**What is Binning?**

Binning is a way to group a number of more or less continuous values into a smaller number of "bins". For example, if you have data about a group of people, you might want to arrange their ages into a smaller number of age intervals. Numeric columns can also be temporarily grouped by right-clicking on a column selector and clicking Auto-bin Column.

There is also an option to group categorical values into bins. This is useful when you have more categorical values in a column than you find necessary. Your visualization may for example show sales of apples, pears, oranges and limes, but you are interested in citrus fruit sales compared to

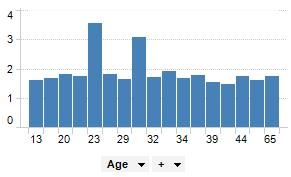
apples and pears sales. Then oranges and limes can be grouped into a bin.

**Note:** A special use case of this binning method is grouping values that are misspelt or differ due to other reasons. For example, if a column contains values like “apple” and “appel”, or “UK” and “United Kingdom”, you can group these values into bins.

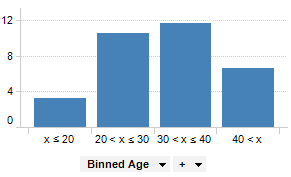
There are several different binning methods available. See [Details on Insert Binned Column](https://docs.tibco.com/pub/spotfire/7.0.1/doc/html/bin/bin_details_on_insert_binned_column.htm) for more information.

**Example of binning continuous data:**

The data table contains information about a number of persons.

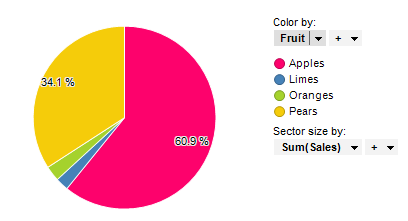


By binning the age of the people into a new column, data can be visualized for the different age groups instead of for each individual.

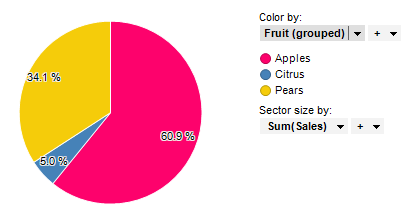


**Example of binning categorical data**

The pie chart shows sales per apples, limes, oranges and pears.



Below oranges and limes have been grouped into a bin called “Citrus”.



**Insert Binned Column and Calculated Columns**

The binning tool available under Insert > Binned Column... will create a new column in your data table, specifying the bin each row belongs to. The same results are obtained by using the [Binning Functions](https://docs.tibco.com/pub/spotfire/7.0.1/doc/html/ncfe/ncfe_binning_functions.htm) in the Insert > Calculated Column tool. Calculated bins are always static and each row belongs to the same bin at all times, independent on any filtering.

**Binning Using Custom Expressions**

When binning functions (such as BinByEvenIntervals or BinByEvenDistribution) are applied as custom expressions, the results from a binning operation is depending on whether the axis expression is evaluated on all data or on the currently filtered data only. If all data is used, you will get a result similar to the calculated bin columns above, with a specific set of bins. However, if current filtering is taken into account, then the bins are recalculated with the filtering and the bin limits will be dynamically updated.

The default setting is that all data is taken into account when the axes are evaluated, but you can change this setting in the Visualization Properties dialog, by clicking on the Settings... button for the axis of interest. See [Details on Advanced Settings](https://docs.tibco.com/pub/spotfire/7.0.1/doc/html/vis/vis_details_on_advanced_settings.htm) for more information.

# How to Use Binning

* **To use the binning tool:**

1. Select **Insert > Binned Column...**.
2. If you have more than one data table in the document, select the **Data table** to work on.
3. Select a **Column** to bin.
4. Select a **Bin method**:

**Specific Limits**

Allows you to explicitly enter values, separated by semicolons, of the limits to use for each bin.

**Even Intervals**

Allows you to specify the desired number of bins and divides the value range into equal intervals.

**Even Distribution of Unique Values**

Allows you to specify the desired number of bins and divides the bins so that each one contains an equal number of unique values.

**Based on Standard Deviation**

Allows you to divide the range into sections as described by the selected standard deviation multipliers.

**Substring**

Allows you to group the values by the first or last characters in the column to be binned.

