

# Tutorial para desarrolladores de Slicer: Programación en Slicer

Sonia Pujol, Ph.D.

Profesora adjunta de radiología

Directora de formación y educación en 3D Slicer

Hospital de mujeres de Brigham

Facultad de Medicina de Harvard

Steve Pieper, Ph.D.

Arquitecto jefe de 3D Slicer

Isomics Inc.

# Objetivo del tutorial



```
def threshold(t):
    n=getNode('T2')
    a=array('T2')
    a[a<t]=0
    arrayFromVolumeModified(n)
    print('Thresholding done')
```



Este tutorial es una introducción a la consola de Python y al widget Qt de la versión 5 de 3D Slicer



```
b=qt.QPushButton('Toggle')
b.connect('clicked()', toggle)
b.setStyleSheet = "font-size: 24pt; color: aqua; margin: 20px"
b.show()
```

# Esquema del Tutorial



Parte 1: Visión general de los módulos de 3D Slicer



Parte 2: Familiarizarse con el entorno Python en 3D Slicer

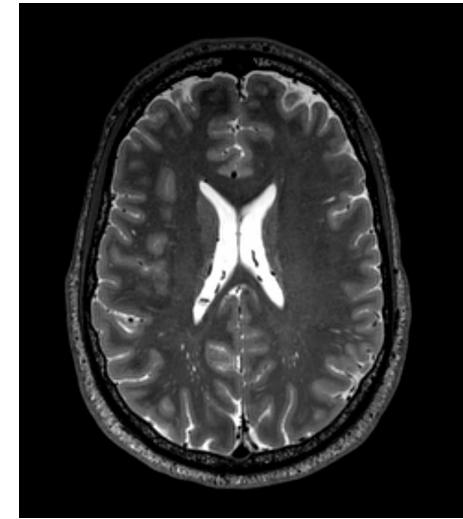
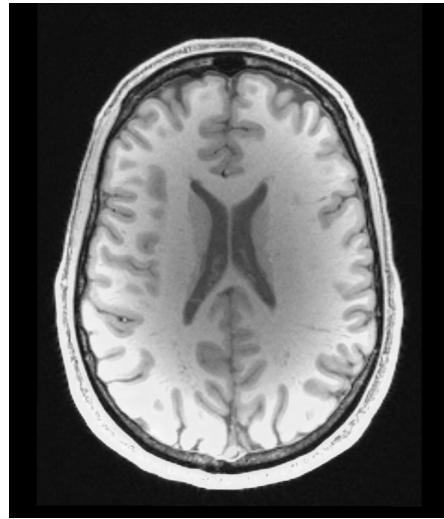
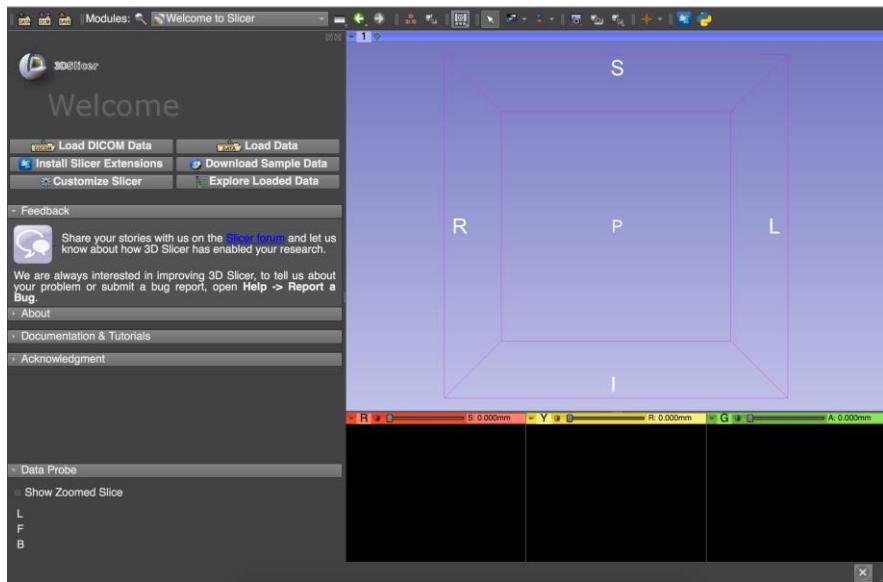


Parte 3: Familiarizarse con el conjunto de herramientas de widgets Qt en 3D Slicer

# Descargo de responsabilidad

- 3D Slicer es una aplicación de software libre de código abierto distribuida bajo una licencia de estilo BSD.
- El software no cuenta con la aprobación de la FDA ni el marcado CE y es para uso exclusivo en investigación.

# Materiales didácticos

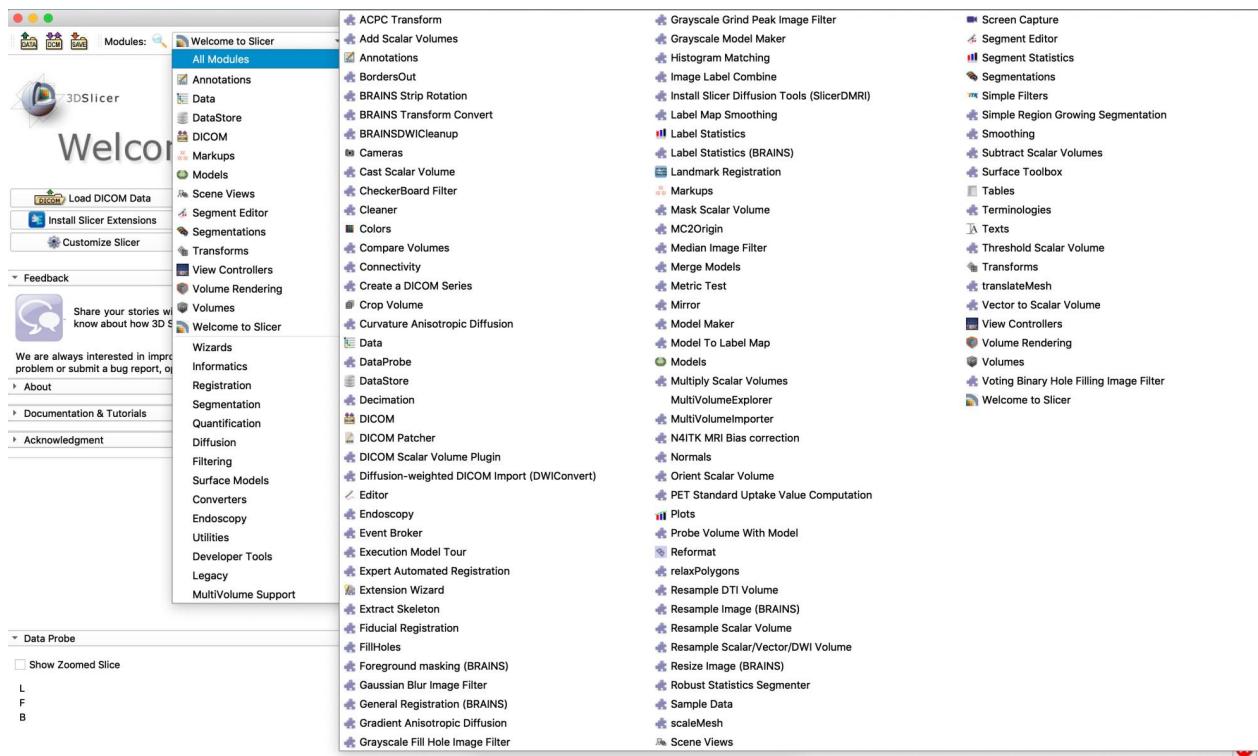


SlicerProgrammingTutorialData.zip  
[link](#)

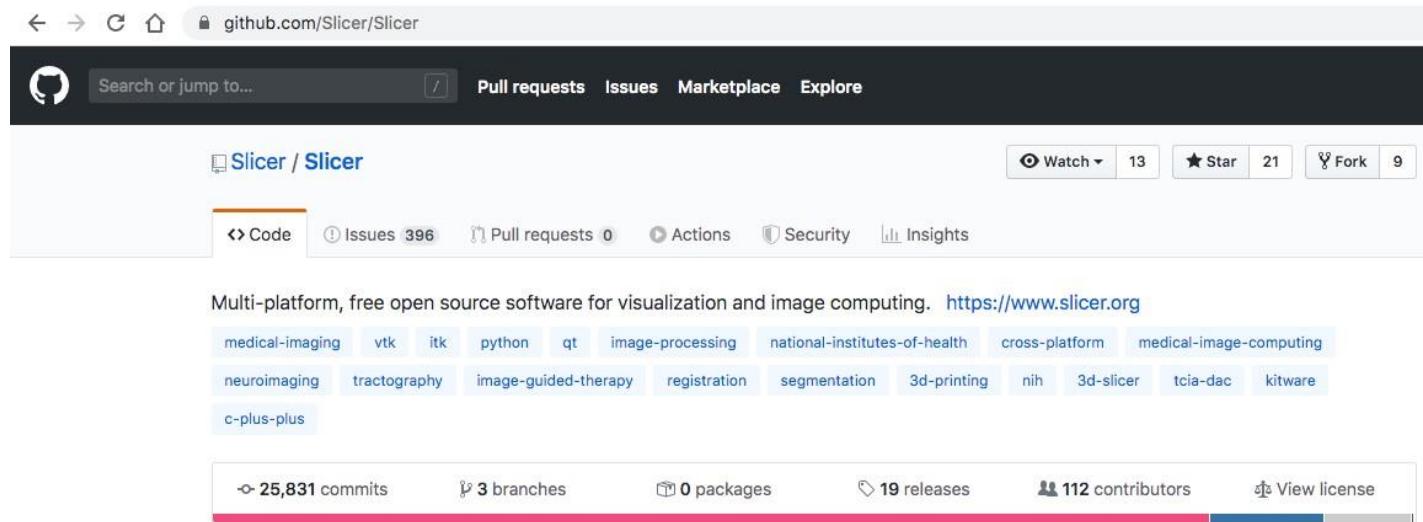
3D Slicer versión 4.11

# Parte 1

# Visión general de los módulos de Slicer



# 3D Slicer



- 3D Slicer es una plataforma de código abierto para el análisis y la visualización de datos de imágenes médicas.
- 3D Slicer se compila y se prueba a diario en plataformas Windows, Mac y Linux
- El código fuente está disponible gratuitamente en GitHub en <http://github.com/Slicer/Slicer>

# Módulos Slicer

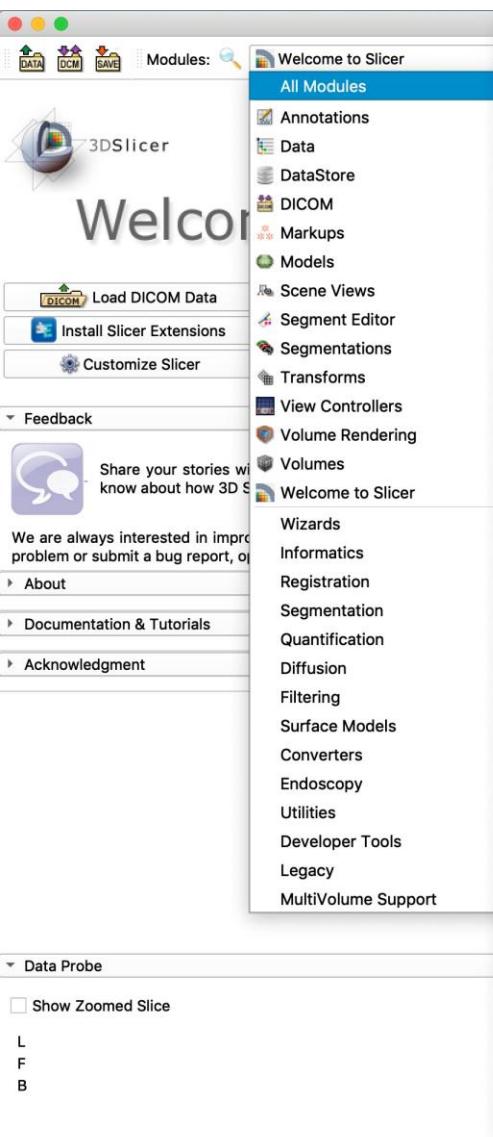
3D Slicer admite tres tipos de módulos:

- Interfaz de línea de comandos (CLI): ejecutor independiente con argumentos de entrada/salida limitados.
- Módulos cargables (C++ Plugins): optimizados para cálculos pesados.

