

# Fundamental Differences in Analytical Tools

## Exploratory, Custom, or Customizable

Stephen Few, Perceptual Edge  
*Visual Business Intelligence Newsletter*  
September/October 2009

Software tools for data analysis are designed for various purposes. Some support exploratory data analysis, allowing us to become familiar with and make sense of data with little direction or constraint, quickly and easily taking each step that's necessary to get to the next insight. Some are customized to support a specific set of related analytical tasks, simply and efficiently, guiding us through a series of steps in a prescribed manner. Some support the development of custom analytical applications, providing the means to make the application work precisely as needed. Some are designed to support multiple purposes, such as exploratory data analysis along with the ability to build custom analytical applications. When you shop for analytical software, it's important to know which of these objectives you're hoping to achieve so you can carefully confirm that products aren't just good, but are a good fit for you. Data analysis software vendors should understand the differences in functionality and in the interface that these different purposes require so they can intentionally and carefully build what's needed into their products.

### An Aside

What I'm calling "exploratory analytics" in this article is often also called "ad hoc analytics." I'm in the habit of using the term "ad hoc" to describe this flexible, exploratory analytical process, but I've discovered that the term "ad hoc" is somewhat ambiguous. In its general usage, the Latin term "ad hoc" translates into English as "for this purpose." Unfortunately, when applied to analytics it suggests an application that was developed for a specific purpose, which is the opposite of what we mean. Different from its general use, in the context of information technology, *ad hoc* means something that is not anticipated but can be customized in the moment to achieve any objective. Having worked in information technology throughout my career (not counting a brief stint in the ministry), I use the term to mean the latter, but have recently discovered that some people find this confusing, which is certainly understandable. For this reason, I'll stick with the term "exploratory" and avoid "ad hoc" entirely.

## Different Analytical Audiences

The audience for analytical tools of these three types differs. Tools should be built with a clear understanding of their audiences (that is, the people who will use them) and their particular needs.

### Exploratory Analytics Audiences

Data analysts (those who work with information to make sense of it, spanning a broad range of analytical expertise) often need exploratory tools. With them, analysts approach data with questions—sometimes open ended questions, such as "What might be hiding in here that's interesting?", and sometimes specific questions, such as "How did sales do during the last quarter?" or "What caused sales to take a dive last month?" As questions arise during the analytical process, these exploratory tools allow them to pursue each question immediately, without waiting for someone to build an application that is customized for the task.

## Custom Analytics Audiences

People often analyze information in routine ways, performing the same series of analytical tasks regularly as a part of their jobs. Having an analytical application that was designed specifically for these tasks makes the job easier and more efficient. Consequently, anyone who is familiar with the information that's being analyzed and the business processes it relates to can use a custom analytical application, even if that person's analytical skills don't extend to more open-ended, exploratory analysis. Customization removes (or tucks into some out-of-the-way corner) functionality that isn't relevant to the analytical objective or exceeds the analytical expertise of its users.

## Customizable Analytics Audiences

Developers (those who possess the skills that are needed to build custom applications for themselves or others) are the target audience for customizable tools. Depending on the tool and the nature of the application, this sometimes requires a high degree of programming skill and sometimes relatively little. Tools that require relatively little programming prowess usually replace the complexities of writing code with long lists of parameters, typically accessed through dialog boxes. Whether constructed with programming code or by wading through dialog boxes, these operations would get in the way of exploratory analysis, making it too onerous to pursue. Consequently, many tools that work fine for customizing analytical applications are not suited for exploratory analysis.

## Different Analytical Requirements

Let's take a look at these different analytical purposes and determine the features and functions that are required for each.