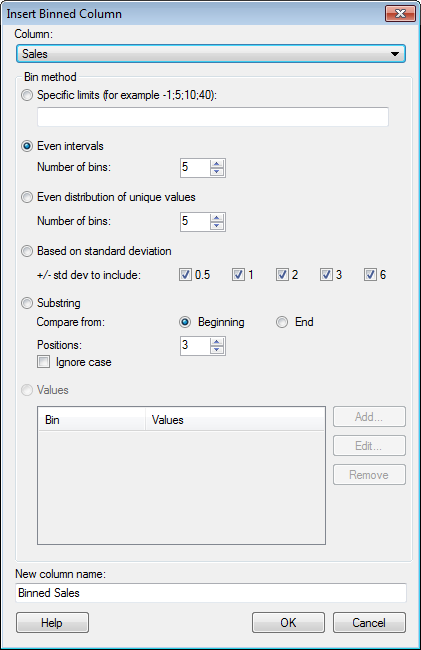
1. Type a **New column name** for the binned column.
2. Click **OK**.

**Tip**: Right-click on a column selector and select [Auto-bin Column](https://docs.tibco.com/pub/spotfire/6.5.2/doc/html/bin/bin_the_binning_slider.htm) to create a temporary, automatic binning on an axis. [Binning functions](https://docs.tibco.com/pub/spotfire/6.5.2/doc/html/ncfe/ncfe_binning_functions.htm) can also be used in custom expressions.

**Details on Insert Binned Column**

* **To reach the Insert Binned Column dialog:**

Select **Insert > Binned Column...**.

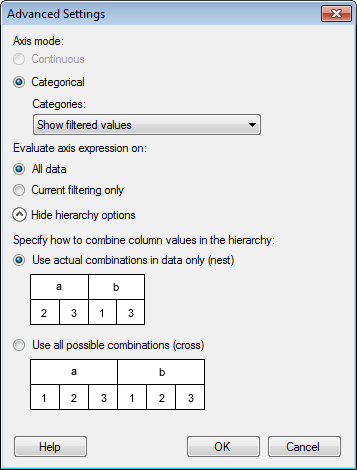


|  |  |
| --- | --- |
| **Option** | **Description** |
| **Data table** | Only available when more than one data table is present in the analysis and the dialog has been opened via the main menu.  Specifies the data table where the binned column will be inserted. |
| **Column** | Displays the available columns on which you can perform binning. It is possible to search for columns by typing in the field provided when the drop-down list is expanded. The values from the selected column will be sorted into several bins or categories based on your selections. |
| **Specific limits** | Allows you to explicitly enter the values of the limits to use for each bin.    Enter the values you wish to use for the limits of your bins and separate them with a semicolon. For example, typing "20;30;40" will result in the  following bins:  x ≤ 20  20 <  x ≤ 30  30 < x ≤ 40  40 < x |
| **Even intervals** | Allows you to specify the desired number of bins and divides the value range into equal intervals.    This method works for all data types except string. The current data range is divided up into the specified number of bins. Empty values will be empty in the new column, and when loading linked data tables, new values will be placed inside one of the available bins. |
| **Even distribution of unique values** | Allows you to specify the desired number of bins and divides the bins so that each one contains an equal number of unique values.    The suggested division works for all data types and is made so that the bins each contain an equal number of unique values. Extra values are placed in the final bins, so if you have four values and you want three bins with one value in each bin, your final value will be placed in the third bin. Empty values will be empty in the new column, and when loading linked data tables the bin ranges will be modified to fit the new data range. |
| **Based on standard deviation** | Divides the range into sections as described by the selected standard deviation multipliers.    This method works for numeric columns only. The range is divided into sections as described by the selected standard deviation multipliers. Bins are created using any of the standard deviations +/- 0.5, 1, 2, 3, 6. In the example below, the range is divided into the following six subsections (µ denoting the average value for the column and s the corresponding standard deviation):  lower limit -> (µ-3s)  (µ-3s) -> (µ-s)  (µ-s) -> µ  µ -> (µ+s)  (µ+s) -> (µ+3s)  (µ+3s) -> upper limit  Empty values will be empty in the new column, and when loading linked data tables the standard deviation will be recalculated. |
| **Substring** | Groups the rows by the first or last characters of the values in the column to be binned. The exact number of characters to take into account must be supplied.  Example:  Suppose the column to be binned contains family names, beginning with Adams and ending with Winter. To bin the rows according to the first letter in the name, use the Substring option considering one character from the beginning. Bin names are generated from the substring, and if **Ignore case** is used, the bin names are all formatted as upper case.    Empty values will be empty in the new column, and when loading linked data tables the new values will be placed in suitable bins, taking the substrings into consideration. |
| **Values** | [The method is applicable for categorical columns only.]  Allows you to group values from the selected column into bins. Each added bin is listed, and so are the included values.  To illustrate, suppose there are four values in a column; Boston, Los Angeles, Seattle and New York. You can, for example, let Boston and New York form a bin, and give the new bin value a name.  **Tip:** You can quickly create a categorical grouping by marking items from the categories of interest and selecting [Group from Marked Categories](https://docs.tibco.com/pub/spotfire/7.0.1/doc/html/bin/bin_details_on_group_from_marked_categories.htm) from the pop-up menu. |
| **Add...** | Opens the Add Bin dialog, where you specify the column values to bin. |
| **Edit...** | Opens the Edit Bin dialog, where you can make changes to an already added bin. |
| **Remove** | Removes the selected bin from the list. |
| **New column name** | The name of the new, binned column. |