Es el enfoque de este tutorial

- Módulos con script (Python): recomendados para la creación rápida de prototipos y el desarrollo de flujo de trabajo.

# Módulos Slicer



- ACPC Transform
- Add Scalar Volumes
- Annotations
- BordersOut
- BRAINS Strip Rotation
- BRAINS Transform Convert
- BRAINSDWICleanup
- Cameras
- Cast Scalar Volume
- CheckerBoard Filter
- Cleaner
- Colors
- Compare Volumes
- Connectivity
- Create a DICOM Series
- Crop Volume
- Curvature Anisotropic Diffusion
- Data
- DataProbe
- DataStore
- Decimation
- DICOM
- DICOM Patcher
- DICOM Scalar Volume Plugin
- Diffusion-weighted DICOM Import (DWIConvert)
- Editor
- Endoscopy
- Event Broker
- Execution Model Tour
- Expert Automated Registration
- Extension Wizard
- Extract Skeleton
- Fiducial Registration
- FillHoles
- Foreground masking (BRAINS)
- Gaussian Blur Image Filter
- General Registration (BRAINS)
- Gradient Anisotropic Diffusion
- Grayscale Fill Hole Image Filter
- Grayscale Grind Peak Image Filter
- Grayscale Model Maker
- Histogram Matching
- Image Label Combine
- Install Slicer Diffusion Tools (SlicerDMRI)
- Screen Capture
- Segment Editor
- Segment Statistics
- Segmentations
- Simple Filters

El tipo de módulo es transparente para el usuario final

- Model Maker
- Model To Label Map
- Models
- Multiply Scalar Volumes
- MultiVolumeExplorer
- MultiVolumeImporter
- N4ITK MRI Bias correction
- Normals
- Orient Scalar Volume
- PET Standard Uptake Value Computation
- Plots
- Probe Volume With Model
- Reformat
- relaxPolygons
- Resample DTI Volume
- Resample Image (BRAINS)
- Resample Scalar Volume
- Resample Scalar/Vector/DWI Volume
- Resize Image (BRAINS)
- Robust Statistics Segmente
- Sample Data
- scaleMesh
- Scene Views

# Extensiones Slicer

Una extensión Slicer es un paquete de entrega que incluye uno o varios módulos de Slicer



SwissSkullStripper  
Bill Lorensen (Noware...)  
★★★★★ (0)

[INSTALL](#)



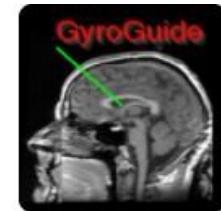
PETTumorSegmenta...  
Christian Bauer (Univ...)  
★★★★★ (0)

[INSTALL](#)



SlicerOpenIGTLINK  
Junichi Tokuda (SPL), ...  
★★★★★ (0)

[INSTALL](#)



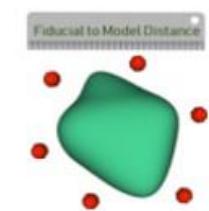
GyroGuide  
Rufeng Chen, Luping...  
★★★★★ (0)

[INSTALL](#)



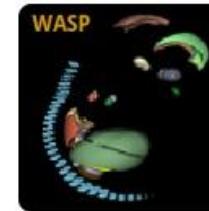
PET-IndiC  
Ethan Ulrich (Universi...)  
★★★★★ (0)

[INSTALL](#)



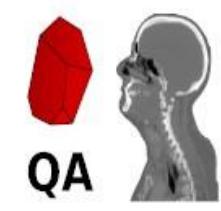
FiducialsToModelDi...  
Jesse Reynolds (Cante...)  
★★★★★ (0)

[INSTALL](#)



WASP  
Thomas Lawson (MR...  
★★★★★ (0)

[INSTALL](#)

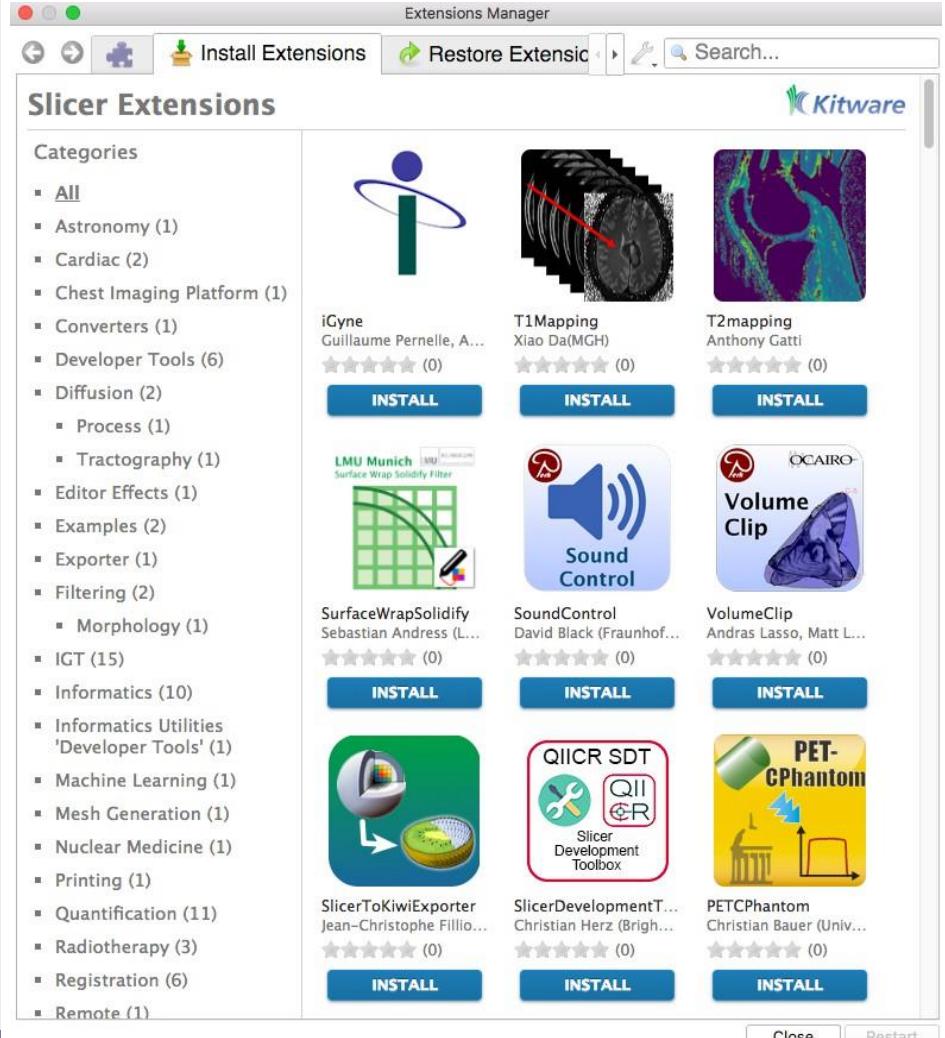


ImageCompare  
Paolo Zaffino (Magna ...  
★★★★★ (0)

[INSTALL](#)

# Administrador de extensiones de Slicer

- El administrador de extensiones de Slicer proporciona una plataforma de “tienda de aplicaciones” para el ecosistema de 3D Slicer
- El Extension Manager facilita la creación e instalación de extensiones de Slicer.
- La versión 5 de Slicer incluye más de 130 extensiones.



# Parte 2

## Familiarizarse con el entorno Python en 3D Slicer

.....

Consola Python

```
Python 3.9.10 (main, Jan 23 2025, 23:04:06) [MSC v.1942 64 bit (AMD64)] on win32
>>>
```

# Python en Slicer

Slicer v.4.11 funciona con Python3 y un conjunto de bibliotecas estándar



NumPy

NumPy, el cual es el paquete fundamental para la computación científica con Python.



VTK es una biblioteca de código abierto para la manipulación y visualización de datos científicos.



ITK es una biblioteca de código abierto para el análisis de imágenes.



CTK es una biblioteca de código abierto para el cálculo de imágenes biomédicas.



PythonQT es un enlace de Python para Qt.



Qt es un entorno de multiplataforma utilizado como conjunto de herramientas gráficas.

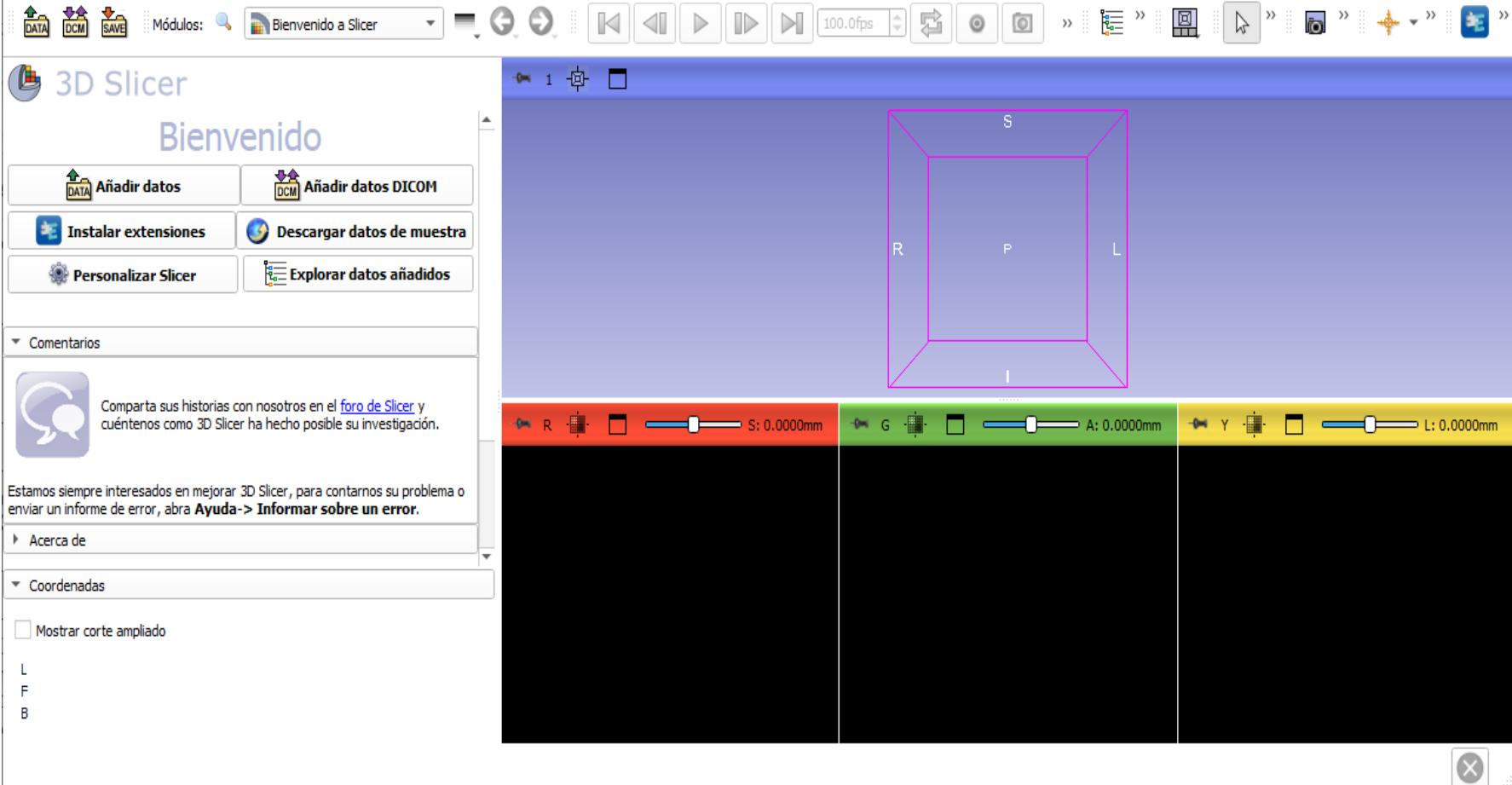
# Python en Slicer

- El índice de paquetes de Python (PyPi) da acceso a más de 200.000 paquetes adicionales de Python (<http://pipy.org>).
- El comando “pip install” en Slicer permite a los desarrolladores instalar las herramientas de computación científica más comunes (por ejemplo, TensorFlow, SciPy, PyTorch, Pandas, etc.).
- Slicer puede utilizarse como núcleo de Jupyter notebook
- PyCharm y otras herramientas de desarrollo de Python se pueden utilizar con Slicer



