**Best Dart package for data visualization**

Graphic : <https://pub.dev/packages/graphic/example>

flutter\_echarts : <https://pub.dev/packages/flutter_echarts>

charts\_flutter : <https://pub.dev/packages/charts_flutter>

fl\_chart : <https://pub.dev/packages/fl_chart>

syncfusion\_flutter\_charts : <https://pub.dev/packages/syncfusion_flutter_charts>

syncfusion\_flutter\_sliders : <https://pub.dev/packages/syncfusion_flutter_sliders>

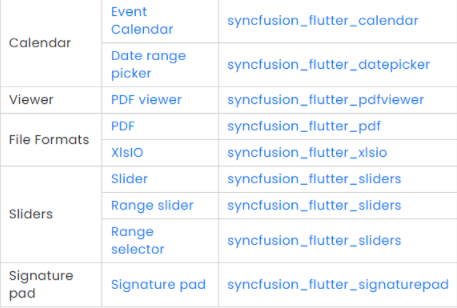
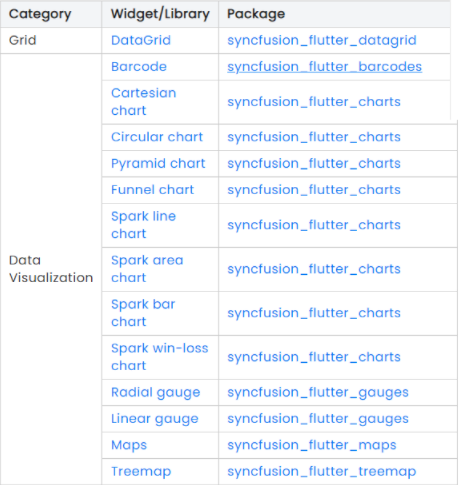
google gallery charts : <https://google.github.io/charts/flutter/gallery.html>

flutterawesome : <https://flutterawesome.com/tag/chart/>

**Syncfusion dart package**

Syncfusion offers over 1,700 components and frameworks for mobile, web, and desktop development for data visualization

**Widgets Catalog**

****

# Flutter DataGrid

The Syncfusion Flutter DataGrid is used to display and manipulate data in a tabular view. It is built from the ground up to achieve the best possible performance even when loading large amounts data.

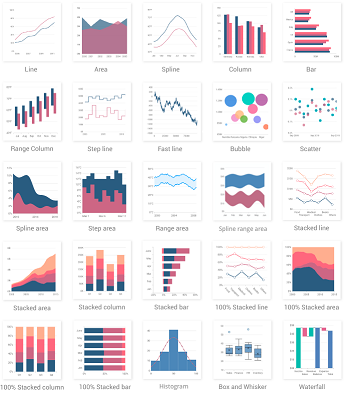
## **Key Features**

* **Column types** - Support to load any widget in a each column.
* **Column sizing** - Set the width of columns with various sizing options. Fit the columns based on the value of the cells to improve readability.
* **Row height** - Set the height for header and data rows. Fit the rows based on the value of the cells to improve readability. Also, set the different height for specific rows.
* **Editing** - Allows users to edit cell values. An editor widget can be loaded based on the column type to edit cell values.
* **Sorting** - Sort one or more columns in the ascending or descending order.
* **Selection** - Select one or more rows. Keyboard navigation is supported for web platforms.
* **Styling** - Customize the appearance of cells and headers. Conditional styling is also supported.
* **Stacked headers** - Show unbound header rows. Unbound header rows span stacked header columns across multiple rows and columns.
* **Load more** - Display an interactive view when the grid reaches its maximum offset while scrolling down. Tapping the interactive view triggers a callback to add more data from the data source of the grid at run time.
* **Paging** - Load data in segments. It is useful when loading huge amounts of data.
* **Freeze Panes** - Freeze the rows and columns when scrolling the grid.
* **Swiping** - Swipe a row right to left or left to right for custom actions such as deleting, editing, and so on. When the user swipes a row, the row will be moved and the swipe view will show the custom actions.
* **Footer** - Show an additional row that can be displayed below to last row. Widgets can also be displayed in the footer row.
* **Pull to refresh** - Allows users to refresh data when the DataGrid is pulled down.
* **Theme** - Use a dark or light theme.
* **Accessibility** - The DataGrid can easily be accessed by screen readers.
* **Right to Left (RTL)** - Right-to-left direction support for users working in RTL languages like Hebrew and Arabic.

