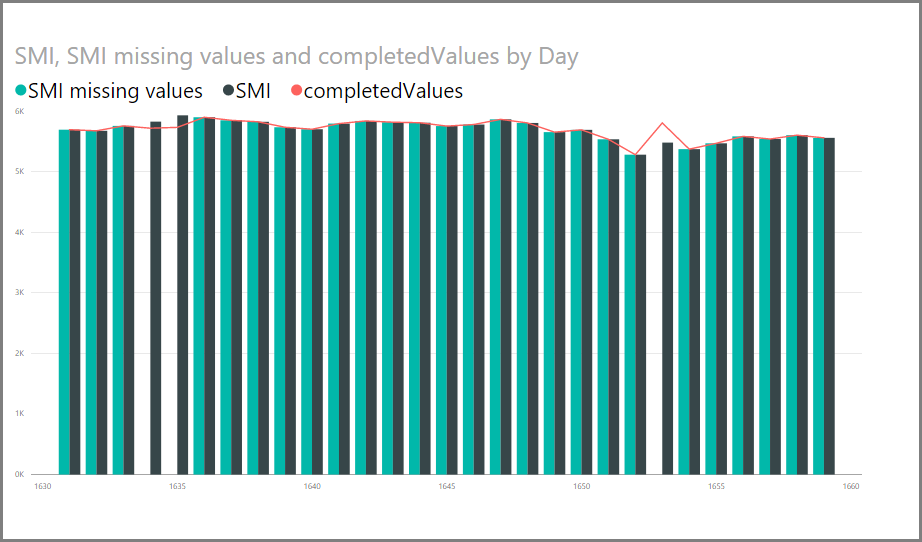
Creating visuals from R script data

Now we can create a visual to see how the R script code using the *mice* library completed the missing values, as shown in the following image:



Once that visual is complete, and any other visuals you might want to create using **Power BI Desktop**, you can save the **Power BI Desktop** file (which saves as a .pbix file) and then use the data model, including the R scripts that are part of it, in the Power BI service.

Note

Want to see a completed .pbix file with these steps completed? You're in luck - you can download the completed **Power BI Desktop** file used in these examples [right here](http://download.microsoft.com/download/F/8/A/F8AA9DC9-8545-4AAE-9305-27AD1D01DC03/Complete%20Values%20with%20R%20in%20PQ.pbix).

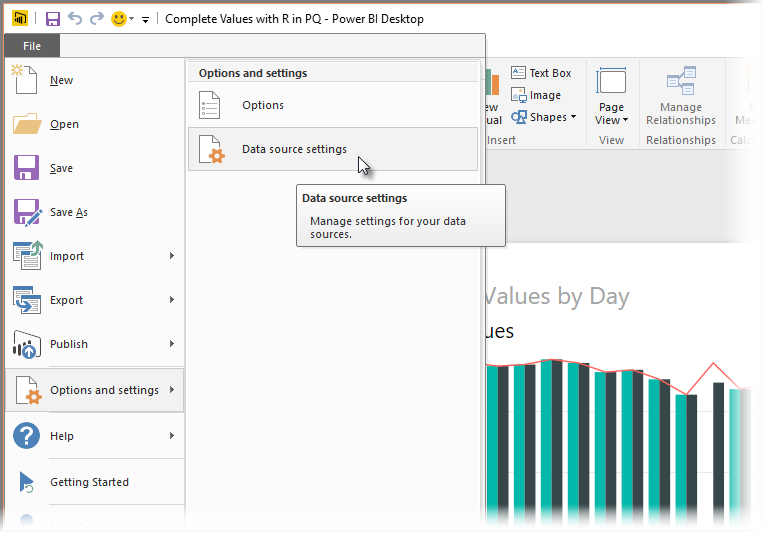
Once you've uploaded the .pbix file to the Power BI service, a couple more steps are necessary to enable data refresh (in the service) and to enable visuals to be updated in the service (the data needs access to R for visuals to be updated). The additional steps are the following:

* **Enable scheduled refresh for the dataset** - to enable scheduled refresh for the workbook that contains your dataset with R scripts, see [Configuring scheduled refresh](https://docs.microsoft.com/en-us/power-bi/refresh-scheduled-refresh), which also includes information about **Personal Gateway**.
* **Install the Personal Gateway** - you need a **Personal Gateway** installed on the machine where the file is located, and where R is installed; the Power BI service must access that workbook and re-render any updated visuals. You can get more information on how to [install and configure Personal Gateway](https://docs.microsoft.com/en-us/power-bi/personal-gateway).

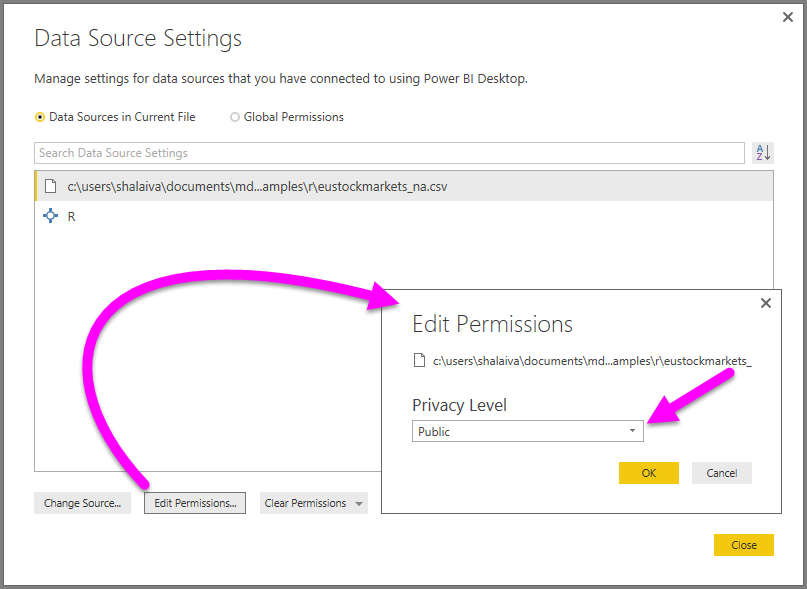
Limitations

There are some limitations to queries that include R scripts created in **Query Editor**:

* All R data source settings must be set to *Public*, and all other steps in a query created in **Query Editor** must also be public. To get to data source settings, in **Power BI Desktop** select **File > Options and settings > Data source settings**.