3D Slicer 5.8.0

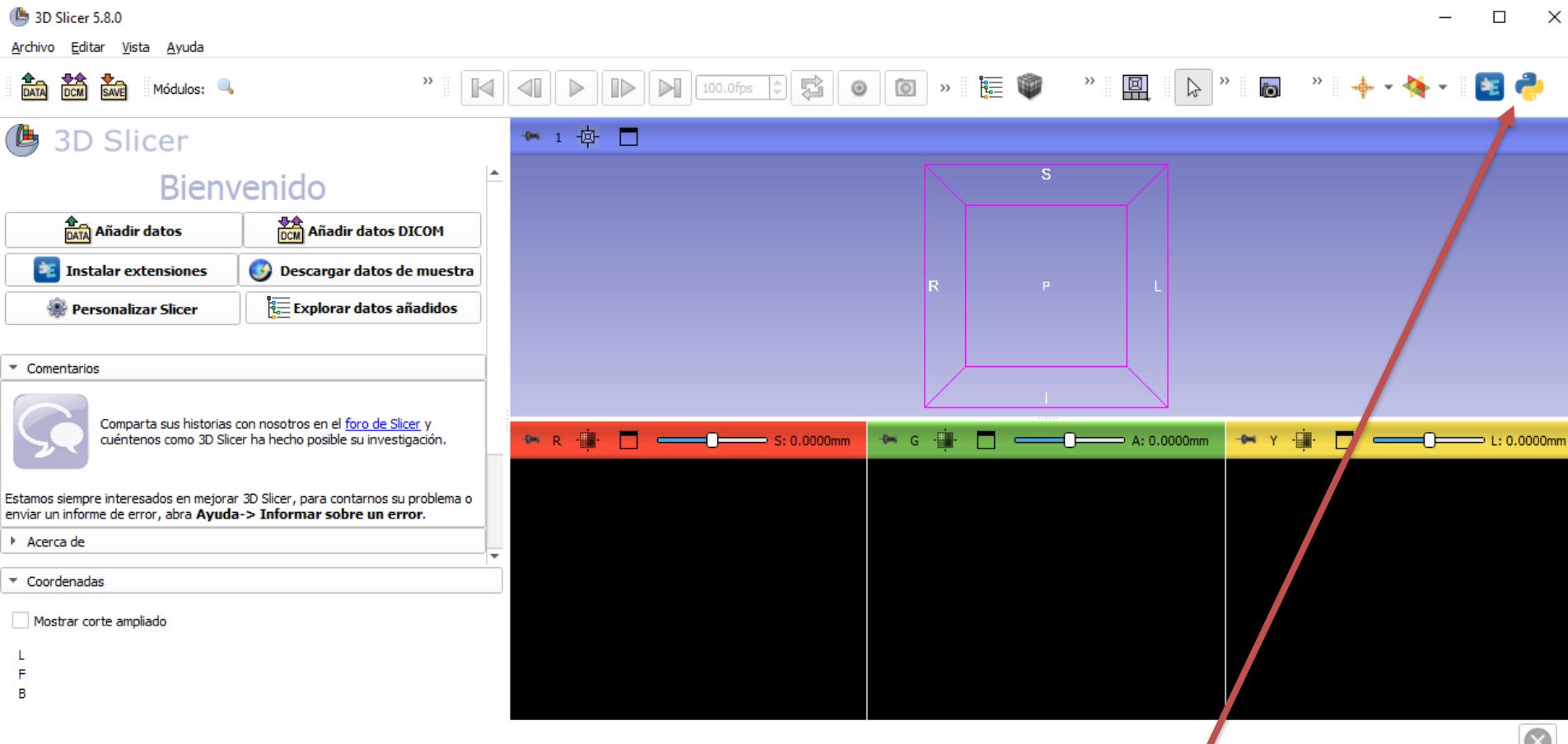
Archivo Editar Vista Ayuda



La version 5 de Slicer integra  
Python3, VTK5 and ITK5

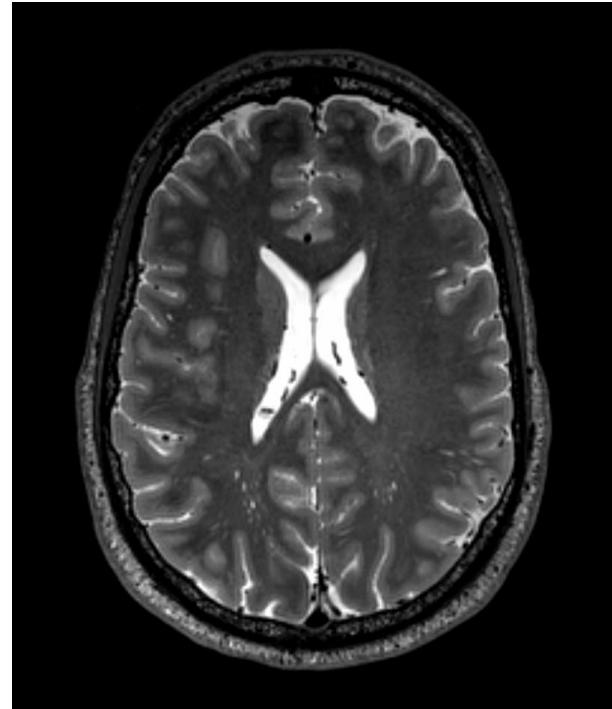
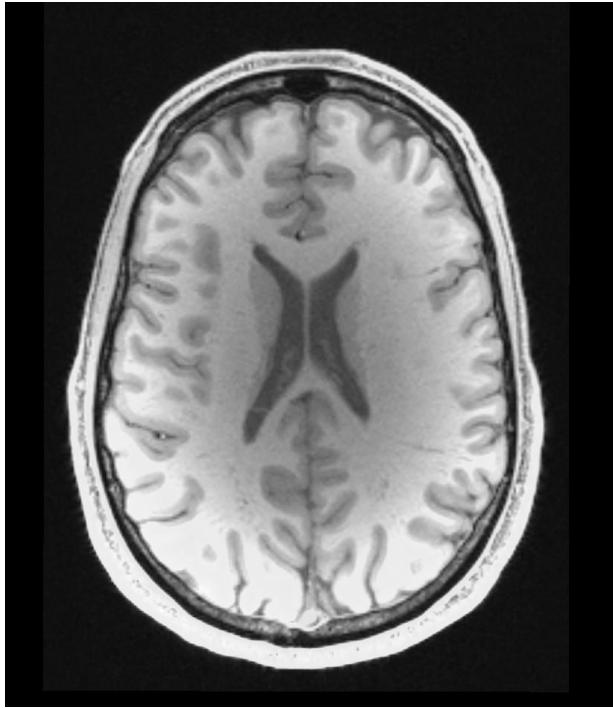
# Consola de Python en Slicer

La consola de Python esta basada en Qt y permite un acceso directo a Slicer MRML Nodos, librerias (NumPy, VTK, ITK, CTK) y Qt.



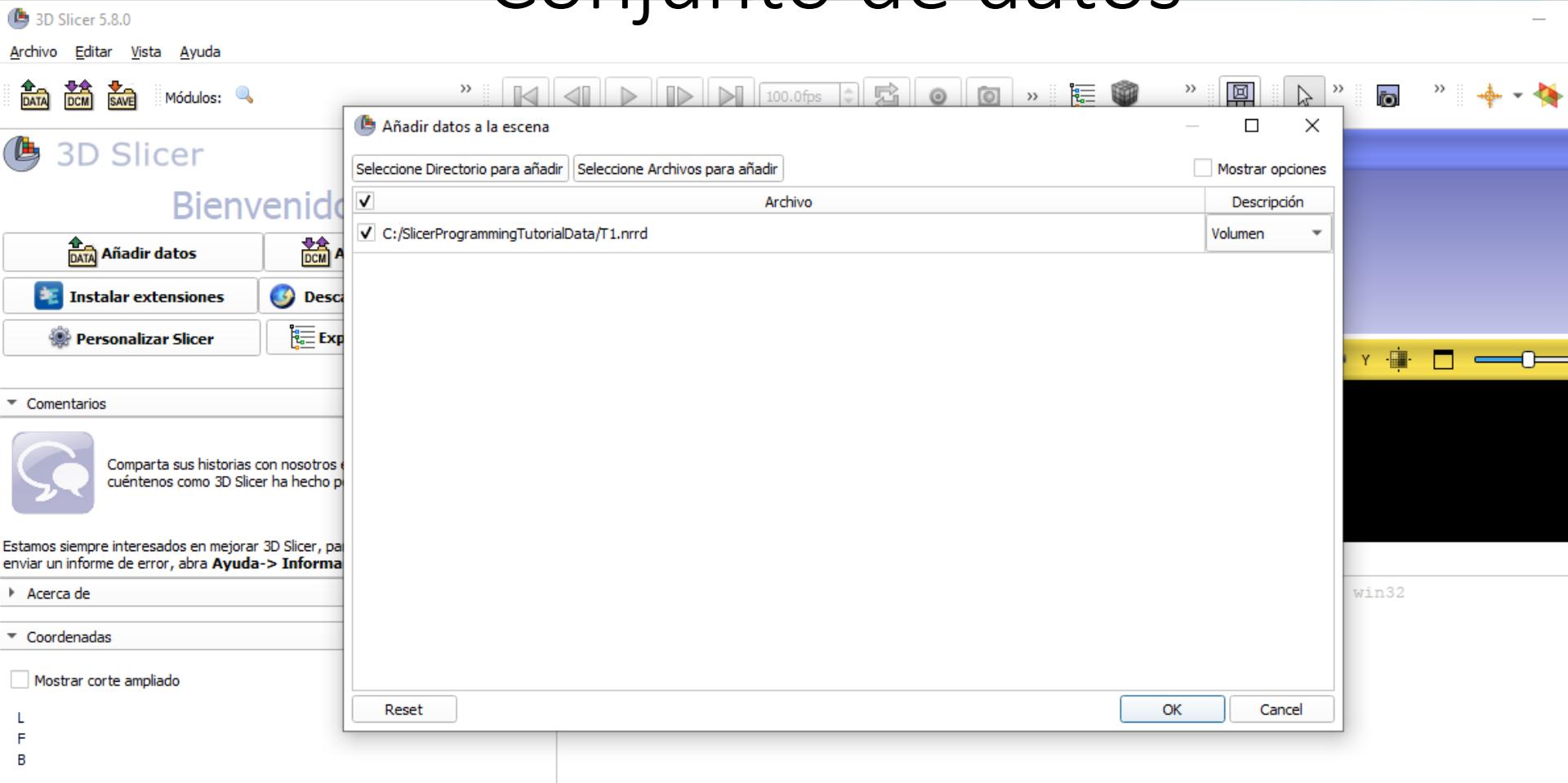
Para acceder a la Consola de Python, haga clic en el icono python en el menú de la barra superior de Slicer





El conjunto de datos del tutorial de programación Slicer incluye una resonancia magnética ponderada en T1 y otra en T2 de un sujeto sano.

