

5 ways to approach custom visualisation in Power BI

Many business-oriented data visualisation needs can and should be met with judicious use of standard visual types – tables & matrix visuals, KPI cards, bar charts, line charts etc are our default go-to reporting tools because they are highly adaptable and easy to read.

In fact, my typical reaction to business stakeholders requiring something bespoke is to ask “Why do you need to go outside the standard data visualisation toolset to represent this data? What is the outcome you are looking for and should we be looking for a less complex way to get there?”

Non-standard visuals require a learning curve to read and more often than not they come with development overhead that makes them not worthwhile.

But custom visualisation can't be completely ignored. Eventually we all run into a scenario where we absolutely need to develop our own visualisation solution, for example:

1. Like-for-like replacement of existing non-standard report visuals
2. Industry specific renderings for operational data
3. Infographic style displays

Sooner or later, we all most likely will encounter situations where we need to go custom, as the standard Power BI tools and the marketplace visuals won't help us. Fortunately, there has never been so many approaches available to us to bridge this gap.

Here are 5 popular approaches to custom visualisation used today to address the ever-evolving requirements of our increasingly data-driven world.

1. Write it from scratch

[Building from scratch](#) is the most direct way to create your own bespoke visual. Using developer tools to create a .pbviz custom visual file is still the most powerful option available. There are many online tutorials available to help you get started and in theory you are limited only by your imagination as to what you can create.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users> pbiviz

+syys+/
oms/+osyhdhyso/
ym/      /+oshddhys+/
ym/      /+oyhddhyo+/
ym/      /osyhdho
ym/
ym/      yddy      sm+
ym/      shho /mmm/  om+
/      oys/ +mmmm /mmm/  om+
oso ommmh +mmmm /mmm/  om+
ymmy smmmh +mmmm /mmm/  om+
ymmy smmmh +mmmm /mmm/  om+
ymmy smmmh +mmmm /mmm/  om+
+dmd+ smmmh +mmmm /mmm/  om+
      /hmdo +mmmm /mmm/ /so//ym/
      /dmmh /mmm/ /osyhhy/
      //      dmm
      ++

PowerBI Custom Visual Tool

Usage: pbiviz [options] [command]

Options:
-V, --version      output the version number
--install-cert    Creates and installs localhost certificate
-h, --help        output usage information

Commands:
new [name]        Create a new visual
info             Display info about the current visual
start           Start the current visual
package         Package the current visual into a pbiviz file
update [version] Updates the api definitions and schemas in the current visual. Changes the version if specified
help [cmd]       display help for [cmd]

