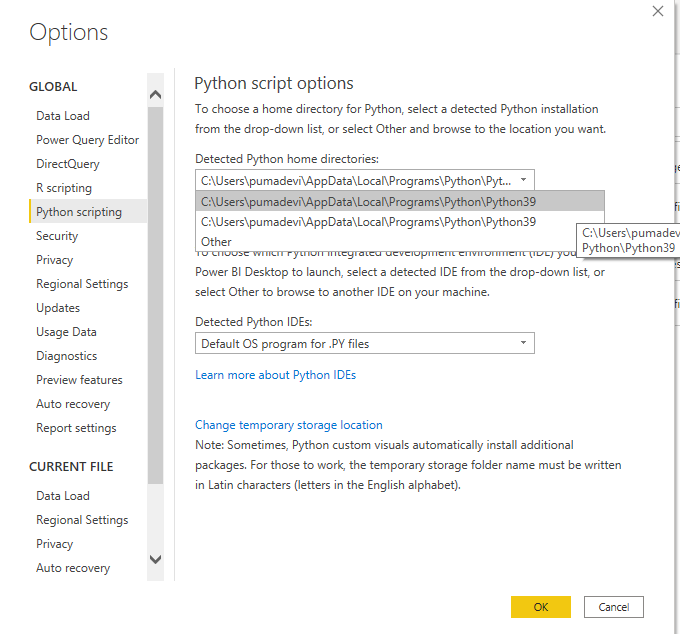
**Run Python scripts in Power BI Desktop:**

We need to enable python scripting

**Enable Python scripting:**

1. In Power BI Desktop, select **File** > **Options and settings** > **Options** > **Python scripting**. The **Python script options** page appears.



1. Specify your local Python installation path in **Detected Python home directories.**
2. Select **OK**.

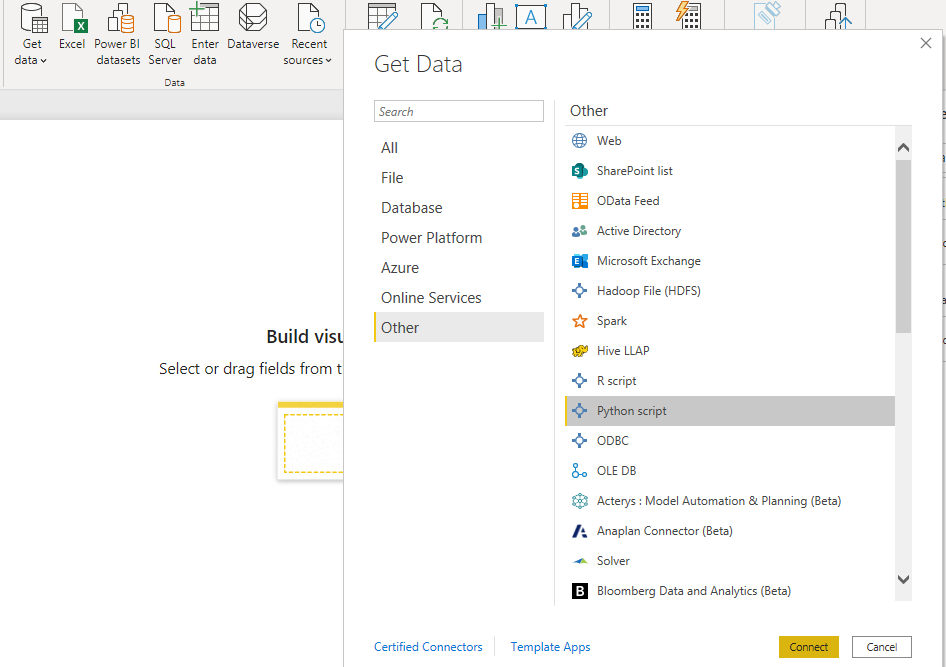
Once you specify your Python installation, you’re ready to begin running Python scripts in Power BI Desktop.