**Details on Advanced Settings**

* **To reach the Advanced Settings dialog:**

1. In the Visualization Properties dialog, go to the page of interest (for example, X-axis, Y-axis, Trellis...).
2. Click on the **Settings...** button next to the column selector of the property of interest.



|  |  |
| --- | --- |
| **Option** | **Description** |
| **Axis mode** | Specifies whether the column or hierarchy should be treated as continuous or categorical. |
| **Continuous** | Select this option for numerical columns when you want the values to be treated as continuous, numerical values.  This option is not available for properties that are always categorical (like string columns). |
| **Categorical** | Select this option when you want the values to be seen as categories rather than numerical values.  For example, if you have the years 2001, 2002 and 2003 in a column, you would probably like to treat the different values as categories rather than numbers. |
| **Categories** | Determines how to display the categories.  **Note:** [Subsets](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/subsets/subsets_what_are_subsets.htm) work as any other category, and you can use the settings below to specify whether or not [empty subsets](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/subsets/subsets_what_are_subsets.htm#Empty_Subsets) should be displayed in a visualization. |
| **Show filtered values** | Select this option to display values available after current filtering only. This means that those categories that are visible will change with the filtering and that categories currently without any values will be hidden from the visualization. |
| **Show filtered range** | Select this option to hide empty categories on both ends of the currently visible range. This means that those categories that are visible will change with the filtering, and that empty categories within the currently visible range will remain visible. |
| **Show all values** | Select this option to keep all categories visible even if data for some categories have been filtered out. |
| **Evaluate axis expression on** | This setting is only applicable when the range of the available values on the axis is determined by the expression itself, that is, for column-based expressions such as ranking or binning. (For row-based expressions there will be no difference between the options.)  In the column-based cases, you will get different results upon filtering depending on whether the expression is evaluated before or after the filtering.  For example, if you have an expression like <BinByEvenIntervals([Age],3)> on an axis where the age distribution ranges between 13 and 65 years, the unfiltered visualization shows the age values split into three bins, as seen below:  vis_axis_evaluation_example.png  If the expression is evaluated on **all data**, the resulting bin ranges will  remain the same upon filtering. However, if the evaluation is done on the **current filtering only**, the resulting bin ranges will change with the filtering.  See below for descriptions of the different options. |
| **All data** | Select this option to evaluate the axis expression before filtering is applied and use all of the data to calculate the available ranges.  In the binning example above, the distribution of the available bins is calculated on the entire data volume. Filtering will not change the available bins on the axis, only the values therein:  vis_axis_evaluation_all_data.png  If one bin becomes empty due to filtering, it may be hidden depending on the settings you have specified under Categories above. |
| **Current filtering only** | Select this option to take the current filtering (as specified on the Data page) into account when evaluating axis expressions.  Using the same binning expression as above, filtering on Age will recalculate the available bins with new bin intervals for each filtering when this option is selected:  vis_axis_evaluation_current_filtering.png  If you are using in-db dynamic sets on the axis, then you must use this option, since the filtering affects which categories should be available (the content of the selected column varies with the filtering). |
| **Show/Hide hierarchy options** | For hierarchical categories only, but not available for Date, Time or DateTime hierarchies.  Shows or hides the options used to specify how hierarchies should be built, that is, which combinations of the different categories should be available. |
| **Specify how to combine column values in the hierarchy** | Specifies how many categories will be available, e.g., on an axis or in a list of colors in the legend. |
| **Use actual combinations in data only (nest)** | Use this option to display all combinations available in the data. For more information about the nest and cross alternatives, see [General Syntax](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/ncfe/ncfe_general_syntax.htm). |
| **Use all possible combinations (cross)** | Use this option to display all possible combinations, even showing categories that are currently not included in the data. |

# Binning Functions

**Note:** If an axis using a custom expression with binning has been set up to **Evaluate axis expression on: Current filtering only**, then the binning will be recalculated with each filtering. See [Details on Advanced Settings](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/vis/vis_details_on_advanced_settings.htm) for more information.

