

SQL Server Technical Article

**Title**: Statistical Analysis in Excel and SQL Server Analysis Services: An Example using Boxplot

**Author:** David He, Senior Program Manager, Microsoft

**Contributors:** Chris Lucas, Michael Murstein, Jon Klein

**Published:** October, 2013

**Applies to:** SQL Server 2012, SQL Server 2008 R2

**Summary:** To offset the lack of commonly-used statistical calculations in Analysis Services, the author describes an approach for creating statistical calculations in SSAS and visualizing the results using Excel pivot tables, to combine slice-and-dice filtering with statistical functions in customizable charts. The boxplot is used as an example to demonstrate the technique, and tips are provided for how to extend the technique to other statistical functions, such as the probability density function.

Copyright

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2013 Microsoft. All rights reserved.

Contents

[Introduction 4](#_Toc369784719)

[Scenario 4](#_Toc369784720)

[Design the SQL Server Analysis Services Cube 5](#_Toc369784721)

[Data Model 5](#_Toc369784722)

[Boxplot Calculations 5](#_Toc369784723)

[Visualize Boxplot Charts in Excel 7](#_Toc369784724)

[Connect Excel to the SSAS Cube 7](#_Toc369784725)

[Customize Excel Pivot Table 9](#_Toc369784726)

[Visualize the Boxplot 11](#_Toc369784727)

[Compare Two Data Sets 18](#_Toc369784728)

[Pivot Two Data Sets 18](#_Toc369784729)

[Visualize Boxplots Side by Side 22](#_Toc369784730)

[Compare a Single Value to a Set 24](#_Toc369784731)

[Handling Outliers 27](#_Toc369784732)

[Extending the Technique to Other Visualizations 28](#_Toc369784733)

[Conclusion 29](#_Toc369784734)

# Introduction

Microsoft Excel and SQL Server Analysis Services (SSAS) both provide powerful business intelligence tools for exploring, analyzing and visualizing data. However, the current version of SSAS does not provide built-in functions for commonly-used statistical calculations, such as boxplots and probability density function (PDF). And although there is an [Analysis Toolpak](http://office.microsoft.com/en-us/excel-help/use-the-analysis-toolpak-to-perform-complex-data-analysis-HA102748996.aspx?CTT=1) add-in that installs with Excel 2013, Excel doesn’t offer visualizations for boxplots. In addition, in certain scenarios it would be desirable to have slicing and dicing capabilities when applying statistical methodologies on analytics.

To address these limitations, this paper describes an approach for creating statistical calculations in SSAS and visualizing results using Excel pivot tables, to combine slice-and-dice filtering with statistical functions. The boxplot will be used as an example to demonstrate the technique.

**Note:** This paper assumes some familiarity with Excel, SSAS, and common statistical terminology, methods and practices.

# Scenario

The scenario that we will use in this paper is based on the performance of software development teams. The goal is to identify significant variations between individual and team performance. For example, a software development manager may need to answer the following questions:

* What does the typical performance of a developer look like in my team?
* How does the performance of my team differ from the typical performance of other teams?
* How does the performance of any individual compare to the typical performance of their team, or of other teams?

In descriptive statistics, a boxplot is a convenient way to graphically depict groups of numerical data through their quartiles. Boxplots are useful for comparing distributions between groups or sets of data, such as the distribution of “work items resolved by person” on a development team.

We are going to produce graphs for the following two comparisons as examples, and compare them visually:

* Comparison of a distribution of values for a group against a distribution of numbers for another group.
* Comparison of a summary value for an individual against a distribution of numbers for a group.

Of course, these are not the only metrics you might use. You may need to assess the differences manually or run statistical tests such as the *t*-test on the dataset. This paper is not intended to provide an exhaustive list of metrics or statistical tests. Rather, our goal is to provide a method for filling in one piece of the puzzle, by demonstrating how statistical calculations, such as the distribution of “Work Items Resolved by Person”, can be visualized in Excel as boxplots for easy visual comparison.

# Design the SQL Server Analysis Services Cube

The data for the analyses is stored in an SSAS cube. The raw data may come from a data warehouse environment, with key metrics loaded into a cube. We have created several default measures based on these metrics, and will use custom MDX calculations to generate the additional values needed for boxplots, as described in the next section. The custom calculations are then used in Excel for visualization.

## Data Model

First, create a simple dimensional model in an SSAS cube, with one fact table and two dimensions, as shown in the diagram.

* Fact table: Work Items Resolved.
* Dimension: Date
* Dimension: Person

Dimensions are useful for selecting an arbitrary set of data filtered by date and person.



Figure 1. Fact and dimension data model

Using this model, you can create different data sets, and slice and dice the work items by date and by person in Excel.

## Boxplot Calculations

SSAS supports many mathematical operations but it does not support statistical calculations. Depending on what you are trying to do, you may have to write your own MDX code for statistical calculations. For the boxplot, you can create the following calculations in the cube:

* Maximum (Max)
* 1st Quartile (Q1)
* Median
* 3rd Quartile (Q3)
* Minimum (Min)
* Interquartile Range (IQR)

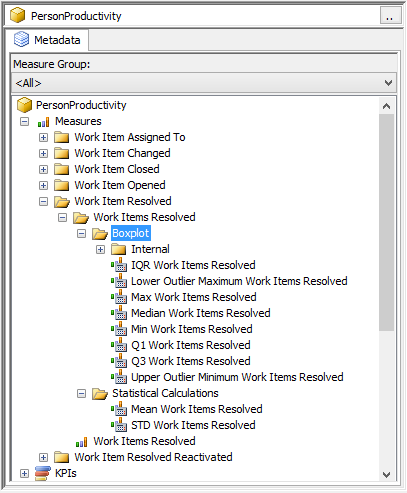


Figure 2. Boxplot calculations in the SSAS cube

The following MDX statement is one example of how to calculate First Quartile (Q1). The code is designed solely to illustrate the concept, so it is may not be optimized for performance.

