

Level of detail Expressions:

=>LoD's support aggregation at dimensionalities other than view level.

=>You can also attach one or more dimensions to any aggregate expression.

Row level : Expressions referecing unaggregated datasource columns are computed for each row.

=>calculates the ratio of profit and sales for every row in the data source, and then sums the numbers.

=>That is, the division is performed before the aggregation.

Ex: [Sales] / [Profit]

cal: profitratio : [Sales] / [Profit]

=>However, this is almost certainly not what you would have intended because summing ratios is generally not useful.

sum(profitratio)

=>Instead, you probably want to know the sum of all profits divided by the sum of all sales.

=>That formula is shown below.

Margin = SUM([Profit]) / SUM([Sales])

=>In this case, the division is performed after each measure is aggregated.

=>An aggregate calculation allows you to create formulas like this.

=>When you create an aggregate calculation, no further aggregation of the calculation is possible.

=>Therefore, the field's context menu does not offer any aggregation choices. However, you can disaggregate the field.

=>The rules that apply to aggregate calculations are:

1.For any aggregate calculation, you cannot combine an aggregated value and a disaggregated value.

For example, SUM(Price)*[Items] is not a valid expression because SUM(Price) is aggregated and Items is not.

However, SUM(Price*Items) and SUM(Price)*SUM(Items) are both valid.

2.Constant terms in an expression act as aggregated or disaggregated values as appropriate.

For example: SUM(Price*7) and SUM(Price)*7 are both valid expressions.

3.All of the functions can be evaluated on aggregated values.

However, the arguments to any given function must either all be aggregated or all disaggregated.

For example: MAX(SUM(Sales),Profit) is not a valid expression because Sales is aggregated and Profit is not.

However, MAX(SUM(Sales),SUM(Profit)) is a valid expression.

4.The result of an aggregate calculation is always a measure.

5.Like predefined aggregations, aggregate calculations are computed correctly for grand totals.

View level : Expressions referencing aggregated data source columns are computed at the dimensionality.

Ex: SUM(Sales) / SUM(Profit)

Tableau encloses it in an AGG function: AGG(SUM(Sales) / SUM(Profit))

Why you go for LODs ?

or

How to use aggregate and non-aggregate measures in calculation?

[Sales] – AVG([Sales])

Tableau displays the error message: “Cannot mix aggregate and non-aggregate arguments with this function”

The user’s intent in this case was to compare store sales for each individual store to the average of sales for all stores.

This can now be accomplished with a level of detail expression:

[Sales] - {AVG([Sales])}

Table-Scoped Level of Detail Expressions

It is possible to define a level of detail expression at the table level without using any of the scoping keywords.

For example, the following expression returns the minimum (earliest) order date for the entire table:

{MIN([Order Date])} o/p: 1/4/2011

This is equivalent to a FIXED level of detail expression with no dimension declaration:

{FIXED : MIN([Order Date])}

Drag: +year(cal) o/p: 2011 Q1 January 4

FIXED level of detail expressions:

FIXED level of detail expressions compute a value using the specified dimensions, without reference to the dimensions in the view.

The following FIXED level of detail expression computes the sum of sales per region:

`{FIXED [Region] : SUM([Sales])}`

This level of detail expression, named **[Sales by Region]**, is then placed on Text to show total sales per region:

Region	State	Sales by region
Central	Illinois	501,240
	Indiana	501,240
East	connecticut	678,781

The view level of detail is **[Region]** plus **[State]**, but because FIXED level of detail expressions do not consider the view level of detail, the calculation only uses the **[Region]** dimension, and so the values for the individual states in each region are identical.

Expression's granularity:

Coarser : `{FIXED [Region] : SUM([Sales])}`

Finer : `{FIXED [Segment], [Category] : SUM([Sales])}`

Ex 1:

Segment	Finer
Consumer	387,133.781666667
Corporate	235,382.122266667
Home Office	143,217.716166667

Ex2:

Segment	Category	Avg. finer
Consumer	Furniture	363,952.136000001
Consumer	Office supplies	391,049.312000001
Consumer	Home Appliances	406,399.897

Ex3:

Segment	Category	Shipmode	Avg. finer
Consumer	Furniture	First	363,952.136000001
Consumer	Furniture	Same Day	363,952.136000001
Consumer	Furniture	Second class	363,952.136000001
Consumer	Furniture	Standard	363,952.136000001

Note:

sales by region :{FIXED [Region] : SUM([Sales])}

drag into text mark: sum(sales by region)

=>if u change the sum into avg also the result set will not be changed(only for fixed).

