

Ejecución de scripts de Python en Power BI Desktop

Para ejecutar scripts de Python en Power BI Desktop, deberá instalar Python en el equipo local. Puede descargar Python en el [sitio web de Python](#). La versión actual de scripting de Python admite caracteres Unicode y espacios en la ruta de instalación.

Para la integración de Python con Power BI, es necesario instalar los dos paquetes de Python siguientes. En una consola o un shell, use la herramienta de línea de comandos [pip](#) para instalar los paquetes. La herramienta pip se incluye con las versiones recientes de Python.

[Pandas](#) es una biblioteca de software para manipular y analizar datos. Pandas ofrece operaciones y estructuras de datos para manipular series temporales y tablas numéricas. Para importarlos en Power BI, los datos de Python deben estar en una [trama de datos pandas](#). Una trama de datos es una estructura de datos bidimensional, como una tabla con filas y columnas.

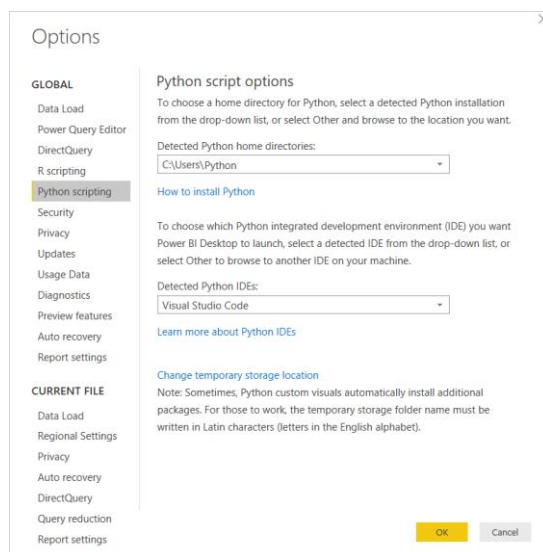
[Matplotlib](#) es una biblioteca de trazado para Python y su extensión matemática numérica [NumPy](#).

```
pip install pandas
pip install matplotlib
```

Habilitación de la creación de scripts de Python

Para habilitar la creación de scripts de Python en Power BI:

1. En Power BI Desktop, seleccione **Archivo>Opciones y configuración>Opciones>Creación de scripts de Python**. Aparece la página **Opciones de script de Python**.



2. Si es necesario, proporcione o edite la ruta de la instalación de Python local en **Directorios principales de Python detectados**. En la imagen anterior, la ruta de instalación local de Python es *C:\Python*. Si tiene más de una instalación local de Python, asegúrese de seleccionar la que quiera usar.
3. Seleccione **Aceptar**.

Creación de un script de Python

Este es un script de Python sencillo de ejemplo en el que se importa pandas y se usa una trama de datos:

```
import pandas as pd
data = [['Alex',10],['Bob',12],['Clarke',13]]
df = pd.DataFrame(data,columns=['Name','Age'])
print (df)
```

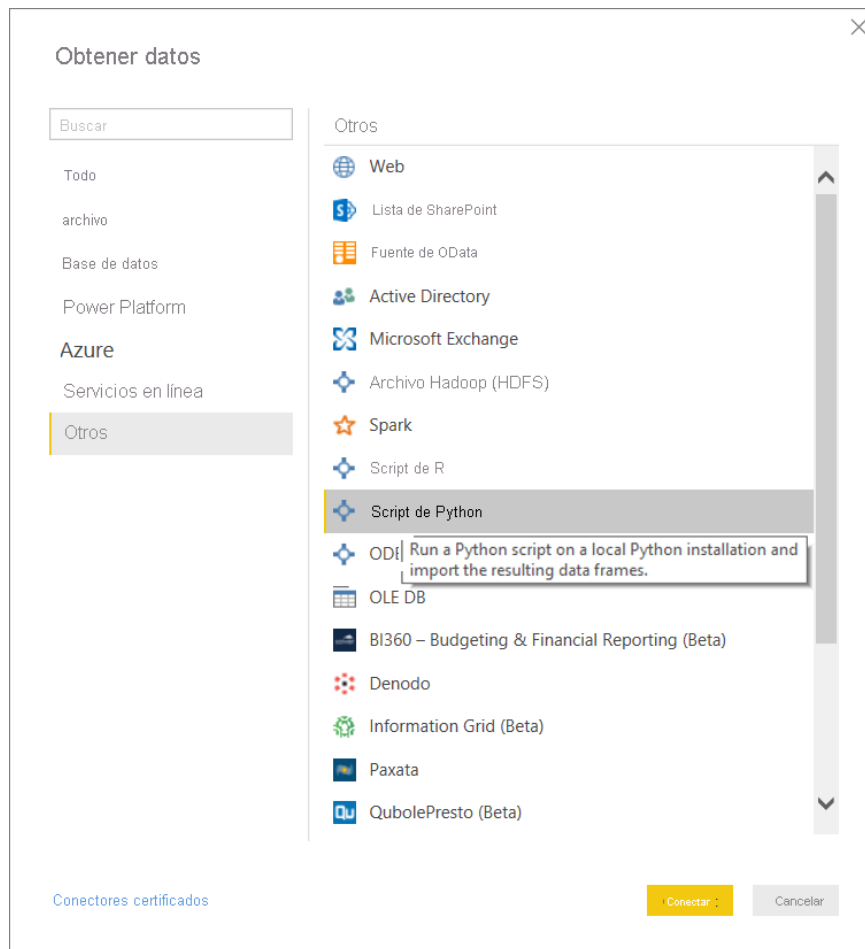
Cuando se ejecuta, este script devuelve:

```
   Name  Age
0  Alex  10.0
1   Bob  12.0
2 Clarke 13.0
```