From the **Data Source Settings** dialog, select the data source(s) and then select **Edit Permissions...** and ensure that the **Privacy Level** is set to *Public*.



* To enable scheduled refresh of your R visuals or dataset, you need to enable **Scheduled refresh** and have a **Personal Gateway** installed on the computer that houses the workbook and the R installation. For more information on both, see the previous section in this article, which provides links to learn more about each.

There are all sorts of things you can do with R and custom queries, so explore and shape your data just the way you want it to appear.

## Run R scripts

With just a few steps in Power BI Desktop, you can run R scripts and create a data model, from which you can create reports, and share them on the Power BI service. R scripting in Power BI Desktop now supports number formats that contain decimals (.) and commas (,).

### **Prepare an R script**

To run an R script in Power BI Desktop, create the script in your local R development environment, and make sure it runs successfully.

To run the script in Power BI Desktop, make sure the script runs successfully in a new and unmodified workspace. This means that all packages and dependencies must be explicitly loaded and run. You can use source() to run dependent scripts.

When preparing and running an R script in Power BI Desktop, there are a few limitations:

* Only data frames are imported, so make sure the data you want to import to Power BI is represented in a data frame
* Columns that are typed as Complex and Vector are not imported, and are replaced with error values in the created table
* Values that are N/A are translated to NULL values in Power BI Desktop
* Any R script that runs longer than 30 minutes times out
* Interactive calls in the R script, such as waiting for user input, halts the script’s execution
* When setting the working directory within the R script, you must define a full path to the working directory, rather than a relative path

### **Run your R script and import data**

1. In Power BI Desktop, the R Script data connector is found in **Get Data**. To run your R Script, select **Get Data > More...**, then select **Other > R script** as shown in the following image:
2. If R is installed on your local machine, the latest installed version is selected as your R engine. Simply copy your script into the script window and select **OK**.
3. If R is not installed, is not identified, or if there are multiple installations on your local machine, expand **R Installation Settings** to display installation options, or to select which installation you want to run the R script.

If R is installed and is not identified, you can explicitly provide its location in the text box provided when you expand **R Installation Settings**. In the above image, the path C:\Program Files\R\R-3.2.0 is explicitly provided in the text box.

R installation settings are centrally located in the R Scripting section of the Options dialog. To specify your R installation settings, select **File > Options and settings** and then **Options > R scripting**. If multiple installations of R are available, a drop-down menu appears that allows you to select which installation to use.

1. Select **OK** to run the R Script. When the script runs successfully, you can then choose the resulting data frames to add to the Power BI model.

### **Refresh**

You can refresh an R script in Power BI Desktop. When you refresh an R script, Power BI Desktop runs the R script again in the Power BI Desktop environment.

Creating visuals from R script data

Now we can create a visual to see how the R script code using the *mice* library completed the missing values, as shown in the following image:

Once that visual is complete, and any other visuals you might want to create using **Power BI Desktop**, you can save the **Power BI Desktop** file (which saves as a .pbix file) and then use the data model, including the R scripts that are part of it, in the Power BI service.

Note

Want to see a completed .pbix file with these steps completed? You're in luck - you can download the completed **Power BI Desktop** file used in these examples [right here](http://download.microsoft.com/download/F/8/A/F8AA9DC9-8545-4AAE-9305-27AD1D01DC03/Complete%20Values%20with%20R%20in%20PQ.pbix).

Once you've uploaded the .pbix file to the Power BI service, a couple more steps are necessary to enable data refresh (in the service) and to enable visuals to be updated in the service (the data needs access to R for visuals to be updated). The additional steps are the following:

* **Enable scheduled refresh for the dataset** - to enable scheduled refresh for the workbook that contains your dataset with R scripts, see [Configuring scheduled refresh](https://docs.microsoft.com/en-us/power-bi/refresh-scheduled-refresh), which also includes information about **Personal Gateway**.
* **Install the Personal Gateway** - you need a **Personal Gateway** installed on the machine where the file is located, and where R is installed; the Power BI service must access that workbook and re-render any updated visuals. You can get more information on how to [install and configure Personal Gateway](https://docs.microsoft.com/en-us/power-bi/personal-gateway).

Limitations

There are some limitations to queries that include R scripts created in **Query Editor**:

* All R data source settings must be set to *Public*, and all other steps in a query created in **Query Editor** must also be public. To get to data source settings, in **Power BI Desktop** select **File > Options and settings > Data source settings**.

From the **Data Source Settings** dialog, select the data source(s) and then select **Edit Permissions...** and ensure that the **Privacy Level** is set to *Public*.

* To enable scheduled refresh of your R visuals or dataset, you need to enable **Scheduled refresh** and have a **Personal Gateway** installed on the computer that houses the workbook and the R installation. For more information on both, see the previous section in this article, which provides links to learn more about each.

There are all sorts of things you can do with R and custom queries, so explore and shape your data just the way you want it to appear.