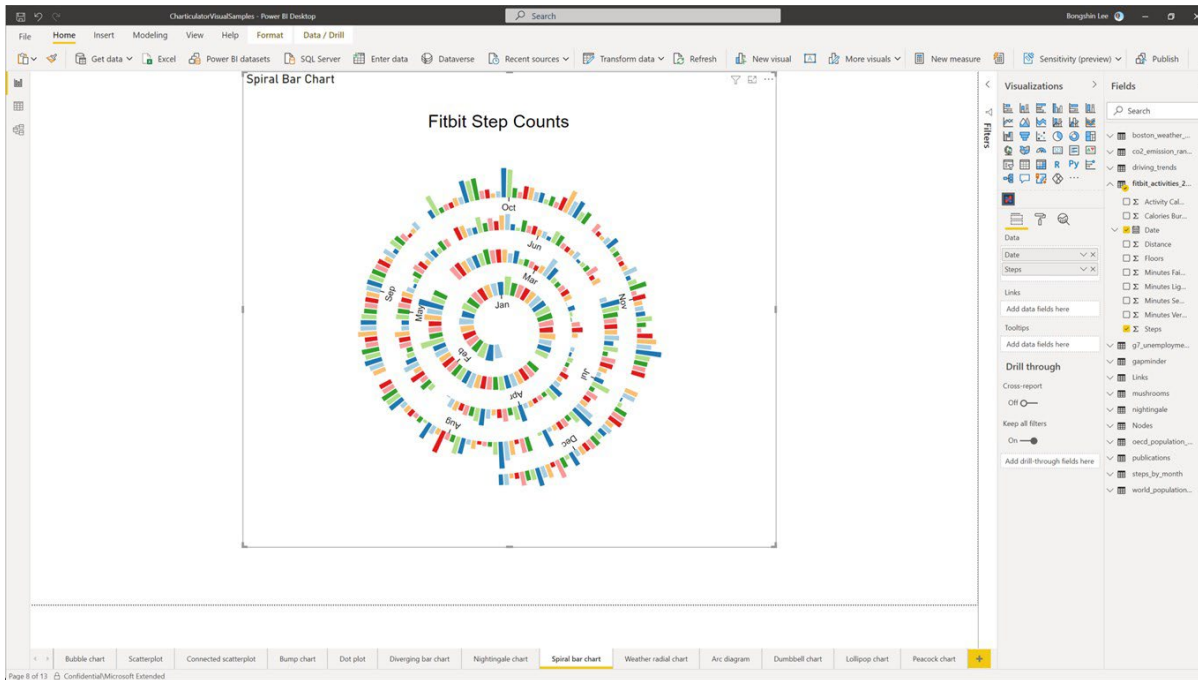
PS C:\Users>
```

However, custom visual development has a steep learning curve, and it will take a significant setup, coding, testing and ongoing support effort to maintain, even for a relatively simple visual. This is especially true for people who don't come from a coding background.

2. Charticulator

[Charticulator](#) is a Microsoft solution for building custom visuals. Historically, charticulator was a web app that you could use to build and export your visual to a .pbiviz file which could be imported into Power BI and used to render your data. This was essentially a shortcut to building from-scratch, within the scope of what Charticulator can do.

Charticulator has changed recently, and the old web-based pbiviz creator is gone. Charticulator is now a custom visual within Power BI. The same web interface designer is used in the same ways to create your visuals, but (instead of building online in the web app and exporting to a file) you now build and run your visual directly within the Charticulator visual in Power BI.



Charticulator is a great option for low-code custom visual development and as such it opens up a range of possibilities for novice developers. Pre-built templates offer a jumping off point and now that the visual is integrated directly into the Power BI environment, concerns about longer-term support for your visual through Power BI version changes are resolved.

3. R & Python visuals

Statistics gurus will be familiar with R and Python, the open-source data and statistics scripting languages that have risen to prominence as an alternatives to minitab and other such desktop statistics tools. Power BI has custom visuals that allow users to code their own solutions and render them to the screen using these two scripting packages.