=>Level of detail expressions are always automatically wrapped in an aggregate when they are added to a shelf in the view **unless they're used as dimensions**. So if you double-click on a shelf and type

{FIXED[Segment], [Category] : SUM([Sales])}

and then press Enter to commit the expression, what you now see on the shelf is

SUM({FIXED[Segment], [Category] : SUM([Sales])})

But if you double-click into the shelf to edit the expression, what you see in edit mode is the original expression.

unless they're used as dimensions: {fixed :min(order date)}

INCLUDE level of detail expressions :

INCLUDE level of detail expressions compute values using the specified dimensions in addition to whatever dimensions are in the view.

EXCLUDE level of detail expressions :

EXCLUDE level of detail expressions declare dimensions to omit from the view level of detail.

EXCLUDE level of detail expressions are useful for 'percent of total' or 'difference from overall average' scenarios. They are comparable to such features as Totals and Reference Lines.

EXCLUDE level of detail expression cannot be used in row-level expressions (where there are no dimensions to omit), but can be used to modify either a view level calculation or anything in between (that is, you can use an EXCLUDE calculation to remove dimension from some other level of detail expression).

Ex: The following level of detail expression excludes **[Region]** from a calculation of the sum of **[Sales]**:

{EXCLUDE [Region]: SUM([Sales])}

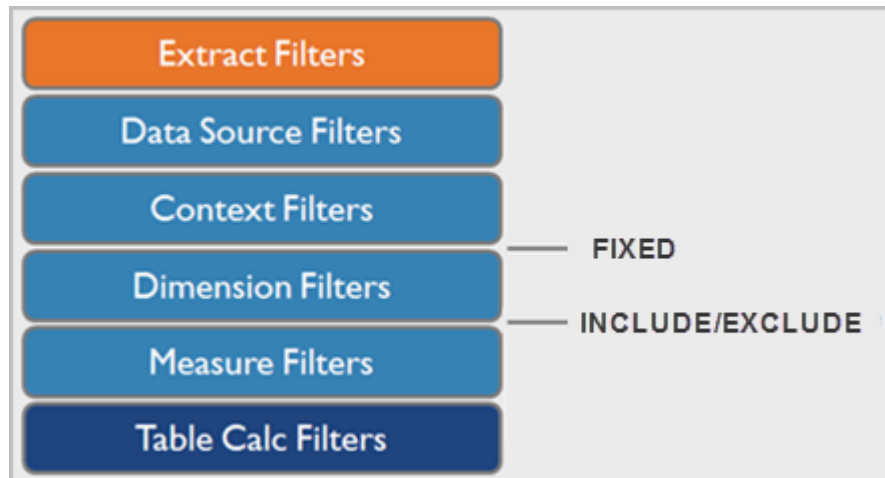
The expression is saved as **[ExcludeRegion]**.

To illustrate how this expression might be useful, first consider the following view, which breaks out the sum of sales by region and by month:

Dropping **[ExcludeRegion]** on Color shades the view to show total sales by month but without the regional component.

Filters and Level of Detail Expressions:

There are several different kinds of filters in Tableau and they get executed in the following order from top to bottom.



The text on the right shows where level of detail expressions are evaluated in this sequence.

Extract Filters (in orange) are only relevant if you're creating a Tableau Extract from a data source.

Table calculations filters (dark blue) are applied after calculations are executed and therefore hide marks without filtering out the underlying data used in the calculations.

If you're familiar with SQL, you can think of measure filters as equivalent to the HAVING clause in a query, and dimension filters as equivalent to the WHERE clause.