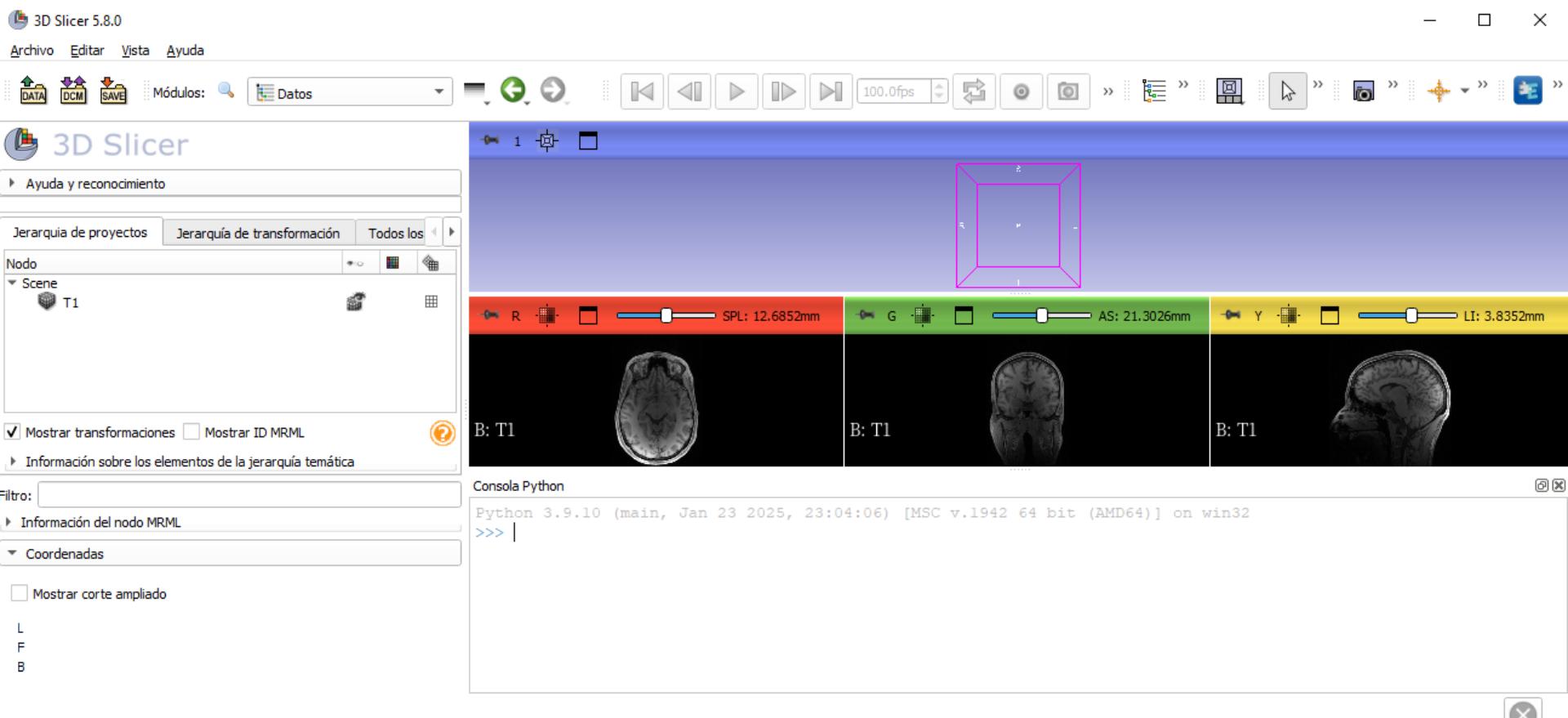
# Conjunto de datos



Arrastre y suelte el archivo T1.nrrd

Haga clic en Aceptar para cargar el archivo en Slicer

# Tutorial dataset



# Panorama general

- Slicer es un software gratuito y de código abierto
- Hay miles de imágenes médicas sofisticadas disponibles en Internet que puede visualizar y analizar con 3D Slicer.

# Modelo de datos de Slicer



El modelo de datos de Slicer se basa en la estructura de datos de escena de Slicer.



Una escena Slicer es una colección de imágenes, anotaciones, modelos 3D, transformaciones espaciales, fiduciales y cámaras.



El lenguaje MRML (Medical Reality Markup Language) es un lenguaje basado en XML que se utiliza para serializar el contenido de la escena Slicer en disco (scene.mrml).



Cada elemento de una escena se denomina nodo MRML

# Nodos MRML de Slicer: Tipos Básicos



Nodo de datos:  
Almacena los datos  
en bruto.

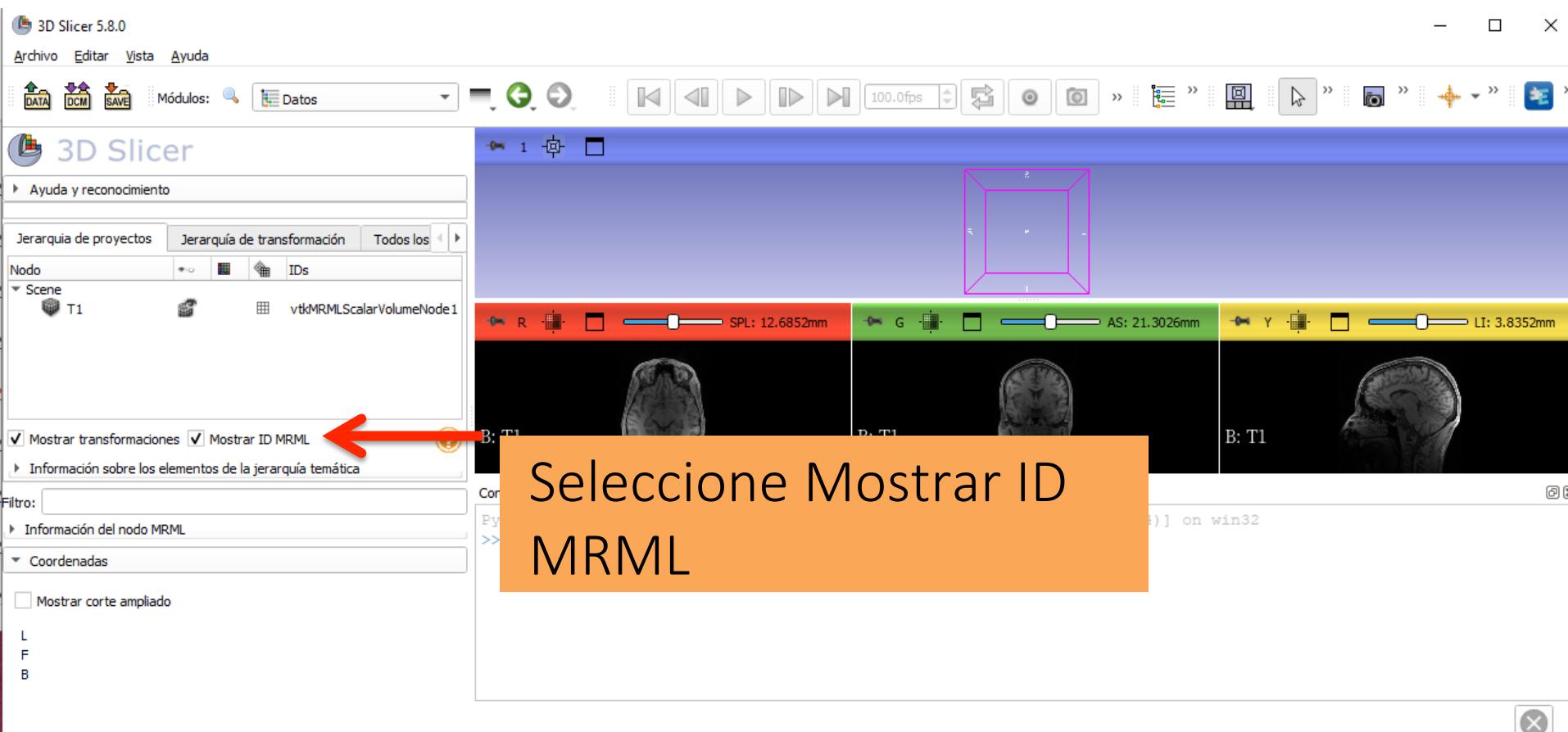


Nodo de  
visualización:  
Describe cómo  
deben visualizarse los  
datos.



Nodo de  
almacenamiento:  
Describe cómo deben  
almacenarse los  
datos en el disco.

