



Power[R] BI

Enhance your
Microsoft Power BI
experience with R



As a specialized, open source statistical environment, R represents the primary analysis language for a large number of data scientists and statisticians. In recent years, R has also undergone a significant shift in user base by gaining wider adoption in the business world.

By extending Power BI with R, Microsoft has opened up numerous opportunities to enhance your Business Intelligence solutions. In addition to its versatility for data science, R is a great language and ecosystem for work related to both data visualization and data processing. By incorporating R into its products, Microsoft has signaled a strong commitment from Microsoft not only to data science, but the R platform in general.



Microsoft began incorporating R into Power BI in December 2015 and has been adding additional functionality ever since!

Over the following pages, you will explore **three ways** to use R in Microsoft Power BI:

1. **R-powered Custom Visuals** – utilize pre-built R Custom Visuals
2. **R Visuals** – visualize your data with full control over your R code
3. **R Scripts** – use R in your data preparation

R-powered Custom Visuals

R-powered visuals work like any other Power BI *Custom Visual*. Apart from requiring a local connection to R from Power BI Desktop, the use of R in Power BI's R-powered visuals is hidden from users. They allow users to take advantage of R without knowing R or even seeing any R code.



If you would like to easily use advanced R visualizations, you can use what other developers have created. Examples include plotting the correlation between different fields, different types of clustering, advanced forecasting, and more.

Developers use the same tools to create R-powered visuals as they do for standard Power BI *Custom Visuals*. In fact, most of the code used to create R-powered visuals often will not be R. It will be TypeScript! Anyone who wants to create R-powered visuals embeds the R code to visualize data within the Custom Visual framework. Additional code for handling data and format options utilizes TypeScript.

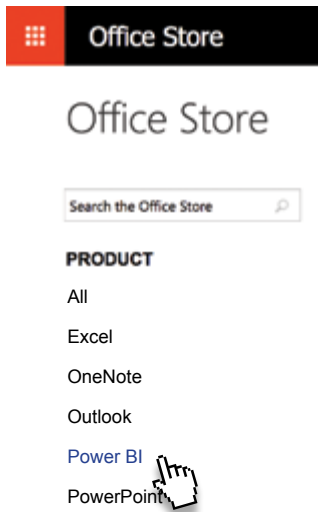
Advantages of using R-powered visuals include:

- R-powered visuals are good options for teams who seek a quick visual solution with the same level of ease as any other Power BI visual.
- You can unlock the power of R without knowing R.

Disadvantages of using R-powered visuals include:

- Additional customization is limited since you download and import a visual created by another developer. You do not have direct control over the R code to change the visual's appearance. The R code that drives the visual is bundled with Power BI's .pbiviz file and is not directly exposed to users. Some visuals may include formatting options in the Format tab, but it is at the discretion of the R-powered Visual developer.
- Unlike native visuals or custom visuals, R-powered visuals display static images. The R-powered visual will interact with other visuals that you may include on the report page, but no interaction is available in the R-powered visual itself. Features such as visual highlighting or using slicers on the report canvas will affect the appearance of R-powered visuals, but clicking on R visuals will not update the R visual or update other visuals on-screen.
- Tooltips are not available.

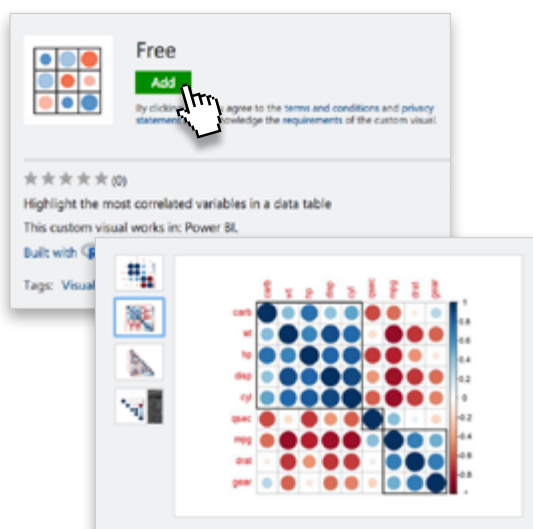
Acquiring R-powered visuals



Obtain R-powered visuals from the Office Store as you would with any Power BI Custom Visual. Navigate to <https://store.office.com> and find **Power BI** in the menu.



