint{0.5}{\a}{1}{0})
  arc[
  start angle=\WCangle{0.5}{\a}{1}{0},
  end angle=\WCangle{1}{0}{1}{0},
  radius=\WCradius{1}{0}
  ]
  },slices style=(
draw=\WCvarB,fill=\WCvarB!25,ultra thick
 ),value=1]\exampleforthismanual\end{tikzpicture}

\wheelchart/slices inner angle reduce={\langle angle\rangle}
(no default)

This key sets slices inner end angle shift to $-\langle angle\rangle$ and slices inner start angle shift to $\langle angle\rangle$.

\wheelchart/slices inner angle shift={\langle angle\rangle}
(no default)

This key sets slices inner end angle shift and slices inner start angle shift to $\langle angle\rangle$. 
\begin{tikzpicture}
\wheelchart[
  data=, middle=, slices inner, angle shift=90
]
\end{tikzpicture}

\begin{tikzpicture}
\wheelchart[
  data=, gap, radius={1}{3}, slices arc={0.5}{0}, slices inner angle shift=45,
  value=1, wheel data={\WCvarC}, wheel data angle pos=0.8
]
\end{tikzpicture}
\begin{tikzpicture}
\foreach\a/\x in {-60/0,0/5.6,60/10}
\wheelchart[
at=\{(\a,0)\},
radius={0.66}{2},
slices arc inner start,
slices inner angle shift=\a,
slices style={fill=\WClistcolors},
total count=40,
WClistcolors={RedOrange,none}]
}{\}
\end{tikzpicture}

This key is similar to the key slices end arc but sets the inner part of the slice.

/wheelchart/slices inner arc tangent=(boolean) (default true, initially false)

If true then the key slices inner arc is set such that the arc is tangent to the end and start of the slice if possible. Note that this is not possible for all settings for keys such as plot and slices inner angle shift.

\begin{tikzpicture}
\wheelchart[
clockwise,
data=,
gap=0.1,
middle=slices inner\arc tangent,
middle style={font=\ttfamily},
slices inner arc tangent,
slices style={
draw=\WCvarB,
fill=\WCvarB!50, ultra thick
},
value=I
]\exampleforthismanual
\end{tikzpicture}

/wheelchart/slices inner arrow=\{(value 1)\}{\{(value 2)\}} (no default)

This key is similar to the key slices end arrow but sets the inner part of the slice.

\begin{tikzpicture}
\wheelchart[
counter clockwise,
data=,
gap=0.1,
middle=slices inner\arc tangent,
middle style={font=\ttfamily},
slices inner arc tangent,
slices style={
draw=\WCvarB,
fill=\WCvarB!50, ultra thick
},
value=I
]\exampleforthismanual
\end{tikzpicture}
The end angle of the inner part of the slice will be modified such that the angle between the end and the inner part of the slice is shifted with \( \langle \text{angle} \rangle \) (taking into account the key \text{counterclockwise}). The behavior of this key depends on whether a plot is used.

This key is similar to the key \text{slices inner start angle shift} but modifies the start angle of the inner part of the slice.

This key sets the \text{to} path operation for the inner part of the slice. The angle at the start is determined by \( \langle \text{value 1} \rangle \) and the angle at the end is determined by \( \langle \text{value 2} \rangle \).

This key determines the outer part of the slice. Initially, this is an arc from the outer start to the outer end of the slice.

This key sets \text{slices outer end angle shift to} \( -\langle \text{angle} \rangle \) and \text{slices outer start angle shift to} \( \langle \text{angle} \rangle \).
/wheelchart/slices outer angle shift={⟨angle⟩}  
(no default)

This key sets slices outer end angle shift and slices outer start angle shift to ⟨angle⟩.

/wheelchart/slices outer arc={⟨value 1⟩}{⟨value 2⟩}  
(no default)

This key is similar to the key slices end arc but sets the outer part of the slice.

/wheelchart/slices outer arc tangent={boolean}  
(default true, initially false)

If true then the key slices outer arc is set such that the arc is tangent to the end and start of the slice if possible. Note that this is not possible for all settings for keys such as plot and slices inner angle shift.
slices outer arc tangent

\begin{tikzpicture}
\wheelchart[
  data=, gap=0.1, middle=slices outer\arc tangent, middle style={font=\ttfamily}, slices outer arc tangent, slices style={
    draw=\WCvarB, fill=\WCvarB!50, ultra thick}, value=1]
\exampleforthismanual
\end{tikzpicture}

\begin{tikzpicture}
\pgfkeys{
  /wheelchart, data=, radius=1\{1.5\}, value=1}
\wheelchart[
  slices inner arrow=0\{0\}]
\exampleforthismanual
\wheelchart[
  at={(0,0)}, slices outer arrow=0\{0\}]
\exampleforthismanual
\wheelchart[
  at={(0.25,0)}, slices outer arrow=0\{0\}]
\exampleforthismanual
\wheelchart[
  at={(6.5,0)}, slices outer arrow=0\{0\}]
\exampleforthismanual
\end{tikzpicture}

/wheelchart/slices outer arrow={\{value 1\}}{\{value 2\}}

This key is similar to the key \texttt{slices end arrow} but sets the outer part of the slice.

\begin{tikzpicture}
\foreach\a/\x in {1/0, \{tan(180/\WCtotalcount)\}/5}{
\wheelchart[
  at={\a}, data=, gap, radius=0.66\{2\}, slices outer arrow={\a}{0}, start half, value=J, wheel data=\WCvarC]
\exampleforthismanual
}\end{tikzpicture}
\begin{tikzpicture}
\foreach \r/\s/\a in 
{3/0/0.5,2/15/1,1/30/0.7} {
\wheelchart[
  radius=\r cm,
  slices outer arrow=\a \s,
  slices style={
    fill=\WClistcolors!20,
    draw=\WClistcolors,
    ultra thick,
    double
  },
  start half=\s
  total count=12,
  WClistcolors={CarnationPink,Orchid}
]{}
}
\end{tikzpicture}

\begin{tikzpicture}
\wheelchart[
  data=,
  radius={0}{2.5},
  slices arc={0.4}{0},
  slices outer to={70}{70},
  start half, 
  value=1,
  wheel data=\WCvarC,
  wheel data pos=1
]\exampleforthismanual
\end{tikzpicture}

/wheelchart/slices outer end angle shift={⟨angle⟩} (no default, initially 0)

The end angle of the outer part of the slice will be modified such that the angle between the end and the inner (not the outer) part of the slice is shifted with ⟨angle⟩ (taking into account the key **counterclockwise**). The behavior of this key depends on whether a plot is used.

/wheelchart/slices outer start angle shift={⟨angle⟩} (no default, initially 0)

This key is similar to the key **slices outer end angle shift** but modifies the start angle of the outer part of the slice.

/wheelchart/slices outer to={⟨value 1⟩}{⟨value 2⟩} (no default)

This key sets the **to** path operation for the outer part of the slice. The angle at the start is determined by ⟨value 1⟩ and the angle at the end is determined by ⟨value 2⟩.

/wheelchart/slices pos={⟨value⟩} (no default, initially 0.5)

This key determines the position of the slices if the key **slices** is used similar as the corresponding key for the key **data**.
This key accepts a list of keys which will be applied to the scope in which the slices of the wheelchart, the wheel lines determined by the key \texttt{wheel lines} and the different kinds of data are placed.

This key determines the position of the slices if the key \texttt{slices} is used similar as the corresponding key for the key \texttt{data}.

This key determines the start of the slice. Initially, this is a line segment from the inner start to the outer start of the slice.

This key is similar to the key \texttt{slices end arc} but sets the start of the slice.

This key is similar to the key \texttt{slices end arrow} but sets the start of the slice.
This key sets the to path operation for the start of the slice. The angle at the inner side is determined by \langle value 1\rangle and the angle at the outer side is determined by \langle value 2\rangle.

This key defines the style of the slices of the wheelchart.

This sets slices end to and slices start to but uses the opposite respective values for slices start to.

This key accepts a list of keys from the wheelchart key family. The \langle range\rangle is mandatory and must be non-empty. It is processed with \texttt{\foreach} with the option \texttt{parse=true}. Hereafter the elements are processed with \texttt{\fp_eval:n}. The \langle options\rangle will only be applied to a slice if the number of the slice is in the \langle range\rangle. The \langle range\rangle only makes sense for a key which is processed for each slice. For example, the \langle range\rangle does not make sense for the key \texttt{middle}.

This key defines the \langle angle\rangle in degrees at which the first slice of the wheelchart starts.

This key sets the start angle such that the middle of the first slice of the wheelchart is positioned at \langle angle\rangle in degrees.

This contains the \langle text\rangle which will be placed above the wheelchart. The \langle text\rangle is placed in a node. The \texttt{x} coordinate of this node is the \texttt{x} coordinate of the center of the wheelchart, which is defined by the key \texttt{at}. In general, this is not the same as the \texttt{x} coordinate of the center of the local bounding box around the wheelchart. The \texttt{y} coordinate of this node is at a value determined by the key \texttt{title sep} above the north of the local bounding box around the wheelchart. The style of this node is given as follows. First, the options \texttt{anchor=south,align=center} are given. Thereafter, the style of the key \texttt{title style} is added.
This key contains the \( \langle \text{text} \rangle \) which will be placed above left of the wheelchart. The \( \langle \text{text} \rangle \) is placed in a node. This node is placed at a value determined by the key \( \text{title left sep} \). The style of this node is given as follows. First, the options \( \text{anchor=south west, align=left} \) are given. Thereafter, the style of the key \( \text{title left style} \) is added.

\[ /\text{wheelchart/title left sep}={\langle \text{value} \rangle} \]

The node where the contents of the key \( \text{title left} \) is placed is at \( \langle \text{value} \rangle \) above the north west of the local bounding box around the wheelchart.

\[ /\text{wheelchart/title left style}={\langle \text{options} \rangle} \]

This key accepts a list of keys which will be applied to the node where the contents of the key \( \text{title left} \) is placed.

\[ /\text{wheelchart/title sep}={\langle \text{value} \rangle} \]

The \( y \) coordinate of the node where the contents of the key \( \text{title} \) is placed is at \( \langle \text{value} \rangle \) above the north of the local bounding box around the wheelchart.

\[ /\text{wheelchart/title style}={\langle \text{options} \rangle} \]

This key accepts a list of keys which will be applied to the node where the contents of the key \( \text{title} \) is placed.

\[ /\text{wheelchart/total angle}={\langle \text{angle} \rangle} \]

This key defines the total \( \langle \text{angle} \rangle \) in degrees of the wheelchart.

\[ /\text{wheelchart/total count}={\langle \text{number} \rangle} \]

If this key is set then the number of slices of the wheelchart is determined by \( \langle \text{number} \rangle \). Moreover, \( \text{WCvarA} \) is defined as 1 and \( \text{WCvarB} \) and \( \text{WCvarC} \) are defined to be empty.

\[
\begin{tikzpicture}
\def\n{57}
\wheelchart[
  gap=0.015,
  middle={$\Large\qty{\n}{\percent}$},
  slices style=Gray,
  slices style{1,...,\n}=Cyan,
  total count=100
]{}
\end{tikzpicture}
\]

\[ /\text{wheelchart/triangle proportional area}={\langle \text{width} \rangle}{\langle \text{height} \rangle} \]

This key configures the plot such that a triangular shape is obtained. The value is proportional to the area and \textit{not} to the height. Moreover, it sets \texttt{samples=2} and \texttt{wheel data pos=0.5}. The point \((0,0)\) is at the top. This can be shifted with the key \texttt{at}.
This key configures the plot such that a triangular shape is obtained. The value is proportional to the height and not to the area. Moreover, it sets `samples=2` and `wheel data pos=0.5`. The point (0,0) is at the top. This can be shifted with the key at.

This key defines the ⟨value⟩ which corresponds to the size of each slice of the wheelchart.

This key locally defines a macro \WClist⟨name⟩ which gives the element in the ⟨list⟩ with as index \WCcount modulo the length of the ⟨list⟩. The ⟨list⟩ is expanded once and processed using a clist. In particular, blank arguments are ignored. An empty argument in the ⟨list⟩ can be obtained with \{\}. Items containing a , can be obtained by surrounding it with \{ and \} such as \WClistA={\{a,b\},\{c,d\}}. If \def\mylist{a,b,c} and \WClistA=\mylist then \WClistA gives a,b,c for each slice. On the other hand, if \WClistA/.expanded=\mylist then \WClistA alternates between a, b and c.

This key contains the ⟨text⟩ which will be placed on top of each slice of the wheelchart. The ⟨text⟩ is placed in a node. The style of this node is given as follows. First, the option align=left is given. Thereafter, the style of the key wheel data style is added.

These keys determine the position of the contents of the key wheel data similar as the corresponding keys for the key data.

This key accepts a list of keys which will be applied to the node where the contents of the key wheel data is placed.

(style, no default, initially empty)
If this key is set then lines with the style determined by this key will be drawn inside the slices of the wheelchart. The number of these lines depends on the value of the key `value`.

Below is the example from [5, Subsection 7.6] recreated with the package `wheelchart`.

```
\usepackage{siunitx}
\begin{tikzpicture}
\colorlet{good}{green!75!black}
\colorlet{bad}{red}
\colorlet{neutral}{black!60}
\colorlet{none}{white}
\wheelchart[
  anchor xsep=15,
  contour=gray,
  data="\WCvarC\" : \WCvarA (\WCperc),
  middle=Ratings given by \WCtotalnum participants,
  radius=(1.8)(2.2),
  start half=270,
  wheel lines={black!15,thick}
]{
  10/neutral/ok,
  9/good!60!white/good,
  3/good/very good,
  20/none/none,
  0/bad/very bad,
  8/bad!60!white/bad
}
\end{tikzpicture}
```

\hspace{1in}

```
\wheelchart/xbar={\langle\text{width}\rangle}{\langle\text{height}\rangle}
\begin{tikzpicture}
\colorlet{good}{green!75!black}
\colorlet{bad}{red}
\colorlet{neutral}{black!60}
\colorlet{none}{white}
\wheelchart[
  anchor xsep=15,
  contour=gray,
  data="\WCvarC\" : \WCvarA (\WCperc),
  middle=Ratings given by \WCtotalnum participants,
  radius=(1.8)(2.2),
  start half=270,
  wheel lines={black!15,thick}
]{
  10/neutral/ok,
  9/good!60!white/good,
  3/good/very good,
  20/none/none,
  0/bad/very bad,
  8/bad!60!white/bad
}
\end{tikzpicture}
```

\hspace{1in}

```
\wheelchart/ybar={\langle\text{width}\rangle}{\langle\text{height}\rangle}
\begin{tikzpicture}
\colorlet{good}{green!75!black}
\colorlet{bad}{red}
\colorlet{neutral}{black!60}
\colorlet{none}{white}
\wheelchart[
  anchor xsep=15,
  contour=gray,
  data="\WCvarC\" : \WCvarA (\WCperc),
  middle=Ratings given by \WCtotalnum participants,
  radius=(1.8)(2.2),
  start half=270,
  wheel lines={black!15,thick}
]{
  10/neutral/ok,
  9/good!60!white/good,
  3/good/very good,
  20/none/none,
  0/bad/very bad,
  8/bad!60!white/bad
}
\end{tikzpicture}
```

This key sets `domain=0:{\langle\text{width}\rangle}, plot={\langle\text{#1}\rangle,\langle\text{#2}\rangle}, radius=0:{\langle\text{height}\rangle}, samples=2` and also `wheel data pos=0.5`. The point (0,0) is below left of the bar. This can be shifted with the key `at`.