FIXED calculations are applied before dimension filters, so unless you promote the fields on your Filter shelf to [Improve View Performance with Context Filters](#), they will be ignored. For example, consider if you have the following calculation on one shelf in a view, along with **[State]** on a different shelf:

$$\text{SUM}([\text{Sales}]) / \text{ATTR}(\{\text{FIXED} : \text{SUM}([\text{Sales}])\})$$

This calculation will give you the ratio of a state's sales to total sales.

If you then put **[State]** on the Filters shelf to hide some of the states, the filter will affect only the numerator in the calculation. Since the denominator is a FIXED level of detail expression, it will still divide the sales for the states still in the view against the total sales for all states—including the ones that have been filtered out of the view.

INCLUDE and EXCLUDE level of detail expressions are considered after Dimension filters. So if you want filters to apply to your FIXED level of detail expression but don't want to use Context Filters, consider rewriting them as INCLUDE or EXCLUDE expressions.

Extract Your Data

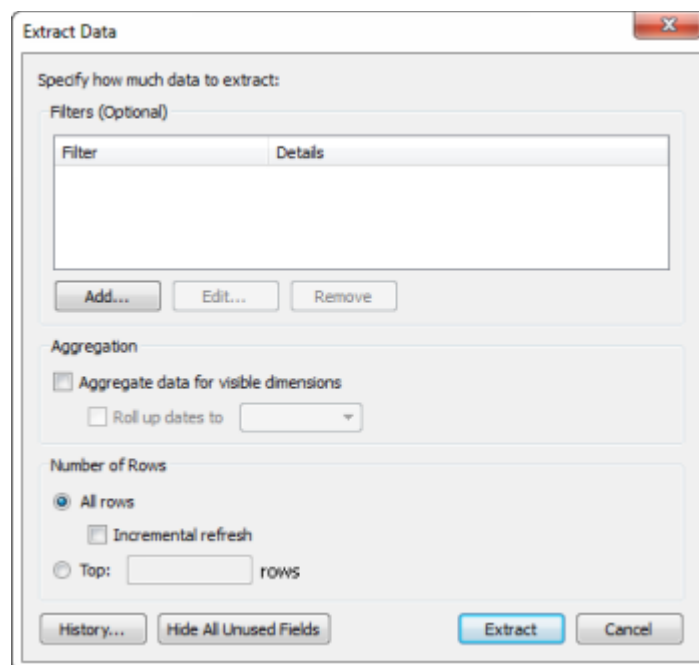
Extracts are saved subsets of a data that you can use to improve performance or to take advantage of Tableau functionality not available or supported in your original data. When you extract your data to create an extract, you can reduce the total amount of data by using filters and defining other limits. After you create an extract you can refresh it with data from the original data. When refreshing the data, you have the option to either do a full refresh, which replaces all of the extract contents, or you can an incremental refresh, which only adds rows that are new since the previous refresh.

Extracts can do the following:

- Improve performance. For file based data sources such as Excel or Access, a full extract takes advantage of the Tableau data engine. For large data sources, a filtered extract can limit the load on the server when you only need a subset of data.
- Take advantage of Tableau functionality that is not available in the original data source, such as the ability to compute Count Distinct.
- Provide offline access to your data. If your data source is not available (for example, because you are traveling), you can extract the data to a local data source.

Though there are number of points during your Tableau workflow where you can create an extract, the primary method is described below.

1.In the upper-right corner of the Data Source page, select **Extract**, and then click **Edit** to open the Extract Data dialog box.



2.(Optional) Do one or more of the following to define filters and limit the amount of data that is extracted:

- Click **Add** to define one or more filters to limit how much data gets extracted based on fields and their values.
- Select **Aggregate data for visible dimensions** to aggregate the measures using their default aggregation. Aggregating the data consolidates rows and can minimize the size of the extract file and increase performance.

When you choose to aggregate the data, you can also select **Roll up dates** to a specified date level such as Year, Month, etc.

The examples below show how the data will be extracted for each aggregation option.