IIF(

**//Mod function**

([Measures].[Row Count Work Items Resolved] - (INT([Measures].[Row Count Work Items Resolved] / 4)\*4)) = 0

,

**//if even, take average**

(

(

MAX(

BOTTOMCOUNT(

NONEMPTY([Person Breakdown] \* [Measures].[Work Items Resolved])

,[Measures].[Row Q1 Work Items Resolved]

,[Measures].[Work Items Resolved]

),

[Measures].[Work Items Resolved])

)

+

(

MAX(

BOTTOMCOUNT(

NONEMPTY([Person Breakdown] \* [Measures].[Work Items Resolved])

,[Measures].[Row Q1 Work Items Resolved] + 1

,[Measures].[Work Items Resolved]

),

[Measures].[Work Items Resolved])

)

) /2

,

**//if odd, take value**

MAX(

BOTTOMCOUNT(

NONEMPTY([Person Breakdown] \* [Measures].[Work Items Resolved])

,[Measures].[Row Q1 Work Items Resolved]

,[Measures].[Work Items Resolved]

),

[Measures].[Work Items Resolved])

)

# Visualize Boxplot Charts in Excel

After the boxplot calculations have been defined in the cube, you can access the values from Excel and pivot by dimensions, such as work items resolved by person in a specific period.

## Connect Excel to the SSAS Cube

To use the boxplot calculations in Excel, first connect to Analysis Services.

1. In Excel, click **Data** > **From Other Sources** > **From Analysis Services**.
2. In the **Data Connection Wizard**, type the name of the server that hosts the SSAS cube and click **Next**.

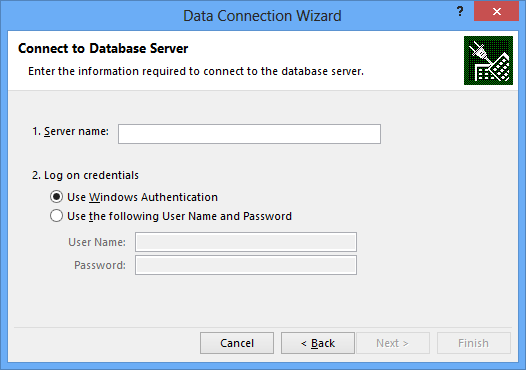


Figure 3. Data Connection Wizard dialog box

1. Follow the steps in the wizard to select an SSAS instance and a specific cube.
2. In the **Save Data Connection File** dialog box, click **Finish**.
3. In the **Import Data** dialog box, select **PivotTable Report** and then click **OK**.

You should be able to see a list of cube measures and dimensions, such as those in the following diagram.

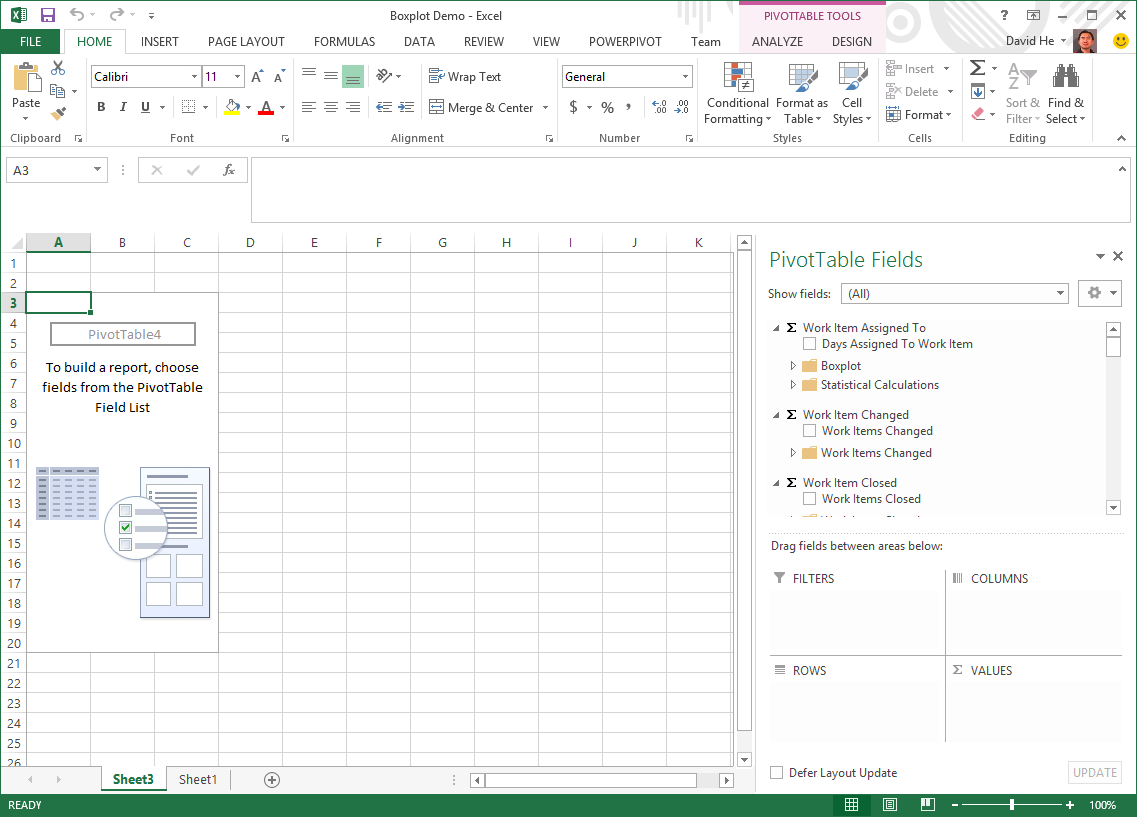


Figure 4. Cube measures and dimensions in Excel

## Customize Excel Pivot Table

Excel Pivot Tables and Pivot Charts allow users to slice and dice data from a SSAS cube, but Excel doesn’t include the boxplot chart type. In order to display a boxplot, you need to create a customized Pivot Table.

1. In the Excel worksheet, add the Date and Person dimensions to the **Filters** area of the field list.
2. Add formulas to the Pivot Table that correspond to the boxplot calculations (Max, Q3, median, Q1, Min). You create these calculations by using the Excel [CUBEVALUE function](http://office.microsoft.com/en-us/excel-help/cubevalue-function-HA010083018.aspx) and calling the measure defined earlier.

For example, the following expression retrieves the Min value of the boxplot calculation defined earlier in the cube. The expression also applies two filters, based on the dimensions you added as filters: one on Person (cell B2) and one on Date (cell B3).