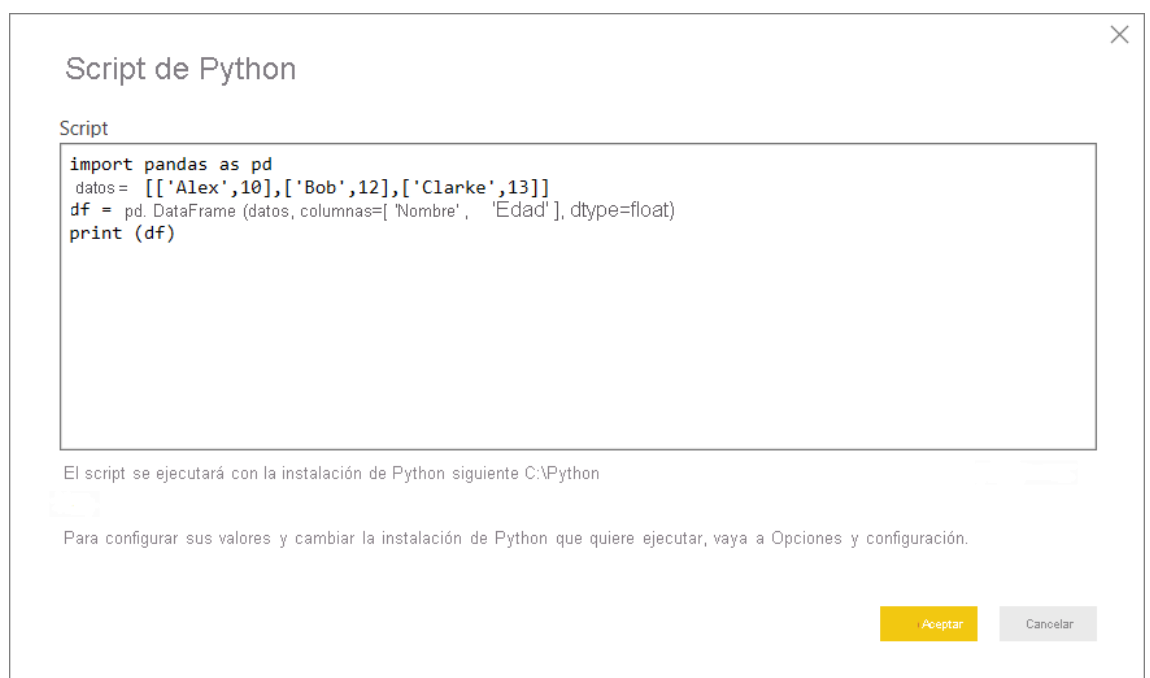
Ejecución del script e importación de los datos

Para ejecutar el script de Python:

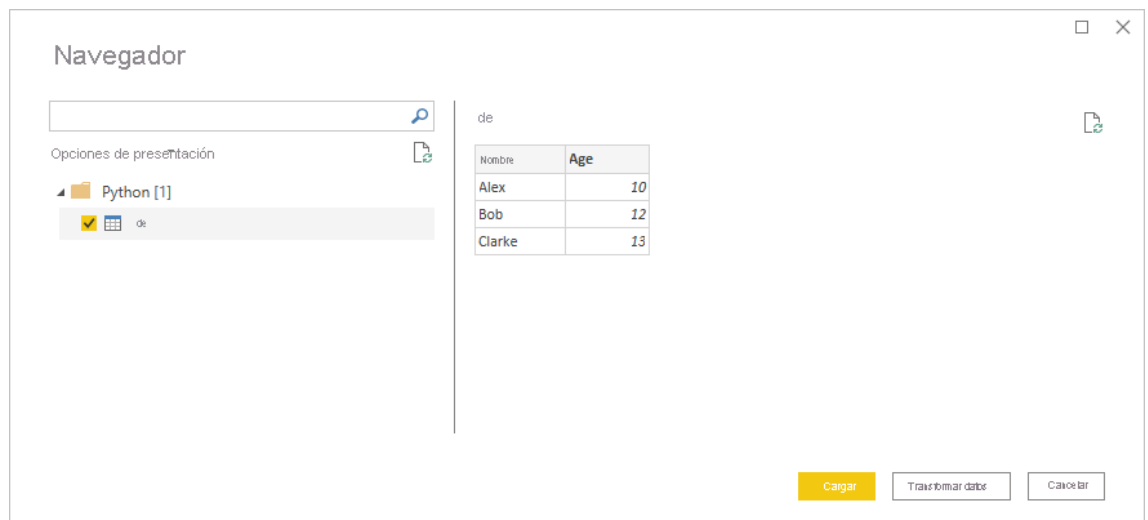
1. En el grupo **Inicio** de la cinta de Power BI Desktop, seleccione **Obtener datos**.
2. En el cuadro de diálogo **Obtener datos**, seleccione **Otros>Script de Python** y después **Conectar**. Power BI usa la versión de Python instalada más reciente como motor de Python.



3. En la pantalla **Script de Python**, pegue el script de Python en el campo **Script** y seleccione **Aceptar**.



4. Si el script se ejecuta correctamente, aparecerá la ventana **Navegador** y podrá cargar los datos. Seleccione la tabla **df** y después **Cargar**.



Power BI importa los datos y puede usarlos para crear visualizaciones e informes. Para actualizar los datos, seleccione **Actualizar** en el grupo **Inicio** de la cinta de Power BI Desktop. Al realizar la actualización, Power BI vuelve a ejecutar el script de Python.

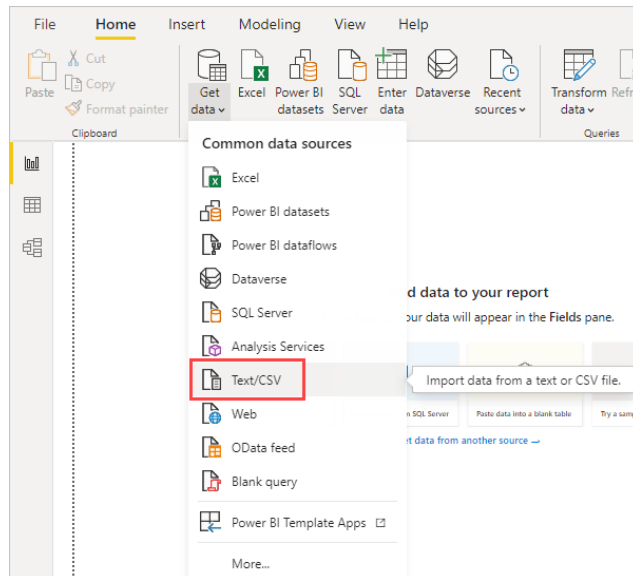
Uso de Python en el Editor de Power Query

Puede utilizar *Python*, un lenguaje de programación usado ampliamente por estadistas, científicos de datos y analistas de datos, en el **Editor de Power Query** de Power BI Desktop. Esta integración de Python en el **Editor de Power Query** permite limpiar datos con Python y realizar acciones avanzadas, como analizar y dar forma a los datos en conjuntos de datos, incluida la operación de completar datos que faltan, las predicciones y la agrupación en clústeres, por nombrar algunas. Python es un lenguaje eficaz y se puede usar en el **Editor de Power Query** para preparar el modelo de datos y crear informes.

