# Use Case 3

PREREQUISITES

- This tutorial is intended for Power BI users already familiar with using Power BI Desktop to create more advanced models. You should already be familiar with using Get Data and Query Editor to import data, work with multiple related tables, and add fields to the report canvas.

- This tutorial uses the Contoso Sales Sample for Power BI Desktop file, which includes online sales data from the fictitious company, Contoso. Because this data is imported from a database, you can't connect to the datasource or view it in Query Editor. Download and extract the file on your computer.

AUTOMATIC MEASURES

When Power BI Desktop creates a measure, it's most often created for you automatically. To see how Power BI Desktop creates a measure, follow these steps:

1. In Power BI Desktop, select **File** > **Open**, browse to the *Contoso Sales Sample for Power BI Desktop.pbix* file, and then select **Open**.

2. In the **Fields** pane, expand the **Sales** table. Then, either select the check box next to the **SalesAmount** field or drag **SalesAmount** onto the report canvas. A new column chart visualization appears, showing the sum total of all values in the **SalesAmount** column of the **Sales** table.
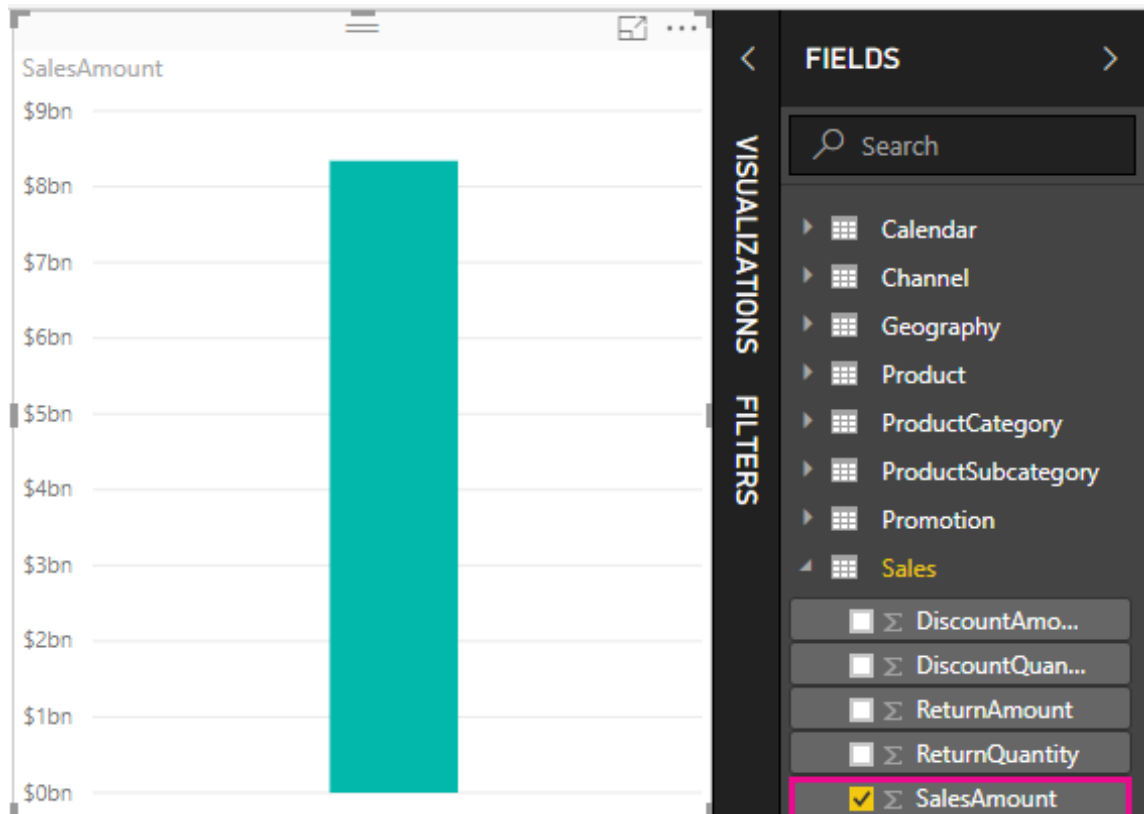
Any field (column) in the **Fields** pane with a sigma icon Σ is numeric, and its values can be aggregated. Rather than display a table with many values (two million rows for **SalesAmount**), Power BI Desktop automatically creates and calculates a measure to aggregate the data if it detects a numeric datatype. Sum is the default aggregation for a numeric datatype, but you can easily apply different aggregations like average or count. Understanding aggregations is fundamental to understanding measures, because every measure performs some type of aggregation.

To change the chart aggregation, follow these steps:

1. Select the **SalesAmount** visualization in the report canvas.
2. In the **Value** area of the **Visualizations** pane, select the down arrow to the right of **SalesAmount**.
3. From the menu that appears, select **Average**.