Within the vast libraries of R is a graphing tool called ggplot2. Microsoft bundles an R custom visual that can integrate R scripting with your Power BI report and render custom visualisations with ggplot2. Similarly, Python developers may use the Py libraries to render their ideas directly into the report.

Since both are widely used statistics solutions, there are many freely available scripts out there for you to start with. If you want to create statistics-oriented visualisations from your data, this (in particular R) may be the option for you. Once you have installed R or Python on your PC, you will be able to use the visuals and code directly to them in Power BI.

However, it should be noted that there are limitations. Most importantly, you will not be able to use R or Python visuals to cross-filter native Power BI visuals or drill through to other pages. There are also limits to how much data can be processed, making it unsuitable for processing large, granular data sources.

[Custom Visualisation with R](#)

[Custom Visualisation with Python](#)

4. PlotlyJS

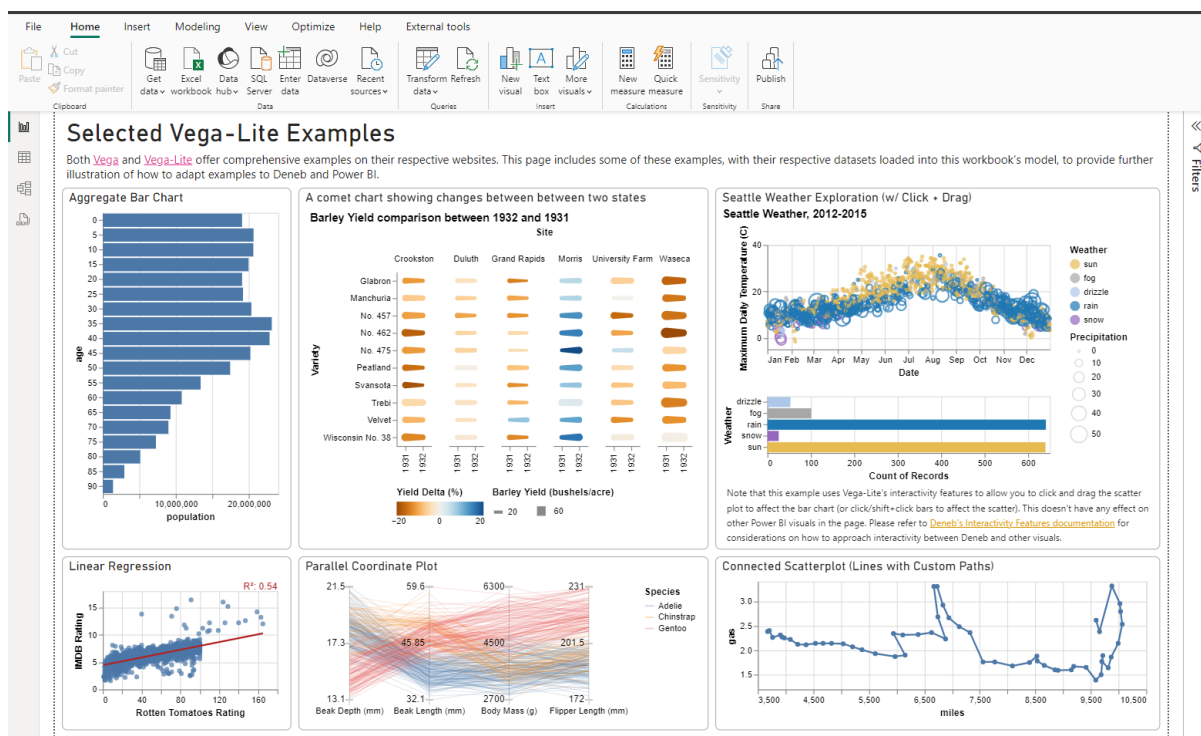
[Plotly.js](#) visual by Akvelon combines Plotly.js, Chart Studio and Power BI together. It allows users to visualize data using the capabilities of plotly.js and provides a chart editor that helps users to create and customize over 40 types of charts in Power BI. You are however limited to these chart types, which may not suite everyone.

5. Deneb

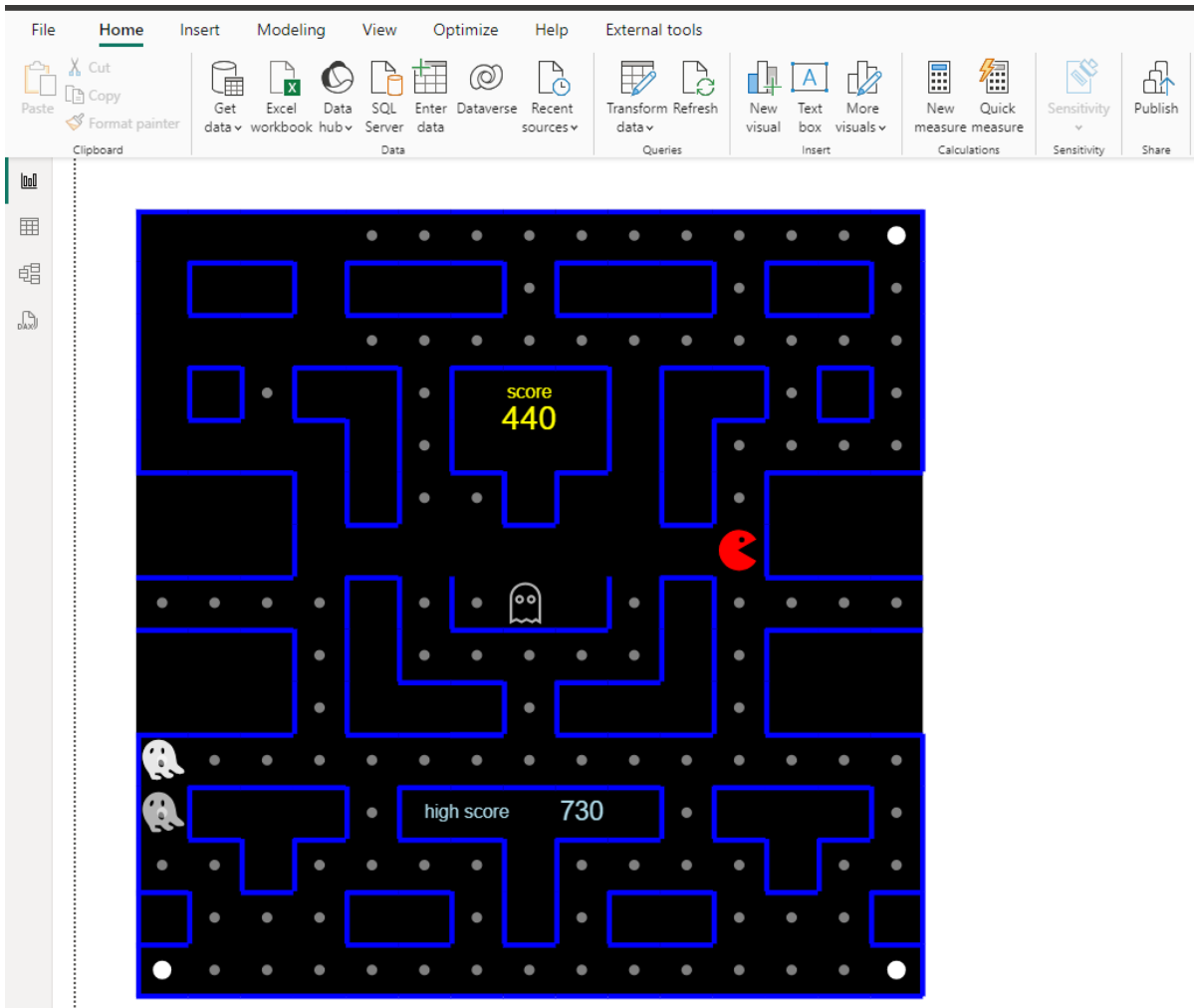
[Deneb](#) has become my personal go-to for custom visualisation in Power BI. Written by data visualisation guru and Microsoft MVP [Daniel Marsh-Patrick](#), Deneb enables you to write fully interactive visuals using either Vega or Vega-lite JSON scripting!

Visuals built with Deneb are fully integrated into the Power BI slice, dice & drill operations – meaning your end users will have a fully seamless experience operating your custom work along-side and with native Power BI controls.