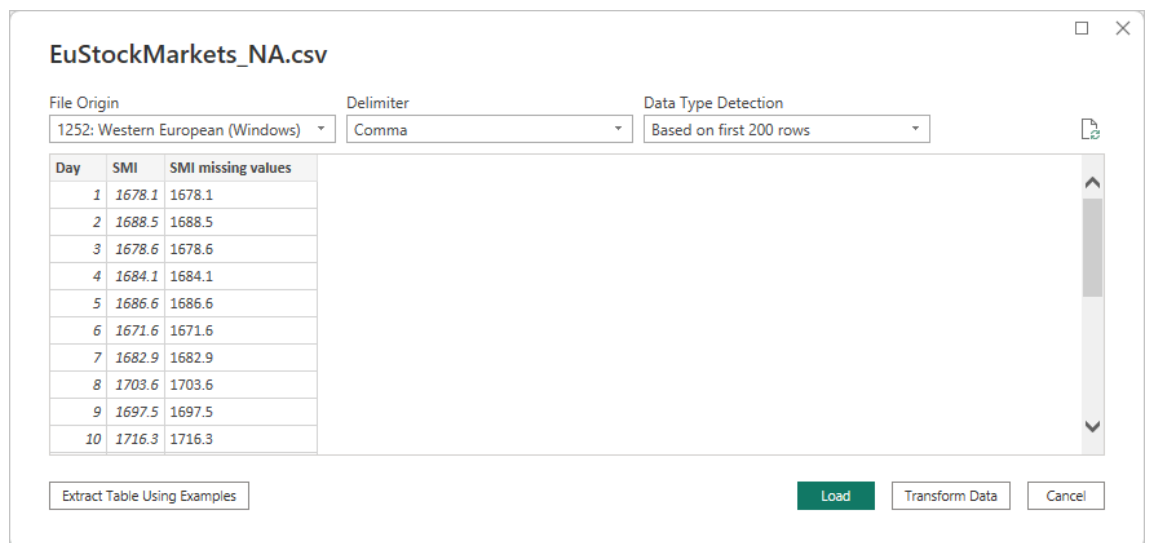
Uso de Python con el Editor de Power Query

Para ver cómo se utiliza Python en el **Editor de Power Query**, use este ejemplo de un conjunto de datos de la bolsa, basado en un archivo .csv que puede [descargar aquí](#), y sígalo. El siguiente procedimiento muestra los pasos de este ejemplo:

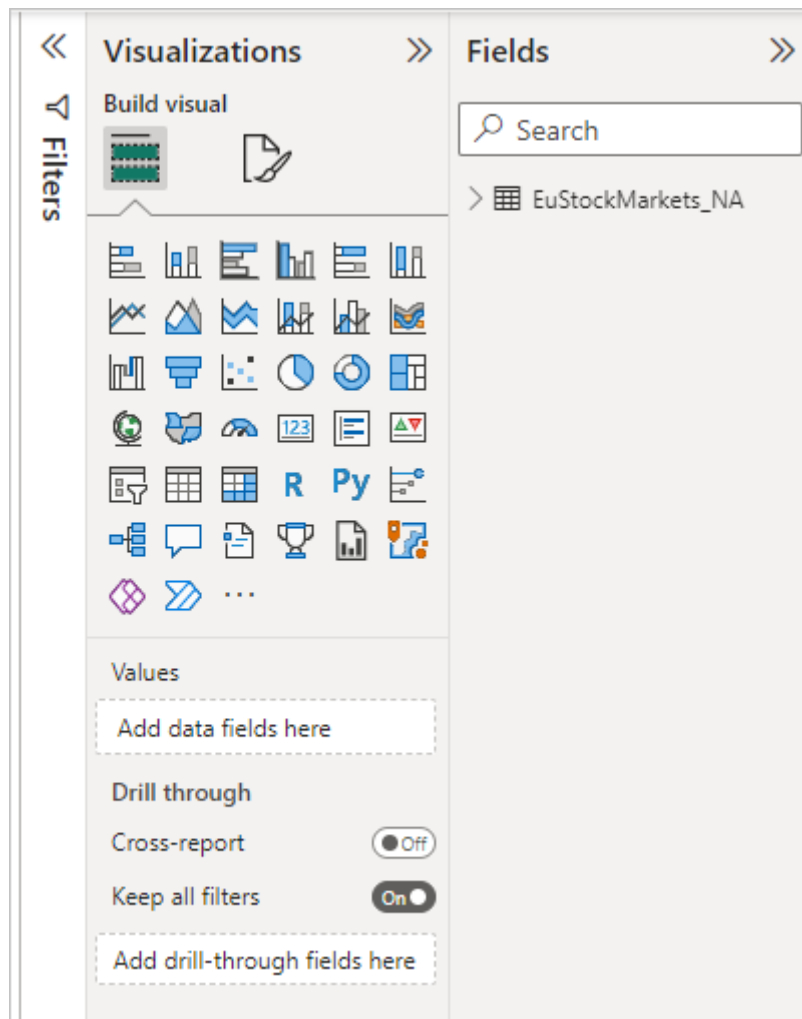
1. En primer lugar, cargue los datos en **Power BI Desktop**. En este ejemplo, cargue el archivo *EuStockMarkets_NA.csv* y seleccione **Obtener datos>Texto o CSV** en la cinta **Inicio** de **Power BI Desktop**.