=CUBEVALUE("ServerName DatabaseName CubeName", $B$2,$B$3, "([Measures].[Min Work Items Resolved])")

1. Add similar formulas for the other boxplot calculations.

**1st Quartile (Q1)**

=CUBEVALUE("ServerName DatabaseName CubeName", $B$2,$B$3, "([Measures].[Q1 Work Items Resolved])")

**Median**

=CUBEVALUE("ServerName DatabaseName CubeName", $B$2,$B$3, "([Measures].[Median Work Items Resolved])")

**3rd Quartile (Q3)**

=CUBEVALUE("ServerName DatabaseName CubeName", $B$2,$B$3, "([Measures].[Q3 Work Items Resolved])")

**Maximum (Max)**

=CUBEVALUE("ServerName DatabaseName CubeName", $B$2,$B$3, "([Measures].[Max Work Items Resolved])")

1. Now drag the measure for the boxplot functions to the **Values** area of the PivotTable. You should have a list like that shown below. Note that, by default, the values for the Person and Date filters are set to **All**.
2. Now, for Person, select the group Software Development Engineer, and for Date, select the year 2013.

The results of the boxplot formulas created using the CUBEVALUE function are updated using the filters on the Pivot Table, to show the calculation over that subset.

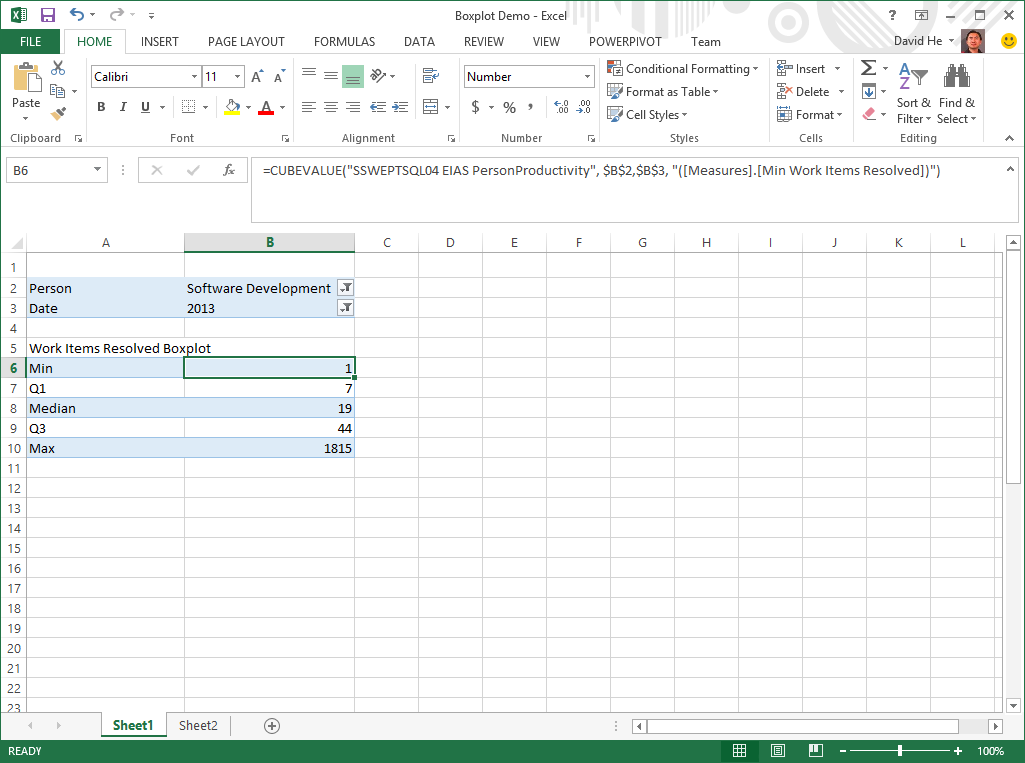


Figure 5. Retrieving boxplot values from a SSAS Cube

## Visualize the Boxplot

Now that you have retrieved the boxplot values from the cube and created local calculations to use those values, the next step is to visualize the results of the boxplot formulas. Excel doesn’t provide boxplot charts, but you can customize a stacked column chart to display as a boxplot.

However, the boxplot values that you obtained are not stacked. Therefore, to show the correct values using a stacked column chart, you need to take one more step, and calculate the difference between some of the values from the cube.

You can do this by adding some simple Excel formulas just below the other formulas, as shown in the following graphic:

* Q1 minus Min
* Median minus Q1
* Q3 minus Median
* Max minus Q3

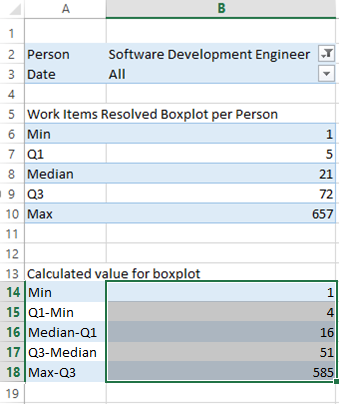


Figure 6. Adding derived boxplot values

Now that you have all the values you need, you can create the stacked column chart.

1. Select all the values for the boxplot that you want include in the chart, as shown above. Be sure to include Min, Q1-Min, Median-Q1, Q3-Median, and Max-Q3.
2. Click **Insert** >**Charts** >**Stacked Column Chart**. When the preview appears, click the chart type that stacks all values in a single column.

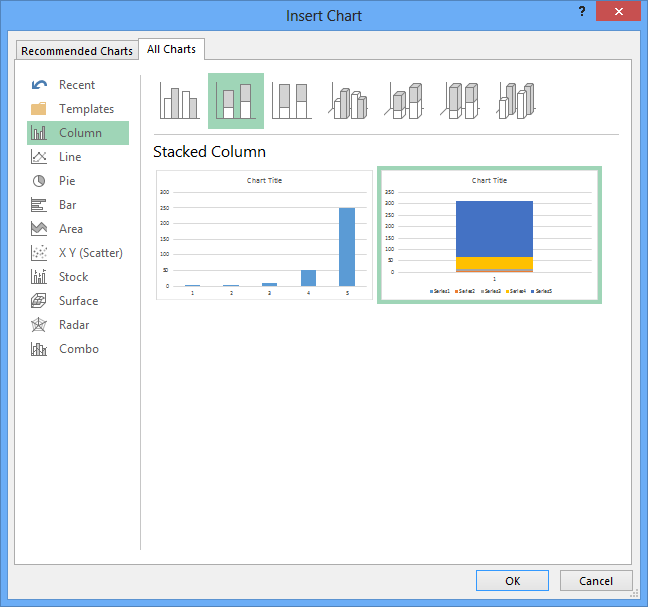


Figure 7. Insert Chart dialog box

1. After the stacked column chart has been added to the Excel worksheet, select the first column from the top (Series 5).
2. Right-click and use the shortcut menu to change the fill color to **No Fill**
3. Change the border style (**Shape Outline**) to **No Outline**.