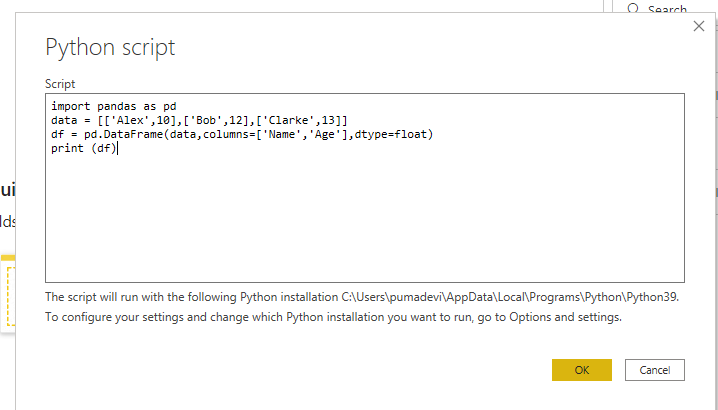
**Run Python Scripts:**

**Prepare a Python scripts:**

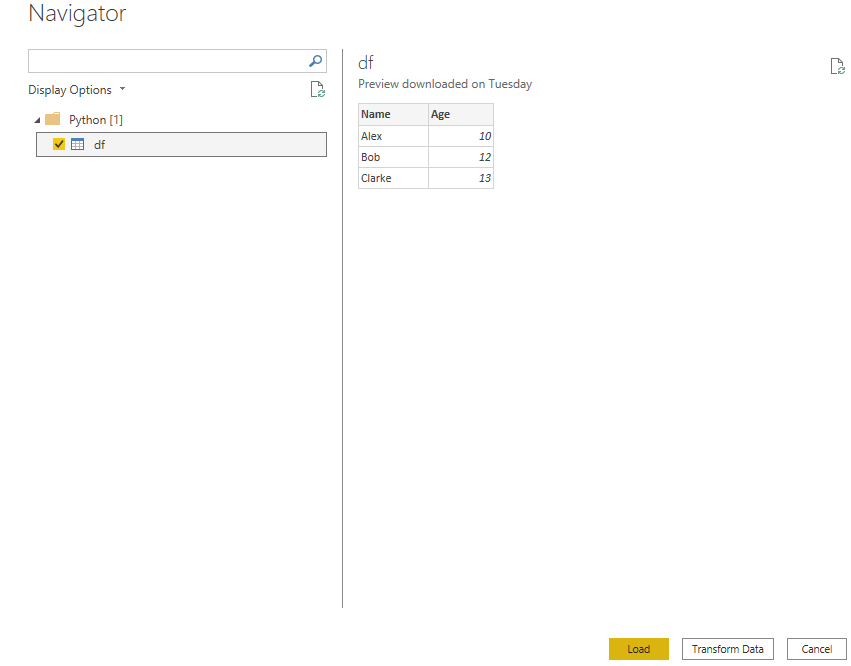
**Create a Data Frame through python script in power bi desktop:**

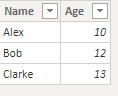
First, we need to create a script in your local Python development area. For example, here's a simple Python script that imports pandas and uses a data frame:





When run, this script returns



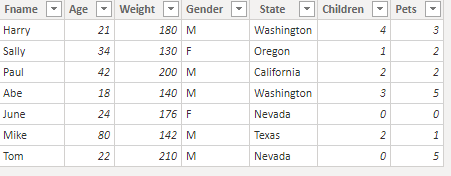


When preparing and running a Python script in Power BI Desktop, there are a few limitations:

* Only **pandas data frames** are imported, so make sure the data you want to import to Power BI is represented in a data frame
* Any Python script that runs longer than 30 minutes times out
* Interactive calls in the Python script, such as waiting for user input, halts the script’s execution
* When setting the working directory within the Python script, you *must* define a full path to the working directory, rather than a relative path
* Nested tables are currently not supported

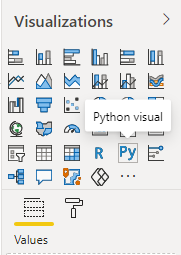
**Add the Custom Visuals in Power Bi Desktop:**

Create Data frame on the power bi desktop.



**Create Python Visuals in power bi desktop:**

1. Select the **Python visual** icon in the **Visualizations** pane.

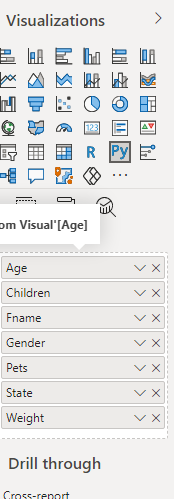


1. In the **Enable script visuals** dialog box that appears, select **Enable**.

When you add a Python visual to a report, Power BI Desktop will show

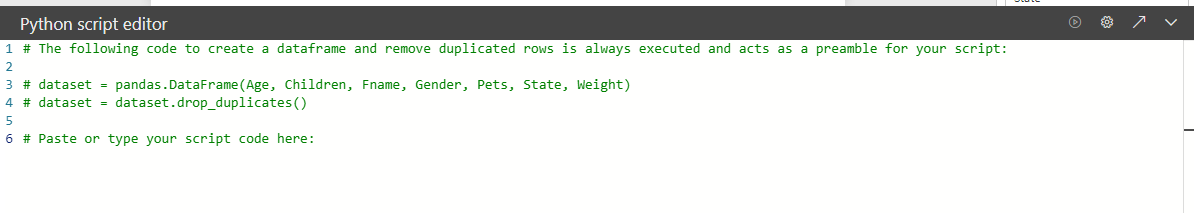
1. A placeholder Python visual image appears on the report canvas.
2. The **Python script editor** appears along the bottom of the center pane.

Next, drag the **Age**, **Children**, **Fname**, **Gender**, **Pets**, **State**, and **Weight** fields to the **Values** section where it says **Add data fields here**.



Now we can use the data you selected to create a plot.