**Charts :**

**Cartesian\_charts**

**Chart types** - Supports more than 30+ different chart types . Each type represents data in a unique style.

****

**Axis types** - Plot various types of data in a graph with the help of numeric, category, date-time, date-time category and log axis types. The built-in axis features allow to customize an axis elements further to make the axis more readable.

Charts typically have two axes that are used to measure and categorize data: a vertical (Y) axis, and a horizontal (X) axis.

Vertical(Y) axis always uses numerical scale. Horizontal(X) axis supports the following types of scale:

* Category

Category axis displays text labels instead of numbers. When the string values are bound to x values, then the x-axis must be initialized with CategoryAxis.

* Numeric

Numeric axis uses numerical scale and displays numbers as labels. By default, [NumericAxis](https://pub.dev/documentation/syncfusion_flutter_charts/latest/charts/NumericAxis-class.html" \t "_blank) is set to both horizontal axis and vertical axis.

* Date-time(auto, years months days hours minutes seconds milliseconds)

The date-time axis uses date-time scale and displays date-time values as axis labels in specified format

* Date-time category(auto, years months days hours minutes seconds milliseconds)

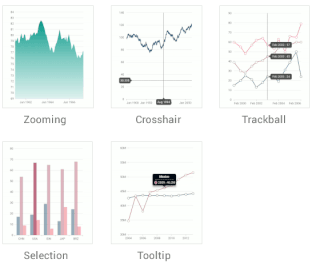
Date-time category axis is a combination of both DateTimeAxis and CategoryAxis. Date-time category axis is used to display the date-time values with non-linear intervals. For example, the business days alone have been depicted in a week here.

* Logarithmic

Logarithmic axis uses logarithmic scale and displays numbers as axis labels.

****

**User interaction** - The end-user experience is greatly enhanced by including the user interaction features such as zooming and panning, crosshair, trackball, callbacks, selection, tooltip, and auto-scrolling in chart.

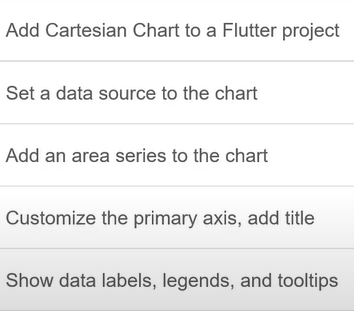
****

**Legends** - Display additional information about the chart series. The chart legend can also be used to collapse the series. The legends can be wrapped or scrolled if an item exceeds the available bounds.

Example we can add (Tunisia as name for the first line, Algeria as a name for the second line in a line chart)

**Dynamic update** - Updates the chart dynamically or lazily with live data that changes over seconds or minutes like stock prices, temperature, speed, etc.