   The visualization changes to an average of all sales values in the **SalesAmount** field.
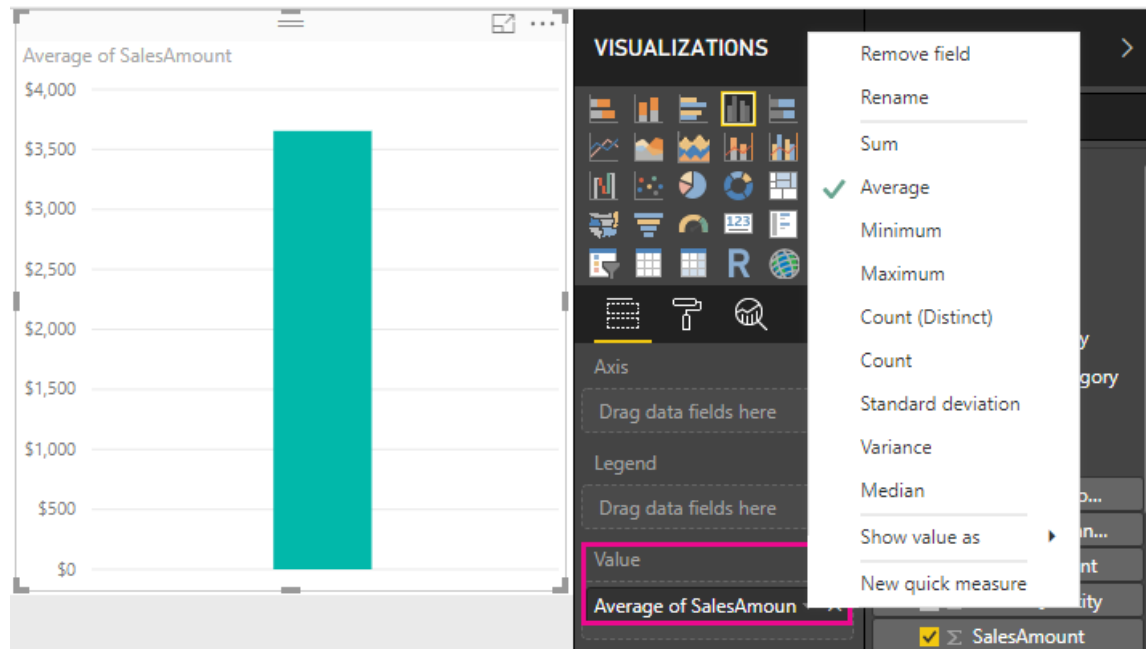
Depending on the result you want, you can change the type of aggregation. However, not all types of aggregation apply to every numeric datatype. For example, for the **SalesAmount** field, Sum and Average are useful, and Minimum and Maximum have their place as well. However, Count doesn't make sense for the **SalesAmount** field, because while its values are numeric, they're really currency.

Values calculated from measures change in response to your interactions with your report. For example, if you drag the **RegionCountryName** field from the **Geography** table onto your existing **SalesAmount** chart, it changes to show the average sales amounts for each country.
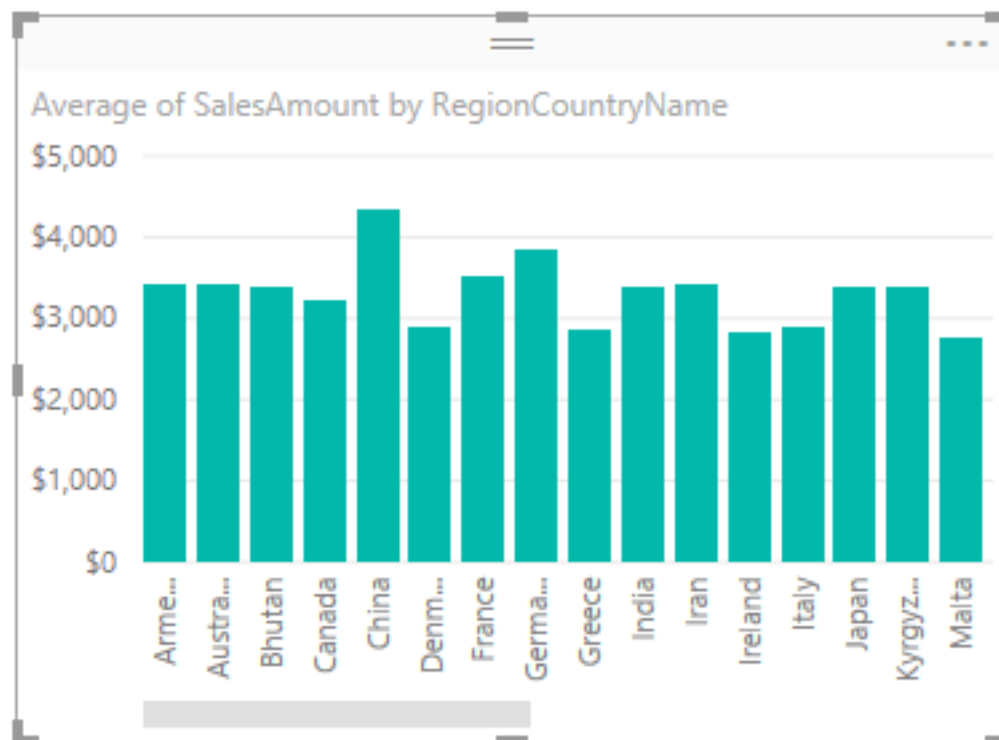
When the result of a measure changes because of an interaction with your report, you've affected your measure's *context*. Every time you interact with your report visualizations, you're changing the context in which a measure calculates and displays its results.

CREATE AND USE YOUR OWN MEASURES

In most cases, Power BI Desktop automatically calculates and returns values according to the types of fields and aggregations you choose. However, in some cases you might want to create your own measures to perform more complex, unique calculations. With Power BI Desktop, you can create your own measures with the Data Analysis Expressions (DAX) formula language.

DAX formulas use many of the same functions, operators, and syntax as Excel formulas. However, DAX functions are designed to work with relational data and perform more dynamic calculations as you interact with your reports. There are over 200 DAX functions that do everything from simple aggregations like sum and average to more complex statistical and filtering functions. There are many resources to help you learn more about DAX. After you've finished this tutorial, see DAX basics in Power BI Desktop.

When you create your own measure, it's called a *model* measure, and it's added to the **Fields** list for the table you select. Some advantages of model measures are that you can name them whatever you want, making them more identifiable; you can use them as arguments in other DAX expressions; and you can make them perform complex calculations quickly.

QUICK MEASURES

Starting with the February 2018 release of Power BI Desktop, many common calculations are available as *quick measures*, which write the DAX formulas for you

based on your inputs in a window. These quick, powerful calculations are also great for learning DAX or seeding your own customized measures.

Create a quick measure using one of these methods:

- From a table in the **Fields** pane, right-click or select **More options (...)**, and then select **New quick measure** from the list.
- Under **Calculations** in the **Home** tab of the Power BI Desktop ribbon, select **New Quick Measure**.

For more information about creating and using quick measures, see Use quick measures.

CREATE A MEASURE

Suppose you want to analyze your net sales by subtracting discounts and returns from total sales amounts. For the context that exists in your visualization, you need a measure that subtracts the sum of DiscountAmount and ReturnAmount from the sum of SalesAmount. There's no field for Net Sales in the **Fields** list, but you have the building blocks to create your own measure to calculate net sales.