Note that since this key sets in particular the outer parts of the slices, keys such as `slices outer arc` must be placed after the key `xbar` to be applied.

```
\begin{tikzpicture}
\wheelchart[
  anchor xsep=15,
  contour=gray,
  data="\WCvarC\" : \WCvarA (\WCperc),
  middle=Ratings given by \WCtotalnum participants,
  radius=(1.8)(2.2),
  start half=270,
  wheel lines={black!15,thick}
]{
  10/neutral/ok,
  9/good!60!white/good,
  3/good/very good,
  20/none/none,
  0/bad/very bad,
  8/bad!60!white/bad
}
\end{tikzpicture}
```

This key sets `domain=0:{\langle\text{height}\rangle}, plot={\langle\text{#2}\rangle,\langle\text{#1}\rangle}, radius=0:{\langle\text{width}\rangle}, samples=2` and also `wheel data pos=0.5`. The point (0,0) is below left of the bar. This can be shifted with the key `at`.

```
\begin{tikzpicture}
\wheelchart[
  anchor xsep=15,
  contour=gray,
  data="\WCvarC\" : \WCvarA (\WCperc),
  middle=Ratings given by \WCtotalnum participants,
  radius=(1.8)(2.2),
  start half=270,
  wheel lines={black!15,thick}
]{
  10/neutral/ok,
  9/good!60!white/good,
  3/good/very good,
  20/none/none,
  0/bad/very bad,
  8/bad!60!white/bad
}
\end{tikzpicture}
```
5 Additional examples

The following example is an answer to the question on https://tex.stackexchange.com/questions/433848/is-there-a-way-to-make-sunburst-charts-multi-level-pie-charts-in-latex.
The following example is an answer to the question on https://tex.stackexchange.com/questions/447920/pie-chart-with-color-palette-info-inside-and-legend.
The following example is an answer to the question on https://tex.stackexchange.com/questions/477310/cyclic-flowchart-in-tikz.

```latex
\usetikzlibrary{decorations.text}
\begin{tikzpicture}
\sffamily
\wheelchart[
  data=,
  middle=Optimized\vibrating\systems,
  middle fill=RoyalBlue,
  middle style=white,
  radius=(1,2)(4),
  slices=\{0,0\} circle[radius=0.8]\;,
  slices style=\WCvarA,
  start half,
  value=1,
  wheel data=\WCvarB,
  wheel data pos=0.5,
  wheel data style={
    white,
    align=center
  }
]{%}
\foreach\n in {-30,90,210}{
  \draw[->,MidnightBlue,ultra thick]
  (\n:1.7)--(\n:1.3);
}\fill[
  top color=Gray!50,
  bottom color=Gray,
  draw,
  even odd rule
] (0,0) circle[radius=3.5]
circle[radius=4.2];
\wheelchart[
  arc(2)={
    <,
    ultra thick
  },
  arc around text,
  arc data=\WCvarA,
  arc data pos=0.5,
  arc pos=0.5,
  data=,
  gap polar=10,
  radius=(3.5)(4.2),
  slices style=\{fill=none\},
  start half=180,
  value=1
]{%}
\foreach\n in {Mass M, Damping D, Stiffness K}{
  Dynamic model,
  Frequency response functions H,
  \}
\end{tikzpicture}
```
6 Version history

Version 1.0 (2022/09/11) First version.

Version 2.0 (2023/12/03)

- The package now mainly uses \LaTeX\ syntax.
- Improved the definition of the path of the slices.
- Many internal computations are now performed with \texttt{\textbackslash fp\_eval\texttt{\textunderscore}n} instead of \texttt{pgfmath} for higher accuracy and to allow larger values. This applies in particular to the computation of \texttt{\WC\textunderscore}percentage, \texttt{\WC\textunderscore}percentage\texttt{\textunderscore}rounded and \texttt{\WC\textunderscore}totalnum. Hence \texttt{\WC\textunderscore}percentage\texttt{\textunderscore}rounded can be parsed by \texttt{siunitx} since its definition does not involve \texttt{\pgfmath\textunderscore}print\texttt{\textunderscore}number\texttt{\textunderscore}to anymore and \texttt{\WC\textunderscore}totalnum does not end with \texttt{.0} if it is an integer.
- The number of data which can be given to each slice of the wheelchart and accessed by \texttt{\WC\textunderscore}var\texttt{\textunderscore}A and so on is not limited to 26 anymore.
- Added the macros \texttt{\WC\textunderscore}count\textunderscore}discrete, \texttt{\WC\textunderscore}etoct\textunderscore}helinked\textunderscore}name, \texttt{\WC\textunderscore}etoct\textunderscore}helinked\textunderscore}number, \texttt{\WC\textunderscore}etoct\textunderscore}helinked\textunderscore}page, \texttt{\WC\textunderscore}etoct\textunderscore}hename, \texttt{\WC\textunderscore}etoct\textunderscore}hnumber, \texttt{\WC\textunderscore}etoct\textunderscore}hnumber\textunderscore}of\textunderscore}pages, \texttt{\WC\textunderscore}etoct\textunderscore}hpage, \texttt{\WC\textunderscore}legendend}, \texttt{\WC\textunderscore}list\texttt{\textunderscore}name} and \texttt{\langle prefix\rangle\langle name\rangle}.
- Added the keys \texttt{after\ slices}, \texttt{arc}, \texttt{arc\ around\ text}, \texttt{arc\ data}, \texttt{arc\ data\ align}, \texttt{arc\ data\ angle\ pos}, \texttt{arc\ data\ angle\ shift}, \texttt{arc\ data\ dir}, \texttt{arc\ data\ pos}, \texttt{arc\ data\ sep}, \texttt{arc\ data\ style}, \texttt{arc\ first\ half}, \texttt{arc\ pos}, \texttt{arc\ second\ half}, \texttt{arc\ sep}, \texttt{before\ slices}, \texttt{caption\ left\ sep}, \texttt{caption\ sep}, \texttt{data\ angle\ pos}, \texttt{data\ pos}, \texttt{discrete}, \texttt{discrete\ factor}, \texttt{discrete\ partitioning}, \texttt{discrete\ pic}, \texttt{discrete\ sort}, \texttt{discrete\ space\ at\ borders}, \texttt{domain}, \texttt{etoc\ code}, \texttt{etoc\ count\ total\ pages}, \texttt{etoc\ level}, \texttt{etoc\ name}, \texttt{etoc\ use\ name}, \texttt{expand\ list\ items}, \texttt{for\ loop\ end}, \texttt{for\ loop\ start}, \texttt{gap\ max\ angle}, \texttt{gap\ radius}, \texttt{header}, \texttt{header\ prefix}, \texttt{inner\ data\ angle\ pos}, \texttt{inner\ data\ angle\ shift}, \texttt{inner\ data\ pos}, \texttt{inner\ plot}, \texttt{inner\ plot\ style}, \texttt{legend\ columns}, \texttt{legend\ only}, \texttt{legend\ row}, \texttt{lines\ angle\ pos}, \texttt{lines\ angle\ shift}, \texttt{lines\ ext\ dir}, \texttt{lines\ ext\ fixed\ left}, \texttt{lines\ ext\ fixed\ right}, \texttt{lines\ pos}, \texttt{outer\ plot}, \texttt{outer\ plot\ style}, \texttt{parse\ plot}, \texttt{plot\ style}, \texttt{samples}, \texttt{separator\ columns}, \texttt{separator\ rows}, \texttt{slices\ angle\ pos}, \texttt{slices\ angle\ shift}, \texttt{slices\ arc\ inner\ end}, \texttt{slices\ arc\ inner\ start}, \texttt{slices\ arc\ inner\ start\ end\ slices\ arc\ inner\ end\ start}, \texttt{slices\ arc\ outer\ end\ start}, \texttt{slices\ arc\ outer\ end\ start\ slices\ arc\ outer\ end\ start\ slices\ arc\ outer\ start}, \texttt{slices\ arc\ outer\ start\ end}, \texttt{slices\ end\ to}, \texttt{slices\ inner\ angle\ reduce}, \texttt{slices\ inner\ angle\ shift}, \texttt{slices\ inner\ arc}, \texttt{slices\ inner\ arc\ tangent}, \texttt{slices\ inner\ arrow}, \texttt{slices\ inner\ angle\ end\ shift}, \texttt{slices\ inner\ start\ angle\ shift}, \texttt{slices\ inner\ to}, \texttt{slices\ outer\ angle\ reduce}, \texttt{slices\ outer\ angle\ shift}, \texttt{slices\ outer\ arc}, \texttt{slices\ outer\ arc\ tangent}, \texttt{slices\ outer\ arrow}, \texttt{slices\ outer\ end\ angle\ shift}, \texttt{slices\ outer\ start\ angle\ shift}, \texttt{slices\ outer\ to}, \texttt{slices\ pos}, \texttt{slices\ scope}, \texttt{slices\ sep}, \texttt{slices\ start\ to}, \texttt{slices\ to}, \texttt{\langle range\rangle}, \texttt{\langle\ prefix\rangle\langle\ name\rangle}, \texttt{\WC\percentagerounded}, \texttt{\WC\textunderscore}list\textunderscore}data}, \texttt{\WC\textunderscore}wheel\ data\ angle\ pos}, \texttt{\WC\textunderscore}wheel\ data\ sep}, \texttt{xbar} and \texttt{ybar}.
- Added the possibility to give a \texttt{\langle range\rangle} to the keys such that the options given to the key will only be applied to a slice if the number of the slice is in the \texttt{\langle range\rangle}.
- Added the possibility to give a \texttt{\langle list\rangle} to the keys.
- The \texttt{\langle wheelchart\ data\rangle} are not processed with \texttt{\foreach} anymore but instead with one of \texttt{\seq\set\split\texttt{\textunderscore}Ne}, \texttt{\seq\set\split\texttt{\textunderscore}Ne\texttt{\textunderscore}Ne} or \texttt{\seq\set\split\texttt{\textunderscore}Ne\texttt{\textunderscore}Ne} depending on the keys \texttt{expand\ list} and \texttt{expand\ list\ items}. Thus syntax which is specific to how \texttt{\foreach} processes a list does not work anymore, such as the dots notation and the repeating of the last entry if some entry in the list has fewer entries than required.
- If the key \texttt{\begin{small}start\ angle\end{small}} is set after the key \texttt{\begin{small}start\ half\end{small}} then v1.0 preserved the setting of the key \texttt{\begin{small}start\ half\end{small}}. In v2.0, the setting is determined by the key which is set last.
– In v1.0, the value of the key **data angle shift** was also applied to **inner data**, **lines** and **wheel data**. In v2.0, this is not the case anymore. Instead there are now separate keys **inner data angle shift**, **lines angle shift**, **wheel data angle shift** and also **arc data angle shift**.

– In v1.0, the key **data sep** was not applied if the key **lines ext** was used. In v2.0, this is not the case anymore.

– In v1.0, a negative value for the key **lines** was not applied. In v2.0, this is not the case anymore.

**Version 3.0 (2024/03/08)**

– Improved the parametrization of the slices in the case that no plot is used. In particular, the **arc** and **arc data** are placed with an arc if no plot is used whereas in v2.0, these were placed with a plot even if no plot was used. Also, the computation of \texttt{WCdataangle} and \texttt{WCmidangle} is more precise than in v2.0.

– Optimized the code. The compilation is faster than in v2.0.

– Added the commands \texttt{WCangle}, \texttt{WCcoordinate}, \texttt{WCpoint} and \texttt{WCradius}.

– Added the keys **arc around line**, **arc data expand**, **arc data line sep factor**, **slices Arrow**, **slices end**, **slices inner**, **slices outer** and **slices start**.

– Changed the definition of \texttt{WCperc} in the key **arc data** so that \texttt{WCperc} follows the arc or plot.

– Added the possibility that the contents of the key **arc data** consists of multiple lines separated by `\`.

– Reduced the functionality of the keys **contour** and **middle fill** to require a fixed inner and outer radius for all slices.

– Removed the key **parse**. The values of applicable keys are parsed with \texttt{pgfmathparse}. If a value should be parsed with \texttt{l3fp} then \texttt{fpeval} can be used.

– In v2.0, the key **arc data angle shift** was not taken into account for the key **arc** in combination with the key **arc around text**. This is fixed in v3.0.

– In v2.0, the number of items for each slice in the ⟨wheelchart data⟩ which can be accessed with the macros \texttt{WCvarA} and so on was determined by the number of items for the last slice. For example, \texttt{data{1}=`WCvarD} in combination with the ⟨wheelchart data⟩ 1/black/A/a,2/gray/B was not possible with v2.0. This is not a limitation anymore with v3.0.

**References**


Index

\langle prefix \rangle \langle name \rangle, 8

after slices key, 11
anchor xsep key, 11
anchor ysep key, 11
arc key, 12
arc around line key, 12
arc around text key, 12
arc data key, 12
arc data align key, 12
arc data angle pos key, 12
arc data angle shift key, 12
arc data dir key, 13
arc data expand key, 13
arc data line sep factor key, 13
arc data pos key, 13
arc data sep key, 13
arc data style key, 13
arc first half key, 13
arc pos key, 13
arc second half key, 13
arc sep key, 13
at key, 14
before slices key, 14
caption key, 14
caption left key, 14
caption left sep key, 14
caption left style key, 14
caption sep key, 14
caption style key, 14
countour key, 15
counterclockwise key, 15
data key, 15
data angle pos key, 16
data angle shift key, 16
data pos key, 16
data sep key, 16
data style key, 17
discrete key, 17
discrete factor key, 17
discrete partitioning key, 17
discrete pic key, 18
discrete sort key, 19
discrete space at borders key, 20
domain key, 21
etoc code key, 21
etoc count total pages key, 21
etoc level key, 21
etoc name key, 21
etoc use name key, 22
\exampleforthismanual, 3
expand list key, 22
expand list items key, 23
explode key, 24
for loop end key, 24
for loop start key, 24
gap key, 24
gap max angle key, 24
gap polar key, 25
gap radius key, 26
header key, 26
header prefix key, 26
inner data key, 26
inner data angle pos key, 26
inner data angle shift key, 26
inner data pos key, 26
inner data sep key, 26
inner data style key, 26
inner plot key, 27
inner plot style key, 27
inner radius key, 27
legend key, 27
legend columns key, 27
legend entry key, 27
legend only key, 27
legend row key, 28
lines key, 29
lines angle pos key, 29
lines angle shift key, 29
lines ext key, 29
lines ext bottom dir key, 29
lines ext dir key, 29
lines ext dirsep key, 30
lines ext fixed key, 30
lines ext fixed left key, 30
lines ext fixed right key, 30
lines ext left anchor key, 30
lines ext right anchor key, 30
lines ext top dir key, 31
lines pos key, 31
lines sep key, 31
lines style key, 31
middle key, 33
middle fill key, 33
middle style key, 33
name key, 33
outer plot key, 33
outer plot style key, 35
outer radius key, 35
perc precision key, 35
pie key, 35
plot key, 35
plot style key, 37
radius key, 37
samples key, 37
separator columns key, 37
separator rows key, 37
slices key, 37
slices angle pos key, 38
slices angle shift key, 38
slices arc key, 38
slices arc inner end key, 39
slices arc inner end start key, 39
slices arc inner start key, 40
slices arc match key, 40
slices arc outer end key, 41
slices arc outer end start key, 41
slices arc outer start key, 41
slices arc outer start end key, 42
slices Arrow key, 42
slices arrow key, 42
slices end key, 43
slices end arc key, 44
slices end arrow key, 45
slices end to key, 45
slices inner key, 45
slices inner angle reduce key, 46
slices inner angle shift key, 46
slices inner arc key, 48
slices inner arc tangent key, 48
slices inner arrow key, 48
slices inner end angle shift key, 49
slices inner start angle shift key, 49
slices inner to key, 49
slices outer key, 49
slices outer angle reduce key, 49
slices outer angle shift key, 50
slices outer arc key, 50
slices outer arc tangent key, 50
slices outer arrow key, 51
slices outer end angle shift key, 52
slices outer start angle shift key, 52
slices outer to key, 52
slices pos key, 52
slices scope key, 53
slices sep key, 53
slices start key, 53
slices start arc key, 53
slices start arrow key, 53
slices start to key, 54
slices style key, 54
slices to key, 54
slice{\langle range\rangle} key, 54
start angle key, 54
start half key, 54
title key, 54
title left key, 54
title left sep key, 55
title left style key, 55
title sep key, 55
title style key, 55
total angle key, 55
total count key, 55
triangle proportional area key, 55
triangle proportional height key, 56
value key, 56
\WCangle, 4
\WCcoordinate, 5
\WCcount, 5
\WCcountdiscrete, 5
\WCdataangle, 5
\WCetocthelinkedname, 5
\WCetocthelinkednumber, 5
\WCetocthepage, 6
\WCetocthenumber, 6
\WCetocthenumberofpages, 6
\WCetocthepage, 6
\WCLegend, 6
\WClist(name) key, 56
\WClist(name), 6
\WCmiddangle, 6
\WCperc, 6
\WCPercents, 7
\WCPercentsaggerounded, 7
\WCpoint, 7
\WCradius, 7
\WCtotalcount, 7
\WCtotalnum, 7
\WCVarA, 8
\WCVarB, 8
\WCVarC, 8
wheel data key, 56
wheel data angle pos key, 56
wheel data angle shift key, 56
wheel data pos key, 56
wheel data sep key, 56
wheel data style key, 56
wheel lines key, 56
\wheelchart, 3
/wheelchart/
after slices, 11
anchor xsep, 11
anchor ysep, 11
arc, 12
arc around line, 12
arc around text, 12
arc data, 12
arc data align, 12
arc data angle pos, 12
arc data angle shift, 12
arc data dir, 13
arc data expand, 13
arc data line sep factor, 13
arc data pos, 13
arc data sep, 13
slices inner to, 49
slices outer, 49
slices outer angle reduce, 49
slices outer angle shift, 50
slices outer arc, 50
slices outer arc tangent, 50
slices outer arrow, 51
slices outer end angle shift, 52
slices outer start angle shift, 52
slices outer to, 52
slices pos, 52
slices scope, 53
slices sep, 53
slices start, 53
slices start arc, 53
slices start arrow, 53
slices start to, 54
slices style, 54
slices to, 54
slice({range}), 54
start angle, 54
start half, 54
title, 54