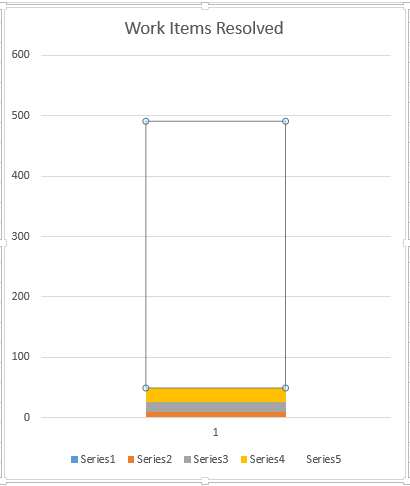


Figure 8. Customizing a stacked column chart to be a boxplot

1. Select the first and second column from the bottom (Series 1 and Series 2).
2. For these series, change the shape property to **No Fill** and change the outline style to **No Outline**.

Tip: If it is difficult to select a part of the chart because it is too small, select the series name in the legend, right-click, and select **Format Data Series**. In some menus, the option might be **Border Styles**, rather than **Outline**, but they do the same thing.

At this point, the series displayed in the chart should be as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Series** | **Formula** | **Value** | **Fill** | **Outline** |
| Series 1 | Min | 1 | No | No |
| Series 2 | Q1-Min | 4 | No | No |
| Series 3 | Median-Q1 | 16 | Yes | Yes |
| Series 4 | Q3-Median | 51 | Yes | Yes |
| Series 5 | Max-Q3 | 585 | No | No |

By making Series 1, 2 and 5 **No Fill** and **No Outline**, you can add whisker plots.

1. Select the second column from the top (Series 4).
2. Select **Design** >**Add Chart Element** >**Error Bars** >**More Error Bar Options**. (In Excel 2010, you can find the **Error Bars** option in the **Chart Tools** ribbon, on the **Layout** tab.)

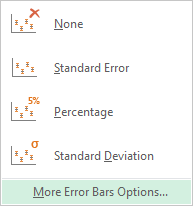


Figure 9. Error Bar Options dialog box

1. In the **Format Error Bars** dialog box, for **Direction** select **Plus**, and for **End Style** select **Cap**.

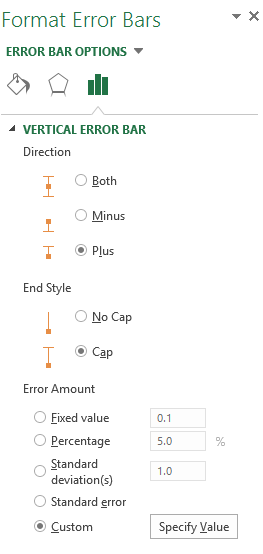


Figure 10. Formatting the top whisker plot

1. Select the **Custom** option, and then click **Specify Value**.
2. In the **Custom Error Bars** dialog box, for **Positive Error Value**, select the cell that holds the “Max-Q3” value and click **OK**. You can ignore the Negative Error Value.

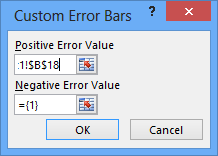


Figure 11. Selecting the value for the top whisker plot

Now, you will create another whisker at the bottom of the chart, using a slightly different method.

1. Select the second column from the bottom (Series 2, or Q1-Min).
2. Select **Design >Add Chart Elements >Error Bars >More Error Bars Options…**
3. In the **Format Error Bars** pane, select **Minus, Cap**, and **Percentage.** Enter 100 as the percentage, and then press Enter.

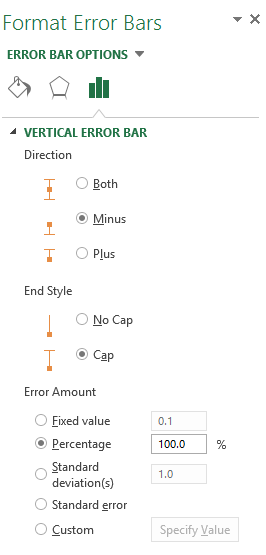


Figure 12. Formatting the bottom whisker plot

You will see a boxplot chart as shown below.

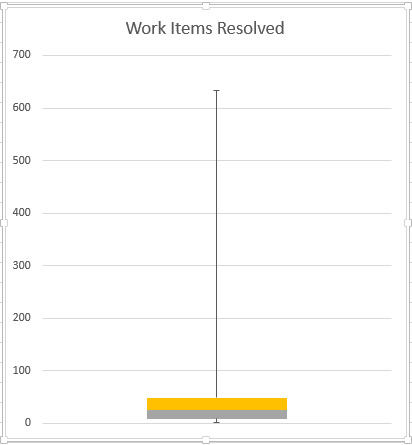


Figure 13. A finished boxplot chart

You can further customize the boxplot chart for a different look and feel by using the chart properties.

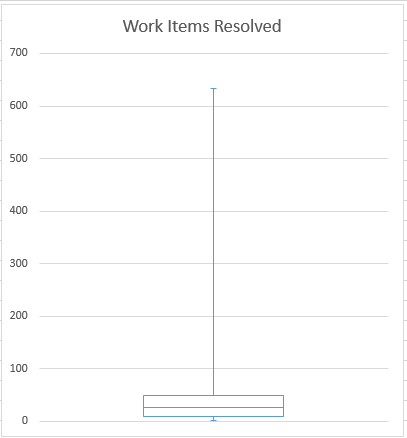


Figure 14. Customizing the look and feel of the boxplot

# Compare Two Data Sets

In the preceding sections, we have shown how to connect to an SSAS cube, create a customized pivot table and visualize a dataset as a boxplot. Next we want to show a more complex scenario.

## Pivot Two Data Sets

In this scenario, you need to visualize and compare two sets of data. To illustrate the techniques, you’ll compare data from the same SSAS cube that was used in the previous example, but apply different filters to slice and dice the data. The two sets of data could also come from different data sources.

* Step 1. Add a second pivot table to the worksheet.
* Step 2. Get the boxplot values from the cube as described earlier.
* Step 3. Apply filters to retrieve different groups in the side-by-side Pivot Tables.