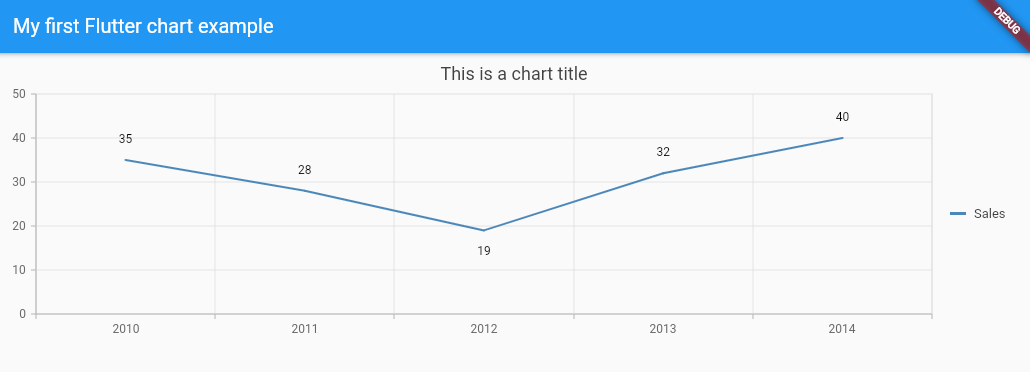
architecture of creating a chart



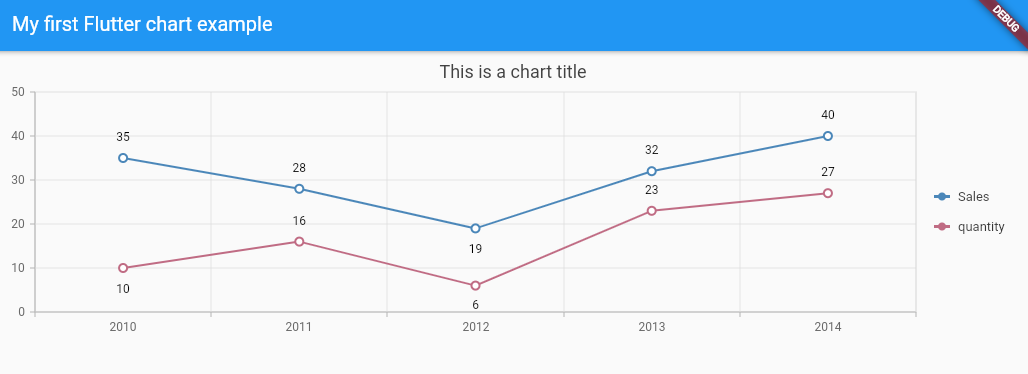
**Linechart**

Accept (int, double, num) as axis(y) and any type as axis(x)

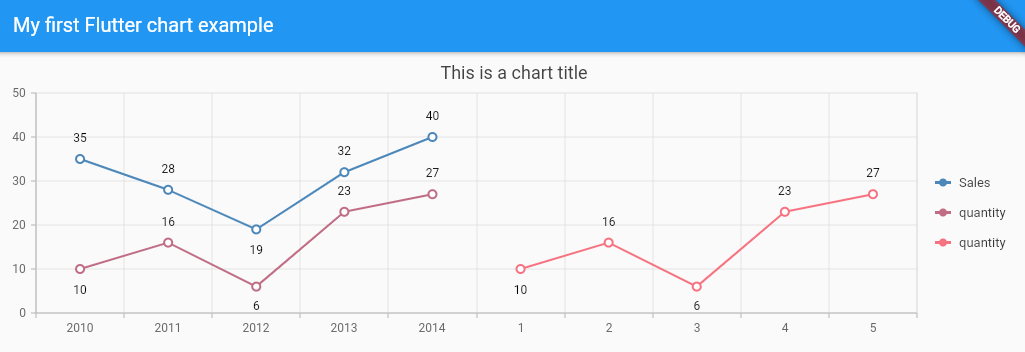
Num is an integer or floating-point number. ( It is a compile-time error for any type other than [int] or [double] to attempt to extend or implement num.)

****

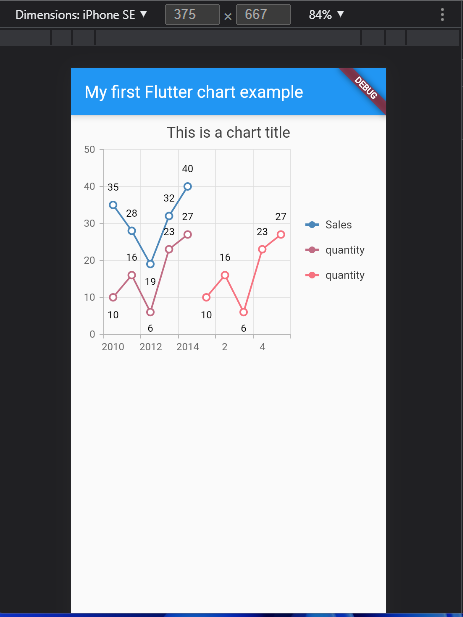
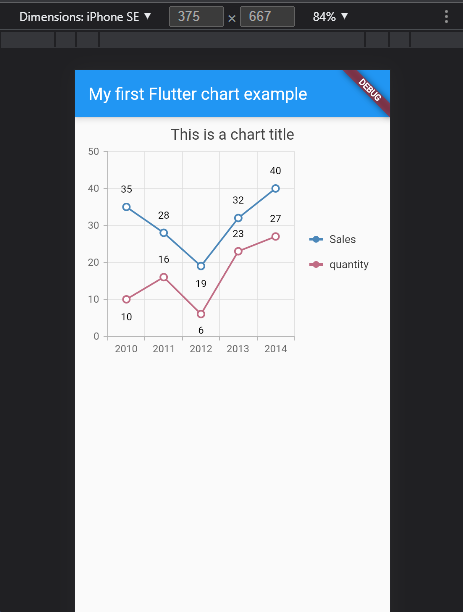
Two LineSeries with the same axis(x) and different (Y)