[Veta-lite](#) is a simplified JSON specification that utilises the more common chart types as a jumping off point for creating your own highly customisable visuals. If you want to build, say a customised version of a waterfall chart with features not supported in standard Power BI visuals or marketplace offerings, vega-lite offers a streamlined way to do this fast.



[Vega](#) (full) on the other hand is no-holes-barred, fully customisable and capable of creating app-like experiences for your end users. While it presents a steeper learning curve, it is one well worth the journey, as solutions built with the full Vega specification are capable in ways otherwise only matched by complex from-scratch pbviz development. For an extreme example, it is possible to build complete games in Power BI using Vega in a Deneb visual, such as a Pacman example from the Vega template library.



Conclusion

While using custom visualisations is not always desirable from an end-user perspective, it is not something we can get away from. When the time comes for an appropriately customised offering however, there has never been more options open to Power BI users.

Are you struggling with custom requirement problems and no one else can help? [Email us](#) today or call **03 9924 3000** to discuss how one of our Power BI specialists can help.

Contact

Fujitsu Data & AI
+61 3 9924 3000

© Fujitsu 2022. All rights reserved. Fujitsu and Fujitsu logo are trademarks of Fujitsu Limited registered in many jurisdictions worldwide. Other product, service and company names mentioned herein may be trademarks of Fujitsu or other companies. This document is current as of the initial date of publication and subject to be changed by Fujitsu without notice. This material is provided for information purposes only and Fujitsu assumes no liability related to its use.