title left, 54
title left sep, 55
title left style, 55
title sep, 55
title style, 55
total angle, 55
total count, 55
triangle proportional area, 55
triangle proportional height, 56
value, 56
WClist(name), 56
wheel data, 56
wheel data angle pos, 56
wheel data angle shift, 56
wheel data pos, 56
wheel data sep, 56
wheel data style, 56
wheel lines, 56
xbar, 57
ybar, 57

xbar key, 57
ybar key, 57
A The source code

\NeedsTeXFormat{LaTeX2e}
\RequirePackage{tikz}
\usetikzlibrary{calc}
\ProvidesExplPackage{wheelchart}{2024/03/08}{3.0}{Diagrams with circular or other shapes using TikZ and LaTeX3}

A.1 Variables

\newcounter{g__wheelchart_WCcount_counter}

\bool_new:N \l__wheelchart_arc_bool
\bool_new:N \l__wheelchart_contour_bool
\bool_new:N \g__wheelchart_def_angle_radius_shift_bool
\bool_new:N \l__wheelchart_discrete_bool
\bool_new:N \l__wheelchart_etoc_use_name_bool
\bool_new:N \l__wheelchart_legend_only_bool
\bool_new:N \l__wheelchart_legend_row_bool
\bool_new:N \l__wheelchart_lines_ext_dir_bool
\bool_new:N \l__wheelchart_middle_fill_bool
\bool_new:N \l__wheelchart_pie_bool
\fp_new:N \l__wheelchart_arc_start_angle_fp
\fp_new:N \l__wheelchart_coord_determinant_fp
\fp_new:N \g__wheelchart_coord_x_fp
\fp_new:N \g__wheelchart_coord_y_fp
\fp_new:N \l__wheelchart_counter_or_clockwise_fp
\fp_new:N \g__wheelchart_def_angle_angle_fp
\fp_new:N \l__wheelchart_discrete_end_length_fp
\fp_new:N \l__wheelchart_discrete_factor_fp
\fp_new:N \l__wheelchart_discrete_inner_length_fp
\fp_new:N \l__wheelchart_discrete_level_fp
\fp_new:N \l__wheelchart_discrete_level_end_length_fp
\fp_new:N \l__wheelchart_discrete_level_start_length_fp
\fp_new:N \l__wheelchart_discrete_levels_sum_fp
\fp_new:N \l__wheelchart_discrete_outer_length_fp
\fp_new:N \l__wheelchart_discrete_start_length_fp
\fp_new:N \l__wheelchart_discrete_sublevel_end_length_fp
\fp_new:N \l__wheelchart_discrete_sublevel_start_length_fp
\fp_new:N \l__wheelchart_gap_fp
\fp_new:N \l__wheelchart_gap_max_angle_fp
\fp_new:N \l__wheelchart_gap_max_angle_def_fp
\fp_new:N \l__wheelchart_gap_polar_fp
\fp_new:N \l__wheelchart_gap_radius_fp
\fp_new:N \g__wheelchart_half_ex_over_one_cm_fp
\fp_new:N \l__wheelchart_inner_data_angle_pos_fp
\fp_new:N \l__wheelchart_inner_data_angle_shift_fp
\fp_new:N \l__wheelchart_inner_data_pos_fp
\fp_new:N \l__wheelchart_inner_data_sep_fp
\fp_new:c { \g__wheelchart_inner~end_x_fp }
\fp_new:c { \g__wheelchart_inner~end_y_fp }
\fp_new:N \l__wheelchart_inner_radius_fp
\fp_new:c { \g__wheelchart_inner~start_x_fp }
\fp_new:c { \g__wheelchart_inner~start_y_fp }
\fp_new:N \l__wheelchart_lines_fp
\fp_new:N \l__wheelchart_lines_angle_pos_fp
\fp_new:N \l__wheelchart_lines_angle_shift_fp
\fp_new:N \l__wheelchart_lines_ext_fp
\fp_new:N \l__wheelchart_lines_ext_dirsep_fp
\fp_new:N \l__wheelchart_lines_ext_fixed_left_fp
\fp_new:N \l__wheelchart_lines_ext_fixed_right_fp
\int_new:N \g__wheelchart_discrete_count_int
\int_new:N \l__wheelchart_discrete_levels_int
\int_new:N \l__wheelchart_discrete_partitioning_first_index_int
\int_new:N \l__wheelchart_discrete_partitioning_second_index_int
\int_new:N \l__wheelchart_discrete_sort_int
\int_new:N \l__wheelchart_discrete_space_at_borders_int
\int_set:Nn \l__wheelchart_discrete_space_at_borders_int { -1 }
\int_new:N \l__wheelchart_discrete_sublevels_int
\int_new:N \l__wheelchart_etoc_count_total_pages_int
\int_new:N \l__wheelchart_legend_columns_int
\int_new:N \l__wheelchart_legend_rows_int
\int_new:N \l__wheelchart_lines_ext_bottom_dir_int
\int_new:N \l__wheelchart_lines_ext_dir_int
\int_const:Nn \c__wheelchart_lines_ext_dir_left_int { -1 }
\int_const:Nn \c__wheelchart_lines_ext_dir_right_int { 1 }
\int_new:N \l__wheelchart_lines_ext_top_dir_int
\int_new:N \l__wheelchart_max_list_items_int
\seq_new:N \l__wheelchart_arc_data_seq
\seq_new:N \l__wheelchart_discrete_coefficients_first_seq
\seq_new:N \l__wheelchart_discrete_coefficients_second_seq
\seq_new:N \l__wheelchart_discrete_points_seq
\seq_new:N \l__wheelchart_list_seq
\seq_new:N \l__wheelchart_list_items_seq
\tl_new:N \WClegend
\tl_const:Nn \c__wheelchart_arc_around_text_aux_tl
{ sign ( \l__wheelchart_arc_data_dir_fp ) *
} \tl_new:N \l__wheelchart_arc_data_align_tl
\tl_new:N \l__wheelchart_data_anchor_tl
\tl_new:N \l__wheelchart_etoc_level_tl
\tl_new:N \l__wheelchart_etoc_name_tl
A.2 Functions

\cs_generate_variant:Nn \seq_set_split:Nnn { Nen , Neo , Nnf } { \Nen \Neo \Nnf }
\cs_generate_variant:Nn \tl_build_put_right:Nn { NV }
\cs_new_protected:Npn \__wheelchart_arc_around_text_plot_false:nn #1#2 #1#2
  { \fp_set:Nn \l__wheelchart_arc_start_angle_fp
  \__wheelchart_def_angle_plot_false:nnnnn
  \__wheelchart_arc_data_angle_pos_fp
  \__wheelchart_slices_tl \l__wheelchart_type_tl
  { \l__wheelchart_type_tl { default } }

  \path [ draw , / wheelchart / arc_style , / wheelchart / arc_#1_half ]
  { \fp_use:N \l__wheelchart_arc_start_angle_fp \c_colon_str \fp_use:N \l__wheelchart_arc_radius_fp } arc
\begin{code}
\begin{verbatim}
[ start-angle = \fp_use:N \l__wheelchart_arc_start_angle_fp ,
end-angle =
  \__wheelchart_def_angle_plot_false:nnnnn
  \{ \WCcount \}
  \{#2\}
  \{ 0 \}
  \{ \l__wheelchart_arc_pos_fp \}
  \{ \l__wheelchart_arc_sep_fp + \g__wheelchart_half_ex_over_one_cm_fp \},
radius = \fp_use:N \l__wheelchart_arc_radius_fp
]

\cs_new_protected:Npn \__wheelchart_arc_around_text_plot_true:nnn #1#2#3
{
  \__wheelchart_convex_comb_coord_plot:nnnnnnn
  \{ draw , / wheelchart / arc_style , / wheelchart / arc_#1_half \}
  \{ 1 \}
  \{ 0 \}
  \{ \l__wheelchart_plot_variable_tl \}
  \fp_eval:n
  \{ \l__wheelchart_arc_data_angle_pos_fp + 0.5 * \g__wheelchart_arc_data_aux_ii_fp *
      ( \c__wheelchart_arc_around_text_aux_tl + #2 )
  \} + (#3) * ( 1 - \l__wheelchart_plot_variable_tl )
  \{ \l__wheelchart_plot_variable_tl * \l__wheelchart_arc_data_angle_shift_fp \}
  \{ \l__wheelchart_arc_pos_fp \}
  \{ \l__wheelchart_arc_sep_fp \}
}

\cs_new_protected:Npn \__wheelchart_caption_and_title:nnnnn #1#2#3#4#5
{
  \__wheelchart_if_text:nn {#1}
  \{
\end{verbatim}
\end{code}
\node [ anchor = #2 , align = #3 , / wheelchart / #1_style ]
at ( $(#4) + (0, \{ #5 \ast (\pgfkeysvalueof{/ wheelchart / #1-sep} ) \})$ ) \{ \pgfkeysvalueof{/ wheelchart / #1} \};
\}
\cs_new:Npn \__wheelchart_convex_comb_coord_aux:n #1 { (#1) }
\cs_generate_variant:Nn \__wheelchart_convex_comb_coord_aux:n { o }
\cs_new:Npn \__wheelchart_convex_comb_coord_def:nnnn #1#2#3#4
{ $\fp_eval:n{\__wheelchart_inner_plot:nn{\fp_eval:n{\__wheelchart_outer_plot:nn{\fp_eval:n{(1 - (#1)) * \cs:w g__wheelchart_slice_inner_start_angle_\WCcount_fp\cs_end: + (#1) * \cs:w g__wheelchart_slice_inner_end_angle_\WCcount_fp\cs_end: + \l__wheelchart_counter_or_clockwise_fp * (#2)}}}{\fp_eval:n{\cs:w g__wheelchart_inner_radius_\WCcount_fp\cs_end: - (#4)}}}{\fp_eval:n{\cs:w g__wheelchart_inner_radius_\WCcount_fp\cs_end: - (#4)}}}}$
}$
\cs_new_protected:Npn \wheelchart_convex_comb_coord_plot_aux:nnnn \#1\#2\#3\#4
\{
\path [\#1] plot
[
  domain = \{\#2\} \colon_str \{\#3\},
  samples = \fp_use:c { g\wheelchart_samples\_WCcount\_fp },
  variable = \lwheelchart_plot_variable_tl
]
( \{\#4\} );
\}
\cs_generate_variant:Nn \wheelchart_convex_comb_coord_plot_aux:nnnn { nnno }
\cs_new_protected:Npn \wheelchart_convex_comb_coord_plot:nnnnnnn \#1\#2\#3\#4\#5\#6\#7
\{
\wheelchart_convex_comb_coord_plot_aux:nnno \{\#1\} \{\#2\} \{\#3\}
\{ \wheelchart_convex_comb_coord_def:nnnn \{\#4\} \{\#5\} \{\#6\} \{ \g\wheelchart_half_ex_over_one_cm_fp + (#7) \} \}
\}
\cs_new_protected:Npn \wheelchart_def_angle_aux:
\{
\fp_gset:Nn \g\wheelchart_def_angle_angle_fp
\{
\atand
( \{
  \pgf@xx * ( \{ y \{ \l\wheelchart_def_angle_2 \} - \{ y \{ \l\wheelchart_def_angle_1 \} \} )
- \pgf@xy * ( \{ x \{ \l\wheelchart_def_angle_2 \} - \{ x \{ \l\wheelchart_def_angle_1 \} \} )
\}) \l\wheelchart_coord_determinant_fp,
( \{
  \pgf@yy * ( \{ x \{ \l\wheelchart_def_angle_2 \} - \{ x \{ \l\wheelchart_def_angle_1 \} \} )
- \pgf@yx * ( \{ y \{ \l\wheelchart_def_angle_2 \} - \{ y \{ \l\wheelchart_def_angle_1 \} \} )
\}) \l\wheelchart_coord_determinant_fp
\}
\}
\cs_new_protected:Npn __wheelchart_def_angle:nnnn \#1\#2\#3\#4
{
  \bool_if:NTF __wheelchart_plot_bool
  {
    \path let \p { __wheelchart_def_angle_1 } = __wheelchart_point_plot_true:nnnnn \WCcount \{\#1\} \{\#2\} \{0\} \{\#4\} ,
    \p { __wheelchart_def_angle_2 } = __wheelchart_point_plot_true:nnnnn \WCcount \{\#1\} \{\#2\} \{1\} \{\#4\}
in [ / utils / exec =
    \bool_gset:Nn g__wheelchart_def_angle_radius_shift_bool
    { \fp_compare_p:n {
        \y { __wheelchart_def_angle_2 } - \y { __wheelchart_def_angle_1 } == 0
        &&
        \x { __wheelchart_def_angle_2 } - \x { __wheelchart_def_angle_1 } == 0
      }
    }
  }
  \bool_if:NT g__wheelchart_def_angle_radius_shift_bool
  {
    \__wheelchart_def_angle_aux: 
  }
}
}\bool_if:NT g__wheelchart_def_angle_radius_shift_bool
{
  \path let \p { __wheelchart_def_angle_1 } = __wheelchart_point_plot_true:nnnnn \WCcount \{\#1\} \{\#2\} \{0\} \{\#4\}
    \p { __wheelchart_def_angle_2 } = __wheelchart_point_plot_true:nnnnn \WCcount \{\#1\} \{\#2\} \{1\} \{\#4\}
in [ / utils / exec = { __wheelchart_def_angle_aux: } ]
\pgfmathparse{\Mod(\fp_use:N\g__wheelchart_def_angle_angle_fp, 360)}
\pgfmathparse{\Mod(\__wheelchart_def_angle_plot_false:nnnnn{\WCcount}{#1}{#2}{#3}{#4}, 360)}
\cs_new:Npn\__wheelchart_def_angle_plot_false_aux_angle:nn#1#2
{\(1 - (#2)\) * (\(\cs:w\g__wheelchart_slice_outer_start_angle_#1_fp\cs_end:\)
- \(\cs:w\g__wheelchart_slice_inner_start_angle_#1_fp\cs_end:\))
+ (#2) * (\(\cs:w\g__wheelchart_slice OUTER\_end\_angle_#1_fp\cs_end:\ - \cs:w\g__wheelchart_slice-inner\_end\_angle_#1_fp\cs_end:\))}
\cs_new:Npn\__wheelchart_def_angle_plot_false:nnnnn#1#2#3#4#5
{\fp_eval:n{\fp_compare:nNnTF{\__wheelchart_def_angle_plot_false_aux_angle:nn#1#2}{0}{0}{
asind{\sqrt{(\cs:w\g__wheelchart_outer_radius_#1_fp\cs_end:^2 + \cs:w\g__wheelchart_inner_radius_#1_fp\cs_end:^2 \times (2 + 2 \times \cosd(\__wheelchart_def_angle_plot_false_aux_angle:nn#1#2) + \cs:w\g__wheelchart_outer_radius_#1_fp\cs_end:^2 \times \cs:w\g__wheelchart_inner_radius_#1_fp\cs_end:^2)}}}}}}
\_\_wheelchart\_inner\_plot:nn
{ \fp_use:N \l__wheelchart\_slices\_orientation\_new\_angle_fp }
{ \fp_use:c { \g__wheelchart\_inner\_radius\_1_fp } }
),
\p { \l__wheelchart\_slices\_orientation\_4 } =
( \_\_wheelchart\_inner\_plot:nn
{ \fp_use:N \g__wheelchart\_angle_fp }
{ \fp_use:c { \g__wheelchart\_inner\_radius\_1_fp } }
)
in
[ / utils / exec =
{ \fp_gset:Nn \g__wheelchart\_slices\_orientation_fp
{ \sign ( 0.1 * \y \l__wheelchart\_slices\_orientation\_1 }