Two LineSeries with same (y) and different (x)

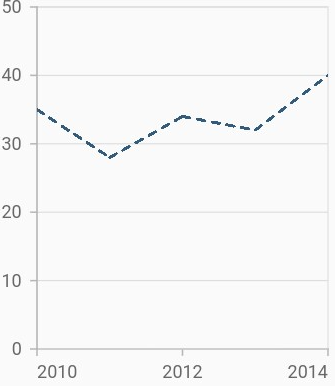


Mobile display

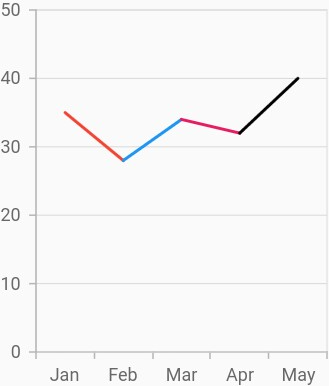


Line types

Dashed line



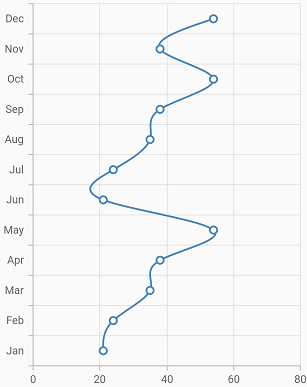
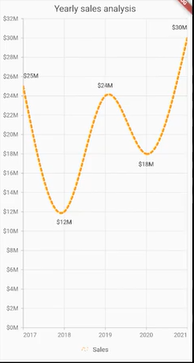
Multi-colored line



FastLineSeries

is a line chart, but it loads faster than LineSeries. You can use this when there are large number of points to be loaded in a chart.

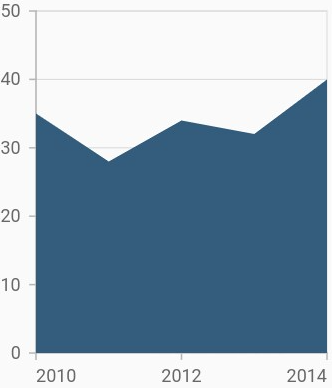
## StepLineSeries SplineLineSeries VerticalSplinechart



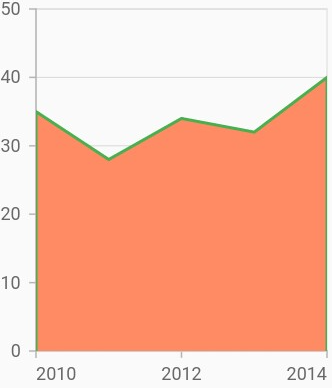
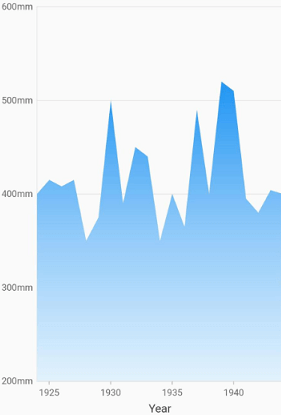
# stacked line chart 100% Stacked line Chart

# Area Chart

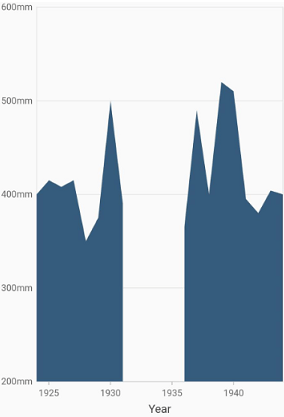
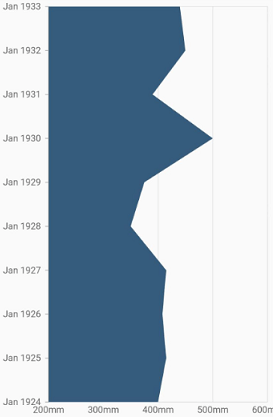
The area chart shows the filled area to represent the data, but when there are more than a series, this may hide the other series. To get rid of this, increase or decrease the transparency of the series.