To create a measure, follow these steps:

1. In the **Fields** pane, right-click the **Sales** table, or hover over the table and select **More options (...)**.
2. From the menu that appears, select **New measure**.
   This action saves your new measure in the **Sales** table, where it's easy to find.
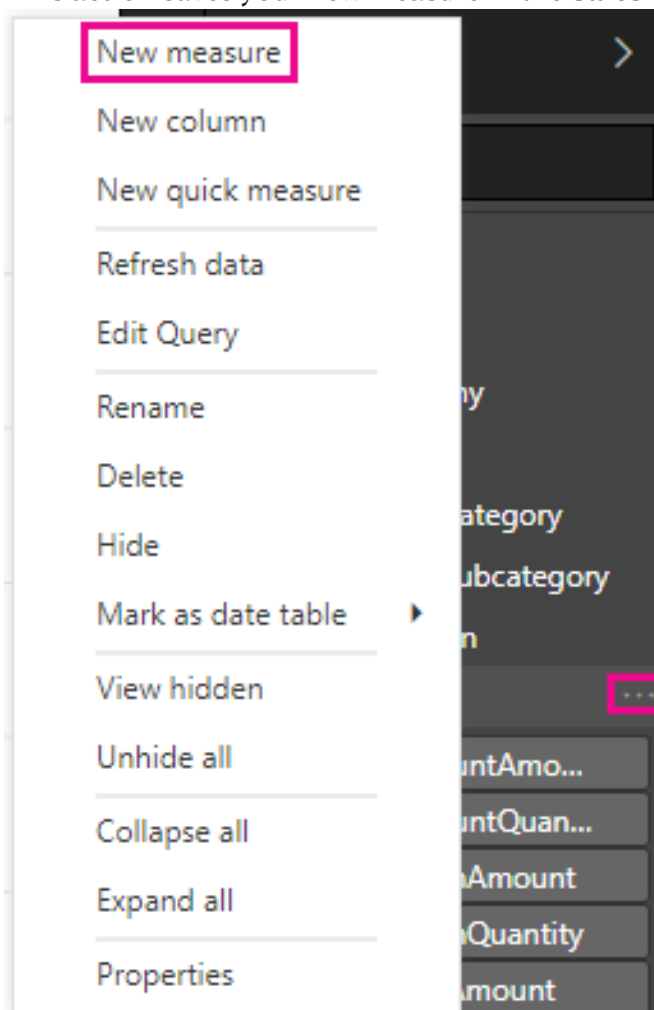
You can also create a new measure by selecting **New Measure** in the **Calculations** group on the **Home** tab of the Power BI Desktop ribbon.
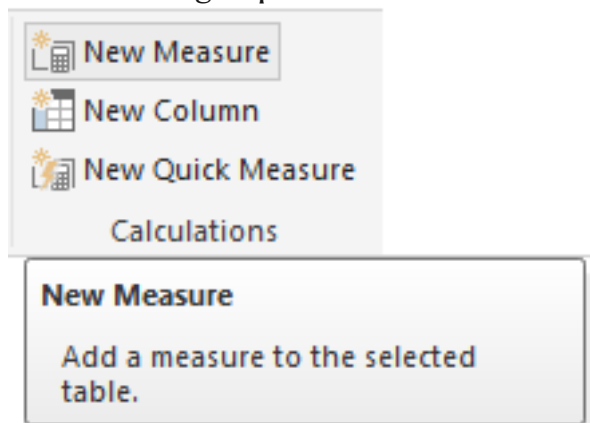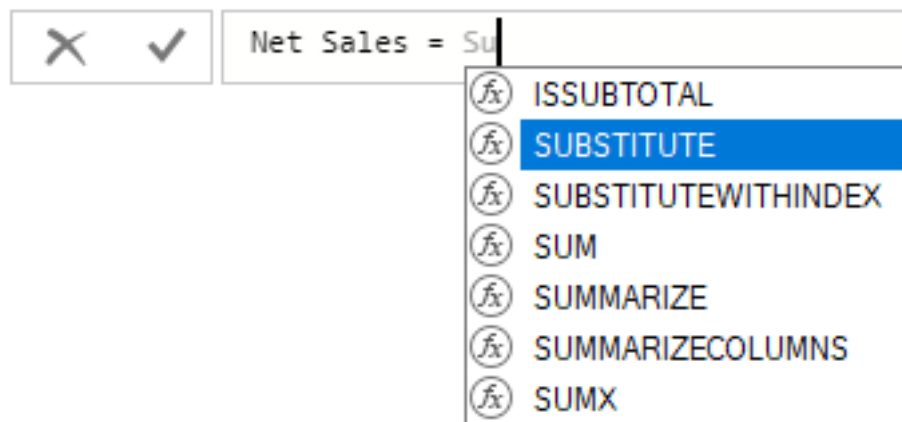
Tip
When you create a measure from the ribbon, you can create it in any of your tables, but it's easier to find if you create it where you plan to use it. In this case, select the **Sales** table first to make it active, and then select **New measure**.