**Create a Scatter Plot:**

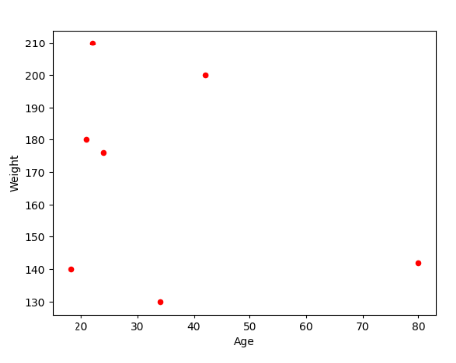
****

import matplotlib.pyplot as plt

dataset.plot(kind='scatter', x='Age', y='Weight', color='red')

plt.show()

****

****

**Create a line plot with multiple columns:**

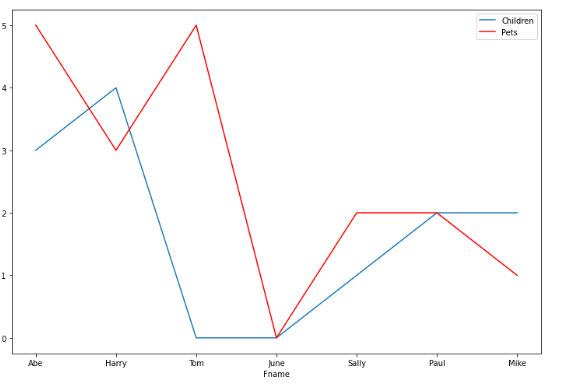
import matplotlib.pyplot as plt

ax = plt.gca()

dataset.plot(kind='line',x='Fname',y='Children',ax=ax)

dataset.plot(kind='line',x='Fname',y='Pets', color='red', ax=ax)

plt.show()

****

**Create Bar Plot:**

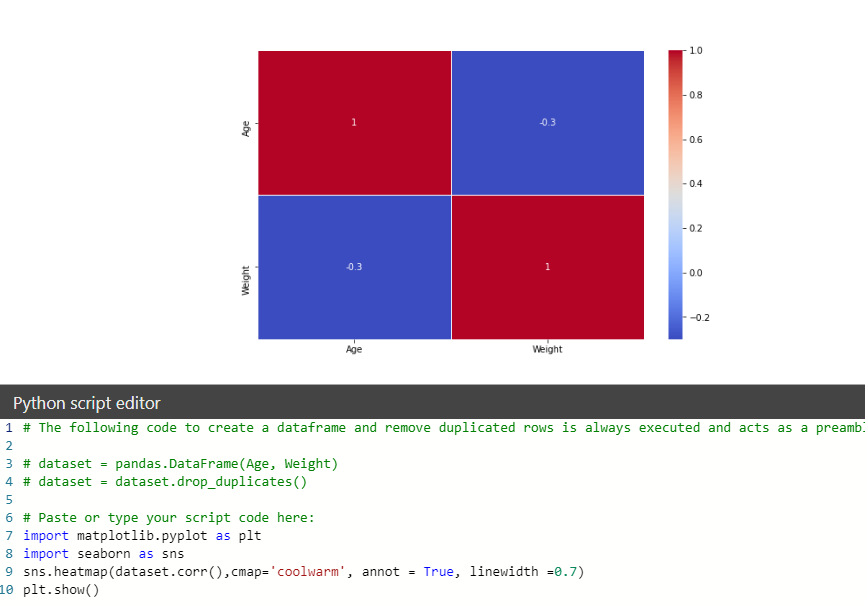
import matplotlib.pyplot as plt

dataset.plot(kind='bar',x='Fname',y='Age')

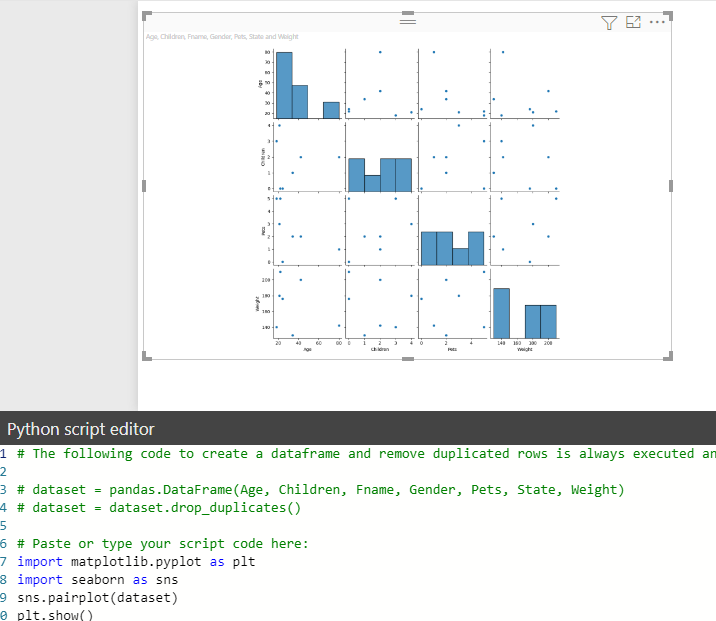
plt.show()

****

**Create heat Map:**

****

**Create a Pair Plot:**

****