## Border customization Area with gradients

## Area with empty points Vertical area chart

# Spline area Chart

# 

# Stacked area Chart

# 

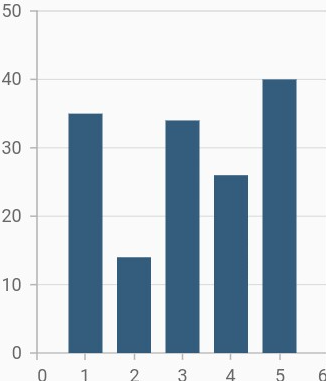
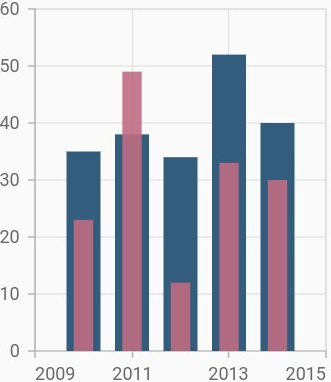
# 100% Stacked area Chart

# 

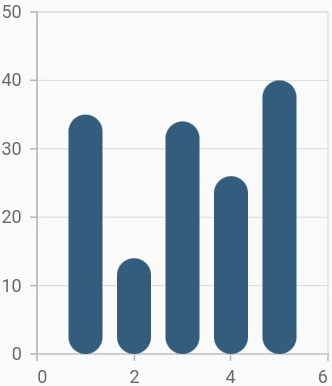
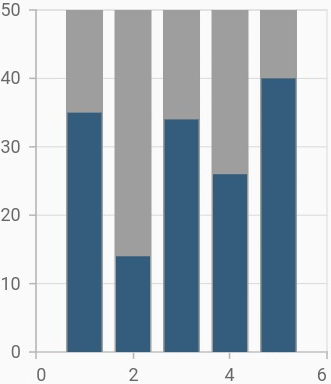
# Column Chart

By default, all the column series that have the same x and y-axes are placed side by side in a chart.

## Side-by-side overlapped

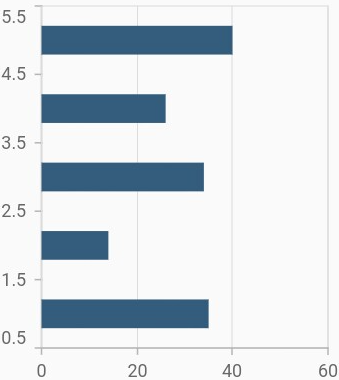
 

## Rounded corners Track customization

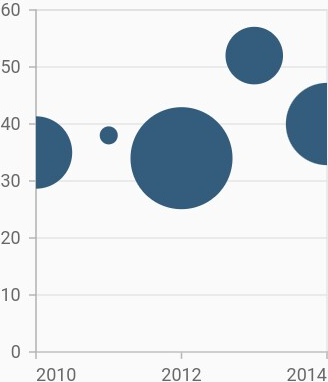
# Bar Chart

Same as Column chart but in vertical axis

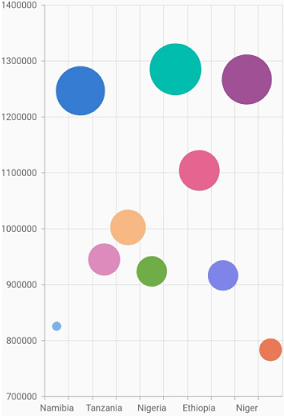
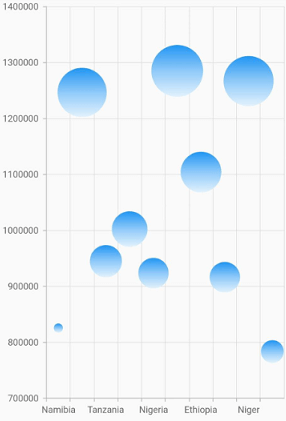


**Bubble Chart**

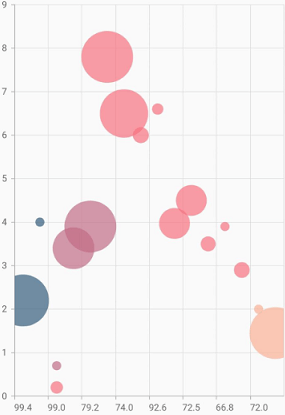
Bubble chart requires three fields (X, Y, and Size) to plot a point.