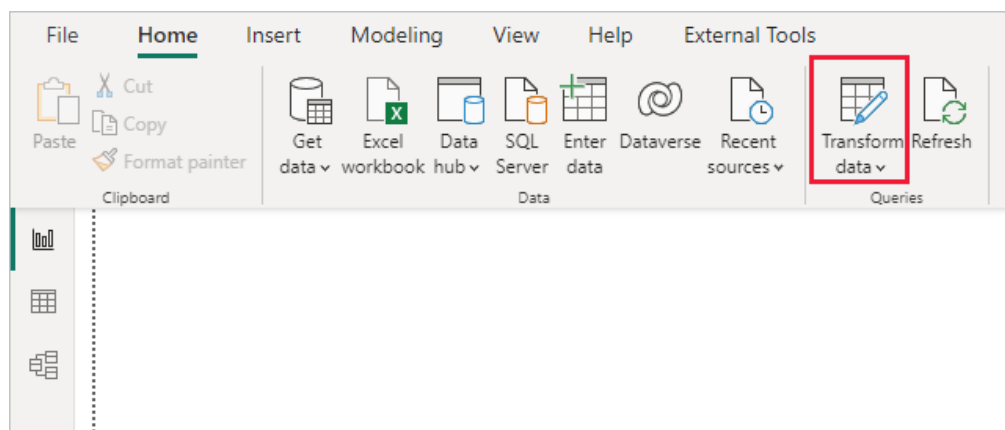
2. Seleccione primero el archivo y, después, **Abrir** para que el archivo .csv se muestre en el cuadro de diálogo **Archivo CSV**.



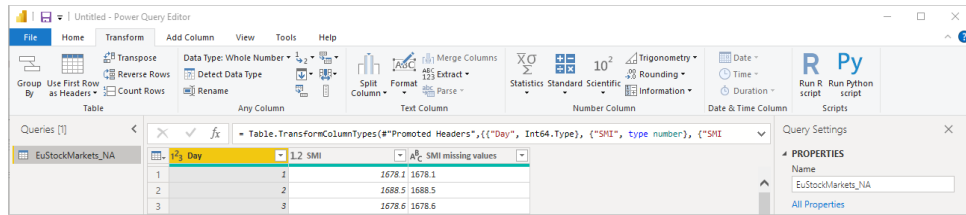
3. Una vez cargados los datos, se muestran en el panel **Campos** de **Power BI Desktop**.



4. Para abrir el **Editor de Power Query**, seleccione **Transformar datos** en la pestaña **Inicio** de **Power BI Desktop**.



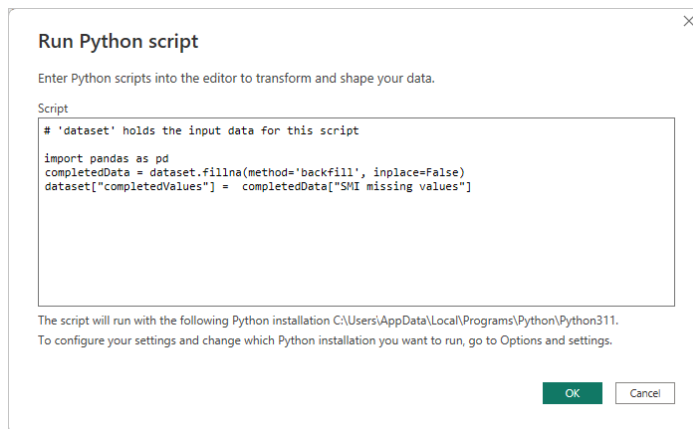
5. En la pestaña **Transformar**, haga clic en **Ejecutar script de Python** para abrir el editor correspondiente, como se muestra en el paso siguiente. En las filas 15 y 20 faltan datos, al igual que en otras filas que no se ven en la siguiente imagen. En los siguientes pasos se muestra cómo Python completa esas filas de forma automática.