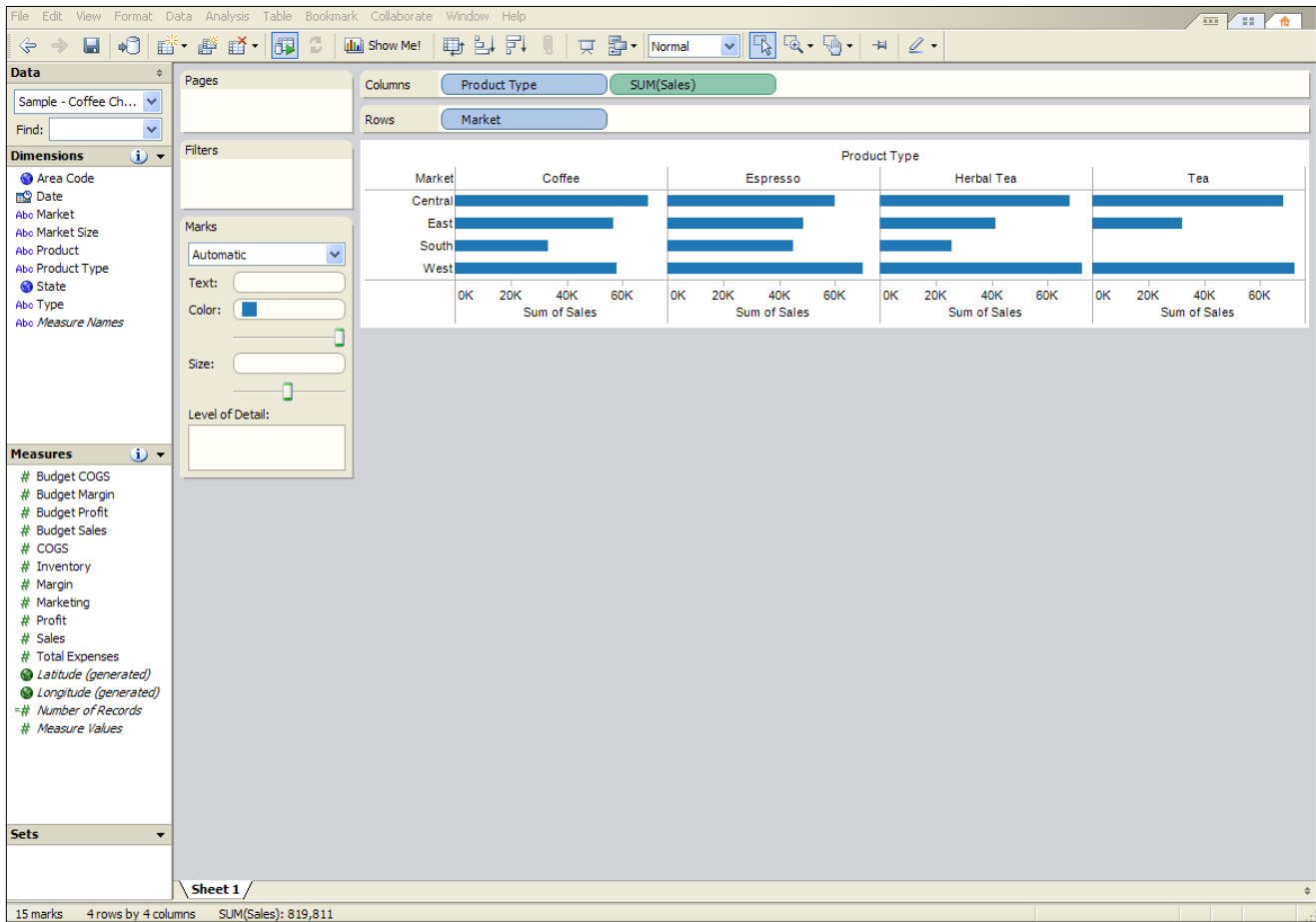
### Exploratory Analytics Requirements

Data exploration requires the ability to easily spot and examine everything that's potentially meaningful in a dataset with relative ease. The tool that we use should allow our minds to remain engaged in thinking about the data without distraction. In other words, ideally the tools should disappear from our awareness so our thoughts can remain focused on data sense-making, without distraction from the mechanics of working the tool.