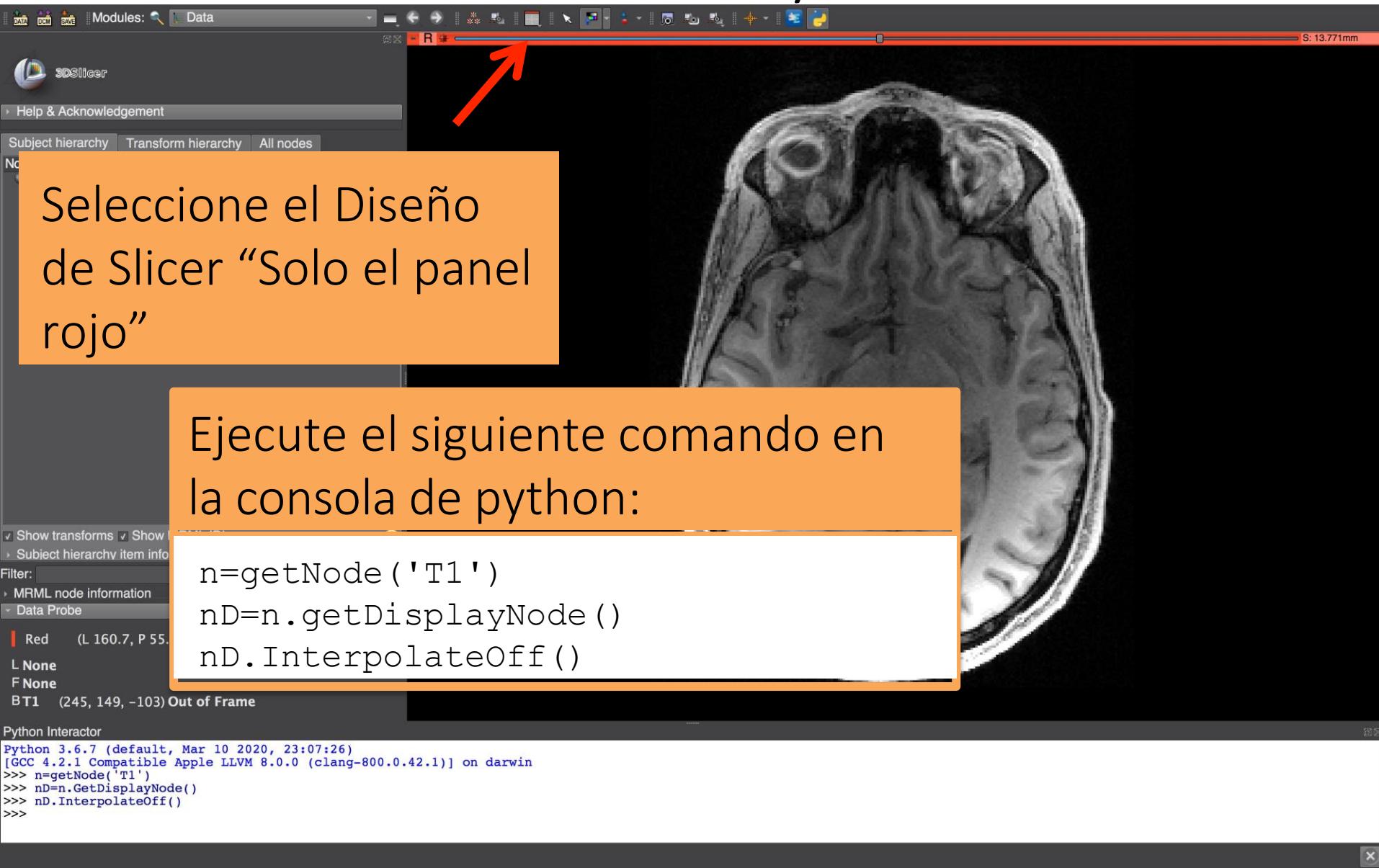
# Conjunto de datos del tutorial



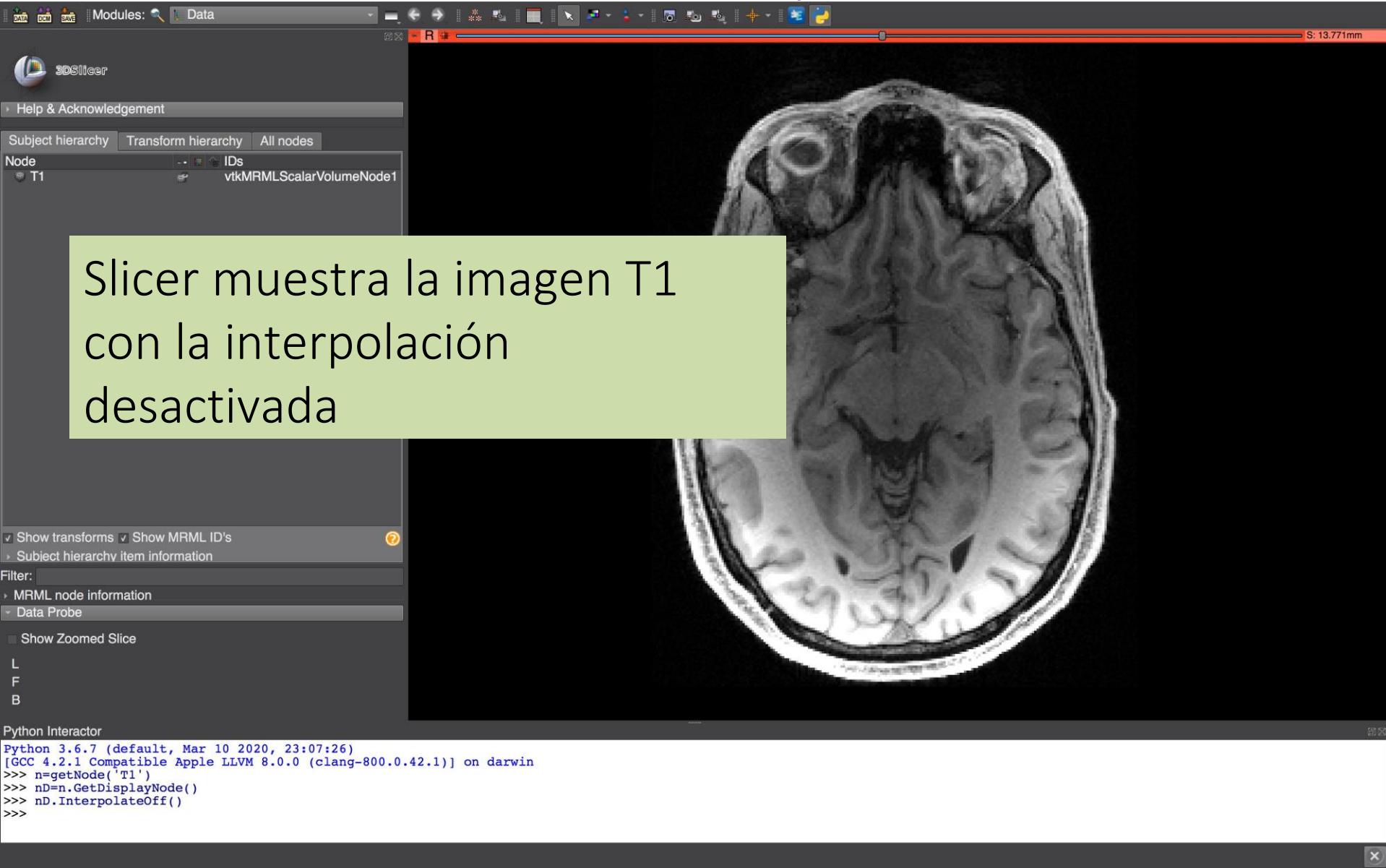
# Slicer Data Model



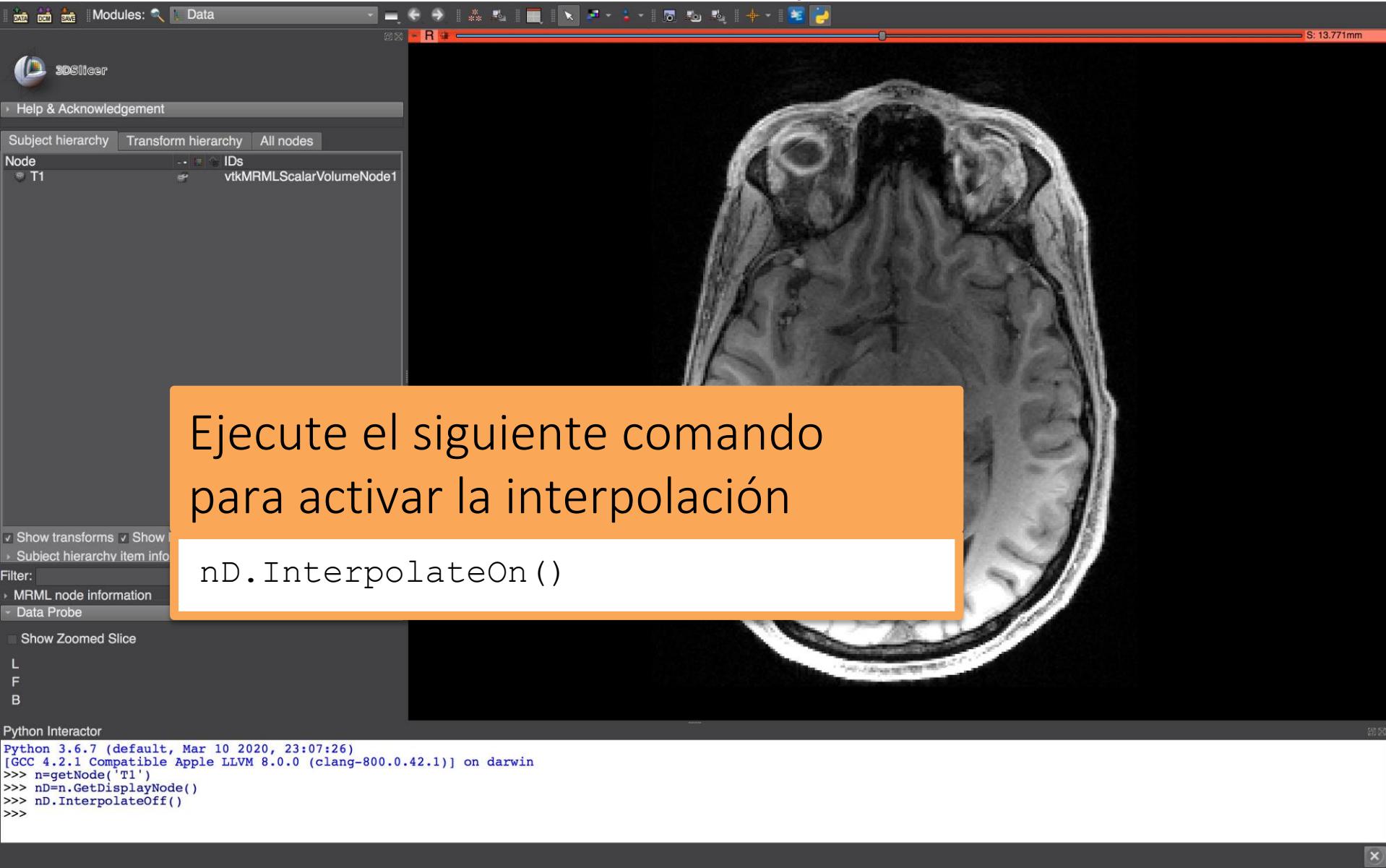
# Acceso a los nodos MRML desde la consola Python



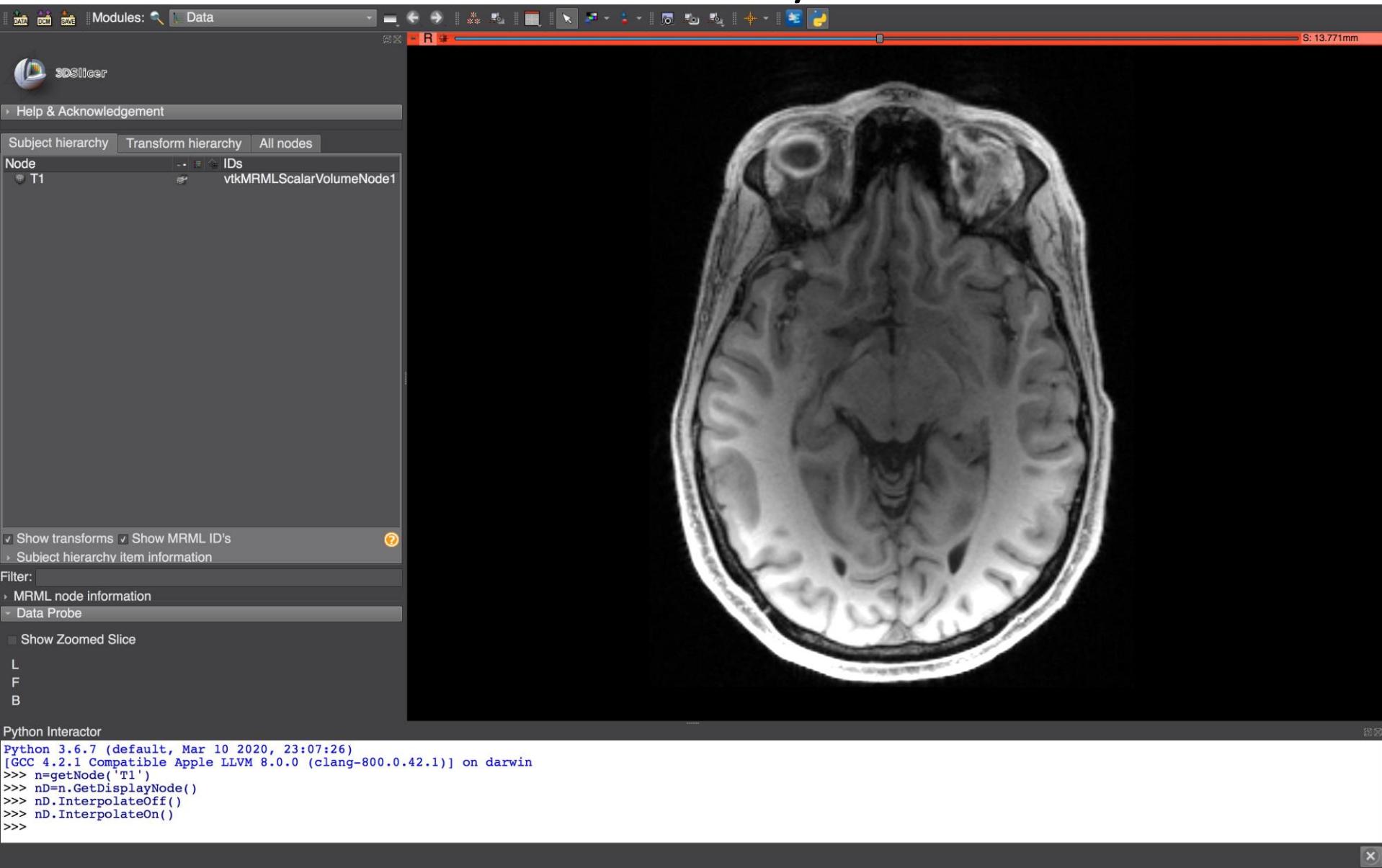
# Acceso a los nodos MRML desde la consola Python