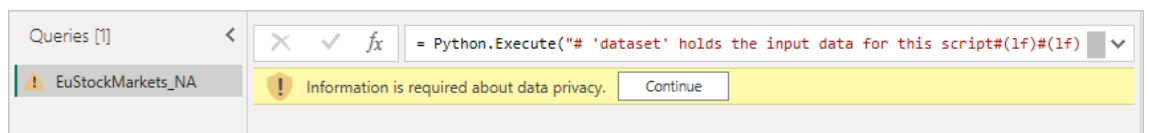
6. En este ejemplo, escriba el siguiente código de script:

```
import pandas as pd
completedData = dataset.fillna(method='backfill', inplace=False)
dataset["completedValues"] = completedData["SMI missing values"]
```

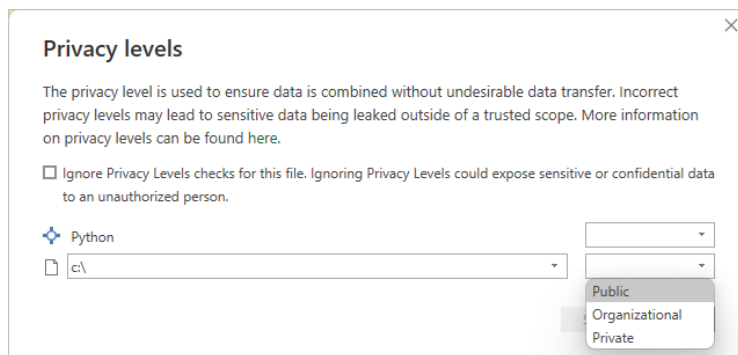
Cuando se incluye el código en el cuadro de diálogo **Ejecutar script de Python**, tiene un aspecto similar al siguiente ejemplo:



7. Después de seleccionar **Aceptar**, en el **Editor de Power Query** se muestra una advertencia sobre la privacidad de los datos.



8. Para que los scripts de Python funcionen correctamente en el servicio Power BI, todos los orígenes de datos se deben establecer como *públicos*. Para obtener más información sobre la configuración de privacidad y sus implicaciones, consulte [Niveles de privacidad](#).



Observe que en el panel **Campos** hay una columna nueva denominada *completedValues*. Observe que algunos elementos de datos faltan, como en las filas 15 y 18. Vea cómo Python lo aborda en la sección siguiente.

Con solo cinco líneas de script de Python, el **Editor de Power Query** ha rellenado los valores que faltaban con un modelo predictivo.

Creación de objetos visuales de Power BI con Python

Importe el siguiente script de Python en Power BI Desktop:

```
import pandas as pd
df = pd.DataFrame({
    'Fname':['Harry','Sally','Paul','Abe','June','Mike','Tom'],
    'Age':[21,34,42,18,24,80,22],
    'Weight': [180, 130, 200, 140, 176, 142, 210],
    'Gender':['M','F','M','M','F','M','M'],

    'State':['Washington','Oregon','California','Washington','Nevada','Texas','Nevada'],
    'Children':[4,1,2,3,0,2,0],
    'Pets':[3,2,2,5,0,1,5]
})
print (df)
```

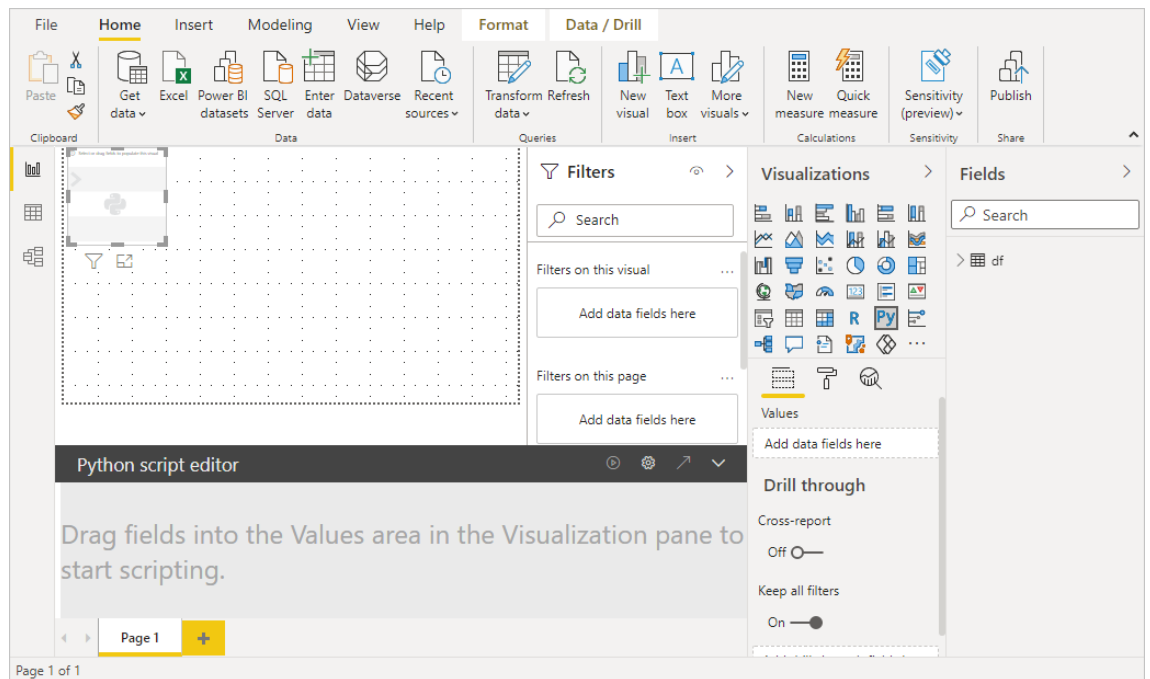
Creación de objetos visuales de Python en Power BI Desktop