Begin by adding a second Pivot Table for the comparison data.

1. Click a cell to use as the starting point for the second pivot table, then select **Insert** >**PivotTable**. Here we’ve positioned the new Pivot Table next to the one created earlier, for easier comparison.

Note: On the first Pivot Table, you didn’t need to go through this manual process, because the Data Connection Wizard automatically creates it at the selected location.

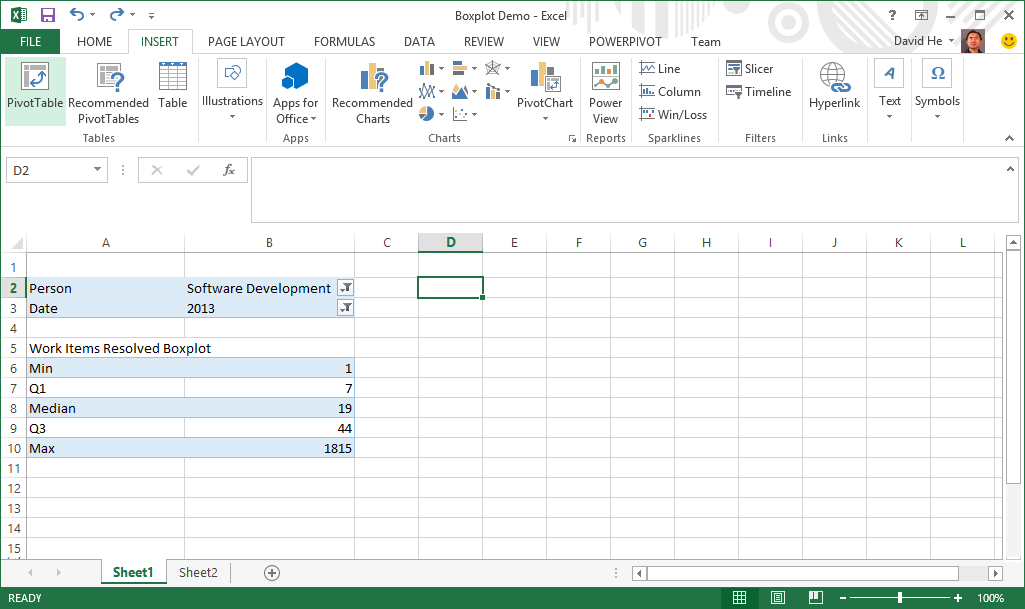


Figure 15. Adding the second PivotTable

1. In the **Create PivotTable Wizard**, select **Use a data source**, and select the connection you created before.

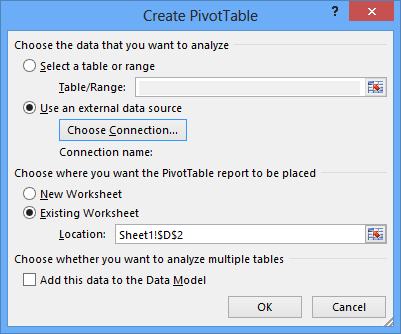


Figure 16. Create PivotTable Wizard

1. Click **OK**. The wizard adds a second pivot table (shown here in orange) to the worksheet.
2. As you did in the previous Pivot Table, drag the Date and Person dimensions to the **Filter** area of the new Pivot Table
3. To differentiate the two Pivot Tables, name the one created earlier ***Group 1***, and name the one you just created ***Group 2***.

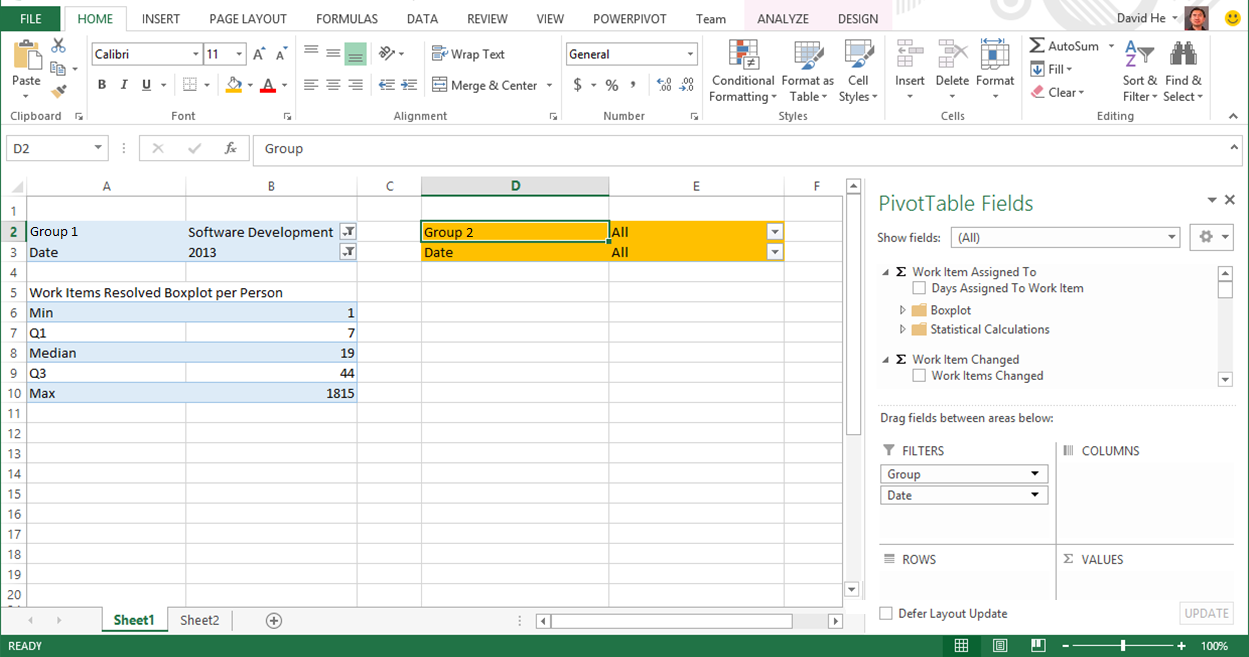


Figure 17. PivotTables for Group 1 and Group 2

To get the boxplot values from the SSAS cube, you will use the same approach described in the previous section.

1. Click a cell opposite the one you created earlier for “Min Work Items Resolved”, and apply the following formula:

=CUBEVALUE("ServerName DatabaseName CubeName", $E$2,$E$3, "([Measures].[Min Work Items Resolved])")

1. Repeat for all formulas that you want to use in the second Pivot Table.

Now you have two Pivot Tables, independent of each other but based on the same data set. With these Pivot Tables, you can view different subsets in each Pivot Table, filter them differently, or slice and dice each set independently.

To understand how to compare two groups using filters, first review the following diagram, which shows the hierarchy of the Person dimension. Notice that “Senior Software Development Engineer” is a sub-group of “Software Development Engineer” in the hierarchy.

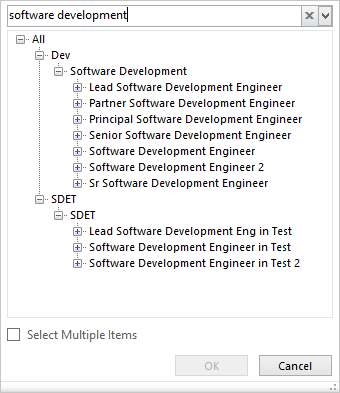


Figure 18. Person hierarchy

You will compare these two groups using the Pivot Table values. The chart on the left, in blue, contains work items resolved by Software Development Engineers. The chart on the right, in orange, contains work items resolved only by Senior Software Development Engineers.

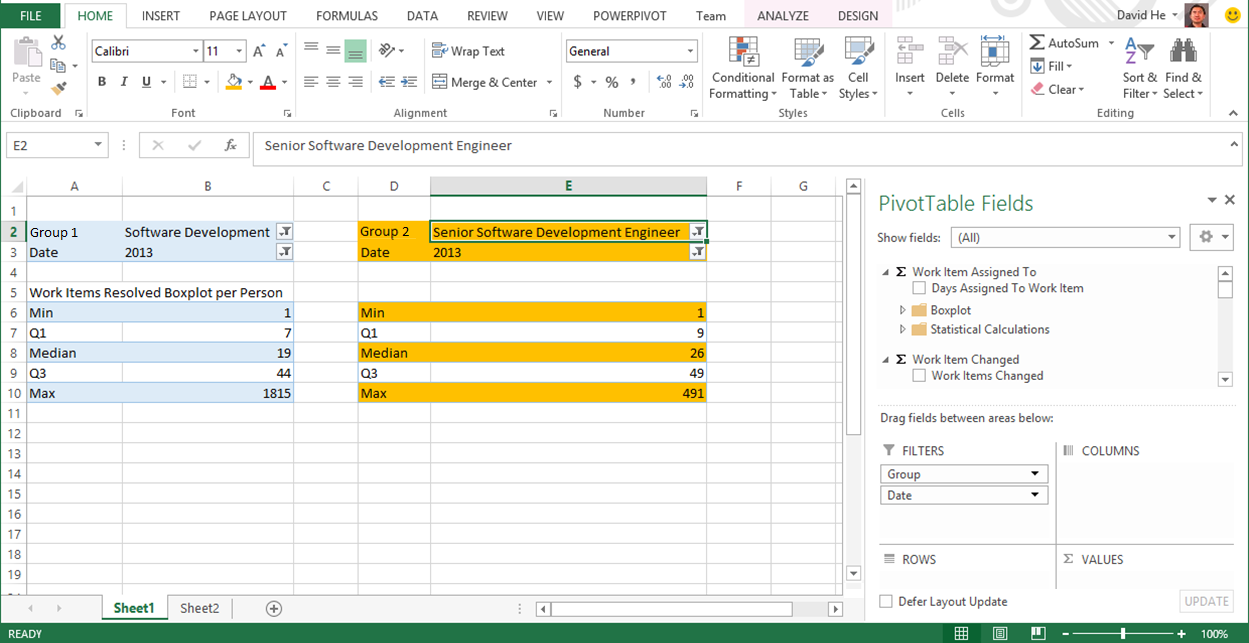


Figure 19. Pivot tables containing boxplot values form two data sets

Now you are ready to compare those two groups using the boxplot charts.

## Visualize Boxplots Side by Side

For easier comparison, you may want to visualize the two boxplots side by side. A simple option is to create two charts, each representing a boxplot, but it can be a challenge to make them display in a way that works well for ad-hoc analytics. For example, they have to be in the exact same scale, which can be hard to control.

A better approach is to show two boxplots in one chart.

To do that, you can insert a combo chart that includes the derived boxplot values (Min, Q1-Min, Median-Q1, Q3-Median, Max-Q3) for both data sets.

1. Select the derived boxplot values for both data sets.
2. Select **Insert** >**Column Chart** >**More Chart Types** >**Combo**.
3. For each series in the combo chart, choose **Stacked Column** as the chart type.

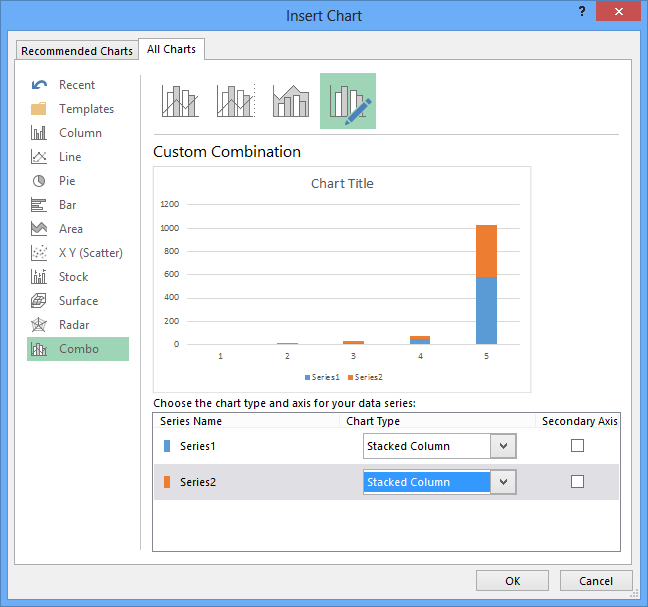


Figure 20. Inserting a Combo chart

However, as you can see from the example below, the default layout of the stacked chart doesn’t keep all values from the same set together.