# Acceso a los nodos MRML desde la consola Python



# Acceso a los nodos MRML desde la consola Python



# Acceso a los vóxeles de un volumen

- El paquete `slicer.util` da acceso a los volúmenes como matrices multidimensionales NumPy
- Los volúmenes pueden modificarse utilizando métodos estándar de NumPy



# Acceso a los vóxeles de un volumen

The screenshot shows the 3DSlicer application interface. On the left, the 'Subject hierarchy' panel lists a node named 'T1'. In the center, a 3D rendering of a brain scan is displayed with three orthogonal slices (R, Y, G) at the bottom. The top right contains two orange-highlighted code snippets. The first snippet is:

```
a=slicer.util.array('T1')
print(a)
```

The second snippet is:

```
a=array('T1') ;# same as above
print(a)
```

Below the code snippets, a note states: 'Nota: en la consola de Python, slicer.util se importa automáticamente'.

At the bottom, the Python Interactor window shows the following command history:

```
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)] on darwin
>>> n=getNode('T1')
>>> nD=n.GetDisplayNode()
>>> nD.InterpolateOff()
>>> nD.InterpolateOn()
>>>
>>> a = slicer.util.array('T1')
>>> print(a)
```

# Acceso a los vóxeles de un volumen

The screenshot shows the Slicer medical image viewer interface. On the left, the Python Interactor window displays the following code and its output:

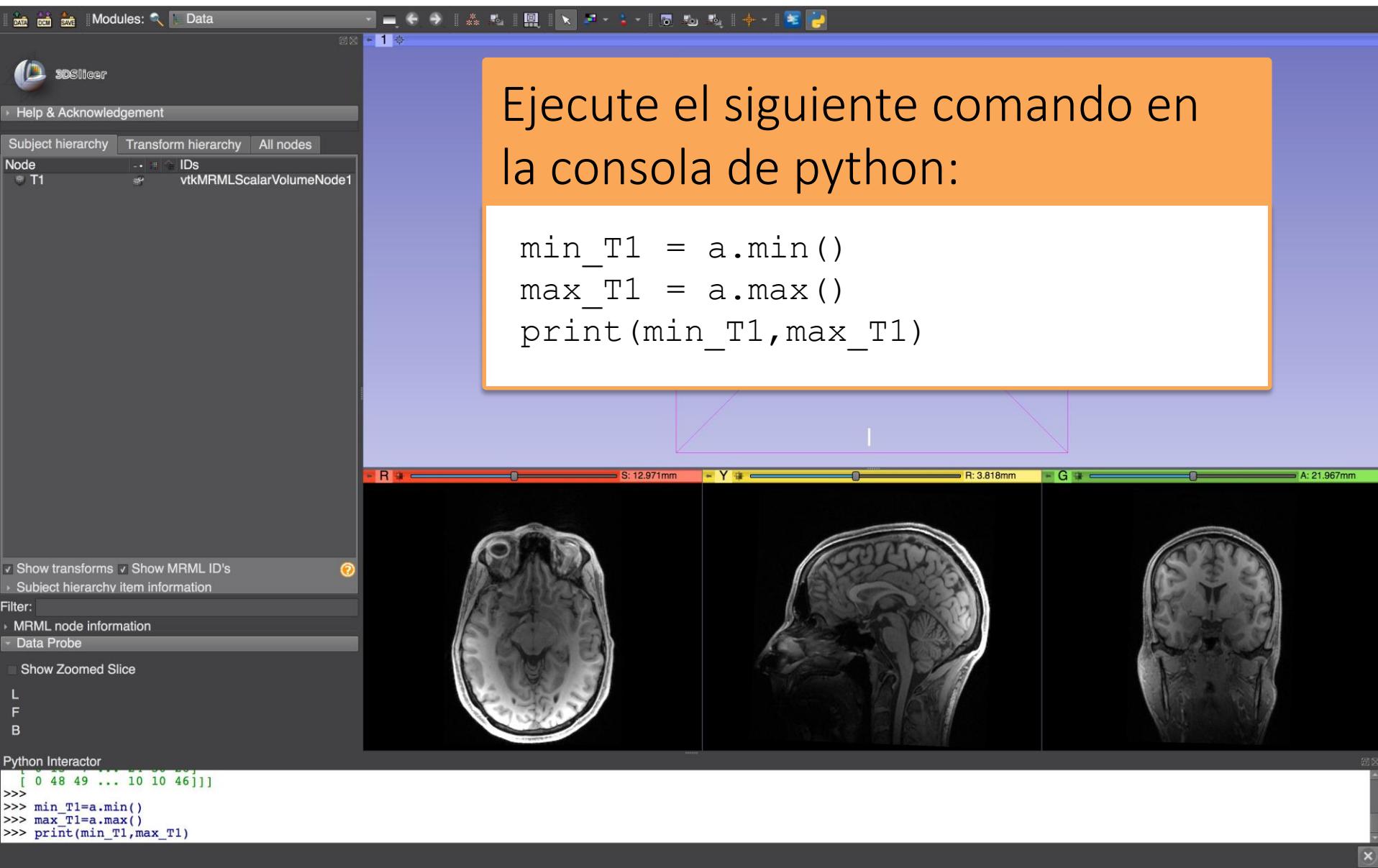
```
>>> a = slicer.util.array('T1')
>>> print(a)
[[[ 0  0  0 ...  0  0  0]
 [ 0 20  6 ... 10 52 27]
 [ 0 24 25 ...  4 32  8]
 ...
 [[ 0 48 14 ... 41 42 21]
 [ 0 15 40 ... 33 38 25]
 [ 0 55 19 ... 21  7 17]]
 ...
 [[ 0  0  0 ...  0  0  0]
 [ 0  4 14 ... 30 17 42]
 [ 0 22  9 ... 11 12 49]
 ...
 [[ 0 86 18 ... 16 66 11]
 [ 0 48 26 ... 14 23 21]
 [ 0 16  3 ... 31 14 33]]
 ...
 [[ 0  0  0 ...  0  0  0]
 [ 0 60 39 ... 7 28 10]
 [ 0 58 19 ... 34 31 29]
 ...
 [[ 0  5 48 ... 39 21 38]
 [ 0 22 55 ... 14 46 15]
 [ 0 17 45 ... 26 20 43]]
 ...
 [[ 0  0  0 ...  0  0  0]
 [ 0  8 26 ... 33 36 44]
 [ 0 27 18 ... 21 21 45]
 ...
 [[ 0 12 22 ... 22 34 14]
 [ 0  2 11 ... 48 65 35]
 [ 0 25  7 ... 7 17 11]]
 ...
 [[ 0  0  0 ...  0  0  0]
 [ 0 34 44 ... 13 41 30]
 [ 0 23 24 ... 28 51 33]
 ...
 [[ 0 18 36 ... 50 14 54]
 [ 0 17 34 ... 42 16 53]
 [ 0 12 30 ... 45 51 36]]
 ...
 [[ 0  0  0 ...  0  0  0]
 [ 0  5 41 ... 11  9 48]
 [ 0 21 64 ... 32 11  9]
 ...
 [[ 0 11 33 ... 30 11 43]
 [ 0 13  7 ... 24 30 26]
 [ 0 48 49 ... 10 10 46]]]
```

The main 3D view shows a grayscale brain scan with a 3D coordinate system overlaid. The R axis points upwards, the S axis points to the right, and the A axis points towards the viewer. The Python console also shows the following status bar text: "S: 13.771mm" and "A: 21.967mm".

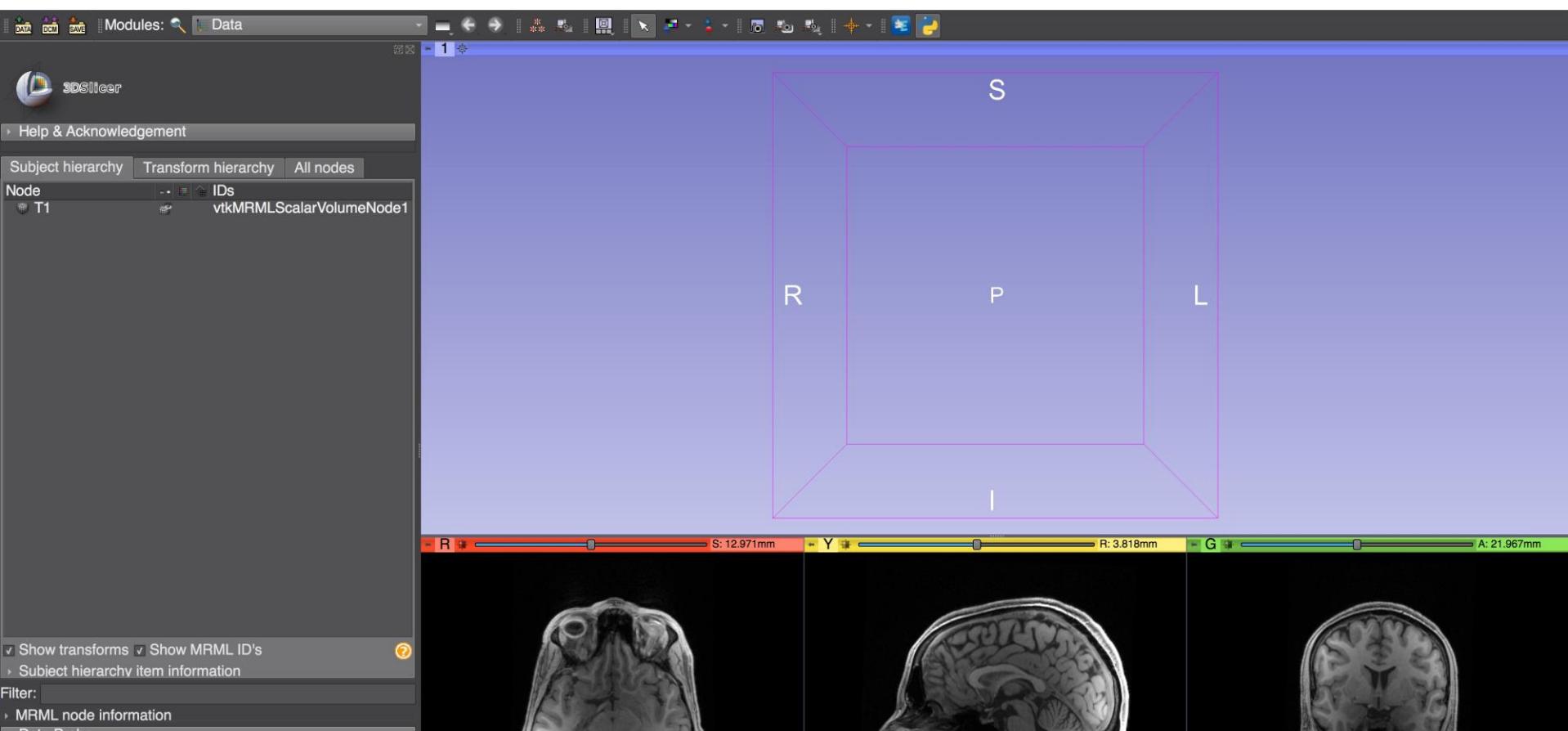
**Slicer imprime los valores de intensidad de la imagen T1**

**La consola python muestra una visualización truncada de una matriz 3D**

# Acceso a los vóxeles de un volumen



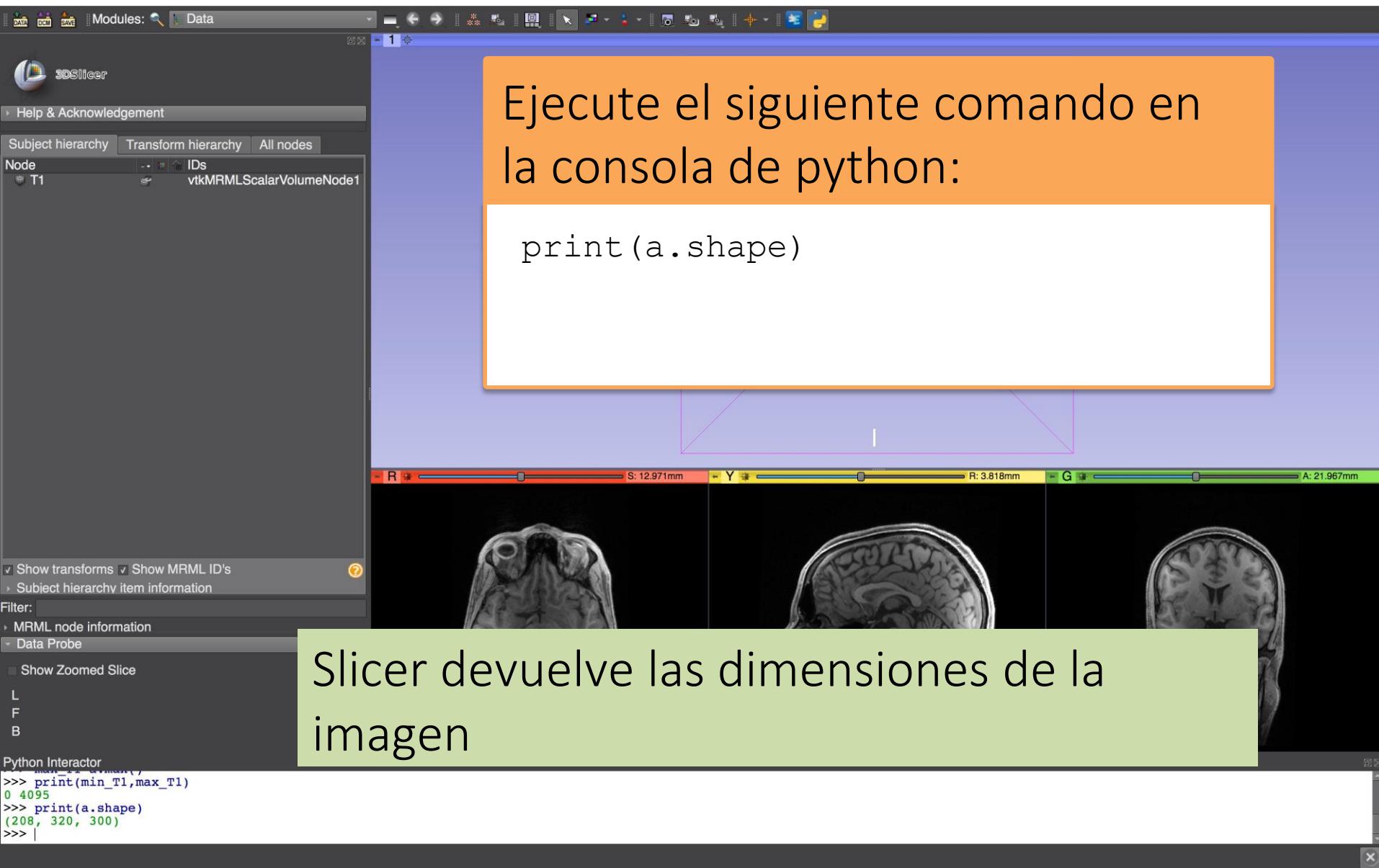
# Acceso a los vóxeles de un volumen



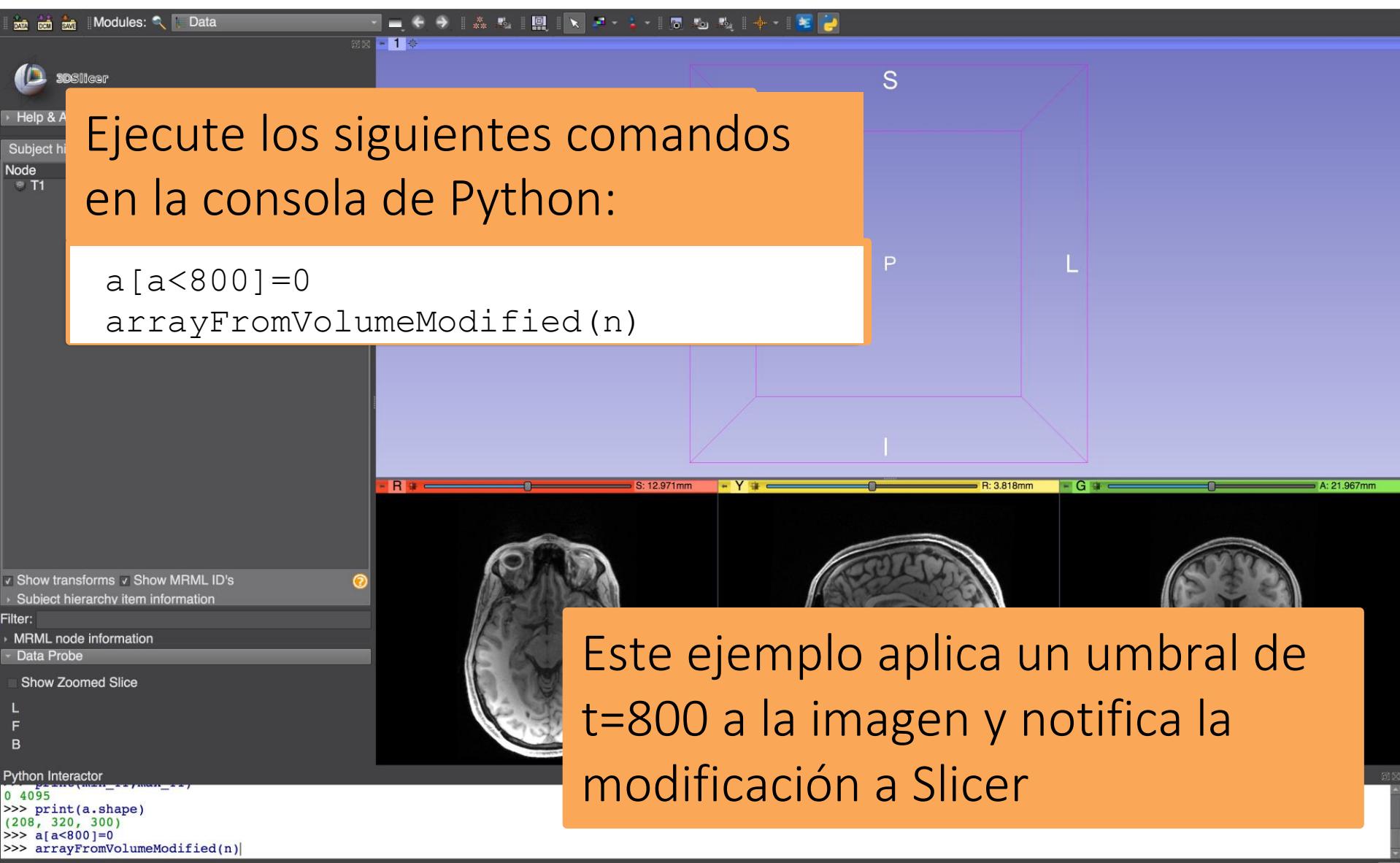
Slicer devuelve los valores mínimo y máximo de la imagen T1

```
>>> min_T1=a.min()  
>>> max_T1=a.max()  
>>> print(min_T1,max_T1)  
0 4095  
>>>
```