1. Después de importar el script de Python, seleccione el icono **visual de Python** en el panel **visualizaciones** de Power BI Desktop.

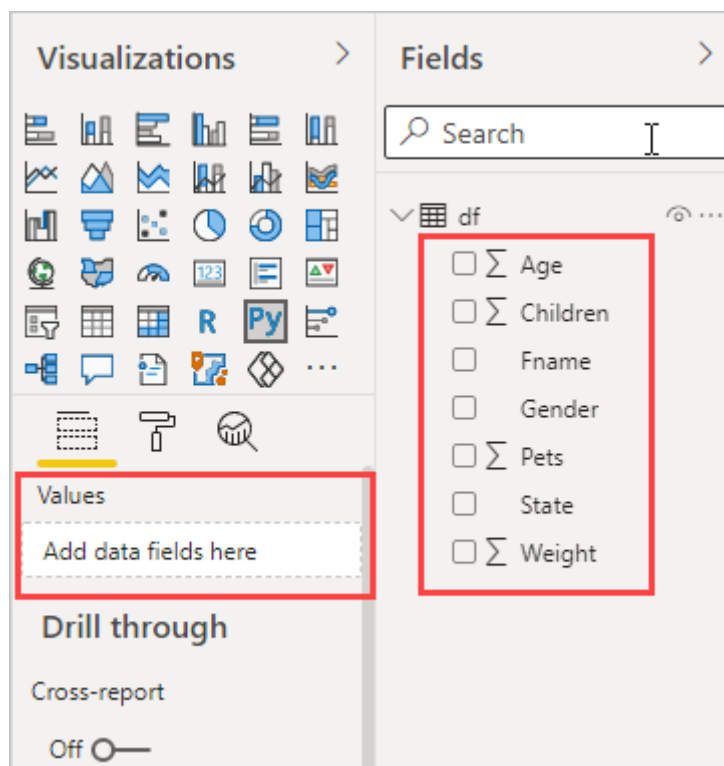


2. En el cuadro de diálogo **Habilitar objetos visuales de script** que aparece, seleccione **Habilitar**.

Aparece una imagen visual de Python de marcador de posición en el lienzo del informe y el **editor de scripts de Python** aparece en la parte inferior del panel central.



3. Arrastre los campos **Age**, **Children**, **Fname**, **Gender**, **Pets**, **State** y **Weight** a la sección **Valores**, donde aparece **"Agregar campos de datos aquí"**.



En función de lo que seleccione, el editor de scripts de Python genera el siguiente código de enlace.

- El editor crea una trama de datos del *conjunto de datos* con los campos que agrega.

- La agregación predeterminada es: **No resumir**.
 - Como sucede con los objetos visuales de tabla, los campos se agrupan y las filas duplicadas aparecen solo una vez.
4. Con la trama de datos generada automáticamente por los campos seleccionados, puede escribir un script de Python que genere un trazado en el dispositivo predeterminado de Python. Una vez completado el script, seleccione el icono **Ejecutar** de la barra de título del **editor de scripts de Python** para ejecutar el script y generar el objeto visual.



Creación de un gráfico de dispersión

Cree un gráfico de dispersión para ver si hay una correlación entre la edad y el peso.