R-powered Visuals will display **Built with R** with the R logo next to it. For example, see the difference between the Histogram Chart and the Correlation Plot. The Histogram Chart is a standard custom visual while the Correlation Plot is R-powered.



After finding your desired R-powered Visual in the Office Store, select it. A new page appears that shows an **Add** button to download the visual.

Visuals also have thumbnails and screenshots that provide a quick glimpse of what you might see in your reports. Some visuals even have videos describing their use.


Acquiring R-powered visuals ...continued

After you click **Add**, you should see **two** options.

Step 1: Download Correlation plot

Select to download Correlation plot

ⓘ This is a custom visual built with R. If you open this custom visual in Power BI Desktop, you may need to install additional R packages and/or an R engine.
[Learn more about R visuals](#)


 PowerBI-visuals-corrplot.1.0.1.0.pbviz

On the left will be an option to **Select to Download**. After clicking that, your browser may prompt you to download a **PBIVIZ file**, or it may automatically start downloading to your default download folder.

Try a sample

Try a sample report to see how to utilize this visual.

You can also [download the sample report](#).

 PowerBI-visuals-corrplot.1.0.1.0.pbix

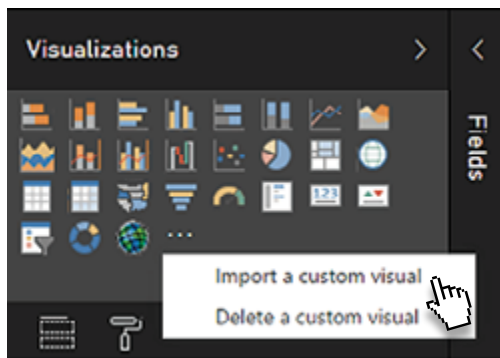
To the right, you may see an option to **Try a Sample**. If you select this option, you can download a **PBIX** report file, which you can open in Power BI Desktop.

Using R-powered Visuals in Power BI Desktop

To use Power BI's *R-powered Visuals* in **Power BI Desktop**, make sure that you have a version of R installed on your workstation. You can download R from <https://www.r-project.org>, or you can use Microsoft R Open and download it from <https://mran.microsoft.com/download>.



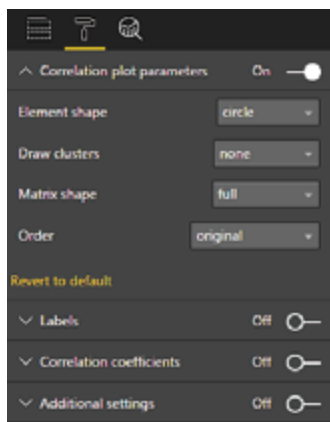
Power BI Desktop should automatically locate your local R installation, but you can be certain by visiting **File → Options** and **Settings → Options → R scripting**.



In the **Visualizations** area, click the [...] icon and select **Import a custom visual** and browse for the **pbiviz** file. A new icon will appear under Visualizations that corresponds to the new R-powered Visual.



Continuing with the Correlation Plot example, the visual will display on your report after you add data.



There are often numerous format options available with R-powered visuals. Format options differ for each visual. Some may offer a wealth of customization while others may offer few options.

Using R-powered Visuals in Power BI Service

To use R-powered visuals in **Power BI Service** at <https://www.powerbi.com>, you do not need R installed on your local workstation. Simply import the appropriate **pbiviz** file in the **Visualization** area as you do in Desktop. Click the [...] icon, select **Import a custom visual**, and browse for the **pbiviz** file.

Building R-powered Visuals

If you are interested in building *R-powered Visuals*, see the guidance from Microsoft's Custom Visual development team on **GitHub**. Navigate to <https://github.com/Microsoft/PowerBI-visuals> and find the section pertaining to **Building R-Powered Custom Visuals**. As previously mentioned, much of the code is not R but TypeScript, so skills useful in building standard Power BI Custom Visuals overlap heavily with building R-powered Visuals.