* ( \x \l__wheelchart\_slices\_orientation\_4 ) } - \x \l__wheelchart\_slices\_orientation\_2 )
+ 0.1 * \y \l__wheelchart\_slices\_orientation\_2 )
* ( \x \l__wheelchart\_slices\_orientation\_1 ) - \x \l__wheelchart\_slices\_orientation\_3 )
+ 0.1 * \y \l__wheelchart\_slices\_orientation\_3 )
* ( \x \l__wheelchart\_slices\_orientation\_2 ) - \x \l__wheelchart\_slices\_orientation\_4 )
+ 0.1 * \y \l__wheelchart\_slices\_orientation\_4 )
* ( \x \l__wheelchart\_slices\_orientation\_3 ) - \x \l__wheelchart\_slices\_orientation\_1 )
)
* \sign ( \l__wheelchart\_coord\_determinant_fp )
}
}

;\%the terms are multiplied with 0.1 to try to avoid an overflow
\fp_compare:nNnT \g__wheelchart\_slices\_orientation_fp \{ 0 \}
{ \fp_gset_eq:NN \g__wheelchart\_slices\_orientation_fp \l__wheelchart\_counter_or\_clockwise_fp }
\cs_new_protected:Npn \__wheelchart\_def\_outer\_radius:
{ \pgfmathparse { \pgfkeysvalueof { / wheelchart / outer-radius } } }
\cs_new_protected:Npn \__wheelchart_def_slice_keys:n #1
\{
  %note the double \{\ldots\} such that the contents is in a group
  %such that pgfkeys which are specific to the current slice are local for this slice
  \clist_if_in:NVT \l__wheelchart_slice_range_local_clist \WCcount
  \{
    \pgfkeys { / wheelchart , slice_final /. expanded = { \exp_not:v { l__wheelchart_slice\WCcount_keys_clist } } }
    \pgfkeys{ / wheelchart , slice_final_style }
  \}
\}
\endgroup}

\cs_new_protected:Npn \__wheelchart_def_WClegend:
\{
  \int_set:Nn \l__wheelchart_legend_columns_int
  \{ \fp_eval:n { ceil ( \WCtotalcount / ceil ( \WCtotalcount / ( \pgfkeysvalueof { / wheelchart / legend~columns } ) ) ) } } \}
  \tl_build_begin:N \WClegend
  \int_compare:nNnTF { \l__wheelchart_legend_columns_int } = { 1 }
    \{ \int_step_inline:nn { 2 } { \WCtotalcount } { \__wheelchart_legend_append:nn {##1} { \ } } \}
    \{ \int_set:Nn \l__wheelchart_legend_rows_int
      \{ \fp_eval:n { ceil ( \WCtotalcount / \l__wheelchart_legend_columns_int ) } \}
      \int_step_inline:nn { \l__wheelchart_legend_rows_int - 1 }
        \int_step_inline:nn { \l__wheelchart_legend_columns_int - 2 }
          \{ \__wheelchart_legend_append:nn {##1 \l__wheelchart_legend_rows_int * \l__wheelchart_legend_columns_int} { & \} \}
      \int_compare:nNnF { ##1 + ( \l__wheelchart_legend_columns_int - 1 ) * \l__wheelchart_legend_rows_int } > { \WCtotalcount }
        \{ \\} \}
  \\}
\endgroup
\__wheelchart_legend_append:nn { \l__wheelchart_legend_rows_int } { & }
\int_step_inline:nn { \l__wheelchart_legend_columns_int - 2 }
{ \__wheelchart_legend_append:nn { \l__wheelchart_legend_rows_int * ( ##1 + 1 ) } { & } }
\int_compare:nNnF { \l__wheelchart_legend_columns_int * \l__wheelchart_legend_rows_int } > { \WCtotalcount }
{ \__wheelchart_legend_append:nn { \l__wheelchart_legend_columns_int * \l__wheelchart_legend_rows_int } { & } }
\__wheelchart_legend_append:nn { 1 } { \ }
%at the moment it is unnecessary to set g__wheelchart_WCcount_counter to 1
%but this is done to be future-proof if the contents of \WClegend would be parsed in a way that prohibits the value for
%g__wheelchart_WCcount_counter to be larger than \WCtotalcount
\tl_build_end:N \WClegend
\cs_set:Npn \WCcount { \theg__wheelchart_WCcount_counter }
\cs_set:Npn \WCpercentage { \cs:w l__wheelchart_WCpercentage_\theg__wheelchart_WCcount_counter\cs_end: }
\cs_set:Npn \WCpercentagerounded { \cs:w l__wheelchart_WCpercentagerounded_\theg__wheelchart_WCcount_counter\cs_end: }
\str_if_eq:eeTF { \l__wheelchart_type_tl } { etoc }
{ \cs_set:Npn \WCetocthelinkedname
  { \cs:w g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_linked_name_\theg__wheelchart_WCcount_counter\cs_end: }
\cs_set:Npn \WCetocthelinkednumber
  { \cs:w g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_linked_number_\theg__wheelchart_WCcount_counter\cs_end: }
\cs_set:Npn \WCetocthelinkedpage
  { \cs:w g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_linked_page_\theg__wheelchart_WCcount_counter\cs_end: }
\cs_set:Npn \WCetocthename
  { \cs:w g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_name_\theg__wheelchart_WCcount_counter\cs_end: }
\cs_set:Npn \WCetocthenumber
  { \cs:w g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_number_\theg__wheelchart_WCcount_counter\cs_end: }
\cs_set:Npn \WCetocthepage
  { \cs:w g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_page_\theg__wheelchart_WCcount_counter\cs_end: }
\cs_set:Npn \WCetocthenumberofpages
  { \cs:w g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_number_of_pages_\theg__wheelchart_WCcount_counter\cs_end: }
}
\clist_if_empty:NTF \l__wheelchart_header_clist
{ \int_step_inline:nn { \l__wheelchart_max_list_items_int }

}
\cs_set:cpn { WCvar\int_to_Alph:n {##1} } 
\{ \cs:w l__wheelchart_item_WCvar\int_to_Alph:n {##1}_\theg__wheelchart_WCcount_counter\cs_end: \}
\}
\int_step_inline:nn { \l__wheelchart_max_list_items_int }
\{ \cs_set:cpn { \pgfkeysvalueof { / wheelchart / header~prefix } \clist_item:Nn \l__wheelchart_header_clist {##1} } 
\{ \cs:w 
  l__wheelchart_item_\pgfkeysvalueof { / wheelchart / header~prefix } 
  \clist_item:Nn \l__wheelchart_header_clist {##1}_\theg__wheelchart_WCcount_counter
\cs_end:
\}
\}
\setcounter { g__wheelchart_WCcount_counter } { 1 }
\}
\cs_new:Npn \__wheelchart_diff_atan:nnnn #1#2#3#4
\{ Mod
\{ \fp_eval:n
\{ \g__wheelchart_slices_orientation_fp * 
\{ atand 
\{ \cs:w \g__wheelchart_#3_y_fp\cs_end: - \cs:w \g__wheelchart_#4_y_fp\cs_end: , \cs:w \g__wheelchart_#3_x_fp\cs_end: - \cs:w \g__wheelchart_#4_x_fp\cs_end: 
\}
- atand 
\{ \cs:w \g__wheelchart_#1_y_fp\cs_end: - \cs:w \g__wheelchart_#2_y_fp\cs_end: , \cs:w \g__wheelchart_#1_x_fp\cs_end: - \cs:w \g__wheelchart_#2_x_fp\cs_end: 
\}
\}
\}
\cs_new_protected:Npn \__wheelchart_discrete_algorithm:
{
\__wheelchart_def_fp:nn { discrete_factor } { discrete-factor }
\__wheelchart_def_fp:nn { gap_radius } { gap-radius }
\__wheelchart_def_outer_radius:
\__wheelchart_def_inner_radius:
\int_compare:nNnT { \l__wheelchart_discrete_space_at_borders_int } = { -1 }
{
\pgfkeys
{
/ errors / boolean-expected /. expanded =
{ discrete-space-at-borders }
{ \pgfkeysvalueof { / wheelchart / discrete-space-at-borders } }
}
\seq_clear:N \l__wheelchart_discrete_points_seq
\bool_if:NTF \l__wheelchart_plot_bool
{
\__wheelchart_def_fp:nn { samples } { samples }
\fp_zero:N \l__wheelchart_discrete_outer_length_fp
\__wheelchart_discrete_def_coord:nn { outer } { 0 }
\int_step_inline:nn { \fp_use:N \l__wheelchart_samples_fp - 1 }
{
\__wheelchart_discrete_def_coord:nn { outer } {##1}
\fp_add:Nn \l__wheelchart_discrete_outer_length_fp
{
\sqrt
{
( \g__wheelchart_coord_x_fp - \g__wheelchart_previous_coord_x_fp ) ^ 2
+ ( \g__wheelchart_coord_y_fp - \g__wheelchart_previous_coord_y_fp ) ^ 2
}
}
\_\_wheelchart_discrete_def_coord:nn \{ inner \} \( \{ l\_\_wheelchart_samples_fp - 1 \} \)
\fp_set:Nn \l\_\_wheelchart_discrete_end_length_fp
\{ sqrt
\{
\g__wheelchart_coord_x_fp - \g__wheelchart_previous_coord_x_fp \}^2
+ \{ \g__wheelchart_coord_y_fp - \g__wheelchart_previous_coord_y_fp \}^2
\}
\fp_zero:N \l\_\_wheelchart_discrete_inner_length_fp
\int_step_inline:nnnn \{ \fp_use:N \l\_\_wheelchart_samples_fp - 2 \} \{ -1 \} \{ 0 \}
\{ \_\_wheelchart_discrete_def_coord:nn \{ inner \} \#
\fp_add:Nn \l\_\_wheelchart_discrete_inner_length_fp
\{ sqrt
\{
\g__wheelchart_coord_x_fp - \g__wheelchart_previous_coord_x_fp \}^2
+ \{ \g__wheelchart_coord_y_fp - \g__wheelchart_previous_coord_y_fp \}^2
\}
\}
\_\_wheelchart_discrete_def_coord:nn \{ outer \} \{ 0 \}
\fp_set:Nn \l\_\_wheelchart_discrete_start_length_fp
\{ sqrt
\{
\g__wheelchart_coord_x_fp - \g__wheelchart_previous_coord_x_fp \}^2
+ \{ \g__wheelchart_coord_y_fp - \g__wheelchart_previous_coord_y_fp \}^2
\}
\}
\fp_set:Nn \l\_\_wheelchart_discrete_outer_length_fp
\{ abs ( \l\_\_wheelchart_total_angle_fp \ast deg \ast \l\_\_wheelchart_outer_radius_fp ) \}
\fp_set:Nn \l\_\_wheelchart_discrete_end_length_fp \{ abs ( \l\_\_wheelchart_outer_radius_fp - \l\_\_wheelchart_inner_radius_fp ) \}
\fp_set:Nn \l\_\_wheelchart_discrete_inner_length_fp
\{ abs ( \l\_\_wheelchart_total_angle_fp \ast deg \ast \l\_\_wheelchart_inner_radius_fp ) \}
%note the abs ( ... ) because \l\_\_wheelchart_total_angle_fp can be negative
\% and $\text{l}_\text{wheelchart}\_outer\_radius\_fp$ can be smaller than $\text{l}_\text{wheelchart}\_inner\_radius\_fp$
\fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_start\_length\_fp \text{l}_\text{wheelchart}\_discrete\_end\_length\_fp
\str_case:enF { \pgfkeysvalueof { / wheelchart / discrete~partitioning } }
{ radius }
{ \int_set:Nn \text{l}_\text{wheelchart}\_discrete\_partitioning\_first\_index\_int \{ 1 \} \int_set:Nn \text{l}_\text{wheelchart}\_discrete\_partitioning\_second\_index\_int \{ 2 \} \fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_level\_start\_length\_fp \text{l}_\text{wheelchart}\_discrete\_inner\_length\_fp \fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_level\_end\_length\_fp \text{l}_\text{wheelchart}\_discrete\_outer\_length\_fp \fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_sublevel\_start\_length\_fp \text{l}_\text{wheelchart}\_discrete\_start\_length\_fp \fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_sublevel\_end\_length\_fp \text{l}_\text{wheelchart}\_discrete\_end\_length\_fp
}{ angle }
{ \int_set:Nn \text{l}_\text{wheelchart}\_discrete\_partitioning\_first\_index\_int \{ 2 \} \int_set:Nn \text{l}_\text{wheelchart}\_discrete\_partitioning\_second\_index\_int \{ 1 \} \fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_level\_start\_length\_fp \text{l}_\text{wheelchart}\_discrete\_start\_length\_fp \fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_level\_end\_length\_fp \text{l}_\text{wheelchart}\_discrete\_end\_length\_fp \fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_sublevel\_start\_length\_fp \text{l}_\text{wheelchart}\_discrete\_inner\_length\_fp \fp_set_eq:NN \text{l}_\text{wheelchart}\_discrete\_sublevel\_end\_length\_fp \text{l}_\text{wheelchart}\_discrete\_outer\_length\_fp \int_set:Nn \text{l}_\text{wheelchart}\_discrete\_sort\_int \{ 3 - \text{l}_\text{wheelchart}\_discrete\_sort\_int \} \text{ \int_set:Nn \text{l}_\text{wheelchart}\_discrete\_levels\_int \fp_eval:n \{ \text{ max } \}
} \pgfkeys
{ / errors / unknown-choice-value /. expanded =
  \pgfkeysvalueof { / wheelchart / discrete~partitioning } }
} \int_set:Nn \text{l}_\text{wheelchart}\_discrete\_levels\_int
round
{
  sqrt
  {
    ( ( \l__wheelchart_discrete_sublevel_start_length_fp + \l__wheelchart_discrete_sublevel_end_length_fp )
      * \l__wheelchart_discrete_factor_fp
    )
    / ( ( \l__wheelchart_discrete_level_start_length_fp + \l__wheelchart_discrete_level_end_length_fp )
    )
    * \WCtotalnum
  
  ),
    1
  }
}
\int_gzero:N \g__wheelchart_discrete_count_int
\fp_zero:N \l__wheelchart_discrete_levels_sum_fp
\int_step_inline:nn { \l__wheelchart_discrete_levels_int - 1 }
{
  \fp_set:Nn \l__wheelchart_discrete_level_fp
  {
    ( #1 - 0.5 * ( 1 + \l__wheelchart_discrete_space_at_borders_int )
      / ( \l__wheelchart_discrete_levels_int - \l__wheelchart_discrete_space_at_borders_int )
    }
  \fp_add:Nn \l__wheelchart_discrete_levels_sum_fp { \l__wheelchart_discrete_level_fp }
\int_set:Nn \l__wheelchart_discrete_sublevels_int
{
  \fp_eval:n
  {
    round
    {
      ( ( #1 ) * \l__wheelchart_discrete_level_start_length_fp
        + \l__wheelchart_discrete_levels_sum_fp
        * ( \l__wheelchart_discrete_level_end_length_fp - \l__wheelchart_discrete_level_start_length_fp )
      )
    }
  
  )
}
\l__wheelchart_discrete_levels_int \* 0.5 \\
* ( ( \l__wheelchart_discrete_level_start_length_fp + \l__wheelchart_discrete_level_end_length_fp ) \\
) \\
* \WCtotalnum - \g__wheelchart_discrete_count_int

\int_gadd:Nn \g__wheelchart_discrete_count_int { \l__wheelchart_discrete_sublevels_int }
\int_compare:nNnTF { \l__wheelchart_discrete_sublevels_int } = { 1 }
{
  \seq_put_right:Ne \l__wheelchart_discrete_points_seq
  { 0.5 \\
    / \fp_use:N \l__wheelchart_discrete_level_fp
  }
}

\int_step_inline:nn { \l__wheelchart_discrete_sublevels_int }
{ \seq_put_right:Ne \l__wheelchart_discrete_points_seq
  { \fp_eval:n \\
    \l__wheelchart_discrete_level_fp \\
    \l__wheelchart_discrete_space_at_borders_int \\
    \fp_eval:n \\
    \l__wheelchart_discrete_space_at_borders_int \\
    \l__wheelchart_discrete_space_at_borders_int \\
    \l__wheelchart_discrete_space_at_borders_int \\
  }
  %the denominator is 0 if \l__wheelchart_discrete_sublevels_int = \l__wheelchart_discrete_space_at_borders_int = 1 \\
  %thus the case when \l__wheelchart_discrete_sublevels_int = 1 is treated separately above
  /
  \fp_use:N \l__wheelchart_discrete_level_fp
}

\int_compare:nNnTF { \l__wheelchart_discrete_levels_int } = { 1 }
\__ wheelchart_outer_plot:nn
\{ \fp eval:n
\{ \fp eval:n
  \l__ wheelchart_start_angle_fp + \l__ wheelchart_counter_or_clockwise_fp * \l__ wheelchart_total_angle_fp *
  \{ \seq item:Nn \l__ wheelchart_discrete_coefficients_first_seq
  \{ \l__ wheelchart_discrete_partitioning_first_index_int \}
  \}
\}
\{ \fp use:N \l__ wheelchart_outer_radius_fp \}
\}
\coordinate (g__wheelchart_slice_##1_####1_coordinate) at
\{ \fp eval:n
\{ \fp eval:n
  \l__ wheelchart_start_angle_fp + \l__ wheelchart_counter_or_clockwise_fp * \l__ wheelchart_total_angle_fp *
  \{ \seq item:Nn \l__ wheelchart_discrete_coefficients_first_seq
  \{ \l__ wheelchart_discrete_partitioning_first_index_int \}
  \}
\}
c_colon_str \fp eval:n
\{
\cs_new_protected:Npn \_\wheelchart_gdef_count_fp:nn #1#2
{
  \fp_gzero_new:c { g_\wheelchart_#1\_count \_fp }
  \pgfmathparse { \pgfkeysvalueof { / wheelchart / #2 } }
  \fp_gset:cn { g_\wheelchart_#1\_count \_fp } { \pgfmathresult }
}
\cs_new_protected:Npn \_\wheelchart_if_text:nn #1#2
{
  \hspace{-2pt}%https://tex.stackexchange.com/questions/42280/expand-away-empty-macros-within-ifthenelse
  \hspace{-2pt}%https://tex.stackexchange.com/questions/44919/proper-way-to-detect-empty-blank-text
  \begin { pgfinterruptpicture }
  \hspace{-2pt}%https://tex.stackexchange.com/questions/459853/savebox-within-tikzpicture-results-in-an-empty-savebox
  \hbox_gset:Nn \g_\wheelchart_if_text_box { \pgfkeysvalueof { / wheelchart / #1 } }
  \end { pgfinterruptpicture }
  \dim_compare:nNnT { \box_wd:N \g_\wheelchart_if_text_box } > { 0 \text{ pt} }
  { \texttt { #2 } }
}
\cs_new_protected:Npn \_\wheelchart_initial:n #1
{
  \str_case:en { \l_\wheelchart_type_tl }
  { default }
  { \tl_if_empty:nTF {#1} }
  { \cs_set:Npn \WCtotalcount { 0 } }
  { \cs:seq_set_split:Ne\l_\wheelchart_list_seq { \pgfkeysvalueof { / wheelchart / separator~rows } } {#1} }
  \cs_set:Npe \WCtotalcount { \seq_count:N \l_\wheelchart_list_seq }
  \int_zero:N \l_\wheelchart_max_list_items_int
  \seq_map_indexed_inline:Nn \l_\wheelchart_list_seq
  { \cs:seq_set_split:Ne\l_\wheelchart_list_items_seq { \pgfkeysvalueof { / wheelchart / list~items~size } } {#1} }
}

99
{\pgfkeysvalueof {/wheelchart/separatorem}}
{##2}
\int_compare:NnT {\seq_count:N \l__wheelchart_list_items_seq} > {\l__wheelchart_max_list_items_int}
{\int_set:Nn \l__wheelchart_max_list_items_int {\seq_count:N \l__wheelchart_list_items_seq}}
%make sure that the namespace \l__wheelchart_item below is unique
\clist_if_empty:NNTF \l__wheelchart_header_clist
{
\seq_map_indexed_inline:Nn \l__wheelchart_list_items_seq
{\cs_set:cpn {l__wheelchart_item_WCvar\int_to_Alph:n {####1}##1} {####2}}
}
{
\seq_map_indexed_inline:Nn \l__wheelchart_list_items_seq
{\cs_set:cpn {l__wheelchart_item_\pgfkeysvalueof{/wheelchart/header~prefix} \clist_item:Nn \l__wheelchart_header_clist {####1}##1} {####2}}
}
}

{totalcount}
{
\cs_set:Npe \WCtotalcount {\fp_use:N \l__wheelchart_total_count_fp}
\cs_set:Npn \WCvarA {1}
\cs_set:Npn \WCvarB {0}
\cs_set:Npn \WCvarC {0}
}
{etoc}
{
\bool_if:NNTF \l__wheelchart_etoc_use_name_bool
{\int_if_exist:CTF {g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _count_int}
{\cs_set:Npe \WCtotalcount {\int_use:c {g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _count_int}}}
{\cs_set:Npn \WCtotalcount {0}}
}
{\IfPackageLoadedTF { etoc } {} \PackageError { wheelchart } { The-package-etoc-must-be-loaded-to-use-the-key-etoc-level } {} } \etocsetlevel { part } { 0 } \etocsetlevel { chapter } { 0 } \etocsetlevel { section } { 0 } \etocsetlevel { subsection } { 0 } \etocsetlevel { subsubsection } { 0 } \etocsetlevel { paragraph } { 0 } \etocsetlevel { subparagraph } { 0 } \etocsetlevel { \l__wheelchart_etoc_level_tl } { -1 } % these level changes are local to the current group \etocsetnexttocdepth { -1 } % only for the next toc \etocsetstyle { \l__wheelchart_etoc_level_tl } {} {} \{ \int_compare:nNnT { \cs:w g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _count_int\cs_end: } > { 0 } \{ \cs_gset:cpe \{ \g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_number_of_pages_\int_use:c { g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _count_int } \} \} \{ \int_eval:n \{ \etocthepage - \cs:w \g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_page_\int_use:c { g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _count_int } \cs_end: \} \} \int_gincr:c { g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _count_int } \cs_gset_eq:cN \{ \g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _the_linked_name_\int_use:c { g__wheelchart_etoc_item_\l__wheelchart_etoc_name_tl _count_int } \} \etocthelinkedname \etocthelinkedname}
\cs_new:Npn \__wheelchart_outer_plot:nn #1#2
  { (#1) \c_colon_str (#2) }

\cs_new:Npn \__wheelchart_point_plot_false:nnnnn #1#2#3#4#5
  { ( \__wheelchart_def_angle_plot_false:nnnn {#1} {#2} {#3} {#4} {#5} \c_colon_str \__wheelchart_def_radius:nnn {#1} {#4} {#5} ) }

\cs_new:Npn \__wheelchart_point_plot_true:nnnnn #1#2#3#4#5
  { \__wheelchart_convex_comb_coord_aux:o { \__wheelchart_convex_comb_coord_def:nnnn {#2} {#3} {#4} {#5} } }
\fp_set:Nn \l__wheelchart_slices_arc_angle_fp
\fp_compare:nNnTF { \l__wheelchart_slices_arc_A_abs_fp } < { 0.01 } 
{#5}
\fp_compare:nNnT { \l__wheelchart_slices_arc_B_fp } < { 1 } 
{ \fp_compare:nNnF { \l__wheelchart_slices_arc_coord_fp } = { 0 } 
{ -- 
 ( g__wheelchart_slice\WCcount _#3_coordinate )
! { \fp_eval:n { \l__wheelchart_slices_arc_coord_fp / 2 } } ! 
( g__wheelchart_slice\WCcount _#4_coordinate )
$ 
) 
} 
arc 
[ start-angle =
\fp_eval:n
{\l__wheelchart_slices_arc_rotate_fp
- \g__wheelchart_slices_orientation_fp * \l__wheelchart_slices_arc_angle_fp}
},
end-angle =
{\fp_eval:n
{\fp_eval:n
{\l__wheelchart_slices_arc_rotate_fp + \g__wheelchart_slices_orientation_fp *
(\sign(\l__wheelchart_slices_arc_A_fp) * 180 + \l__wheelchart_slices_arc_angle_fp)
},
radius =
{\fp_eval:n
{\fp_eval:n
{0.25 * (1 - \l__wheelchart_slices_arc_B_fp)
* \abs((1/\l__wheelchart_slices_arc_A_fp) + \l__wheelchart_slices_arc_A_fp)
* \sqrt{
(\cs:w\g__wheelchart_#3_x_fp\cs_end: - \cs:w\g__wheelchart_#4_x_fp\cs_end:)^2
+ (\cs:w\g__wheelchart_#3_y_fp\cs_end: - \cs:w\g__wheelchart_#4_y_fp\cs_end:)^2
)}}
}}}\fp_compare:nNnF{\l__wheelchart_slices_arc_coord_fp}={0}{#5}
/ utils / exec =
{
\pgfmathparse {#1}
\fp_set:Nn \l__wheelchart_slices_arrow_A_fp { \pgfmathresult }
\fp_compare:nNnF { \l__wheelchart_slices_arrow_A_fp } = { 0 }
{
\pgfmathparse {#2}
\fp_set:Nn \l__wheelchart_slices_arrow_B_fp { \pgfmathresult }
\fp_set:Nn \l__wheelchart_slices_arrow_coord_fp
{
\l__wheelchart_slices_arrow_A_fp < 0 \&\& \l__wheelchart_slices_arrow_B_fp < 0 ? 0 : \l__wheelchart_slices_arrow_B_fp
}
}
\fp_compare:nNnTF { \l__wheelchart_slices_arrow_A_fp } = { 0 }
{#5}
{
\fp_compare:nNnT { \l__wheelchart_slices_arrow_B_fp } < { 1 }
{
\fp_compare:nNnF { \l__wheelchart_slices_arrow_coord_fp } = { 0 }
{
--
($
(g__wheelchart_slice\_\text{#3\_coordinate})
! (\fp_eval:n { \l__wheelchart_slices_arrow_coord_fp / 2 })
!(g__wheelchart_slice\_\text{#4\_coordinate})$
$\)

--
(\fp_eval:n
{
(\cs:w g__wheelchart\_#3\_x_fp\cs_end: + \cs:w g__wheelchart\_#4\_x_fp\cs_end: +

(g__wheelchart_slices_orientation_fp * ( \l__wheelchart_slices_arrow_coord_fp - 1 ))
}
\cs_new:Npn \__wheelchart_slices_to:nn #1#2 {

}
A.3 Pgfkeys

\pgfkeys
{
  / wheelchart /. is-family ,
  / wheelchart ,
  after-slices /. initial = {} ,
  anchor-xsep /. initial = 5 ,
  anchor-ysep /. initial = 5 ,
  arc /. style =
  {
    bool_set_true = \lwheelchart_arc_bool ,
    arc_style /. style = {(#1)}
  },
  arc_style /. style = {} ,
  arc-around-line /. initial = 1 ,
  arc-around-text /. default = true ,%this key is not set up with /is choice or \bool_set:Nn
  %such that for example \WCvarA can be used as value for this key
  arc-around-text /. initial = false ,
}
arc-data /. initial = {} ,
arc-data-align /. code = { \tl_set:Nn \l__wheelchart_arc_data_align_tl {#1} } ,%this key is not set up with /.is choice
%such that for example \WCvarA can be used as value for this key
arc-data-align = center ,
arc-data-angle-pos /. initial = 0.5 ,
arc-data-angle-shift /. initial = 0 ,
arc-data-dir /. initial = 1 ,
arc-data-expand /. initial = n ,
arc-data-line-sep-factor /. initial = 1 ,
arc-data-pos /. initial = 1 ,
arcd-data-sep /. initial = 1 ex / 1 cm ,
arcd-data-style /. style = { arc_data_style /. style = {#1} } ,
arcd_data_style /. style = {} ,
arcd-first-half /. style = { arc_first_half /. style = {#1} } ,
arcd_first_half /. style = {} ,
arcd-pos /. initial = 1 ,
arcd-second-half /. style = { arc_second_half /. style = {#1} } ,
arcd_second_half /. style = {} ,
arcd-sep /. initial = 1 ex / 1 cm ,
at /. initial = { ( 0 , 0 ) } ,
before-slices /. initial = {} ,
bool_set_true /. code = { \bool_set_true:N #1 } ,
caption /. initial = {} ,
caption-left /. initial = {} ,
caption-left-sep /. initial = 0.5 ,
caption-left-style /. style = { caption-left_style /. style = {#1} } ,
caption-left_style /. style = {} ,
caption-sep /. initial = 0.5 ,
caption-style /. style = { caption_style /. style = {#1} } ,
caption_style /. style = {} ,
contour /. style =
{
    bool_set_true = \l__wheelchart_contour_bool ,
    contour_style /. style = {#1}
} ,
contour_style /. style = {} ,
counterclockwise /. is-choice ,
counterclockwise / false /. code = { \fp_set:Nn \l__wheelchart_counter_or_clockwise_fp { -1 } } ,
counterclockwise / false /. value-forbidden ,
counter clockwise / true /. code = { \fp_set:Nn \l__wheelchart_counter_or_clockwise_fp { 1 } },
counter clockwise / true /. value-forbidden,
counter clockwise /. default = true,
counter clockwise = false,
data /. initial = { \WCvarC } ,
data-angle-pos /. initial = 0.5 ,
data-angle-shift /. initial = 0 ,
data-pos /. initial = 1 ,
data-sep /. initial = 0.2 ,
data-style /. style = { data_style /. style = {#1} } ,
data_style /. style = {} ,
discrete /. is-choice,
discrete / false /. code = { \bool_set_false:N \l__wheelchart_discrete_bool } ,
discrete / false /. value-forbidden,
discrete / true /. code = { \bool_set_true:N \l__wheelchart_discrete_bool } ,
discrete / true /. value-forbidden,
discrete /. default = true,
discrete = false,
discrete-factor /. initial = 1 ,
discrete-partitioning /. initial = radius ,
discrete-pic /. initial = {} ,
discrete-sort /. is-choice,
discrete-sort / angle /. code = { \int_set:Nn \l__wheelchart_discrete_sort_int { 1 } } ,
discrete-sort / angle /. value-forbidden,
discrete-sort / radius /. code = { \int_set:Nn \l__wheelchart_discrete_sort_int { 2 } } ,
discrete-sort / radius /. value-forbidden,
discrete-sort = angle,
discrete-space-at-borders /. is-choice ,%this key is not set up with /.is if because an initial value is unwanted for this key
discrete-space-at-borders / false /. code = { \int_set:Nn \l__wheelchart_discrete_space_at_borders_int { 1 } } ,
discrete-space-at-borders / false /. value-forbidden,
discrete-space-at-borders / true /. code = { \int_set:Nn \l__wheelchart_discrete_space_at_borders_int { 0 } } ,
discrete-space-at-borders / true /. value-forbidden,
discrete-space-at-borders /. default = true,
domain /. style-args /. expanded = { #1 \c_colon_str #2 } 
  { 
    clockwise ,
    start-angle = {##1} ,
    total-angle = { (##2) - (##1) }
  },
etoc-code /. initial = { \tableofcontents } ,
etoc-count-total-pages /. code = { \int_set:Nn \l__wheelchart_etoc_count_total_pages_int {#1} } ,
etoc-level /. code =
\tl_set:Nn \l__wheelchart_type_tl { etoc }
\tl_set:Nn \l__wheelchart_etoc_level_tl {#1}
,etoc-name /. code = { \tl_set:Nn \l__wheelchart_etoc_name_tl {#1} } ,
etoc-name = {} ,etoc-use-name /. code =
\tl_set:Nn \l__wheelchart_etoc_name_tl {#1}
\tl_set:Nn \l__wheelchart_type_tl { etoc }
\bool_set_true:N \l__wheelchart_etoc_use_name_bool
,expand-list /. is-choice ,expand-list / false /. code = { \tl_set:Nn \l__wheelchart_expand_list_tl { n } } ,expand-list / false /. value-forbidden ,expand-list / once /. code = { \tl_set:Nn \l__wheelchart_expand_list_tl { o } } ,expand-list / once /. value-forbidden ,expand-list / true /. code = { \tl_set:Nn \l__wheelchart_expand_list_tl { e } } ,expand-list / true /. value-forbidden ,expand-list = once ,expand-list-items /. is-choice ,expand-list-items / false /. code = { \tl_set:Nn \l__wheelchart_expand_list_items_tl { n } } ,expand-list-items / false /. value-forbidden ,expand-list-items / once /. code = { \tl_set:Nn \l__wheelchart_expand_list_items_tl { o } } ,expand-list-items / once /. value-forbidden ,expand-list-items / true /. code = { \tl_set:Nn \l__wheelchart_expand_list_items_tl { e } } ,expand-list-items / true /. value-forbidden ,expand-list-items = false ,explode /. initial = 0 ,explode /. default = 0.2 ,for-loop-end /. initial = {} ,for-loop-start /. initial = {} ,gap /. initial = 0 ,gap /. default = 0.05 ,gap-max-angle /. initial = 180 ,gap-polar /. initial = 0 ,
\begin{verbatim}
gap-polar /. default = 1 ,
gap-radius /. initial = 0 ,
gap-radius /. default = 0.05 ,% the same default value as for gap
header /. code = { \clist_set:Nn \l__wheelchart_header_clist {#1} } ,
header-prefix /. initial = WC ,
inner-data /. initial = {} ,
inner-data-angle-pos /. initial = 0.5 ,
inner-data-angle-shift /. initial = 0 ,
inner-data-pos /. initial = 0 ,
inner-data-sep /. initial = 0.2 ,
inner-data-style /. style = { inner_data_style /. style = {#1} } ,
inner_data_style /. style = {} ,
inner-plot /. style =
{ bool_set_true = \l__wheelchart_plot_bool ,
/ utils / exec = { \cs_set:Npn \__wheelchart_inner_plot:nn ##1##2 {#1} } ,
slices-inner =
{ -- plot
 [ domain = \fp_use:c { g__wheelchart_slice_inner_end_angle_\WCcount _fp } \\
c_colon_str \fp_use:c { g__wheelchart_slice_inner_start_angle_\WCcount _fp } ,
samples = \fp_use:c { g__wheelchart_samples_\WCcount _fp } ,
variable = \l__wheelchart_inner_plot_variable_tl ,
/ wheelchart / inner_plot_style
] 
\__wheelchart_inner_plot:nn
{ \l__wheelchart_inner_plot_variable_tl }
{ \fp_use:c { g__wheelchart_inner_radius_\WCcount _fp } }
}
}
inner-plot-style /. style = { inner_plot_style /. style = {#1} } ,
inner_plot_style /. style = {} ,
inner-radius /. initial = 2 ,
legend /. initial = {} ,
\end{verbatim}
legend-columns /. initial = 1 ,
legend-entry /. initial = {} ,
legend-only /. code = { \bool_set:Nn \l__wheelchart_legend_only_bool { \cs:w c_#1_bool\cs_end: } } ,
legend-only /. default = true ,
legend-only = false ,
legend-row /. code =
  { \bool_set_true:N \l__wheelchart_legend_row_bool
    \tl_set:Nn \l__wheelchart_legend_row_tl {#1}
  },
lines /. initial = 0 ,
lines /. default = 1 ,
lines-angle-pos /. initial = 0.5 ,
lines-angle-shift /. initial = 0 ,
lines-ext /. initial = 0 ,
lines-ext /. default = 0.5 ,
lines-ext-bottom-dir /. code = { \int_set_eq:Nc \l__wheelchart_lines_ext_bottom_dir_int { c__wheelchart_lines_ext_dir_#1_int } } ,
lines-ext-bottom-dir = right ,
lines-ext-dir /. code =
  { \bool_set_true:N \l__wheelchart_lines_ext_dir_bool
    \int_set_eq:Nc \l__wheelchart_lines_ext_dir_int { c__wheelchart_lines_ext_dir_#1_int } }
lines-ext-dir = right ,
lines-ext-dirsep /. initial = 0 ,
lines-ext-fixed /. default = true ,
\%this key is not set up with /.is choice or \bool_set:Nn \%such that for example \WCvarA can be used as value for this key
lines-ext-fixed /. initial = false ,
lines-ext-fixed-left /. initial =
  { \fp_eval:n
      { \l__wheelchart_lines_ext_dir_int *
        ( \cs:w g__wheelchart_outer_radius_\WCcount_fp\cs_end: + \l__wheelchart_lines_sep_fp
          + \l__wheelchart_lines_fp + \l__wheelchart_lines_ext_fp
        )
      }
  },
lines-ext-fixed-right /. initial =
\begin{verbatim}
\fp_eval:n
{\l__wheelchart_lines_ext_dir_int * 
(\cs:w g__wheelchart_outer_radius_\WCcount_fp\cs_end: + \l__wheelchart_lines_sep_fpl + \l__wheelchart_lines_fp + \l__wheelchart_lines_ext_fp)}
}
}
lines-ext-left-anchor /. initial = mid-east,
lines-ext-right-anchor /. initial = mid-west,
lines-ext-top-dir /. code = {\int_set_eq:Nc \l__wheelchart_lines_ext_top_dir_int {c__wheelchart_lines_ext_dir_#1_int} },
lines-ext-top-dir = right,
lines-pos /. initial = 1,
lines-sep /. initial = 0.2,
lines-style /. style = {lines_style /. style = {#1} },
lines_style /. style = {},
middle /. initial = {} ,
middle-fill /. style =
{
  bool_set_true = \l__wheelchart_middle_fill_bool ,
  middle_fill /. style = {#1}
},
middle_fill /. style = {},
middle-style /. style = {middle_style /. style = {#1} },
middle_style /. style = {},
name /. code = {\tl_gset:Ne \g__wheelchart_name_tl \{#1\} },
name = wheelchart@name,
outer-plot /. style =
{
  bool_set_true = \l__wheelchart_plot_bool ,
  /utils/exec = {\cs_set:Npn \__wheelchart_outer_plot:nn #1#2 {#1} },
  slices-outer =
  {
    -- plot
    [ 
    domain = \fp_use:c { g__wheelchart_slice_outer_start_angle_\WCcount_fp }
  }}
\end{verbatim}
\ccolon_str
\fp_use:c { g__wheelchart_slice_outer_end_angle\WCcount _fp } ,
samples = \fp_use:c { g__wheelchart_samples\WCcount _fp } ,
variable = \l__wheelchart_outer_plot_variable_tl ,
/ wheelchart / outer_plot_style
]
(\__wheelchart_outer_plot:nn
 { \l__wheelchart_outer_plot_variable_tl }
 { \fp_use:c { g__wheelchart_outer_radius\WCcount _fp } }
)
)
, outer-plot-style /. style = { outer_plot_style /. style = {#1} } ,
outer_plot_style /. style = {}
, outer-radius /. initial = 3 ,
perc-precision /. initial = 0 ,
pie /. code = { \bool_set:Nn \l__wheelchart_pie_bool { \cs:w c_#1_bool\cs_end: } } ,
pie /. default = true ,
pie = false ,
plot /. style =
 { inner-plot = {#1} ,
 outer-plot = {#1} }
, plot-style /. style =
 { inner-plot-style = {#1} ,
 outer-plot-style = {#1} }
, radius /. style-2-args =
 { inner-radius = {#1} ,
 outer-radius = {#2} }
, samples /. initial = 25 ,%the same number as /tikz/samples
separator-columns /. initial = / ,
separator-rows /. initial = { , } ,
slice_final /. style = { slice_final_style /. style = {#1} } ,
slice_final_style /. style = {},
slices /. code =
   {\
\bool_set_true:N \l__wheelchart_slices_bool
\tl_set:Nn \l__wheelchart_slices_tl {#1}
},
slices-angle-pos /. initial = 0.5 ,
slices-angle-shift /. initial = 0 ,
slices-arc /. style-2-args =
   {\slices-start-arc = { - (#1) } {#2} ,
    \slices-end-arc = {#1} {#2}
},
slices-arc-inner-end /. is-choice ,
slices-arc-inner-end / false /. style = {} ,
slices-arc-inner-end / false /. value-forbidden ,
slices-arc-inner-end / true /. style =
   {\slices-arc-match = { inner } { 1 } { -1 } { 1 } { inner-end } { inner-start } { outer-end } } ,
slices-arc-inner-end / true /. value-forbidden ,
slices-arc-inner-end / . initial = false ,
slices-arc-inner-end / . default = true ,
slices-arc-inner-end-start /. is-choice ,
slices-arc-inner-end-start / false /. style = {} ,
slices-arc-inner-end-start / false /. value-forbidden ,
slices-arc-inner-end-start / true /. style =
   {\slices-arc-match = { inner } { 1 } { 1 } { 1 } { inner-end } { inner-start } { outer-end } } ,
slices-arc-inner-end-start / true /. value-forbidden ,
slices-arc-inner-end-start / . initial = false ,
slices-arc-inner-end-start / . default = true ,
slices-arc-inner-end-start /. is-choice ,
slices-arc-inner-end-start / false /. style = {} ,
slices-arc-inner-end-start / false /. value-forbidden ,
slices-arc-inner-end-start / true /. style =
   {\slices-arc-match = { inner } { 1 } { -1 } { -1 } { inner-start } { outer-start } } ,
slices-arc-inner-end-start / true /. value-forbidden ,
slices-arc-inner-end-start / . initial = false ,
slices-arc-inner-end-start / . default = true ,
slices-arc-inner-end-end /. is-choice ,
slices-arc-inner-end-end / false /. style = {} ,
slices-arc-inner-start-end / false /. value-forbidden ,
slices-arc-inner-start-end / true /. style =
  { slices-arc-match = { inner } { -1 } { -1 } { -1 } { inner-start } { inner-end } { outer-start } } ,
slices-arc-inner-start-end / true /. value-forbidden ,
slices-arc-inner-start-end /. initial = false ,
slices-arc-inner-start-end /. default = true ,
slices-arc-match /. style-n-args = { 7 } {
  slices-end-arc = { (#2) * tan ( \_wheelchart_diff_atan:nnnn {#7} {#6} {#5} {#6} / 2 ) } { 0 } ,
slices-start-arc = { (#3) * tan ( \_wheelchart_diff_atan:nnnn {#7} {#6} {#5} {#6} / 2 ) } { 0 } ,
slices-#1-arc = { (#4) * tan ( \_wheelchart_diff_atan:nnnn {#5} {#7} {#6} {#7} / 2 ) } { 0 } 
},
slices-arc-outer-end /. is-choice ,
slices-arc-outer-end / false /. style = {} ,
slices-arc-outer-end / false /. value-forbidden ,
slices-arc-outer-end / true /. style =
  { slices-arc-match = { outer } { -1 } { 1 } { -1 } { outer-end } { outer-start } { inner-end } } ,
slices-arc-outer-end / true /. value-forbidden ,
slices-arc-outer-end /. initial = false ,
slices-arc-outer-end /. default = true ,
slices-arc-outer-end-start /. is-choice ,
slices-arc-outer-end-start / false /. style = {} ,
slices-arc-outer-end-start / false /. value-forbidden ,
slices-arc-outer-end-start / true /. style =
  { slices-arc-match = { outer } { -1 } { -1 } { -1 } { outer-end } { outer-start } { inner-end } } ,
slices-arc-outer-end-start / true /. value-forbidden ,
slices-arc-outer-end-start /. initial = false ,
slices-arc-outer-end-start /. default = true ,
slices-arc-outer-start /. is-choice ,
slices-arc-outer-start / false /. style = {} ,
slices-arc-outer-start / false /. value-forbidden ,
slices-arc-outer-start / true /. style =
  { slices-arc-match = { outer } { -1 } { 1 } { 1 } { outer-start } { outer-end } { inner-start } } ,
slices-arc-outer-start / true /. value-forbidden ,
slices-arc-outer-start /. initial = false ,
slices-arc-outer-start /. default = true ,
slices-arc-outer-end-end /. is-choice ,
slices-arc-outer-end-end / false /. style = {} ,
slices-arc-outer-end-end / false /. value-forbidden ,
slices-arc-outer-end-end / false /. \_value-forbidden ,
slices-end =
  {\__wheelchart_slices_arrow:nnnnn
  {#1}
  {#2}
  { outer-end }
  { inner-end }
  { -- ( g__wheelchart_slice\_WCcount _inner-end-coordinate ) }
}