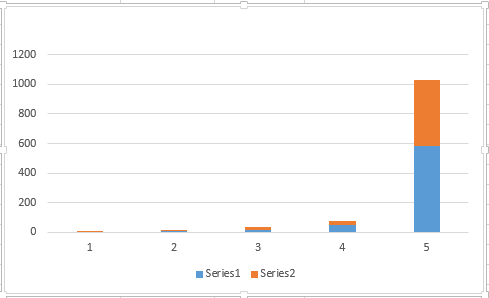
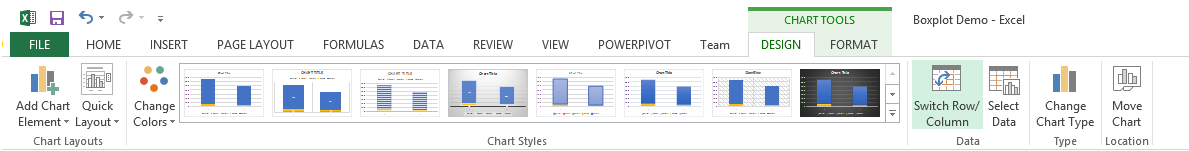


Figure 21. Default layout of the stacked column chart

1. To correct this, in the **Chart Tools** menu, click **Switch Row / Column**.



1. The chart will turn into two stacked columns as shown below.

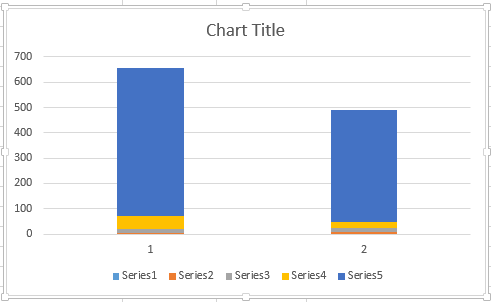


Figure 22. Column chart for two datasets side-by-side

1. Repeat the steps described in previous sections to change each stacked column chart into a boxplot.

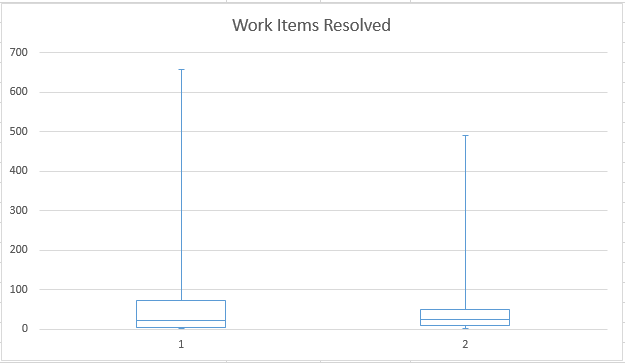


Figure 23. Boxplot chart for two datasets side-by-side

Now you have two boxplots in the same chart for side-by-side comparison.

# Compare a Single Value to a Set

For two sets of data, you can show boxplots side-by-side. However, there are other scenarios in which you may want to compare an individual’s performance (in this case, John Doe) against the performance of a group (such as Senior Software Development Engineers), To do this, you need to visualize a single summary value (representing an individual) against a boxplot distribution (representing a group).

In this section, we will show you how to visualize a summary value and a boxplot together in a single chart. We’ll start with the following example data:

* The individual engineer’s results are in the left-hand Pivot Table, in green.
* The right-hand Pivot Table shows the results and boxplot values for the group you’ll compare the individual to.
* The green stacked chart shows the boxplot values for the group.

Start by setting up the values used in the chart.

1. Apply filters in the Pivot Table, as described earlier, to generate a summary value for an individual.
2. Next select all the derived boxplot values for the comparison group, and select the value from the Count field (‘Total Work Items Resolved”) for the individual.
3. Select **Insert** > **Stacked Column Chart**., and in chart preview, choose the chart that keeps all values in a single column.

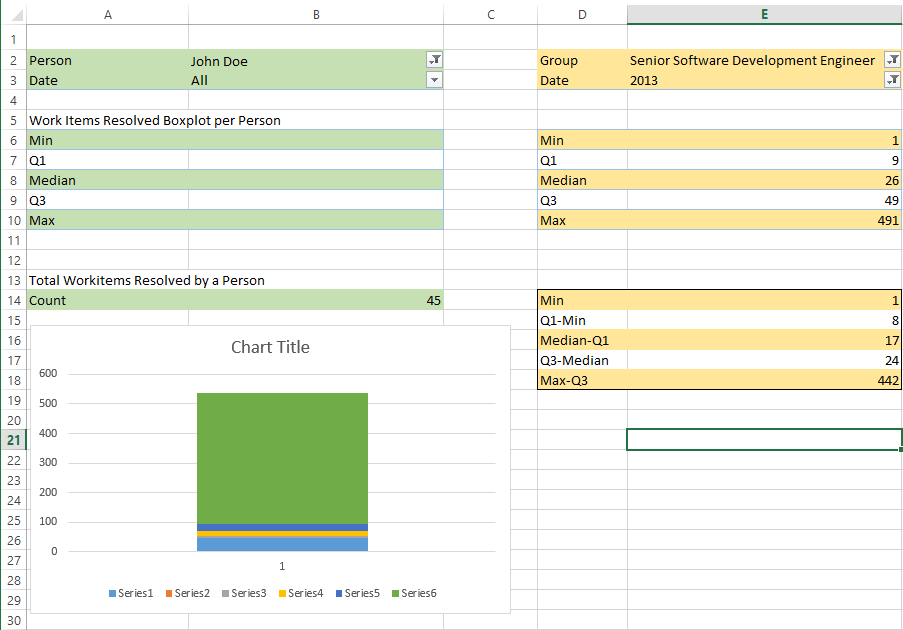


Figure 24. Stacked column chart based on the summary value and boxplot distribution

1. Change the chart type to a combo chart by right-clicking each series in the chart, and then selecting **Change Series Chart Type**.

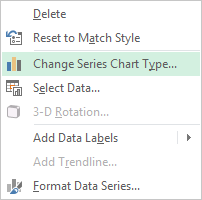


Figure 25. Changing chart type dialog box

1. For Series1, select **Stacked Line with Markers.** This series represents the single value for the individual.
2. Leave the other series as **Stacked Column**. These values represent the boxplot distribution for the comparison group.