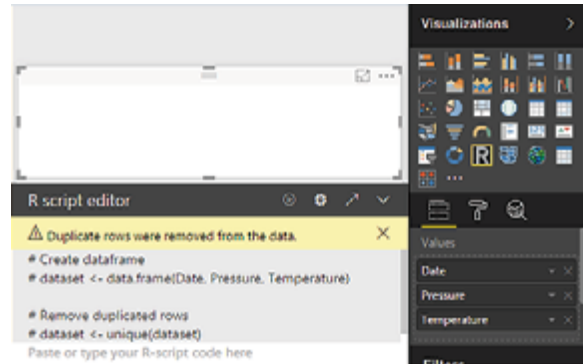
Building R Powered Custom Visual (corrplot)

1. [Creating a new R Powered Custom Visual](#)
2. [Starting a simple R Script](#)
3. [Adding a static property to the property pane](#)
4. [Validate input data in R Script](#)

R Visual

What if you want to do more with R than what the R-powered Visuals allow? In this case, use the built-in **R Visual**. If you have experience plotting with R or are inspired by the ability to expand Power BI's native data visualization capabilities, the R Visual is for you.

Working with the R Visual is a different experience than using other Power BI visuals. Adding the R Visual to the report canvas brings up a Script Editor. After you add data to **Values**, that data becomes available to use in your R Visual's script. The R Visual is completely based on code that you provide. As a result, there are many opportunities to enrich your reports with custom-created plots.



- ◀ If you are seeking inspiration, try starting with **Microsoft's R Script Showcase**. Navigate to <https://community.powerbi.com/t5/R-Script-Showcase/bd-p/RVisuals> and select from the many sample scripts. Use these scripts as a launchpad for your own work.



Advantages of using R Visuals include:

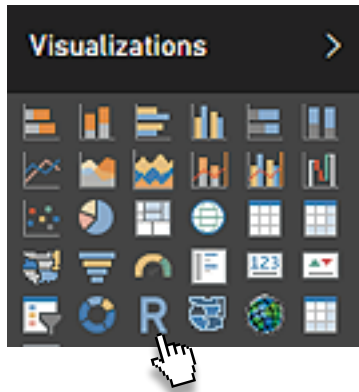
- Expand Power BI's native capabilities by using existing R code for data visualization.
- Unlike with **R-powered Visuals** you download from the Office Store, you have full control over your R code and final output with the R Visual.

Disadvantages of using R Visuals include:

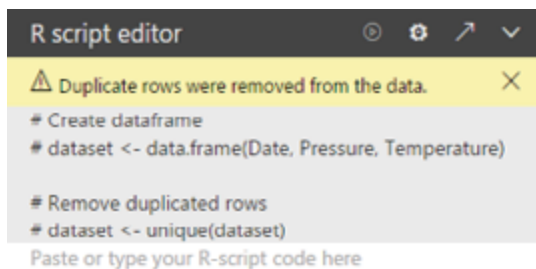
- As with R-powered visuals, self-coded R visuals display static images. The R Visual will interact with other visuals that you may include on the report page, but no interaction is available in the R Visual itself. Features such as visual highlighting or using slicers on the report canvas will affect the appearance of R Visuals, but clicking on R Visuals will not update the R Visual or update other visuals on-screen.
- Tooltips are not available.
- The R Visual currently has a limit of **150,000 records**. If your dataset includes more data than allowed, you could attempt alternative ways of visualizing that data.
- R Visuals do not display in reports that have been distributed using **Publish to Web**.
- Power BI Service has a limited number of R packages that Microsoft deploys and manages. While you can take advantage of common visualization packages like **ggplot2**, you cannot install additional packages for online use like you can when using Power BI Desktop.
- Visuals that take longer to execute than 60 seconds in Power BI Service or 5 minutes in Power BI Desktop will timeout with an error.

Using R Visuals in Power BI Desktop

As with R-powered Visuals, Power BI Desktop requires a local version of R to render R Visuals. You can download R from <https://www.r-project.org>, or you can use Microsoft R Open and download it from <https://mran.microsoft.com/download>. Power BI Desktop should automatically locate your local R installation and any default R development environment. There is no additional configuration necessary to use a local version of R. For example, you do not need to run a special package such as Rserve to access R from Power BI. If needed, however, you may configure a setting if you have multiple versions of R installed locally and want to change which version Power BI uses.

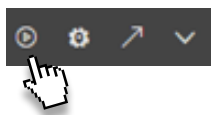


The R Visual appears in the list of **Visualizations** with a blue **R** icon. After adding the R Visual to your report, you should see a message encouraging you to drag fields into the **Values** area.