The tool ideally exhibits the following characteristics:

- Provides every analytical display, interaction, and function that might be needed by those who use it for their analytical tasks.
- Grounds the entire analytical experience in a single, central workspace, with all displays, interactions, and functions within easy reach from there. (More about this is a moment.)
- Supports efficient, seamless transitions from one step to the next of the analytical process, even though the sequence and nature of those steps cannot be anticipated. (Note: This objective isn't achieved if the process is bogged down by lengthy dialog boxes of potential parameters or having to write programming code.)
- Doesn't require a lot of fiddling with things to whip them into shape to support your analytical needs (such as having to take time to carefully position and size graphs on the screen).

When exploring and making sense of data, we need a coherent cognitive space to hang our thoughts on—a desktop of sorts for organizing and navigating our thoughts, with everything that we need within reach. When forced to work with tools that fragment our thoughts into pieces scattered here and there among the clutter of a poor interface, we stumble around disoriented rather than staying focused in analytical pursuit. Perhaps the best example that I've seen of an effective analytical workspace is the one that [Tableau Software](#) provides (see the following page).



With this interface, the information that's being analyzed (coffee and tea sales by regional market in this example) is always front and center, and what we frequently need to explore and make sense of is close by. To manipulate data, we can often interact with the visual representation itself (the set of bar graphs in this case) to take action. For example, to remove a particular Market (Central, East, South, or West), we can click on it with the mouse and discard it from view by selecting "Exclude" from the pop-up menu. To add a field of data, we simply click and drag it from the dimensions or measures lists on the left into the display. The workspace is designed to keep us focused on the data, with little need to stray. No interface for data analysis is perfect, but those that have been designed with care to keep us focused on the data pave the way for exploratory analysis. If we must work too hard to get from one step to the next during the analytical process, we'll likely restrict our exploration to what's convenient and settle for the first answers to our questions that come to mind.

Other analytical tools that have been designed to support this exploratory process effectively, to varying degrees, include [Spotfire](#), [Panopticon Explorer](#), and [Advisor Analyst](#). Another tool that I like—[JMP](#) from SAS—doesn't yet match the coherent interface and seamless interaction provided by these others (however, it's heading this direction with its new Graph Builder), but it excels in statistical horsepower, which might counterbalance the extra effort if that's what you need.

### Custom Analytical Requirements

Routine analytics involving predictable tasks don't require a flexible tool. In fact, you won't want it to do anything that isn't needed, because unnecessary options just get in the way. You want the tool to do precisely what's needed in the most efficient and simplest way possible and nothing more. Whether you're a highly trained statistician who can take on any analytical challenge imaginable or someone who only knows how to perform a specific analytical task that's part of your job, when the task at hand is routine, your work is easier if the tool is customized for the task.

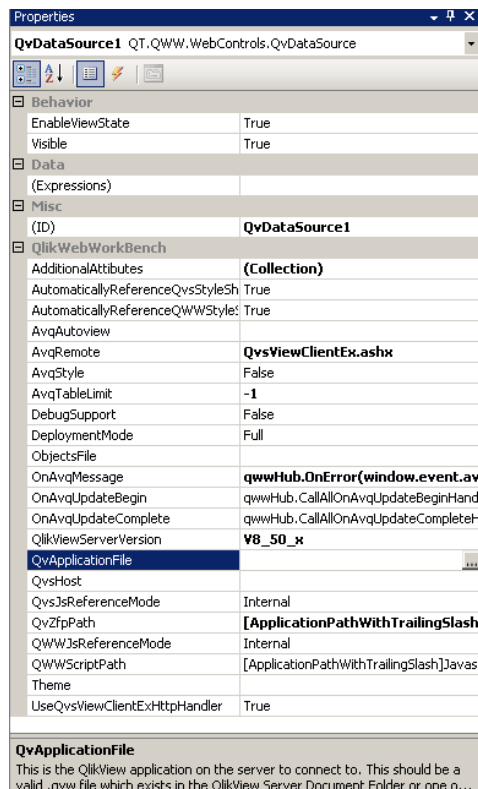
The question that remains is, “Do I build or buy?” There are times when it makes sense to buy an application that has already been customized for the task. If one’s available that does the job well and it’s priced considerably less than it would cost to have it built, it would be difficult to pass up. Just be sure that you’ve demonstrated for yourself, using your own two hands and brain, that it does the job well. On the other hand, if ready-made tools don’t cost considerably less than one you could build in-house (often the case when you have in-house experts in the analytical process, a good tool for building applications, and people who know how to use it), it makes sense to build something that does the job perfectly.

## Customizable Analytics Requirements

To build custom analytical applications, you need programming power. The tool ideally exhibits the following characteristics:

- Provides the means to develop an application that supports precisely what’s needed in the most effective way possible. This requires a high degree of programmability, both in terms of power and flexibility.
- Provides ready-made libraries of useful functions that can be easily plugged into the application with much less effort than it would take to build them from scratch.
- Easy and efficient to use by those who develop the applications.
- Provides the means to remove everything from view in the finished application that isn’t needed.
- Provides the means to guide the analyst step by step through the process.
- Provides the means to coach the user through the process with instructions and examples, as needed.

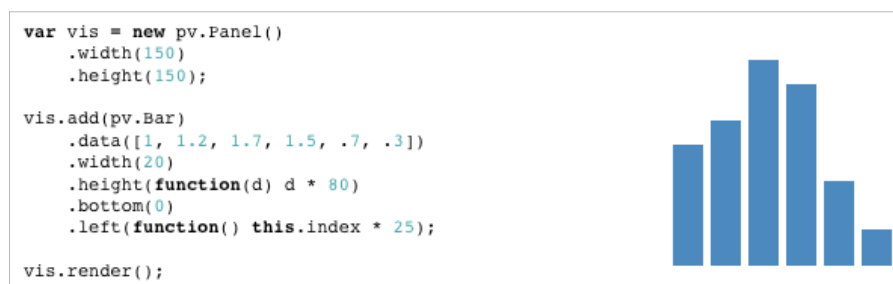
One of the products that I’ve seen that seems to do this fairly well is [QlikView](#). You don’t need to be a professional programmer to work with QlikView. Most of what you need exists as ready-made widgets (for example, particular charts with built-in functionality) that can be easily plugged into the developing application and much of the customization is done by selecting the appropriate parameters from lists that are found in dialog boxes. Programming code might need to be written, but it’s the exception, not the rule.



When you're developing a custom analytical application, you don't mind wading through lists of parameters in dialog boxes or writing a little code. Unlike the process of analysis itself when you must remain immersed in thinking about the data without distraction, these steps are less disruptive to developers. Although even developers benefit from programming interfaces that keep them focused on the task at hand, what they need most is the ability to do everything that's needed, precisely and efficiently. Writing code in this case isn't a distraction, it's the task itself.

Tools such as QlikView are often handy because they have much of the infrastructure that is often needed for data analysis built right into the product, relieving us of the task of creating it, which in some cases would be virtually impossible. For example, QlikView includes a powerful in-memory management infrastructure that makes it possible for data to be manipulated at extremely fast speeds. This is powerful, because when you move a slider control to filter 100,000 rows of data or you drill from the country to the state level, you want the results of that action to appear without delay.

In addition to products such as QlikView, which can be used fairly successfully by non-programmers, there are other tools that can be used to build analytical applications that require a greater degree of programming expertise. For example, Jeff Heer, an information visualization researcher and Assistant Professor at Stanford University, developed a visualization toolkit named [Prefuse](#), which extends the Java programming language. Also, a new tool named [Protovis](#) has recently been developed by Jeff Heer and Mike Bostock (a doctoral student at Stanford working under Heer's supervision), which provides similar functionality but requires less programming skill.