Aggregation

☒ Aggregate data for visible dimensions

☒ Roll up dates to Month

Original data

	A	B	C
1	Date	Region	Sales
2	1/1/2009	South	\$500
3	1/1/2009	West	\$200
4	1/1/2009	West	\$100
5	1/1/2009	East	\$300
6	1/2/2009	South	\$600
7	1/2/2009	South	\$400
8	1/2/2009	East	\$100
9			

Each record is shown as a separate row. There are 7 rows in your data. **Aggregate data for visible dimensions**

(no roll up)

	A	B	C
1	Date	Region	Sales
2	1/1/2009	East	\$300
3	1/1/2009	South	\$500
4	1/1/2009	West	\$300
5	1/2/2009	East	\$100
6	1/2/2009	South	\$1,000
7			

Records with the same date and region have been aggregated into a single row. There are 5 rows in the extracted data source. **Aggregate data for visible dimensions**

(roll up dates to Month)

	A	B	C	D
1	Date	Region	Sales	
2	1/1/2009	East	\$400	
3	1/1/2009	South	\$1,500	
4	1/1/2009	West	\$300	
5				

• Dates have been rolled up to the Month level and records with the same region have been aggregated into a single row. There are 3 rows in the extracted data source. Select the number of rows you want to extract. You can extract **All rows** or the **Top N** rows. Tableau first applies any filters and aggregation and then extracts the number of rows from the filtered and aggregated results.

The number of rows options depend on the type of data source you are extracting from. For example, not all data sources support sampling so that option is not always available.

Note: Any fields that are hidden first in the Data pane on the sheet tab will be automatically be excluded from the extract. Click the **Hide All Unused Fields** button to remove them from the extract.

3. When finished, click **Extract**.

4. In the subsequent dialog box, select a location to save the extract, give the extract file a name, and then click **Save**.

If the Save dialog does not display, see the **Troubleshooting extracts** section, below.

Filter Data from Data Sources :

You can create filters on a data source, thereby reducing the amount of data in the data source.

If you create an extract from a data source that already has data source filters in place, those filters are automatically recommended as extract filters, and will appear in the Extract dialog. Those recommended filters are not required to be part of the Extract filter list, and can safely be removed without affecting the existing set of data source filters.

Data source filters can be useful for restricting the data users can see when you publish a workbook or data source. When you publish a data source to Tableau Server, the data source and any associated files or extracts are transported in entirety to the Server. As you publish a data source you can define access permissions for downloading or modifying the data source, and you can also choose the users and groups who can remotely issue queries through Tableau Server against that data source. When users have query permission and no download permission, you can share a rich data model having calculated fields, aliases, groups, sets and more—but only for querying.

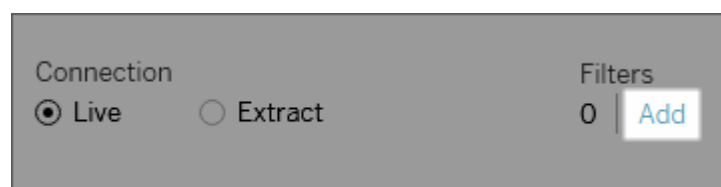
Furthermore, users who query published data source will never be able to see or modify any data source filters present on the originally published data source, but all of the users' queries will be subject to those data source filters. This is a great way to offer a restricted subset of your data, for example by filtering dimensions for specific users and groups, or by defining data source filters based on a fixed or relative date range. This is often useful for data security, and it also allows you to manage performance of the remote database which Tableau Server will ultimately query on a user's behalf. For systems that rely heavily on partitions or indexing, data source filters may yield tremendous control over the performance of queries issued by Tableau.

Create a data source filter

The primary way to create a data source filter is from the data source page.

To create a data source filter

1. On the data source page, click **Add** in the Filters section in the upper-right corner of the page.



To create a data source filter on a worksheet, right-click (control-click on a Mac) the data source and choose **Edit Data Source Filters**.

Whether you start from the Data Source page or from a worksheet, you see an Edit Data Source Filter dialog box, listing any existing data source filters.

2. Click **Add** to open an Add Filter dialog box listing all fields in the data source.

3. Click to select a field to filter; then specify how the field should be filtered, just as you would for a field on the Filters shelf.

To add an additional data source filter, repeat this procedure.

Improve View Performance with Context Filters

By default, all filters that you set in Tableau are computed independently. That is, each filter accesses all rows in your data source without regard to other filters. However, you can set one or more categorical filters as context filters for the view. You can think of a context filter as being an independent filter. Any other filters that you set are defined as dependent filters because they process only the data that passes through the context filter.

You may create a context filter to:

- Improve performance – If you set a lot of filters or have a large data source, the queries can be slow. You can set one or more context filters to improve performance.
- Create a dependent numerical or top N filter – You can set a context filter to include only the data of interest, and then set a numerical or a top N filter.

For example, suppose you're in charge of breakfast products for a large grocery chain. Your task is to find the top 10 breakfast products by profitability for all stores. If the data source is very large, you can set a context filter to include only breakfast products. Then you can create a top 10 filter by profit as a dependent filter, which would process only the data that passes through the context filter.

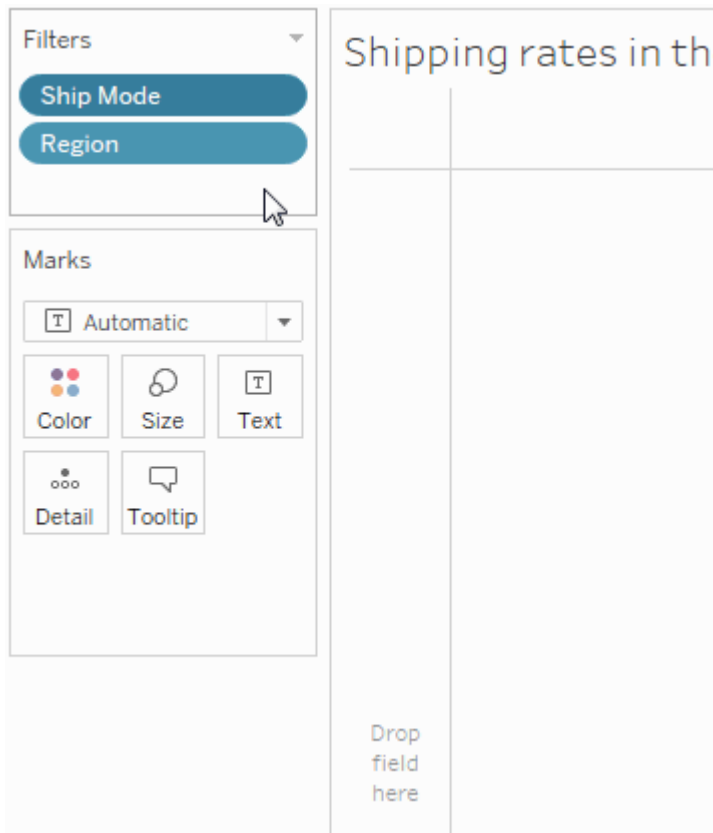
Note: As of Tableau 9.0, context filters no longer create temporary tables, except for generic ODBC data sources and customized data sources.

Create Context Filters

To create a context filter, select **Add to Context** from the context menu of an existing categorical filter. The context is computed once to generate the view. All other filters are then computed relative to the context. Context filters:

- Appear at the top of the Filters shelf.
- Are identified by a gray color on the Filters shelf.
- Cannot be rearranged on the shelf.

As shown below, the **Ship Mode** dimension is set to be the context for a view. The **Region** filter is computed using only the data that passes through **Ship Mode**.



You can modify a context filter by:

- Removing the field from the Filters shelf – If other context filters remain on the shelf, a new context is computed.
- Editing the filter – A new context is computed each time you edit a context filter.
- Selecting **Remove from Context** – The filter remains on the shelf as a standard filter. If other context filters remain on the shelf, a new context is computed.

Speed up Context Filters

To improve performance of context filters, especially on large data sources, follow these general rules.

- Using a single context filter that significantly reduces the size of the data set is much better than applying many context filters. In fact, if a filter does not reduce the size of the data set by one-tenth or more, it is actually worse to add it to the context because of the performance cost of computing the context.
- Complete all of your data modeling before creating a context. Changes in the data model, such as converting dimensions to measures, require recomputing the context.
- Set the necessary filters for the context and create the context before adding fields to other shelves. Doing this work first makes the queries that are run when you drop fields on other shelves much faster.
- If you want to set a context filter on a date you can use a continuous date. However, using date bins like YEAR(date) or context filters on discrete dates are very effective.

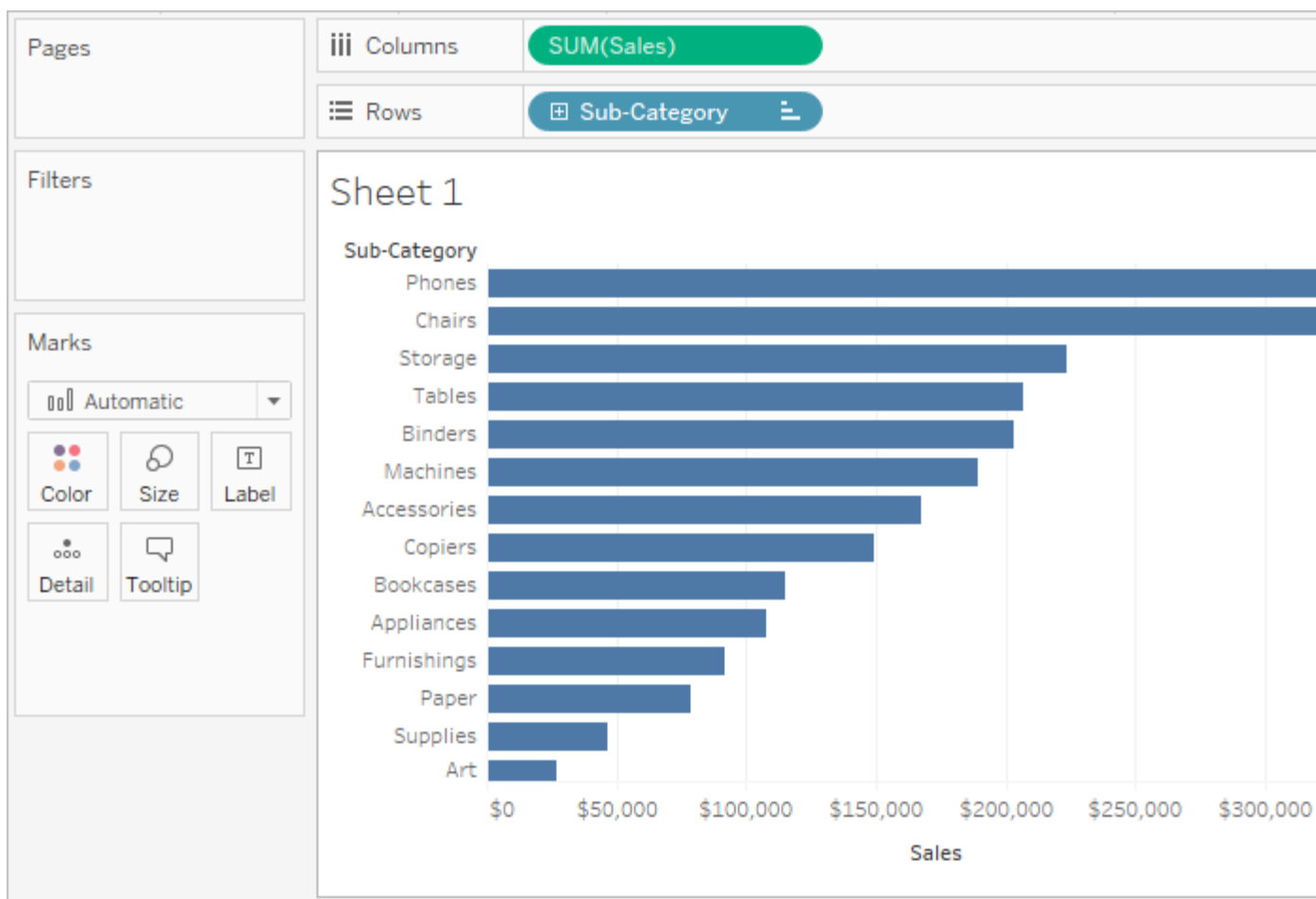
If your data set is heavily indexed, context filters may not provide performance improvement and may actually cause slower query performance.