Once you add your data, Power BI prepares a **data frame**, which is a two-dimensional R data structure consisting of rows and columns. In your subsequent code, reference the name of Power BI's initial data frame, which defaults to **dataset** automatically.

```
library(corrplot)
M <- cor(dataset)
corrplot(M)
```

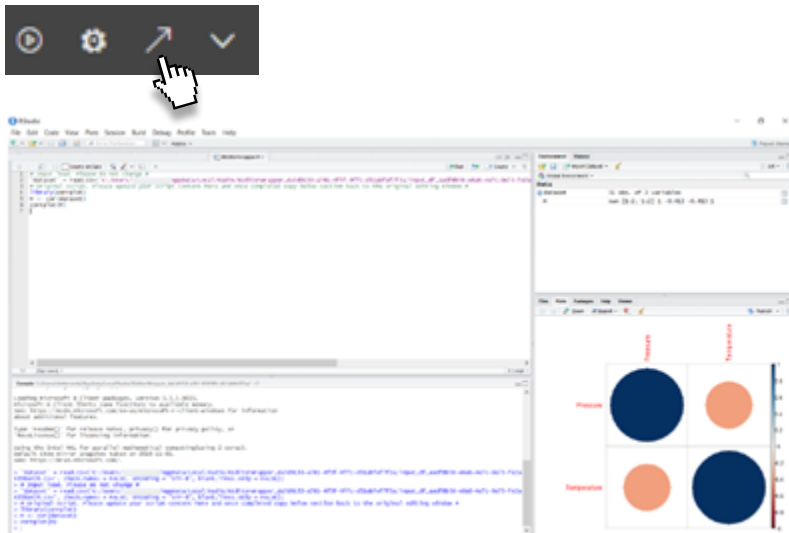


As an example, you could add a correlation plot similar to the one provided by the R-powered Visual of the same name. Provide the code to the R Visual and click the **Run Script** icon in the top right corner. **Run Script** looks like a play button and appears as the leftmost icon.

Using R Visuals in Power BI Desktop ...continued



Here is how everything would appear together on-screen:



Power BI also provides the option to edit scripts in a local R integrated development environment (IDE). Use an IDE such as RStudio or R Tools for Visual Studio by selecting the **Edit Script in External R IDE** icon in the top right corner. It appears as an arrow.

Using R Visuals in Power BI Service

Using R Visuals in the cloud with Power BI Service provides some of the same advantages of Power BI Desktop.

To work with R in Power BI Service, users will need a **Power BI Pro** subscription with a **managed tenant**. With a Pro license, R Visuals should render in your reports as well as any dashboard tile to which you have pinned your R Visual.

Unlike with Power BI Desktop, there are some additional limitations when using the R Visual in Power BI Service. The most noticeable constraint is which R packages are supported. If you use R in Power BI Desktop and need a package that is not currently available to your workspace, you can simply install it. With Power BI Service, however, there is no way to install additional packages, but you can request them from the Power BI team for potential future inclusion. Microsoft has excellent documentation on what packages it supports in Power BI Service. You can find more detail about what packages are supported by navigating to <https://powerbi.microsoft.com/en-us/documentation/powerbi-service-r-packages-support/>.

R Script

In addition to using R for data visualization, Power BI can execute R scripts for data processing tasks. Complementing its native **Power Query / M** functionality, Power BI includes a **Run R Script** option. While Power BI natively handles many advanced data processing tasks, users may still prefer using R Scripts for a few different reasons. These may include performing advanced analytics with R or overcoming feature gaps such as Power BI's lack of regular expressions.

Creating R Scripts requires Power BI Desktop. Reports that use R Scripts can be deployed to Power BI Service, but it requires the additional use of the Power BI *Personal Gateway* to refresh R Scripts that reference on-premises data sources.

As with R Visuals, a local copy of R is required to use R Scripts.