1. En el **editor de scripts de Python**, en **Pegue o escriba aquí el código de script**, introduzca este código:

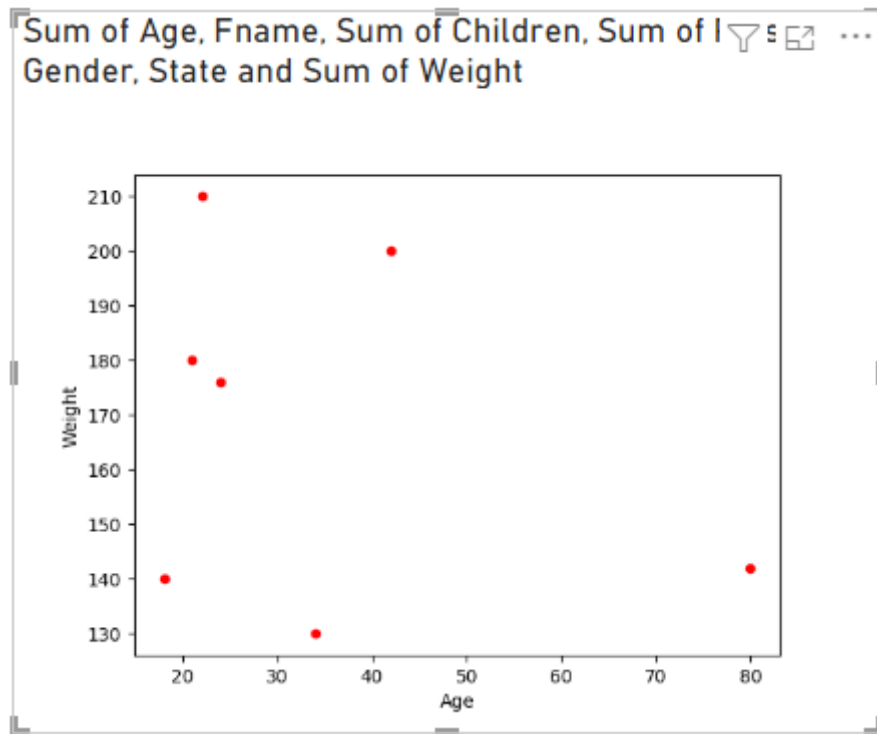
```
import matplotlib.pyplot as plt
dataset.plot(kind='scatter', x='Age', y='Weight', color='red')
plt.show()
```

El panel del editor de scripts de Python debería tener la siguiente imagen:



El código importa la biblioteca Matplotlib, que traza y crea el objeto visual.

2. Seleccione el botón **Ejecutar** script para generar el siguiente gráfico de dispersión en el objeto visual de Python.



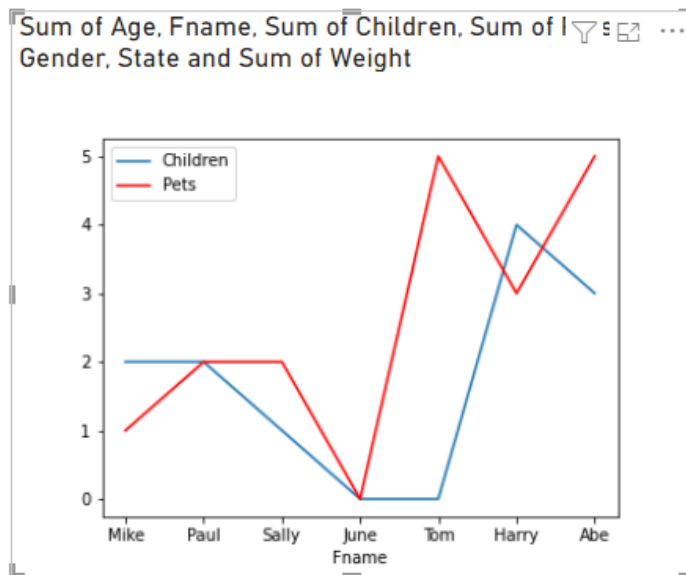
Creación de un gráfico de líneas con varias columnas

Cree un gráfico de líneas para cada persona que muestre el número de hijos y mascotas.

1. En **Pegue o escriba el código de script aquí**, quite o comente el código anterior y escriba el siguiente código de Python:

```
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()
```

2. Seleccione el botón **Ejecutar** para generar el siguiente trazado de líneas con varias columnas:



Creación de un gráfico de barras

Cree un gráfico de barras para la edad de cada persona.

1. En **Pegue o escriba el código de script aquí**, quite o comente el código anterior y escriba el siguiente código de Python:

```
import matplotlib.pyplot as plt
dataset.plot(kind='bar',x='Fname',y='Age')
plt.show()
```

2. Seleccione el botón **Ejecutar** para generar el siguiente trazado de barras:

