iTip	Guidance from the Functional Sizing Standards Committee on topics important to you	

Charts and Graphs

iTip # 2 – (vsn. 1.0 *12/01/2012)*

iTips provide guidance on topics important to the FPA community. They explain the application of IFPUG FPA method in a particular situation. iTips are not rules, but interpretation of the rules, and provide guidance using a realistic example to explain the topic being covered.

This iTip is focused on describing the IFPUG FPA method as it applies to measuring output presented as charts and graphs available in a typical application. This iTip includes a series of examples but is not an exhaustive examination of the subject.

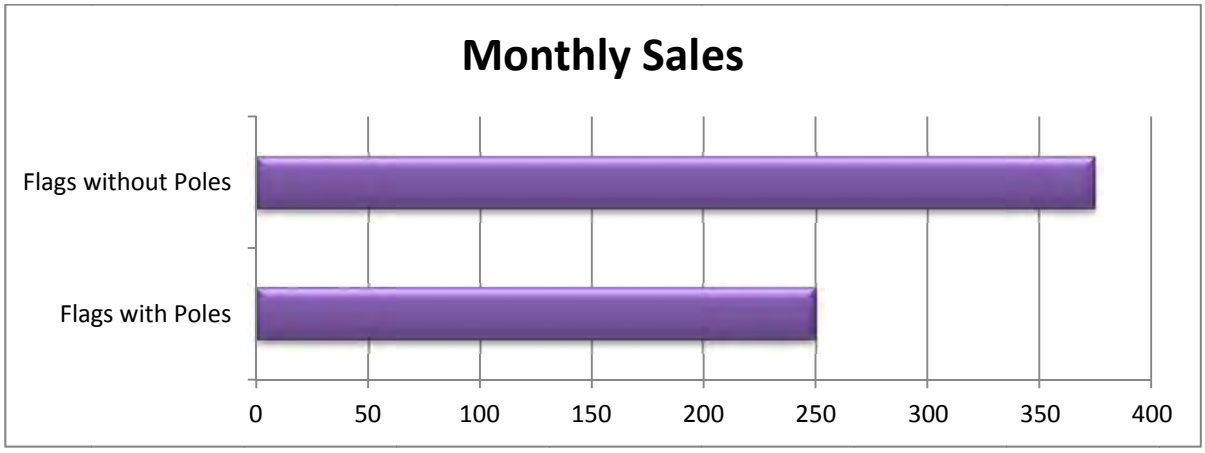
Background

Charts and graphs are effective tools for displaying data to end users. Often it is easier to comprehend data presented in charts and graphs as opposed to a tabular presentation of the same data. It's been said that "a picture is worth a thousand words"; however, what's beyond debate is the simple fact that charts and graphs are measured using the same IFPUG FPA rules applied to other forms of output.

If the data to be displayed in a chart or graph is calculated, the chart or graph is automatically an EO because the primary intent is presenting information to the user and mathematical calculations are performed. Therefore, for the following examples we will limit discussion to cases where the data to be charted or graphed is retrieved from a data function without calculation of those data values. To keep things simple, we will assume that the data retrieved in each example is from a single data function, that there is a requirement for the ability to initiate the chart or graph, and that there are no application response messages.

Example – Simple Bar Chart

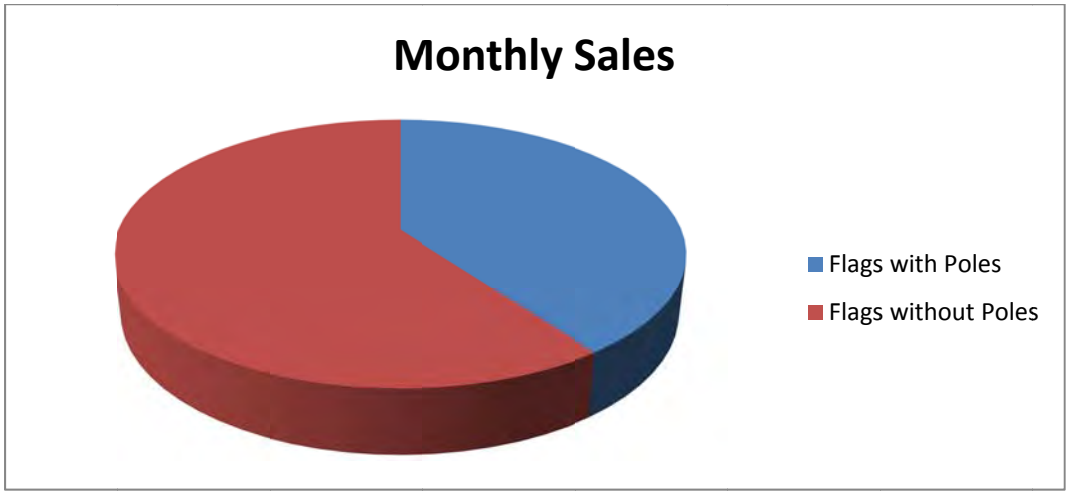
The Patriot Flag Company sells two products: flags with poles and flags without poles. They record their sales of each product monthly. They wish to display a bar chart of a month's sales.



This chart has two unique user recognizable, non-repeated attributes that cross the boundary during the processing of the transactional function retrieved from the data function: Type of Product Sold and Quantity of Product Sold for the Month. Because the user specifically requested a bar chart, there is an additional user recognizable, non-repeated attribute that crosses the boundary – the scale for the chart which is calculated based on the retrieved data values. While this may not seem obvious, it sometimes helps to go back to the manual implementation that would occur if there was no computer. When graphing manually on graph paper, once the values to be graphed are determined the next step is to determine and label the scale to be used for the graph. The values are then graphed against that scale. The bar chart has 3 unique user recognizable attributes and the ability to initiate the chart, resulting in 4 DETs. This bar chart is an example of a low complexity EO.

Example – Pie Chart

In this example, the same flag sales information is requested to be displayed on a pie chart.



Because the user specifically requested a pie chart, there is an implied requirement to display the percentages of each slice as they relate to the whole. Determining the percentage requires a calculation to be performed, which causes this graph to be categorized as an EO. This chart has two unique user recognizable, non-repeated attribute that cross the boundary during the processing of the transactional function retrieved from the data function: Type of Product Sold and Percentage of each Product Sold for the Month and the ability to initiate the chart, resulting in 3 DETs. This pie chart is an example of a low complexity EO.

Summary

While the two examples given are very simplistic, they outline the things to consider when assessing a chart or graph. Look at all of the user requirements. Are there functional user requirements that are implied by the type of chart or graph being requested such as charts with implied cumulative components (e.g., stacked bar charts)? What would be considered as user recognizable if a manual process were being executed rather than an automated one?

Frequently Asked Questions (FAQ)

What if the Bar Chart also displays the graphed quantity?

The rule in effect here is to count "each unique user recognizable, non-repeated attribute that cross the boundary during the processing of the transactional function." The quantity would now be displayed twice – once as a bar on the graph and once as a number. They are different in how they are displayed, but not what is displayed. Any other unique user recognizable, non-repeated attribute that crosses the boundary during the processing of the transactional function would add additional DETs. The same logic would apply if the percentages were displayed in the pie chart. Quantities on the pie chart would be examples of additional DETs.

If the user provides a fixed scale as part of the input in the bar graph, would it be counted the same?

No data would be calculated, so the transaction would be an EQ instead of an EO. The DETs would still be the same.

Are all graphs and charts low complexity EOs?

Most graphs and charts are low complexity EOs. However, the previous question provided an EQ. When data comes from multiple data functions and more user-recognizable fields are input or displayed, the complexity of the function being measured can progress to average or high. (When assessing potential DETs, remember that literal fields such as report titles, screen or panel identifiers, column headings and attribute titles are not counted.) It is up to the function point analyst

to determine the correct elementary process, transactional function, and complexity based on the user requirements for the function being measured.

What if there is messaging, e.g., "There are no values for your selected criteria"?

"Count only one DET per transactional function for the ability to send an application response message even if there are multiple messages" would apply here, so one additional DET would be counted.

Does using a tool (e.g., Excel) vs. creating the output via algorithms (e.g., writing code to create the function) change the way that I consider the outputs?

It does not change the way that you size the functionality. The same rules are applied in the same way to measure the same functional size. It would result in very different productivity measures and if you were estimating the effort to develop or maintain the functionality, the factors applied would be very different.

Further Reading

IFPUG Counting Practices Manual, Part 1, Section 5.5 – Measure Transactional Functions.

IFPUG Counting Practices Manual, Part 2, Chapter 7 – Measure Transactional Functions.

http://office.microsoft.com/en-us/excel-help/available-chart-types-HA010342187.aspx?CTT=1#BMbarcharts