The formula bar appears along the top of the report canvas, where you can rename your measure and enter a DAX formula.
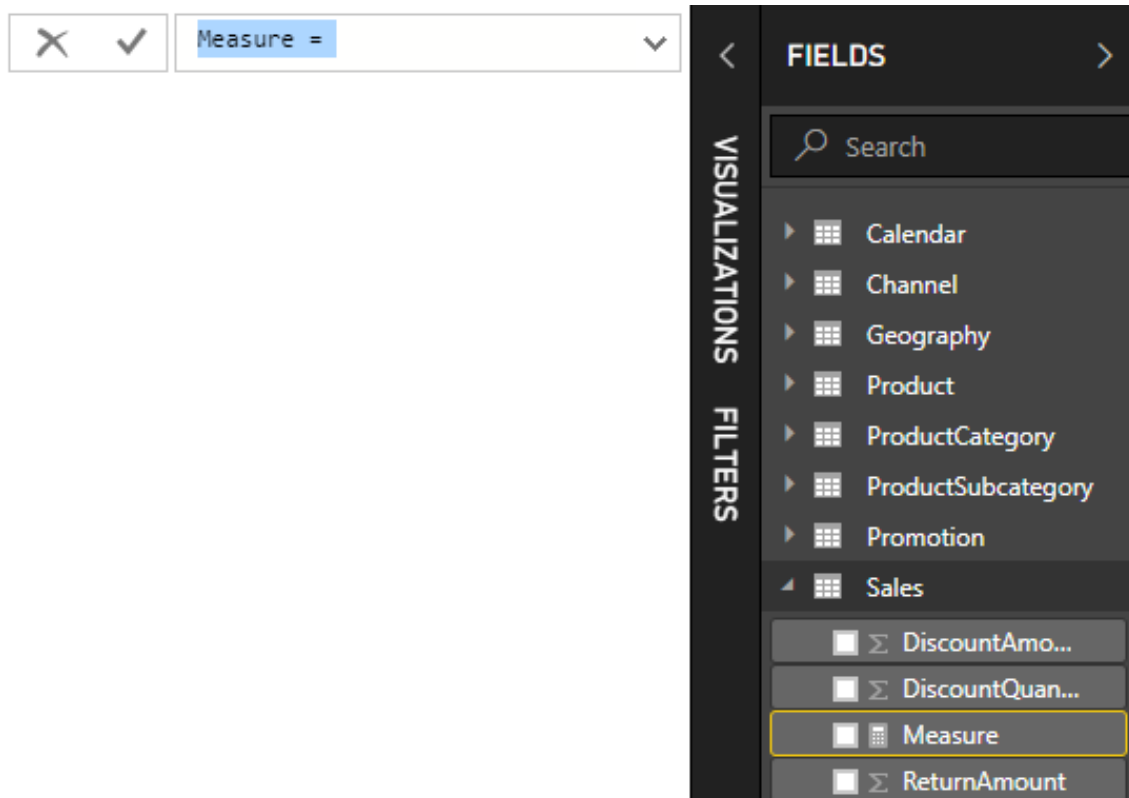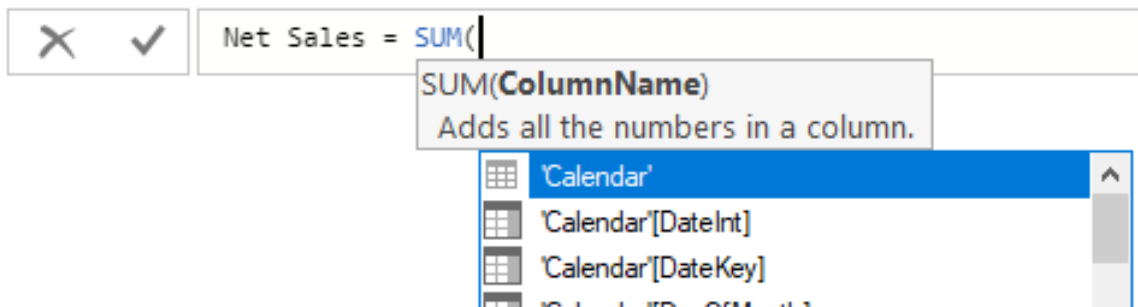

3. By default, each new measure is named *Measure*. If you don't rename it, additional new measures are named *Measure 2, Measure 3,* and so on. Because we want this measure to be more identifiable, highlight *Measure* in the formula bar, and then change it to *Net Sales*.

4. Begin entering your formula. After the equals sign, start to type *Sum*. As you type, a drop-down suggestion list appears, showing all the DAX functions, beginning with the letters you type. Scroll down, if necessary, to select **SUM** from the list, and then press **Enter**.

Net Sales = Su|

- *fx* ISSUBTOTAL
- *fx* **SUBSTITUTE**
- *fx* SUBSTITUTEWITHINDEX
- *fx* SUM
- *fx* SUMMARIZE
- *fx* SUMMARIZECOLUMNS
- *fx* SUMX

An opening parenthesis appears, along with a drop-down suggestion list of the available columns you can pass to the SUM function.

Net Sales = SUM(|

SUM(**ColumnName**)
Adds all the numbers in a column.

- 'Calendar'
- 'Calendar'[DateInt]
- 'Calendar'[DateKey]

5. Expressions always appear between opening and closing parentheses. For this example, your expression contains a single argument to pass to the SUM function: the **SalesAmount** column. Begin typing *SalesAmount* until **Sales(SalesAmount)** is the only value left in the list.

The column name preceded by the table name is called the fully qualified name of the column. Fully qualified column names make your formulas easier to read.

Net Sales = SUM(SalesA|

SUM(**ColumnName**)
Adds all the numbers in a column.
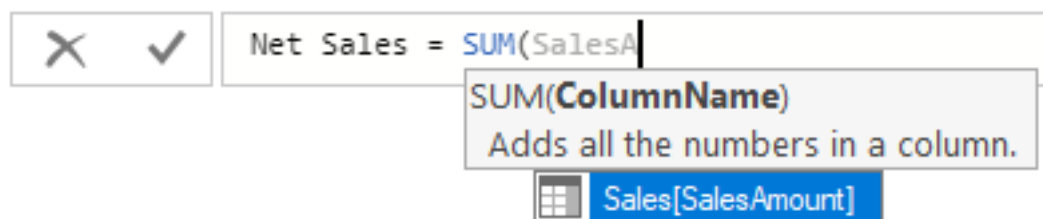
Sales[SalesAmount]

6. Select **Sales[SalesAmount]** from the list, and then enter a closing parenthesis.

Tip
Syntax errors are most often caused by a missing or misplaced closing parenthesis.

7.   Subtract the other two columns inside the formula:

a. After the closing parenthesis for the first expression, type a space, a minus operator (-), and then another space.

b. Enter another SUM function, and start typing *DiscountAmount* until you can choose the **Sales[DiscountAmount]** column as the argument. Add a closing parenthesis.

c. Type a space, a minus operator, a space, another SUM function with **Sales[ReturnAmount]** as the argument, and then a closing parenthesis.