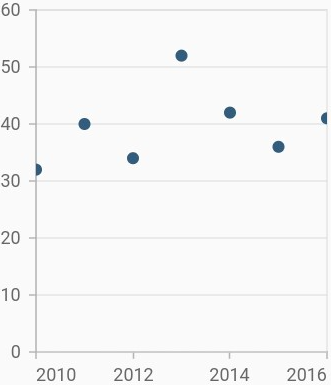
## Bubble with various color Bubble with gradients

## Bubble with multiple series

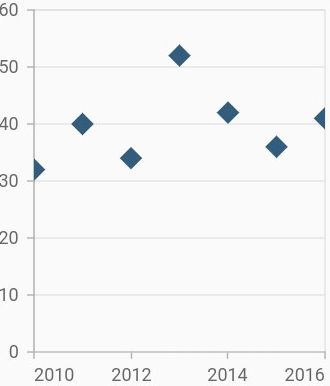


**Scatter Chart**



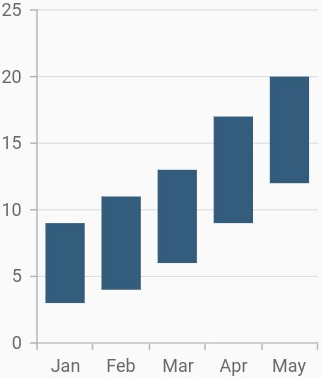
### Change shape and size of the scatter

The available shapes are circle, rectangle, pentagon, verticalLine, horizontalLine, diamond, triangle, image, and invertedTriangle. If image shape is specified, then you can assign the image using the image property.



**Range column Chart**

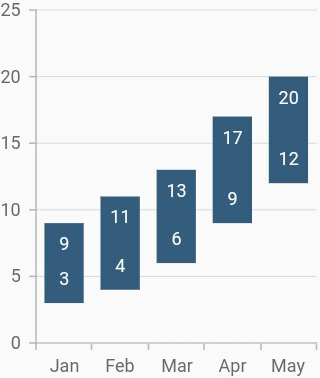
RangeColumnSeries requires two Y values for a point, your data should contain high and low values. High and low value specifies the maximum and minimum range of the point.



## Data label

In the range column chart when data label is enabled, by default there will be two values displayed namely, high and low, but in the other types of charts, only y value will be displayed.

## Data label Transposed range column



isTransposed property of CartesianSeries is used to transpose the horizontal and vertical axes, to view the data in a different perspective. Using this feature, you can render range column chart.

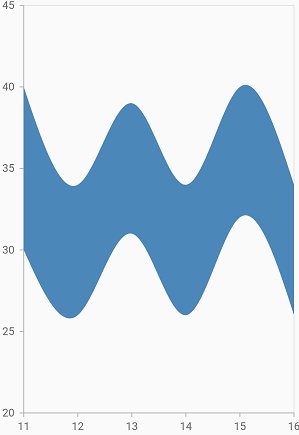
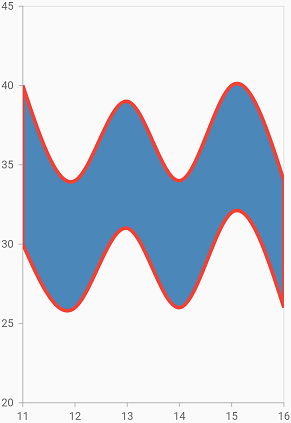
**Range area Chart**

the RangeAreaSeries requires two Y values for a point, your data should contain high and low values. High and low value specifies the maximum and minimum range of the point.



# Spline range area

requires two Y values for a point, data should contain high and low values. The high and low values specify the maximum and minimum ranges of a point.

# Step area Chart

# 

# Histogram Chart

Histogram chart is a graphical representation that organizes a group of data points into user-specified ranges.

requires only (Y) values

The histogram condenses a data series into an easily interpreted visual by taking many data points and grouping them into logical ranges.

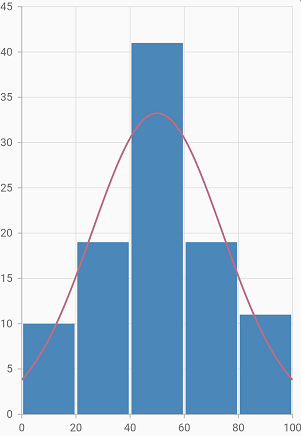
You can customize intervals using the binInterval property. Interval value by which the data points are grouped and rendered as bars, in histogram series.