slices-end-to /. style-2-args =
  {
    slices-end =
    {
      \__wheelchart_slices_to:nn {#2} {#1}
      ( g__wheelchart_slice\_WCcount _inner-end-coordinate )
    }
  }

slices-inner /. initial =
  {
    \fp_compare:nNnT { \cs:w g__wheelchart_inner_radius\_WCcount _fp\cs_end: } > { 0 }
    {
      \fp_compare:nNnT
      { \cs:w g__wheelchart_inner_gap\_WCcount _fp\cs_end: }
        <
      { \cs:w g__wheelchart_abs_half_angle_minus_new_angle\_WCcount _fp\cs_end: }
      {
        arc
        [ start-angle = \fp_use:c { g__wheelchart_slice_inner_end_angle\_WCcount _fp },
        end-angle = \fp_use:c { g__wheelchart_slice_inner_start_angle\_WCcount _fp },
        radius = \fp_use:c { g__wheelchart_inner_radius\_WCcount _fp }
        ]
      }
    }
  }

slices-inner-angle-reduce /. style =
  {
    slices-inner-end-angle-shift = { - (#1) },
  }
slices-inner-arc-tangent / true /. value-forbidden,
slices-inner-arc-tangent /. initial = false,
slices-inner-arc-tangent /. default = true,
slices-inner-arrow /. style-2-args =
  
  slices-inner =
  {
    \_wheelchart_slices_arrow:nnnnn
    {#1}
    {#2}
    { inner-end }
    { inner-start }
    { -- ( g\_wheelchart_slice\_\WCcount\_inner-start-coordinate ) }
  }
}
},
slices-inner-end-angle-shift /. initial = 0,
slices-inner-start-angle-shift /. initial = 0,
slices-inner-to /. style-2-args =
  
  slices-inner =
  {
    \_wheelchart_slices_to:nn {#2} {#1}
    ( g\_wheelchart_slice\_\WCcount\_inner-start-coordinate )
  }
}
},
slices-outer /. initial =
  
  arc
  [ 
    start-angle = \fp\_use:c { g\_wheelchart_slice_\WCcount\_outer-start-angle },
    end-angle = \fp\_use:c { g\_wheelchart_slice_\WCcount\_outer-end-angle },
    radius = \fp\_use:c { g\_wheelchart_\WCcount\_outer-radius }
  ]
}
},
slices-outer-angle-reduce /. style =
  
  slices-outer-end-angle-shift = { - (#1) },
slices-outer-start-angle-shift = {#1}
slices-outer-angle-shift /. style =
{
  slices-outer-end-angle-shift = {#1} ,
  slices-outer-start-angle-shift = {#1}
},
slices-outer-arc /. style-2-args =
{
  slices-outer =
  {
    \__wheelchart_slices_arc:nnnnnn
    {#1}
    {#2}
    { outer-start }
    { outer-end }
    { -- ( g__wheelchart_slice_WCcount _outer-end-coordinate ) }
  }
}
slices-outer-arc-tangent /. is-choice ,
slices-outer-arc-tangent / false /. code = {} ,
slices-outer-arc-tangent / false /. value-forbidden ,
slices-outer-arc-tangent / true /. style =
{
  slices-outer =
  {
    \__wheelchart_slices_arc:nnnnnn
    \__wheelchart_diff_atan:nnnn { outer-start } { inner-start } { outer-end } { inner-end }
    { 0 }
    { outer-start }
    { outer-end }
    { -- ( g__wheelchart_slice_WCcount _outer-end-coordinate ) }
  }
  \fp_compare:nNnTF { \__wheelchart_slices_arc_A_fp } > { 359.99 }
  { \fp_set:Nn \__wheelchart_slices_arc_A_fp { 1 } }
  \fp_compare:nNnTF { \__wheelchart_slices_arc_A_fp } = { 180 }
  { \fp_set:Nn \__wheelchart_slices_arc_A_fp { 0 } }
  \fp_set:Nn \__wheelchart_slices_arc_A_fp { \cotd ( 45 - \__wheelchart_slices_arc_A_fp / 4 ) }
}
slices-outer-arc-tangent/.initial = false,
slices-outer-arc-tangent/.default = true,
slices-outer-arrow/.style-2-args =
{
slices-outer =
{
  \__wheelchart_slices_arrow:nnnnn
  {#1}
  {#2}
  { outer-start }
  { outer-end }
  { -- ( g__wheelchart_slice\_WCcount _outer-end-coordinate ) }
}
},
slices-outer-end-angle-shift/.initial = 0,
slices-outer-start-angle-shift/.initial = 0,
slices-outer-to/.style-2-args =
{
slices-outer =
{
  \__wheelchart_slices_to:nn {#1} {#2}
  ( g__wheelchart_slice\_WCcount _outer-end-coordinate )
}
},
slices-pos/.initial = 0.5,
slices-scope/.style = { slices_scope/.style = {#1} },
slices-scope/.style = {},
slices-scope/.style = {},
slices-start/.initial = 0,
slices-start-arc/.style-2-args =
{
slices-start =
{
  \__wheelchart_slices_arc:nnnnnn
}
\begin{verbatim}
{#1}
{#2}
{ inner-start }
{ outer-start }
{ -- cycle }
{}
},
slices-start-arrow/.style-2-args =
{
slices-start =
{
   \_\_wheelchart_slices_arrow:nnnnn
   {#1}
   {#2}
   { inner-start }
   { outer-start }
   { -- cycle }
}
},
slices-start-to/.style-2-args =
{
slices-start = \_\_wheelchart_slices_to:nn {#1} {#2} cycle }
},
slices-style/.style = { slices_style/.style = {#1} },
slices_style/.style = {} ,
slices-style = \WCvarB ,
slices-to/.style-2-args =
{
slices-end-to = {#1} {#2} ,
slices-start-to = { - (#1) } { - (#2) }
},
start-angle/.initial = 90 ,
start-half/.style =
{
   start-angle =
   {
      (#1) - \\
      \fp_eval:n
      { \l\_\_wheelchart_counter_or_clockwise_fp * \cs:w\ g\_\_wheelchart_value_1_fp\cs_end: * 0.5
         * ( \l\_\_wheelchart_total_angle_fp / \WCtotalnum )
   }
\end{verbatim}
\documentclass{article}
\usepackage{wheelchart}

\begin{document}
\pagecolor{red}
\section{Wheel Chart Example}

\begin{tikzpicture}
\wheelchart[
    total-angle=360,
    total-count=20,
    start-angle=90,
    title=Example Chart,
    title-style={fill=blue!20},
    title-left-style={fill=green!20},
    title-sep=0.5
]
\end{tikzpicture}

\section{Triangle Proportional Area and Height}

\begin{tikzpicture}
\triangleproportionalarea{domain=0:1}{plot={\{ (#2)*sqrt(1-(#1))*(#1)/2 \},\{-sqrt(1-(#1))*(#2)\}}}{radius=-1:1}{samples=2}{wheel-data-pos=0.5}
\end{tikzpicture}

\begin{tikzpicture}
\triangleproportionalheight{domain=0:1}{plot={\{ (#2)*(1-(#1))*(#1)/2 \},\{( (#1)-1)*(#2)\}}}{radius=-1:1}{samples=2}{wheel-data-pos=0.5}
\end{tikzpicture}

\end{document}
\begin{tikzpicture}
\def\n{1}
\wheelchart[
  value=\WClistA,
  WClistA={\n}
]{\exampleforthismanual}
\end{tikzpicture}

https://tex.stackexchange.com/questions/671298/clist-item-and-pgfmathsetmacro-causing-an-error
wheel-data-pos = 0.5
}
}

\pgfkeys
{ / wheelchart /. unknown /. code =
{ \tl_set:Ne \l__wheelchart_key_name_tl { \pgfkeyscurrentname }\t% it is necessary to define \l__wheelchart_key_name_tl %because \pgfkeyscurrentname will be overwritten by / errors / unknown-key / . expanded
\regex_match:nVTF { \{(\+\}\} } \l__wheelchart_key_name_tl
{ \tl_set:Ne \l__wheelchart_key_range_tl { \pgfkeyscurrentname }
\regex_replace_all:nN { \{(\+\}\} } {} \l__wheelchart_key_name_tl
\regex_replace_all:nN { \[\w\s\]+\{(\+\}\} } { \1 } \l__wheelchart_key_range_tl
\str_if_eq:eeTF { \l__wheelchart_key_range_tl } { list }
{ \pgfkeys { / wheelchart , WC_list = { \l__wheelchart_list_\l__wheelchart_key_name_tl } {#1} } \pgfkeys
{ / wheelchart , \l__wheelchart_key_name_tl /. expand_once = { \cs:w \l__wheelchart_list_\l__wheelchart_key_name_tl\cs_end: } }
}
\clist_gclear:N \g__wheelchart_slice_range_for_loop_clist
\foreach \l__wheelchart_slice_range_index_tl [ parse = true ] in \l__wheelchart_key_range_tl
{ \clist_gput_right:Ne \g__wheelchart_slice_range_for_loop_clist
{ \fp_eval:n { \l__wheelchart_slice_range_index_tl } }
}\clist_map_inline:Nn \g__wheelchart_slice_range_for_loop_clist
{ \clist_if_in:NnF \l__wheelchart_slice_range_local_clist {##1}
{ \clist_put_right:Nn \l__wheelchart_slice_range_local_clist {##1} }
\clist_if_exist:cF { \l__wheelchart_slice_##1_keys_clist }
{ \clist_new:c { \l__wheelchart_slice_##1_keys_clist } }
\str_if_eq:eeTF { \l__wheelchart_key_name_tl } { slice }
{ \clist_put_right:cn \l__wheelchart_slice_##1_keys_clist {#1} }\t}
A.4 Additional commands

\NewExpandableDocumentCommand \WCangle { O { \WCcount } m m m m }
{ \__wheelchart_def_angle_plot_false:nnnnn { \__wheelchart_mod:n {#1} } {#2} {#3} {#4} {#5} }

\NewExpandableDocumentCommand \WCcoordinate { O { \WCcount } m }
{ g__wheelchart_slice_\__wheelchart_mod:n {#1}_#2_coordinate }

\NewExpandableDocumentCommand \WPoint { O { \WCcount } m m m m }

\NewExpandableDocumentCommand \WRadius { O { \WCcount } m m }
{ \__wheelchart_def_radius:nnn { \__wheelchart_mod:n {#1} } {#2} {#3} }