8.   Press **Enter** or select **Commit** (checkmark icon) in the formula bar to complete and validate the formula.

The validated **Net Sales** measure is now ready to use in the **Sales** table in the **Fields** pane.
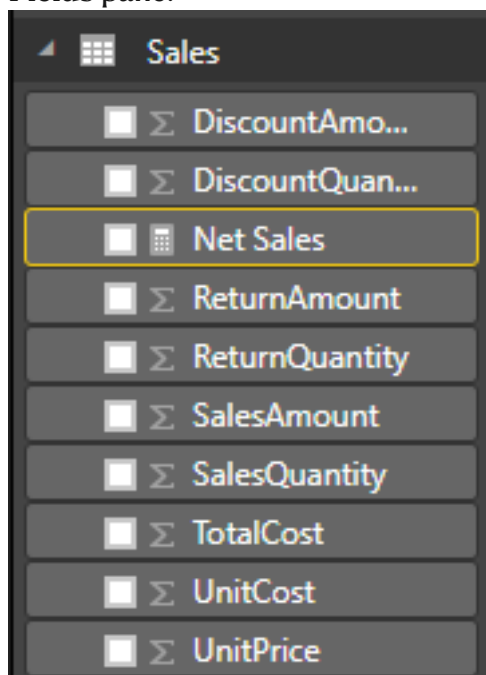
9.   If you run out of room for entering a formula or want it on separate lines, select the down arrow on the right side of the formula bar to provide more space.

The down arrow turns into an up arrow and a large box appears.

10. Separate parts of your formula by pressing **Alt** + **Enter** for separate lines, or pressing **Tab** to add tab spacing.

```
Net Sales = SUM(Sales[SalesAmount])
          - SUM(Sales[DiscountAmount])
          - SUM(Sales[ReturnAmount])
```
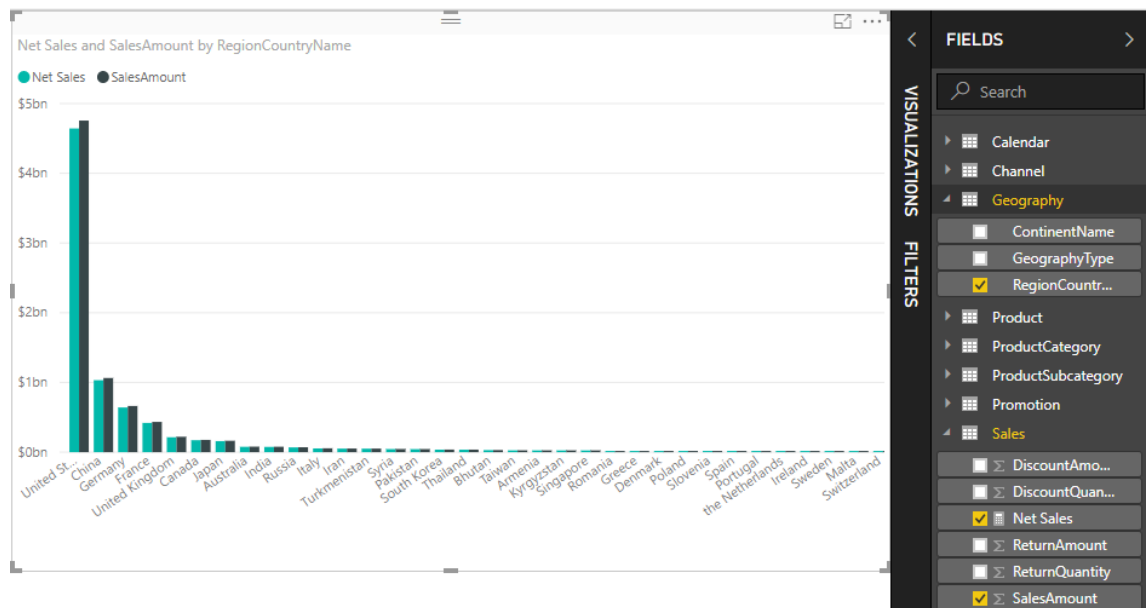
USE YOUR MEASURE IN THE REPORT

Add your new **Net Sales** measure to the report canvas, and calculate net sales for whatever other fields you add to the report.

To look at net sales by country:

1. Select the **Net Sales** measure from the **Sales** table, or drag it onto the report canvas.

2. Select the **RegionCountryName** field from the **Geography** table, or drag it onto the **Net Sales** chart.
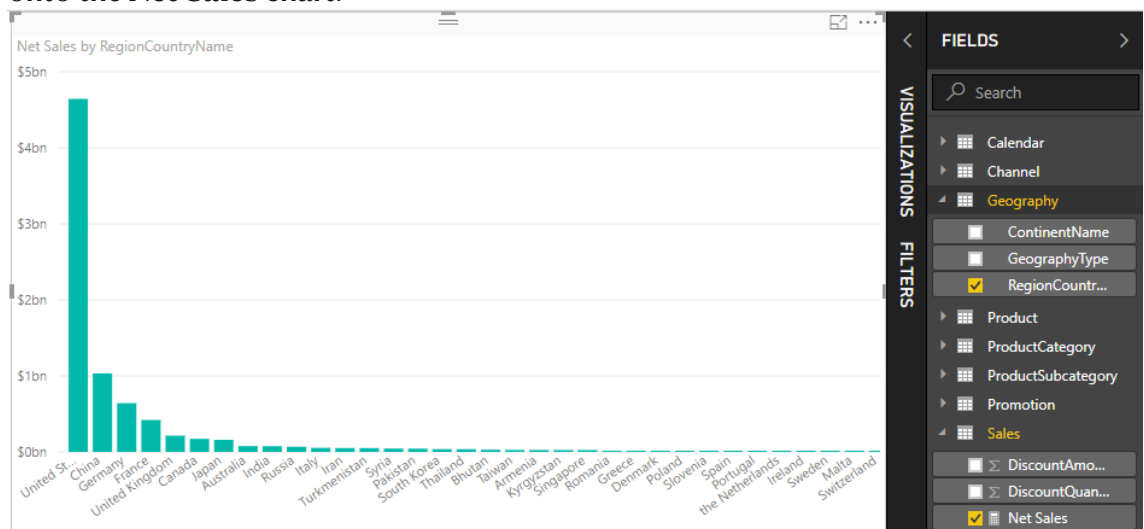
3. To see the difference between net sales and total sales by country, select the **SalesAmount** field or drag it onto the chart.

The chart now uses two measures: **SalesAmount**, which Power BI summed automatically, and the **Net Sales** measure, which you manually created. Each measure was calculated in the context of another field, **RegionCountryName**.

USE YOUR MEASURE WITH A SLICER

Add a slicer to further filter net sales and sales amounts by calendar year:

1. Select a blank area next to the chart. In the **Visualizations** pane, select the **Table** visualization.

    This action creates a blank table visualization on the report canvas.
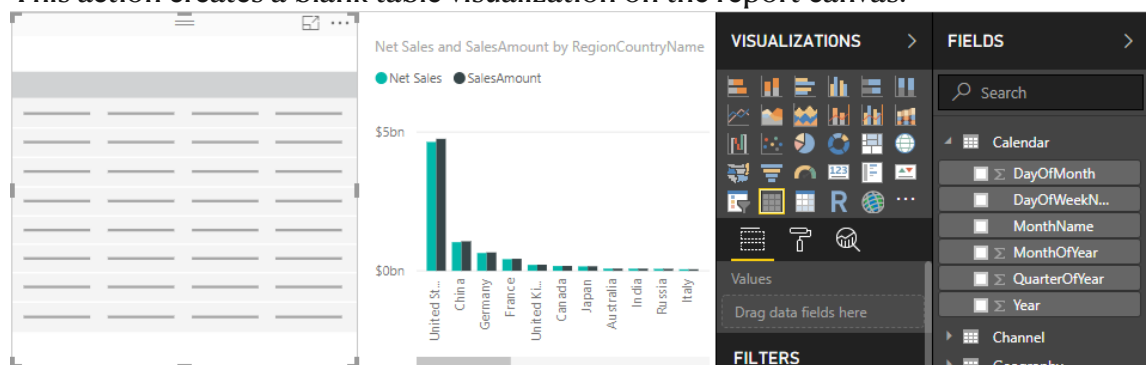
2. Drag the **Year** field from the **Calendar** table onto the new blank table visualization.

    Because **Year** is a numeric field, Power BI Desktop sums up its values. This summation doesn't work well as an aggregation
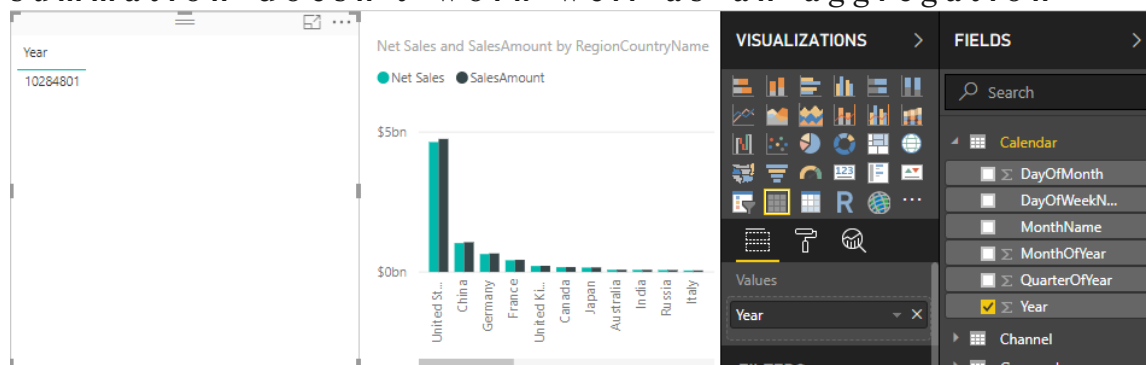
3. In the **Values** box in the **Visualizations** pane, select the down arrow next to **Year**, and then select **Don't summarize** from the list. The table now lists individual years.
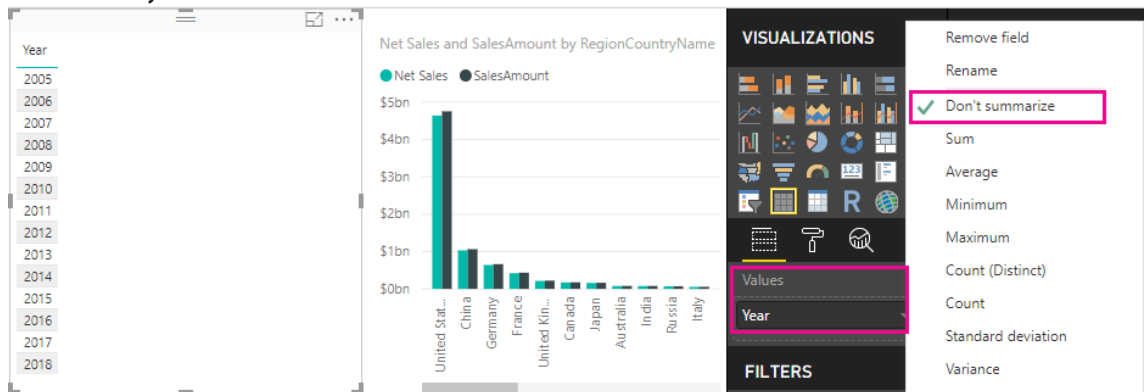
4. Select the **Slicer** icon in the **Visualizations** pane to convert the table to a slicer. If the visualization displays a slider instead of a list, select **List** from the down arrow in the slider.
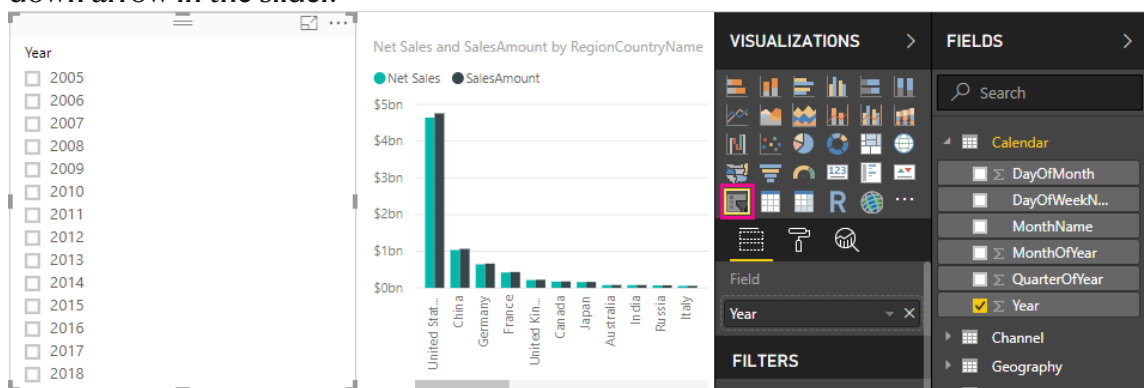
5. Select any value in the **Year** slicer to filter the **Net Sales and Sales Amount by RegionCountryName** chart accordingly. The **Net Sales** and **SalesAmount** measures recalculate and display results in the context of the selected **Year** field.
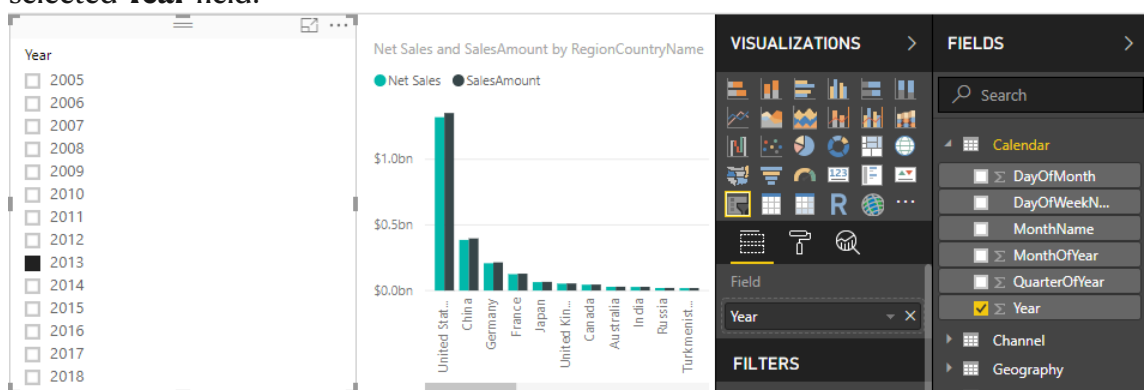