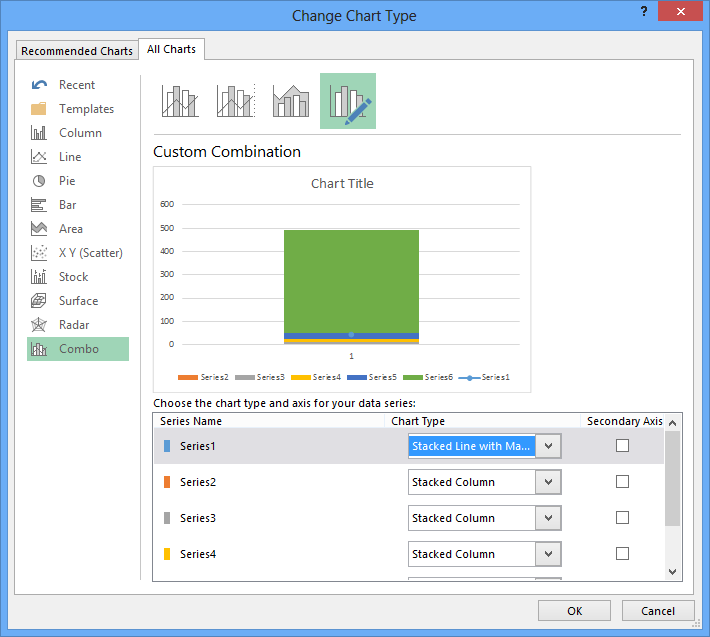


Figure 26. Customizing the Combo chart

1. Finally, repeat the process of removing fill color and outlines to change each of the series in the stacked column blocks to whisker plots, as described earlier.

The result, shown in the final chart below, compares a single value against a distribution. The blue dot represents the performance of the individual and the boxplot represents the distribution of a group’s performance.

Because the blue dot is hard to see, continue changing the appearance of the stacked column series, to make it look more like a box plot, as shown in the following graphic.

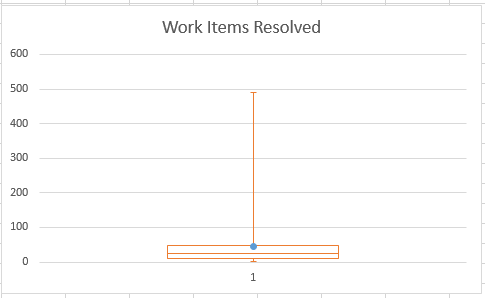


Figure 27. Comparing a summary value to a boxplot

# Handling Outliers

Outliers can skew distributions. Fortunately, if needed, you can easily change the formula to filter out outliers.

The following adjustments to the formulas will generate a Max and Min value for the boxplot visualization without outliers.

Min = MAX(0, MAX(<Min>, <Q1> - 1.5 \* IQR))

Max = MIN(<Max>, <Q3> + 1.5 \* IQR)

Where:

MAX() = math max(a,b) function

MIN() = math min(a,b) function

IRQ = Interquartile Range

In the Excel cell corresponding to “Min Work Items Resolved”, replace the existing formula with the following formula:

=MAX(CUBEVALUE(ServerName DatabaseName CubeName, $B$2,$B$3, "([Measures].[Min Work Items Resolved])"), CUBEVALUE(ServerName DatabaseName CubeName, $B$2,$B$3, "([Measures].[Q1 Work Items Resolved])")-1.5\* CUBEVALUE(ServerName DatabaseName CubeName, $B$2,$B$3, "([Measures].[IQR Work Items Resolved])"))

In the cell corresponding to “Max Work Items Resolved”, replace the existing formula with the following formula:

=MIN(CUBEVALUE(ServerName DatabaseName CubeName, $B$2,$B$3, "([Measures].[Max Work Items Resolved])"),CUBEVALUE(ServerName DatabaseName CubeName, $B$2,$B$3, "([Measures].[Q3 Work Items Resolved])")+1.5\*CUBEVALUE(ServerName DatabaseName CubeName, $B$2,$B$3, "([Measures].[IQR Work Items Resolved])"))

# Extending the Technique to Other Visualizations

The Probability Density Function (PDF) is another commonly used statistical method. You can simulate this function by using a histogram as a graphical representation of the distribution of data.

If you only need to analyze static datasets in Excel, you can of course create histogram charts using the Analysis ToolPak. However, if you want to slice and dice data stored in SSAS, you must be able to dynamically calculate the frequency within each bin.

The key to dynamic histograms is creating a utility dimension in the SSAS cube to represent the bins. You may choose to create static or dynamic utility dimensions.

Designing SSAS cubes to support such histograms is a multiple-step process, with many options and design tradeoffs; therefore, we cannot fully describe it here. However, we will briefly describe the concept of the solution here, to illustrate how the technique described for boxplots can be extended to create other statistical chart types.

The key steps in the solution are as follows:

1. Identify the event for the histogram. For example, in the following example, the event is time to resolve a defect.

2. Define the bin size. In this example, we chose days.

3. Create bins in a utility dimension.

4. Calculate the frequencies of events in each bin.

After you have created a utility dimension, you can easily apply bin values as the X-axis and the frequency values as the Y-axis, to show histogram charts in Excel.

The following diagram shows one example of this technique.

* The X-axis shows the number of days.
* The Y- axis shows the percentage of defects resolved.

Figure 28. A histogram chart based on utility dimensions in a SSAS cube

From this chart, you can see that about 15% of all the defects were resolved within a day in this data sample. By using a histogram chart, you can estimate the Probability Density Function.

# Conclusion

This paper describes how you can use Excel and SSAS for statistical analysis, using boxplots and probability density functions as examples. By using the combined capabilities of Excel and SSAS, you are not limited to statistical calculations on static sets of data, but can select from arbitrary sets of data by slicing and dicing. By using SSAS for statistical calculations and Excel for visualization, you can leverage those products in advanced analytical scenarios.

**About the Author**

David T.Y. He is a Senior Program Manager for Microsoft Advanced Strategy and Research. In his free time, he has written several whitepapers for MSDN on business intelligence and analytics. Contact him at david.he@microsoft.com.

For more information:

<http://www.microsoft.com/sqlserver/>: SQL Server Web site

<http://technet.microsoft.com/en-us/sqlserver/>: SQL Server TechCenter

<http://msdn.microsoft.com/en-us/sqlserver/>: SQL Server DevCenter

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
* Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

[Send feedback](mailto:sqlfback@microsoft.com?subject=White%20Paper%20Feedback:%20[Statistical%20Analysis%20in%20Excel%20and%20SQL%20Server%20Analysis%20Services:%20An%20Example%20using%20Boxplot]).