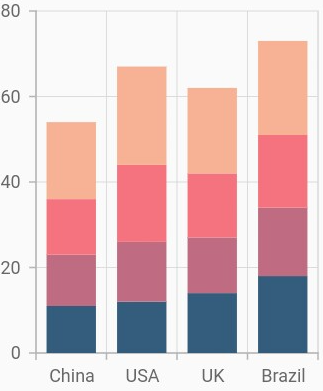
For example, if the binInterval is set to 20, the x-axis will split with 20 as the interval. The first bar in the histogram represents the count of values lying between 0 to 20 in the provided data and the second bar will represent 20 to 40.

If no value is specified for this property, then the interval will be calculated automatically based on the data points count and value.

*binInterval:* **20**,

*yValueMapper:* (ChartData sales, \_) **=>** sales.y)]))));

**  
Stacked column Chart**



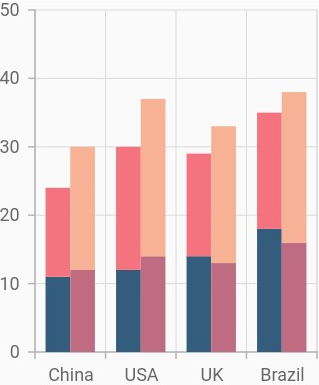
## Grouping series

You can group and stack the similar stacked series types using the groupName property of stacked series. The stacked series that contains the same groupName will be stacked in a single group.

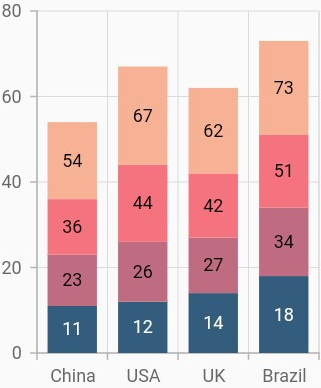
China get two bar

First one grouped by Y1 & Y3

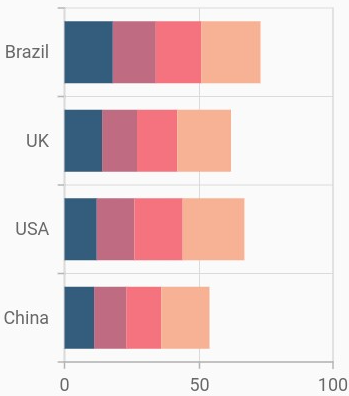
Second one grouped by Y2 & Y4



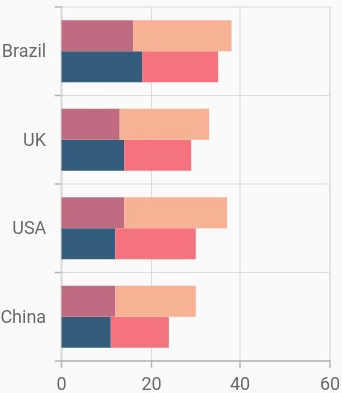
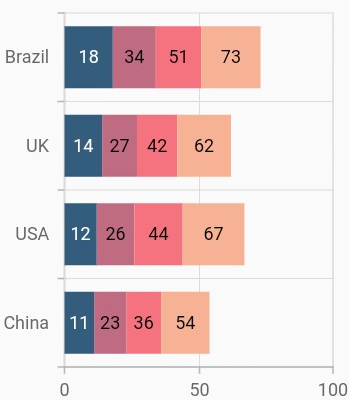
## Display cumulative values



**Stacked bar Chart**

****

## Grouping series Display cumulative values

# 100% Stacked column

# 

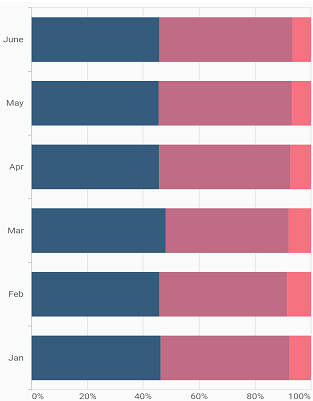
## **Column width and spacing**

The spacing property is used to change the spacing between two segments. The default value of spacing is 0, and the value ranges from 0 to 1. Here, 1 and 0 correspond to 100% and 0% of the available space, respectively.

The width property is used to change the width of the rectangle. The default value of the width is 0.7, and the value ranges from 0 to 1. Here, 1 and 0 correspond to 100% and 0% of the available width, respectively

# 

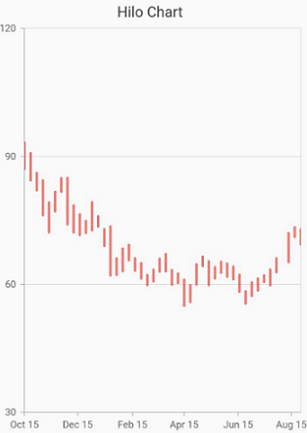
# 100% Stacked bar Chart

****

**Hilo Chart**

HiLo Series illustrates the price movements in stock using the high and low values.

Requires one X values and tow Y as lowValue and highValue

****

**OHLC Chart**

OpenHighLowClose series is used to represent the low, high, open and closing values over time. Requires X value and 4 (Y)

****

In the OHLC series, there is a feature for the datapoints indication when their high and low values are same or open and close values or high, low, open and close values are same for a datapoint

The following are the types of indication when the combination of high, low, open and close values are same for a datapoint.

* In the OHLC chart, if the open and close values are same then a horizontal line will be drawn at that value by default.
* If the high and low values are same and with showIndicationForSameValues property set to true then, a thin vertical line is drawn and if API is set to false, the line will not be drawn.

*Chart1 Open and close values are same*

*Chart2 High and low values are same*

*Chart3 High, low, open, and close values all are same*

****

**Candle Chart**

is similar to HiLo Open Close series, used to represent the low, high, open and closing price over time. Requires X value and 4 (Y)

****

*Chart1 Open and close values are same*

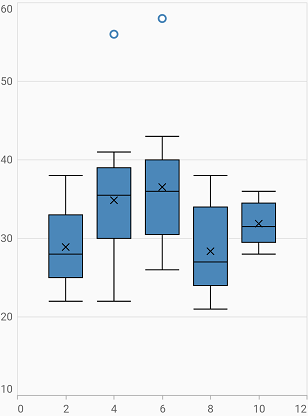
*Chart2 High and low values are same*

*Chart3 High, low, open, and close values all are same*

****

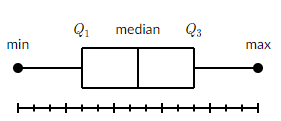
**Box and Whisker Chart**

can be used to visualize a group of numerical data through their quartiles. It is also called box plot. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles. Requires X & Y value

****

A box and whisker plot—also called a box plot—displays the five-number summary of a set of data. The five-number summary is the minimum, first quartile, median, third quartile, and maximum.

In a box plot, we draw a box from the first quartile to the third quartile. A vertical line goes through the box at the median. The whiskers go from each quartile to the minimum or maximum.



Example: Finding the five-number summary

A sample of 10 boxes of raisins has these weights (in grams):

25, 28, 29, 29, 30, 34, 35, 35, 37, 38

Make a box plot of the data.

Step 1: Order the data from smallest to largest.

Our data is already in order.

25, 28, 29, 29, 30, 34, 35, 35, 37, 38

Step 2: Find the median.

The median is the mean of the middle two numbers:

25, 28, 29, 29, {30}, {34}, 35, 35, 37, 38

{30+34}/2=32 The median is 32.

Step 3: Find the quartiles.

The first quartile is the median of the data points to the left of the median.

25, 28, {29}, 29, 30

Q\_1=29

The third quartile is the median of the data points to the right of the median.

34, 35, {35}, 37, 38

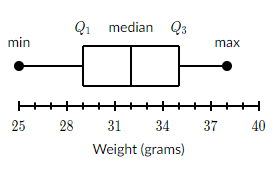
Q\_3=35

Step 4: Complete the five-number summary by finding the min and the max.

The min is the smallest data point, which is 25.

The max is the largest data point, which is 38.

The five-number summary is 25, 29, 32, 35, 38.



showMean - indication for mean value in box plot. It is set to be true, a cross symbol will be displayed at the mean value, for each data point in box plot. Else, it will not be displayed. Mean is not median

**Waterfall Chart**

The waterfall chart explains gradual changes in the quantitative value of an entity that is subject to changes by increments or decrements. Using the waterfall chart, you can quickly illustrate changes in revenues.