For more information about binning in general, see [What is Binning?](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/bin/bin_what_is_binning.htm).

|  |  |
| --- | --- |
| **Function** | **Description** |
| BinByDateTime | Creates a binned column based on a natural date, datetime or time hierarchy.  The first argument is the Date, Time or DateTime column to bin. The second argument is the definition of the levels in the hierarchy. The hierarchy levels should be written in the form of a string containing the desired date parts, separated by dots, for example "Year.Quarter.Month". The third argument is the pruning level which specifies the level of the hierarchy to display.  If you write a custom expression based on a BinByDateTime expression you will see a column selector with all levels of the specified hierarchy available, but with the hierarchy slider handle positioned at the specified pruning level.    Valid arguments for *Arg2* are combinations of:  'year' or 'yy' - The year.  'quarter' or 'qq' - The quarter.  'month' or 'mm' - The month.  'day of year' or 'dy' - The day of year.  'day' or 'dd' - The day.  'week' or 'wk' - The week.  'day of week' or 'dw' - The weekday.  'hour' or 'hh' - The hour.  'minute' or 'mi' - The minute.  'second' or 'ss' - The second.  'millisecond' or 'ms' - The millisecond.    Example:  BinByDateTime([Column],"Year.Quarter.Month.Day",2)    For a date column called Order Date, the expression above would result in the column selector and hierarchy slider shown below:  bin_by_date_time_slider.png  Pruning level 0 would set the slider handle to the year position, 1 would mean the quarter, 2 the month, and 3 the day. |
| BinByEvenDistribution | Creates a binned column where each bin has the same number of unique values as the others. The last bin may have more unique values than the others. The first argument is the column to bin and the second argument is the number of bins. Invalid values will give an invalid result.    Example:  BinByEvenDistribution([Column], 5)  BinByEvenDistribution(Rank([Column])\*Count() + RowId(), 3) |
| BinByEvenIntervals | Creates a binned column where the value range is divided into equal intervals. The first argument is the column to bin and the second argument is the number of bins.    Example:  BinByEvenIntervals([Column], 5) |
| BinBySpecificLimits | Creates a binned column with specific limits for the bins. The first argument is the column to bin and the following arguments are the limits for the bins. All rows which have values larger than the largest limit will have the same bin value. Invalid values will give an invalid result.    Example:  BinBySpecificLimits([Column], 1, 2, 3, 10) |
| BinByStdDev | Creates a binned column where the values are divided into bins depending on the values distance from the mean, measured in standard deviations. The first argument is the column to bin and the following arguments are the number of standard deviations to show from the mean. The standard deviation arguments should be given in ascending order and all values should be positive.    Example:  BinByStdDev([Column], 0.5, 1)  This will create a binning for:  ≤ -1 standard deviation  -1 standard deviation  -0.5 standard deviation  0.5 standard deviation  1 standard deviation  > 1 standard deviation |
| BinBySubstring | Creates a binned column based on beginning or end of value. The first argument is the string column to bin and the following is the number of characters in the substring. If the second argument is negative the substring starts from the end of the value.    Examples:  BinBySubstring([Column], -4)  BinBySubstring(String([Integer Column]), 1) |
| FiscalBinByDateTime | Creates a binned column based on a natural date or datetime hierarchy, with the levels shifted the specified number of months.    The first argument is the Date or DateTime column to bin. The second argument is the hierarchy level definition written in the form of a string containing the desired date parts, separated by dots (e.g., "Year.Quarter.Month"). Only Year, Quarter and Month are supported. The third argument is the pruning level which specifies the level of the hierarchy to display. The fourth argument (optional) is the number of months to shift.    If no fourth argument is specified then the value of the document property FiscalYearOffset will be used.    Examples:  FiscalBinByDateTime([Date Column], "Year.Quarter.Month", 1, 2)  FiscalBinByDateTime([Date Column], "Year.Quarter.Month", 2) |