USE YOUR MEASURE IN ANOTHER MEASURE

Suppose you want to find out which products have the highest net sales amount per unit sold. You'll need a measure that divides net sales by the quantity of units sold. Create a new measure that divides the result of your **Net Sales** measure by the sum of **Sales[SalesQuantity]**.

1. In the **Fields** pane, create a new measure named **Net Sales per Unit** in the **Sales** table.

2. In the formula bar, begin typing *Net Sales*. The suggestion list shows what you can add. Select **[Net Sales]**.
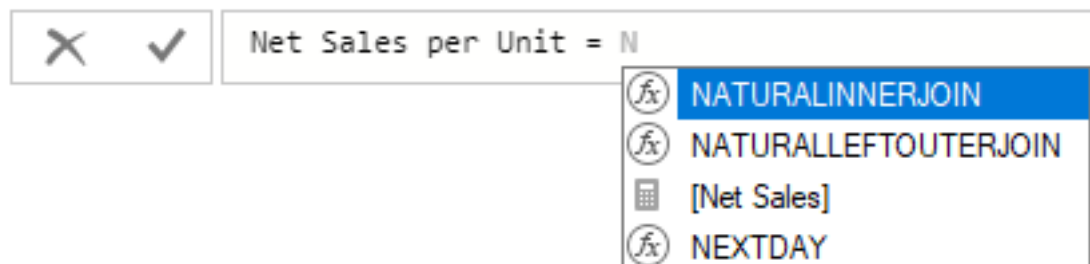
3. You can also reference measures by just typing an opening bracket (**[**). The suggestion list shows only measures to add to your formula.
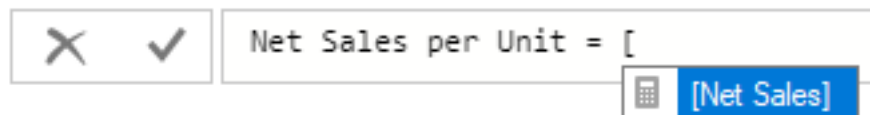

4. Enter a space, a divide operator (/), another space, a SUM function, and then type *Quantity*. The suggestion list shows all the columns with *Quantity* in the name. Select **Sales[SalesQuantity]**, type the closing parenthesis, and press **ENTER** or select **Commit** (checkmark icon) to validate your formula.
   The resulting formula should appear as:
   Net Sales per Unit = [Net Sales] / SUM(Sales[SalesQuantity])