Context filters can adversely affect any query performance improvements when using the **Include joined tables only when referenced** option in the Tables dialog box. See the note at the bottom of [Join Your Data](#).

Example – Create Context Filters

This example walks you through how to create a context filter. First you'll filter a view to show the top 10 products by sales. Then you'll create a context filter on product category so you can see the top 10 furniture products.

1. Use the **Sample - Superstore** data source to create the initial view shown below. The view shows the sales for all sub-categories, sorted with the highest sale at the top.



2. Now create a Top 10 filter to just show the top selling products. You can create this filter by dragging the **Sub-Category** field to the Filters shelf. In the Filter dialog box, switch to the **Top** tab and define a filter that is Top 10 by Sum of Sales. See [Filter Data from Your Views](#) to learn more about defining a Top N filter.

Filter [Sub-Category]

General Wildcard Condition Top

☐ None

☒ By field:

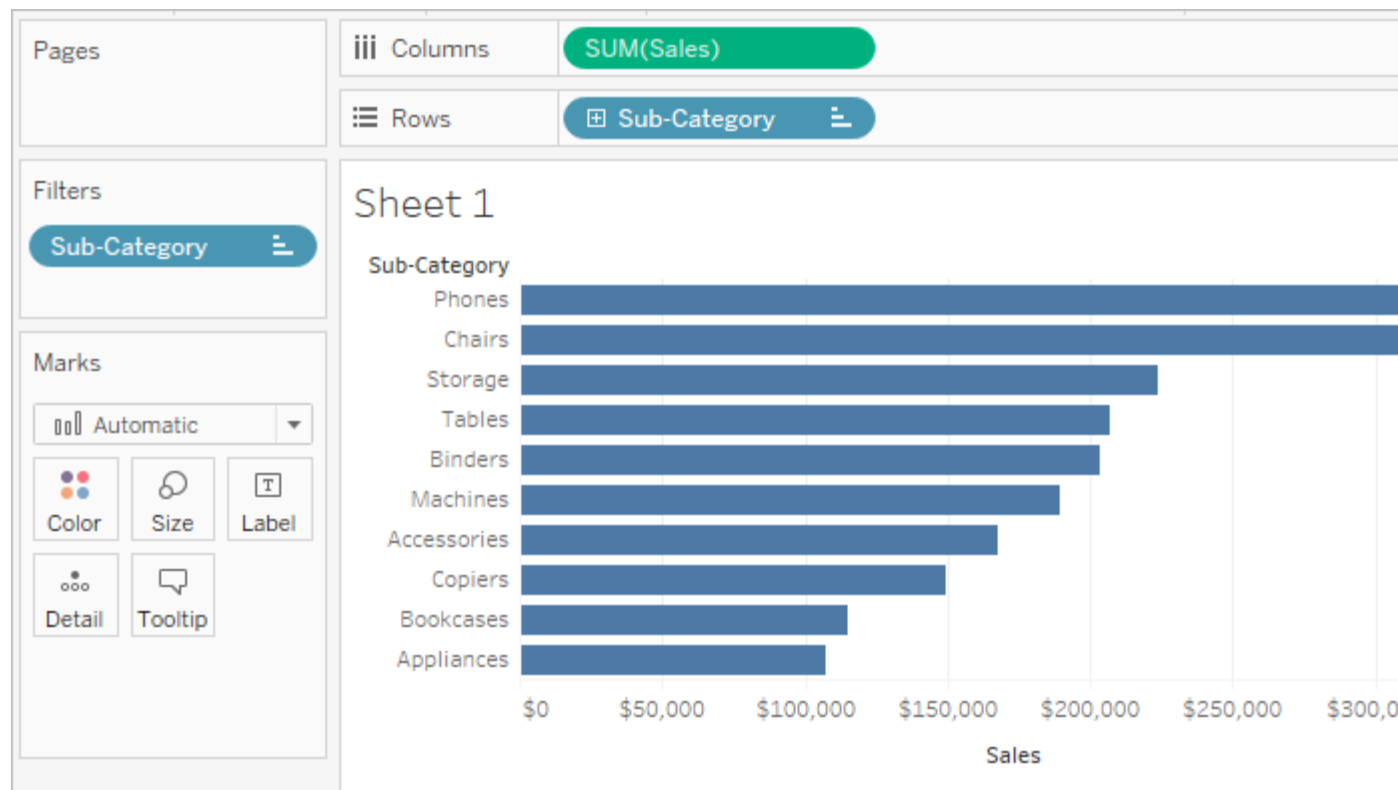