Advantages of using R Scripts include:

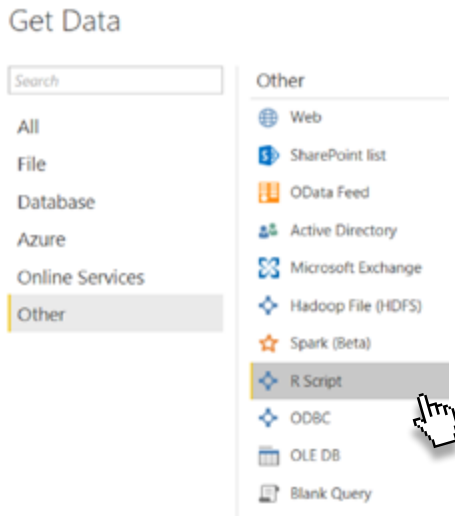
- Import data from sources not currently supported by Power BI Desktop.
- Enrich data directly on the dataset without having to work outside of Power BI.

Disadvantages of using R Scripts include:

- Data must exist as a data frame.
- Data source privacy levels must be set to **Public**.
- Scripts that take longer than 30 minutes will time out.

Using the R Script to Import or Modify Data

R Script appears as an option in **Get Data**. While Power BI has many data connectors, R may help overcome some issues getting data from specific formats. For example, if a user has data stored in format used by statistical applications such as SAS or SPSS, it's not easy to get to that data using native Power BI connectors. It is easy to import it using the R Script option.

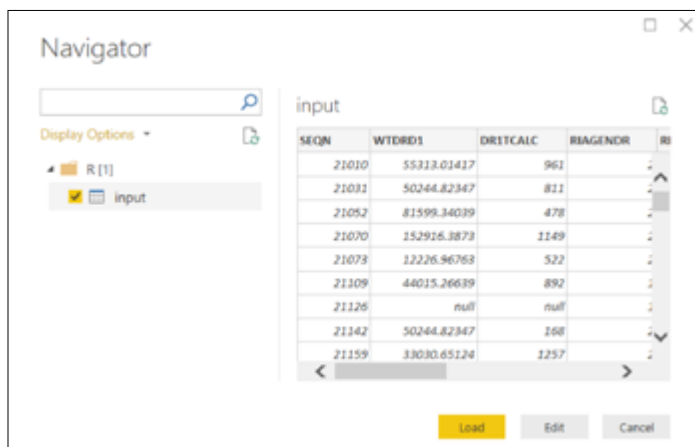


After selecting **Get Data**, simply find the **Other** section and connect using the **R Script** option.



Simply paste a script into the **Execute R Script** window and it's ready. While this example uses data from a file, you could use any source that R connects to such as data online or in a database.

Your scripts could also combine import and additional data processing using R, or you could simply import data using R and continue to wrangle it if needed using Power BI.



After selecting **input** and viewing the initial preview, you may **Load** the data directly as it appears, or you can **Edit** and continue to process the data with Power BI.

Using the R Script to Import or Modify Data ...continued

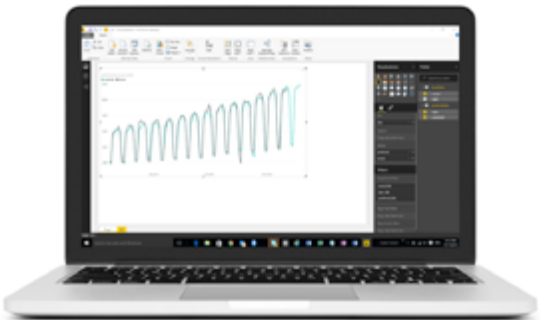
In addition to loading data, the R Script works downstream. You can load data with Power BI's native connectors to get data from numerous sources that it would be difficult to obtain with R (such as Microsoft Azure connectors). Once you've connected to that data source and brought data in, however, you can still enrich that data with R.



In the **Transform** tab, the **Run R Script** option allows you to execute scripts on existing queries. Want to do additional data preparation that is difficult or impossible to do in Power BI (such as use regular expressions)? Want to perform machine learning tasks such as clustering or classification? With R in Power BI, it's all at your fingertips!

Check out BlueGranite's Blog for even more ideas on how to get data using R!

[READ POST →](#)



Final Thoughts

The abundant ways to use R in Power BI help enhance your data and empower your visualizations. Whether you have **broad** experience or **no** experience with R, incorporate R into your reports. Whoever your audience, you can tap into the "Power of R in Power BI".

Interested in More?

BlueGranite has a quick tutorial on using both the R Script and R Visual in Microsoft Power BI. Check it out for free online to work through the hands-on example!

[VIEW TUTORIAL →](#)