5. Select the **Net Sales per Unit** measure from the **Sales** table, or drag it onto a blank area in the report canvas.
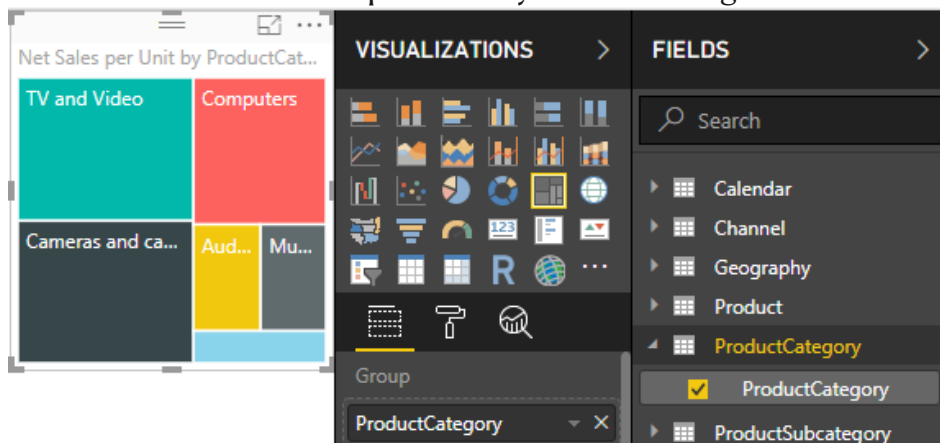   The chart shows the net sales amount per unit over all products sold. This chart isn't very informative; we'll address it in the next step.
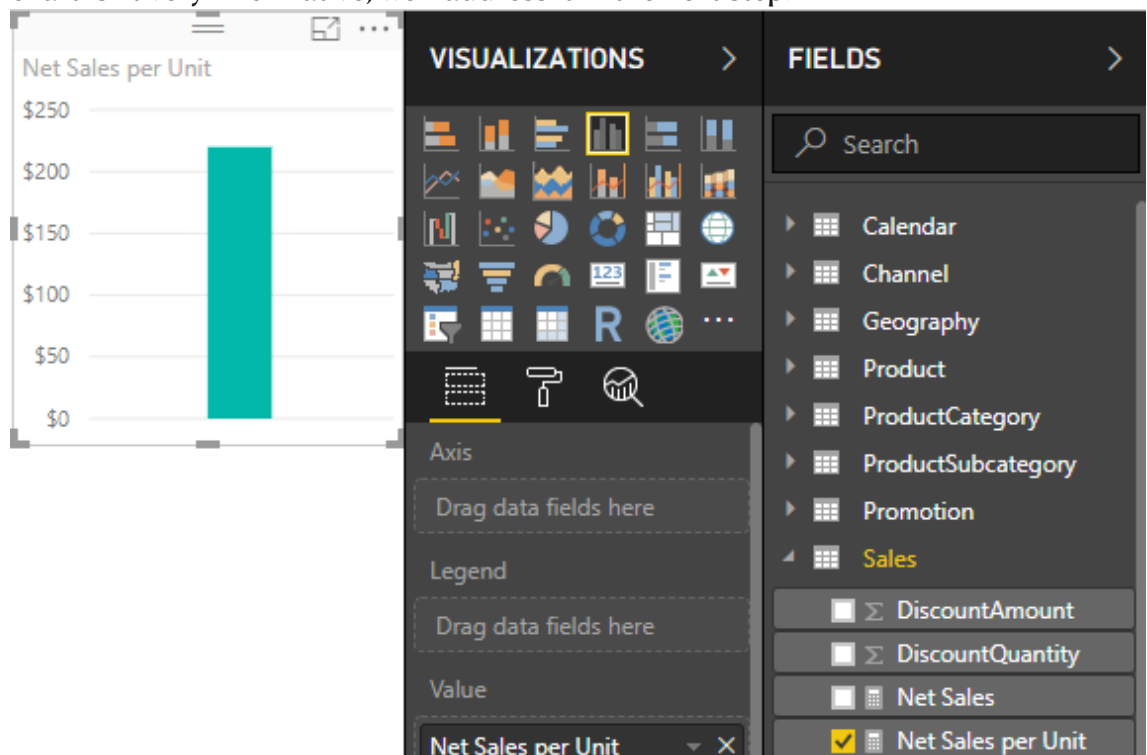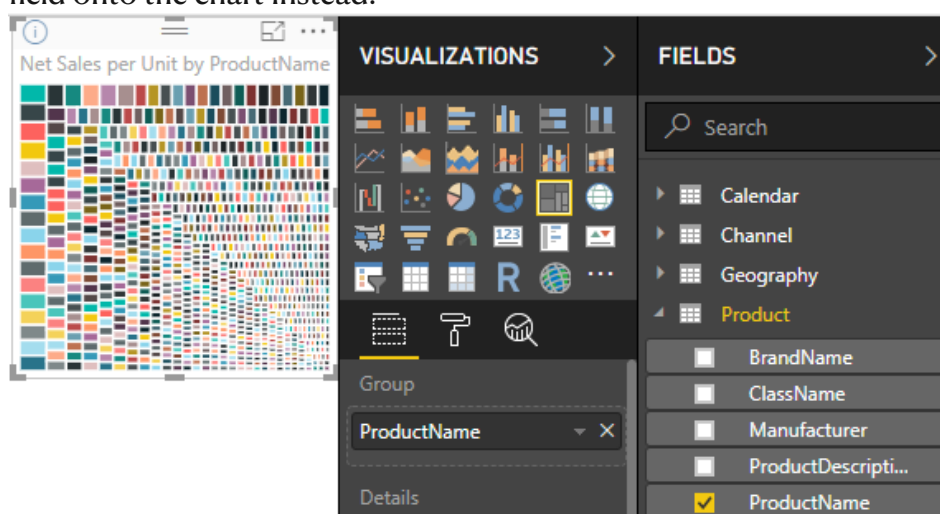
6. For a different look, change the chart visualization type to **Treemap**.
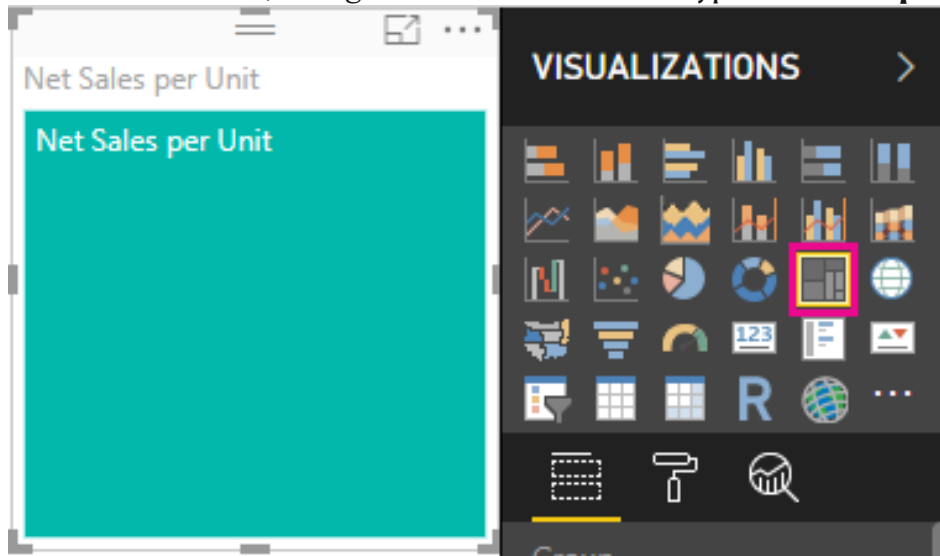
7. Select the **Product Category** field, or drag it onto the treemap or the **Group** field of the **Visualizations** pane. Now you have some good info!

8. Try removing the **ProductCategory** field, and dragging the **ProductName** field onto the chart instead.

Ok, now we're just playing, but you have to admit that's cool! Experiment with other ways to filter and format the visualization.