Top 10 by

Sales Sum

☐ By formula:

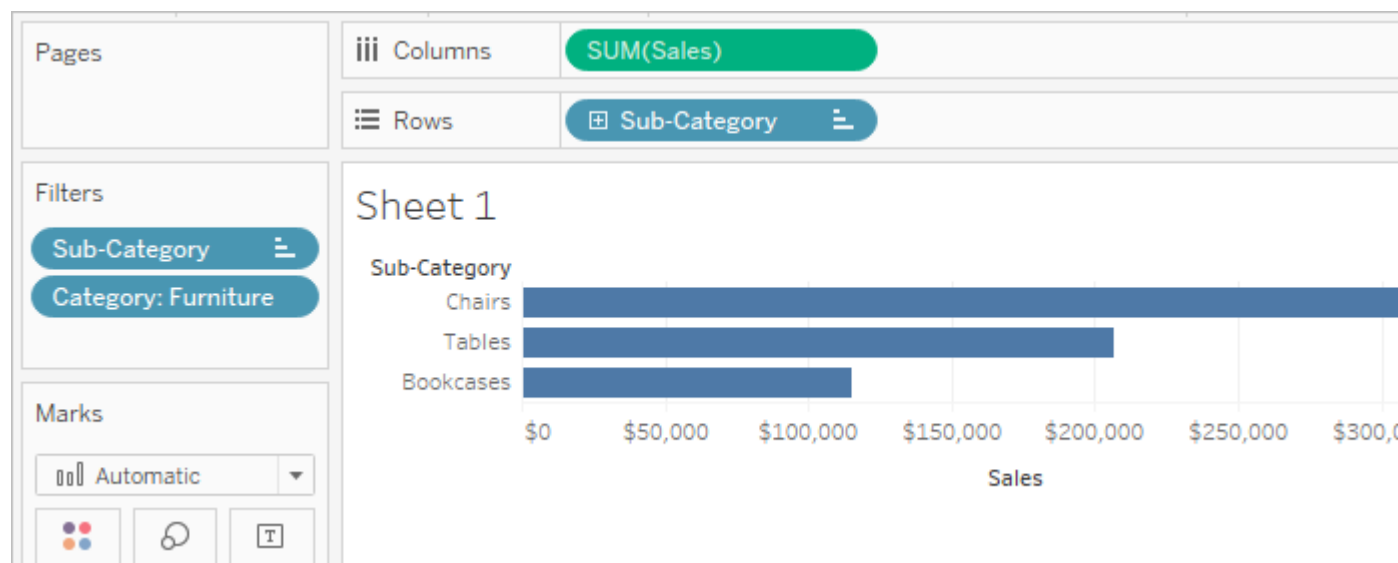
Top 10 by

3. When you click **OK**, you'll see that the view is filtered to show the top 10 product sub-categories in terms of sales.



4. Now, let's add another filter to show only furniture products. Drag the **Category** field to the Filters shelf and select only **Furniture**. When finished, click **OK**.

The view is filtered but instead of 10 products, it now shows 3. This is because by default all filters are evaluated separately and the view shows the intersection of the results. So this view shows that three of the top 10 overall products are furniture products.



5. To find out what the top 10 furniture products are we need to make the Category filter a context filter. Right-click the field on the Filters shelf and select **Add to Context**.

6. The filter is marked as a context filter and the view updates to show the top four furniture products. Why not 10? Because only four of the sub-categories contain furniture. But we now know that the Top 10 filter is being evaluated on the results of that context.