A.5 The command \wheelchart

\NewDocumentCommand \wheelchart { O {} m }
{ %note the double braces {{...}} such that the contents is in a group and such that & can be used in pgfmath in a tabular \pgfkeys { / wheelchart , #1 } \IfPackageLoadedTF { siunitx } }
\__wheelchart_gdef_count_fp:nn \{ data_angle_pos \} \{ data-angle-pos \}
\__wheelchart_def_fp:nn \{ gap \} \{ gap \}
\__wheelchart_def_fp:nn \{ gap_max_angle \} \{ gap-max-angle \}
\__wheelchart_def_fp:nn \{ gap_polar \} \{ gap-polar \}
\__wheelchart_def_fp:nn \{ gap_radius \} \{ gap-radius \}
\__wheelchart_gdef_count_fp:nn \{ samples \} \{ samples \}
\__wheelchart_def_fp:nn \{ slices_inner_start_angle_shift \} \{ slices-inner-start-angle-shift \}
\__wheelchart_def_fp:nn \{ slices_inner_end_angle_shift \} \{ slices-inner-end-angle-shift \}
\__wheelchart_def_fp:nn \{ slices_outer_end_angle_shift \} \{ slices-outer-end-angle-shift \}
\__wheelchart_def_fp:nn \{ slices_outer_start_angle_shift \} \{ slices-outer-start-angle-shift \}
\__wheelchart_def_outer_radius:
\__wheelchart_def_inner_radius:
\fp_gzero_new:c \{ g__wheelchart_inner_radius_\WCcount_fp \}
\fp_gset_eq:cN \{ g__wheelchart_inner_radius_\WCcount_fp \} \l__wheelchart_inner_radius_fp
\fp_gzero_new:c \{ g__wheelchart_outer_radius_\WCcount_fp \}
\fp_gset_eq:cN \{ g__wheelchart_outer_radius_\WCcount_fp \} \l__wheelchart_outer_radius_fp
\fp_gzero_new:c \{ g__wheelchart_abs_half_angle_minus_new_angle_\WCcount_fp \}
\fp_gset:cn \{ g__wheelchart_abs_half_angle_minus_new_angle_\WCcount_fp \}
    \abs ( \g__wheelchart_angle_fp - \g__wheelchart_new_angle_fp ) / 2
\fp_set:Nn \l__wheelchart_abs_half_angle_minus_new_angle_minus_gap_polar_fp
    \cs:w g__wheelchart_abs_half_angle_minus_new_angle_\WCcount_fp \cs_end: - \l__wheelchart_gap_polar_fp
\fp_gzero_new:c \{ g__wheelchart_abs_half_angle_minus_new_angle_\WCcount_fp\cs_end: \}
\fp_gzero_new:c \{ g__wheelchart_outer_gap_\WCcount_fp \}
\fp_gzero_new:c \{ g__wheelchart_inner_gap_\WCcount_fp \}
\bool_if:NTF \l__wheelchart_plot_bool
{\fp_gset_eq:CH \{ g__wheelchart_outer_gap_\WCcount_fp \} \l__wheelchart_gap_polar_fp}
\fp_gset_eq:CH \{ g__wheelchart_inner_gap_\WCcount_fp \} \l__wheelchart_gap_polar_fp
\fp_set:Nn \l__wheelchart_gap_max_angle_def_fp
\{\cs:w g__wheelchart_inner_radius_\WCcount_fp\cs_end: > 0
\?
90
: \}
( \sind ( \l__wheelchart_abs_half_angle_minus_new_angle_minus_gap_polar_fp ) < 0.001
\?
\)
90 < \l__wheelchart_gap_max_angle_fp \&\& \l__wheelchart_gap_max_angle_fp < 180
?
\l__wheelchart_gap_max_angle_fp
:
90
):
:
\l__wheelchart_gap_max_angle_fp < 90 || \l__wheelchart_gap_max_angle_fp > 180
?
180 :
\l__wheelchart_gap_max_angle_fp
)
)
\l__wheelchart_def_gap:nn { outer } { \cs:w g__wheelchart_outer_radius_\WCcount_fp\cs_end: }
\l__wheelchart_def_gap:nn { inner } { \cs:w g__wheelchart_inner_radius_\WCcount_fp\cs_end: }
\fp_compare:nNnT { \l__wheelchart_abs_half_angle_minus_new_angle_minus_gap_polar_fp } > { 0.01 }
{
  \fp_gset:cn { g__wheelchart_inner_radius_\WCcount_fp }
  { max
    ( \l__wheelchart_gap_fp
      / sind
      { min
        ( \l__wheelchart_abs_half_angle_minus_new_angle_minus_gap_polar_fp ,
          \l__wheelchart_gap_max_angle_def_fp
        )
      ,
        \cs:w g__wheelchart_inner_radius_\WCcount_fp\cs_end:
      )
    )
  }
}
\bool_if: NF \l__wheelchart_plot_bool
\{
\__wheelchart_def_inner_radius:
\fill [ /wheelchart/middle_fill ]
\fp_compare:nNnTF \l__wheelchart_total_angle_fp = \{ 360 \}
\{
\( 0, 0 \) circle [ radius = \fp_use:N \l__wheelchart_inner_radius_fp ]
\}
\{
\( 0, 0 \)
\-- ( \fp_use:N \l__wheelchart_start_angle_fp \c_colon_str \fp_use:N \l__wheelchart_inner_radius_fp )
arc
\[
\begin{align*}
\text{start-angle} &= \fp_use:N \l__wheelchart_start_angle_fp, \\
\text{end-angle} &= \fp_eval:n \\
&\{ \\
\l__wheelchart_start_angle_fp \\
+ \l__wheelchart_counter_or_clockwise_fp \times \l__wheelchart_total_angle_fp
\}
\]
\text{radius} = \fp_use:N \l__wheelchart_inner_radius_fp
\}
\-- cycle
\}
\}
\bool_if: NT\F \l__wheelchart_discrete_bool
\{ \__wheelchart_discrete_algorithm: \}
\{ \__wheelchart_for_loop:n \}
\pgfkeysvalueof { /wheelchart/before-slices }
\bool_if: NT\F \l__wheelchart_slices_bool
\{
\__wheelchart_def_fp:nn \{ slices_angle_pos \} \{ slices-angle-pos \}
\__wheelchart_def_fp:nn \{ slices_angle_shift \} \{ slices-angle-shift \}
\__wheelchart_def_fp:nn \{ slices_pos \} \{ slices-pos \}
\__wheelchart_def_fp:nn \{ slices_sep \} \{ slices-sep \}
\__wheelchart_def_angle:nnnn
{ \l__wheelchart_slices_angle_pos_fp }
{ \l__wheelchart_slices_angle_shift_fp }
{ \l__wheelchart_slices_pos_fp }
{ \l__wheelchart_slices_sep_fp }
\fp_set:Nn \l__wheelchart_slices_angle_fp { \pgfmathresult }
\begin { scope }
[ shift /. expanded =
{ \cs:w __wheelchart_point_plot_\bool_to_str:N \l__wheelchart_plot_bool :nnnnn\cs_end: 
  \WCcount 
  \l__wheelchart_slices_angle_pos_fp 
  \l__wheelchart_slices_angle_shift_fp 
  \l__wheelchart_slices_pos_fp 
  \l__wheelchart_slices_sep_fp }
, 
rotate = \fp_use:N \l__wheelchart_slices_angle_fp 
]
\fill [ / wheelchart / slices_style ] \l__wheelchart_slices_tl
\end { scope }
\}
\begin { scope }
\}
\%
We do not use the let operation in the path \fill[/wheelchart/slices_style] ... because then
\%
\n, \p, \x and \y can not be used as macro names inside the argument of a key which is applied
\%
on this path such as the key slices inner arc.
\__wheelchart_def_coord:nnnn { inner-end } { inner } { end }
{ \fp_use:c { g__wheelchart_slice_inner_end_angle_\WCcount _fp } }
\__wheelchart_def_coord:nnnn { inner-start } { inner } { start }
{ \fp_use:c { g__wheelchart_slice_inner_start_angle_\WCcount _fp } }
\__wheelchart_def_coord:nnnn { outer-end } { outer } { end }
{ \fp_use:c { g__wheelchart_slice_outer_end_angle_\WCcount _fp } }
\__wheelchart_def_coord:nnnn { outer-start } { outer } { start }
{ \fp_use:c { g__wheelchart_slice_outer_start_angle_\WCcount _fp } }
\fill [ / wheelchart / slices_style ]
( g__wheelchart_slice_\WCcount _outer-start_coordinate )
\pgfkeysvalueof { / wheelchart / slices-outer }
\pgfkeysvalueof { / wheelchart / slices-end }
\pgfkeysvalueof { / wheelchart / slices-inner }
\bool_if:NT \l__wheelchart_contour_bool
{\!
\bool_if:NF \l__wheelchart_plot_bool
{
\__wheelchart_def_outer_radius:
\__wheelchart_def_inner_radius:
\fp_compare:nNnTF { \l__wheelchart_total_angle_fp } = { 360 }
{\!
\draw [ / wheelchart / contour_style ]
( 0 , 0 ) circle [ radius = \fp_use:N \l__wheelchart_inner_radius_fp ];
\draw [ / wheelchart / contour_style ]
( 0 , 0 ) circle [ radius = \fp_use:N \l__wheelchart_outer_radius_fp ];
}\!
\draw [ / wheelchart / contour_style ]
( \fp_use:N \l__wheelchart_start_angle_fp \cColon_str \fp_use:N \l__wheelchart_inner_radius_fp )
arc[
\!
start-angle = \fp_use:N \l__wheelchart_start_angle_fp ,
end-angle = \fp_eval:n
{\!
\l__wheelchart_start_angle_fp
+ \l__wheelchart_counter_or_clockwise_fp * \l__wheelchart_total_angle_fp
}
\!
, radius = \fp_use:N \l__wheelchart_inner_radius_fp

--
}
\fp_eval:n
{
  \l__wheelchart_start_angle_fp
  + \l__wheelchart_counter_or_clockwise_fp * \l__wheelchart_total_angle_fp
}
\c_colon_str
\fp_use:N \l__wheelchart_outer_radius_fp
)
arc
[
  start-angle =
  \fp_eval:n
  {
    \l__wheelchart_start_angle_fp
    + \l__wheelchart_counter_or_clockwise_fp * \l__wheelchart_total_angle_fp
  }
  ,
  end-angle = \fp_use:N \l__wheelchart_start_angle_fp ,
  radius = \fp_use:N \l__wheelchart_outer_radius_fp
]
-- cycle ;
}
\__wheelchart_for_loop:n
{\__wheelchart_def_fp:nn { lines } { lines }
  \__wheelchart_def_fp:nn { lines_angle_pos } { lines-angle-pos }
  \__wheelchart_def_fp:nn { lines_angle_shift } { lines-angle-shift }
  \__wheelchart_def_fp:nn { lines_ext } { lines-ext }
  \__wheelchart_def_fp:nn { lines_pos } { lines-pos }
  \__wheelchart_def_fp:nn { lines_sep } { lines-sep }
  \fp_compare:nNnF { \l__wheelchart_lines_ext_fp } = { 0 }
  {\bool_if:NF \l__wheelchart_lines_ext_dir_bool 
    {\__wheelchart_def_fp:nn { lines_ext_dirsep } { lines-ext-dirsep }
      \int_set:Nn \l__wheelchart_lines_ext_dir_int
    }
\fp_eval:n
{\fp_compare:nF{\l__wheelchart_lines_fp==0&&\l__wheelchart_lines_ext_fp==0}
{\draw[/wheelchart/lines_style]let\p{l__wheelchart_lines_1}=\cs:w__wheelchart_point_plot\bool_to_str:N\l__wheelchart_plot_bool:nnnn\cs:end:\cs:w__wheelchart_data_angle_pos_\WCcount_fp\cs:end;\cs:wnnnnnnnn\cs:end;}}
coordinate
[shift =
{
{
\fp_eval:n
{
\l__wheelchart_lines_ext_dir_int * \cs:w g__wheelchart_data_sep_\WCcount_\fp\cs_end:
}
},
0
}
]
(g__wheelchart_data_coordinate)

\__wheelchart_if_text:nn { data }
{
\fp_compare:nNnTF { \l__wheelchart_lines_ext_fp } = { 0 }
{
\__wheelchart_def_fp:nn { anchor_xsep } { anchor_xsep }
\__wheelchart_def_fp:nn { anchor_ysep } { anchor_ysep }
\pgfmathparse
{
( \WCdataangle == 0 ? "west" \c_colon_str
( \WCdataangle == 90 ? "south" \c_colon_str
( \WCdataangle == 180 ? "east" \c_colon_str
( \WCdataangle == 270 ? "north" \c_colon_str
( \WCdataangle <= \fp_use:N \l__wheelchart_anchor_ysep_fp ? "west" \c_colon_str
( \WCdataangle < 90 - \fp_use:N \l__wheelchart_anchor_xsep_fp ? "south-west" \c_colon_str
( \WCdataangle <= 90 + \fp_use:N \l__wheelchart_anchor_xsep_fp ? "south" \c_colon_str
\__wheelchart_def_fp:nn { arc\_around\_line } { arc\_around\_line }
\__wheelchart_def_fp:nn { arc\_data\_angle\_pos } { arc\_data\_angle\_pos }
\__wheelchart_def_fp:nn { arc\_data\_angle\_shift } { arc\_data\_angle\_shift }
\__wheelchart_def_fp:nn { arc\_data\_dir } { arc\_data\_dir }
% these are needed for arc data and arc
\pgfinterruptpicture%
\fp_gset:Nn \g__wheelchart\_half\_ex\_over\_one\_cm\_fp { 0.5 \, \text{ex} / 1 \, \text{cm} }
\endpgfinterruptpicture%
\fp_gset:Nn \g__wheelchart\_arc\_data\_aux\_ii\_fp { 0 }
\__wheelchart\_if\_text:nn { arc\_data } {
{ \__wheelchart\_def_fp:nn { arc\_data\_pos } { arc\_data\_pos }
\__wheelchart\_def_fp:nn { arc\_data\_sep } { arc\_data\_sep }
\__wheelchart\_def_fp:nn { arc\_data\_line\_sep\_factor } { arc\_data\_line\_sep\_factor }
\cs:w \seq_set_split:Nn\pgfkeysvalueof { / wheelchart / arc\_data\_expand } { cs_recursive:nn } \l__wheelchart\_arc\_data\_seq
\seq_map_indexed_inline:Nn \l__wheelchart\_arc\_data\_seq {
\fp_set:Nn \l__wheelchart\_arc\_data\_text\_pos\_fp {
\l__wheelchart\_arc\_data\_pos\_fp + 2 \times \g__wheelchart\_slices\_orientation\_fp
\times ( 2 \times \seq_count:N \l__wheelchart\_arc\_data\_seq - 1 ) \times \text{sign} ( \l__wheelchart\_arc\_data\_dir\_fp )
\times \l__wheelchart\_arc\_data\_line\_sep\_factor\_fp \times \g__wheelchart\_half\_ex\_over\_one\_cm\_fp
}/
\cs:w \g__wheelchart\_outer\_radius \WC\count \_fp cs_end:
- \cs:w \g__wheelchart\_inner\_radius \WC\count \_fp cs_end:
+ 2 \times \l__wheelchart\_arc\_data\_sep\_fp
}
% the sign is needed because \l__wheelchart\_arc\_data\_dir\_fp is not necessarily 1 or -1
\hbox_set:Nn \l__wheelchart\_arc\_data\_box
{ \pgfinterruptpicture { cs_recursive:nn } \endpgfinterruptpicture }
\fp_gset:Nn \g__wheelchart\_arc\_data\_slice\_length\_fp { 1 }% this is necessary if the value is 0
\bool_if:NTF \l__wheelchart\_plot\_bool
\__wheelchart_convex_comb_coord_plot:nnnnnnn
{
  overlay ,
  decorate ,
  decoration =
  {
    text-along-path ,
    text =
    {
      \fp_gset:Nn \g__wheelchart_arc_data_slice_length_fp { \pgfdecoratedpathlength } ,
      raise = -0.5 ex ,
      text-align = \l__wheelchart_arc_data_align_tl ,
      \wheelchart / arc_data_style
    }
  }
}

\__wheelchart_plot_variable_tl
0
1
\l__wheelchart_plot_variable_tl
0
\l__wheelchart_arc_data_text_pos_fp
\l__wheelchart_arc_data_sep_fp
\__wheelchart_convex_comb_coord_plot:nnnnnnn
{
  decorate ,
  decoration =
  {
    text-along-path ,
    text =
    {
      \fp_gset:Nn \g__wheelchart_arc_data_aux_i_fp { \pgfdecoratedcompleteddistance } ,
      \wheelchart / arc_data_style
    }
  }
}
\fp_compare:nNnF { \g__wheelchart_arc_data_aux_i_fp } > { 0 } { \PackageWarning { wheelchart } { The~arc~data~in~slice~\WCcount \c_space_tl did~(possibly)~not~fit.~Increase~the~absolute~value~of~arc~data~dir. } %refer to \WCcount and not to \pgfkeysvalueof { / wheelchart / arc~data } %because the latter is not necessarily unique } \int_compare:nNNT {####1} = { \fp_use:N \l__wheelchart_arc_around_line_fp } { \fp_gset_eq:NN \g__wheelchart_arc_data_aux_ii_fp \g__wheelchart_arc_data_aux_i_fp } \bool_if:NT \l__wheelchart_arc_bool { \__wheelchart_def_fp:nn { arc_pos } { arc~pos } } \__wheelchart_def_fp:nn { arc_sep } { arc~sep }
\str_case:enF { \pgfkeysvalueof { / wheelchart / arc-around-text } }
{
{ true }
{
\bool_if:NTF \l__wheelchart_plot_bool
{
\__wheelchart_arc_around_text_plot_true:nnn { first } { -1 } { 0 }
\__wheelchart_arc_around_text_plot_true:nnn { second } { 1 } { 1 }
}
{
\fp_gset:Nn \g__wheelchart_arc_data_aux_ii_fp
{
\g__wheelchart_arc_data_aux_ii_fp
/( \sqrt( \abs( \l__wheelchart_coord_determinant_fp ) ) ) \ast \deg)
}
\fp_set:Nn \l__wheelchart_arc_radius_fp
{
\__wheelchart_def_radius:nnn
{ \WCcount}
{ \l__wheelchart_arc_pos_fp}
{ \l__wheelchart_arc_sep_fp + \g__wheelchart_half_ex_over_one_cm_fp}
}
\__wheelchart_arc_around_text_plot_false:nn { first } { 0 }
\__wheelchart_arc_around_text_plot_false:nn { second } { 1 }
}
}

{ false }
{
\bool_if:NTF \l__wheelchart_plot_bool
{
\__wheelchart_convex_comb_coord_plot:nnnnnnn
{ draw, / wheelchart / arc_style }
{ 0 }
{ 1 }
{ \l__wheelchart_plot_variable_tl }
{ 0 }
{ \l__wheelchart_arc_pos_fp }
{ \l__wheelchart_arc_sep_fp }
}
\fp_set:Nn \l__wheelchart_arc_radius_fp
\__wheelchart_def_radius:nnn { \WCcount } { \l__wheelchart_arc_pos_fp } { \l__wheelchart_arc_sep_fp + \g__wheelchart_half_ex_over_one_cm_fp }
\fp_set:Nn \l__wheelchart_arc_start_angle_fp
\__wheelchart_def_angle_plot_false:nnnnn { \WCcount } { 0 } { 0 } { \l__wheelchart_arc_pos_fp } { \l__wheelchart_arc_sep_fp + \g__wheelchart_half_ex_over_one_cm_fp }
\path [ draw , / wheelchart / arc_style ]
\fp_use:N \l__wheelchart_arc_start_angle_fp \c_colon_str \fp_use:N \l__wheelchart_arc_radius_fp ) arc
[ start-angle = \fp_use:N \l__wheelchart_arc_start_angle_fp ,
end-angle = \__wheelchart_def_angle_plot_false:nnnnn { \WCcount } { 1 } { 0 } { \l__wheelchart_arc_pos_fp } { \l__wheelchart_arc_sep_fp + \g__wheelchart_half_ex_over_one_cm_fp } ,
radius = \fp_use:N \l__wheelchart_arc_radius_fp
] ;
\pgfkeys