With the flexibility of full-fledged programming languages, tools such as these can sometimes get you closer to precisely what you need.

Several statistical analysis tools include rich programming capabilities as well, such as products from SAS and the open source product “R”. Whereas tools such as Prefuse and Protovis focus on visualization capabilities, these tools focus primarily on statistical capabilities, but support flexible charting capabilities as well.

No product has been used to develop custom analytical applications more than Microsoft Excel. An entire industry has grown up around Excel, resulting in what probably totals in the thousands of custom applications and add-in products designed to extend its functionality beyond that of a basic electronic spreadsheet. These range from simple customization through the use of macros to complex applications that rely on a great deal of programming code (mostly Visual Basic) to take Excel places it could never go on its own. It's amazing to see some of the monuments to creative programming that have been constructed using Excel as the foundation. I know from working with some of the greatest Excel programming talent in the world that building applications on top of Excel can be arduous and painful. Various versions of Excel work differently under the hood, which complicates the process. Excel's *application programming interface* (API), which allows other programs to interact with it, permits some things and restricts others, often in annoyingly arbitrary ways, sometimes inconsistently from version to version.

Despite these challenges, because most people who work with data already have a copy of Excel and it's familiar to them, it provides a sensible foundation for many custom applications, but it unfortunately imposes considerable limitations on applications that are used for data analysis. Excel is so mired in the spreadsheet paradigm, it lacks many of the data visualization and data interaction capabilities that are fundamental to analysis. Features that are common in a visual analysis products like Tableau, Spotfire, Panopticon, Advizor Analyst, and JMP, such as slider controls for dynamically filtering data based on quantitative ranges (for example, annual sales revenues between one and two millions dollars) would be difficult to implement in Excel.

Before using Excel as the platform for a custom application, you should take a look at a good visual analysis tool. You'll discover an entirely new world of analytical possibility that will leave Excel looking anemic.

Fortunately, some of the tools that work well for exploratory analytics (Tableau, Spotfire, etc.) can also be used to build most of the custom analytical applications that organizations typically need. Although these tools might lack the extensive customization potential of a full-fledged programming language, they can often make it easy and efficient to implement all the features that you need. By sequencing the steps of a routine analytical process into separate folders (that is, separate screens within the application), with clear instructions appearing right there on the screen for each, and built-in actions to automatically move you through the process based on the choices that you make, these tools are often perfect for organizations that need both exploratory analytics and the ability to customize applications for specific analytical processes.

## Final Thoughts

Whenever you're getting ready to purchase something expensive, it pays to do your homework. Analytical software tools certainly fall into this category. Because they're designed for various purposes—exploratory, custom, or customizable analytics—it makes sense to understand your own purposes before shopping, and to then limit your list of candidates to those that serve your purposes and serve them well.

If you're a software vendor that makes and sells analytical tools, you should understand that different features and interfaces are needed for different analytical purposes. You should define your target user audience clearly in terms of these purposes and design your products to support them effectively.

---

## Discuss this Article

Share your thoughts about this article by visiting the [Fundamental Differences in Analytical Tools: Exploratory, Custom, or Customizable](#) thread in our discussion forum.

---

## About the Author

Stephen Few has worked for over 25 years as an IT innovator, consultant, and teacher. Today, as Principal of the consultancy Perceptual Edge, Stephen focuses on data visualization for analyzing and communicating quantitative business information. He provides training and consulting services, writes the bi-monthly [Visual Business Intelligence Newsletter](#), speaks frequently at conferences, and teaches in the MBA program at the University of California, Berkeley. He is the author of three books: *Show Me the Numbers: Designing Tables and Graphs to Enlighten*, *Information Dashboard Design: The Effective Visual Communication of Data*, and *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. You can learn more about Stephen's work and access an entire [library](#) of articles at [www.perceptualedge.com](http://www.perceptualedge.com). Between articles, you can read Stephen's thoughts on the industry in his [blog](#).