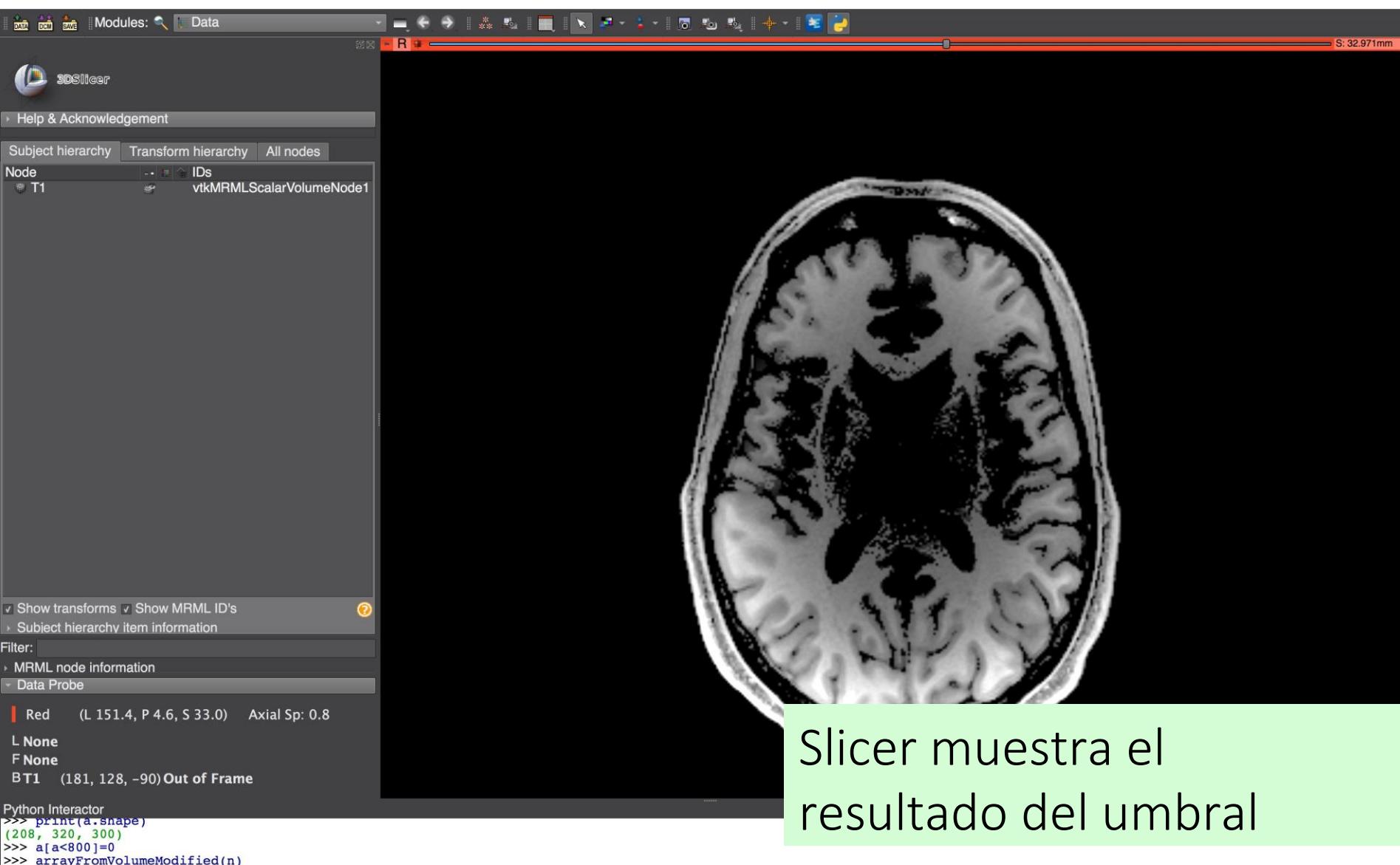
# Modificación de vóxeles en un volumen



# Modificación de vóxeles en un volumen

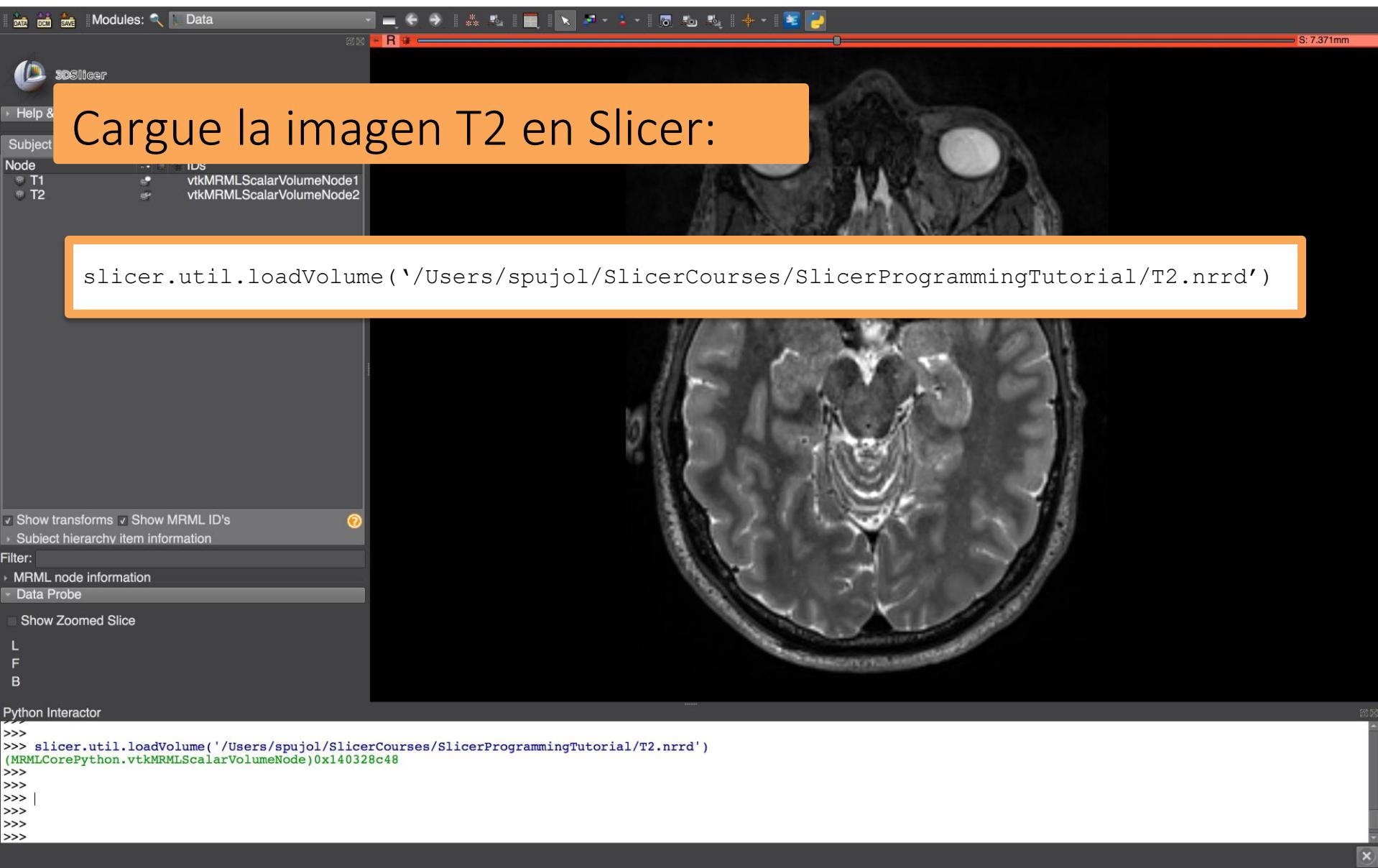


# Modificación de vóxeles en un volumen



Slicer muestra el resultado del umbral

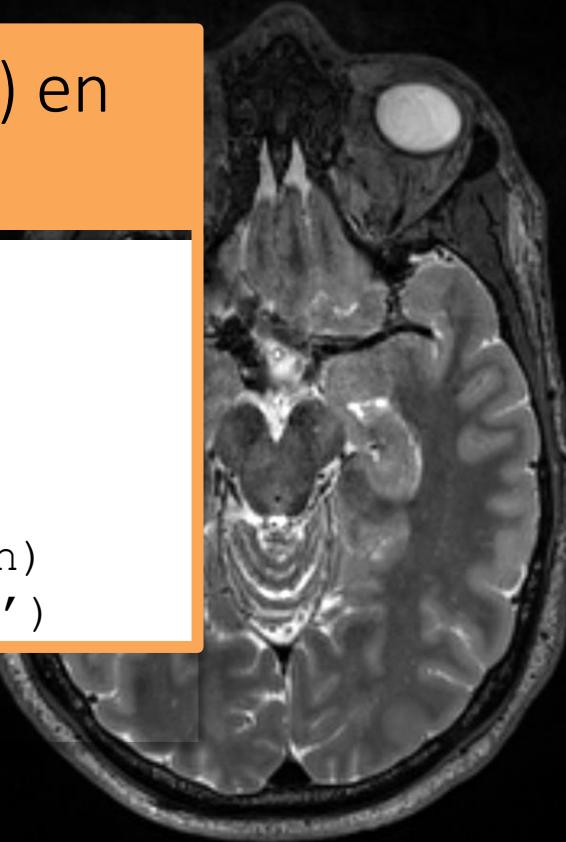
# Cargar el volumen T2



# Función Python: umbral

Crear una función umbral( $t$ ) en la consola Python

```
def threshold(t):  
  
    n=getNode('T2')  
  
    a=array('T2') a[a<t]=0  
        arrayFromVolumeModified(n)  
        print('Thresholding done')
```



SDSlicer

Modules: Data DCM Save R S: 7.371mm

Help & A

Subject hierarchy

Node

T1 T2

Show tra

Subject hierarchy item information

Filter:

MRML node information

Data Probe

Red (R 145.5, P 86.4, S 7.4) Axial Sp: 0.8

L None

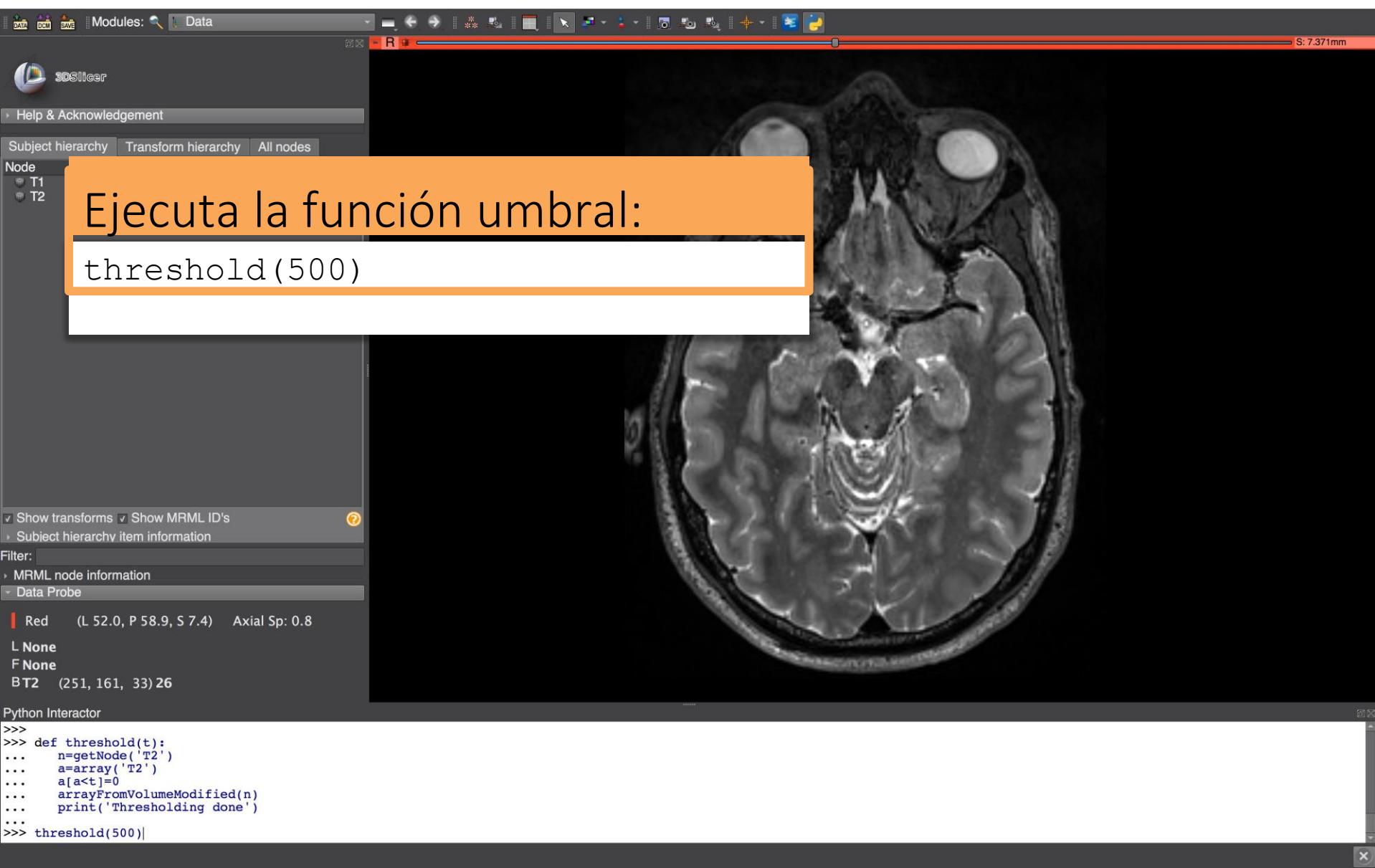
F None

BT2 (286, 167, 280) Out of Frame

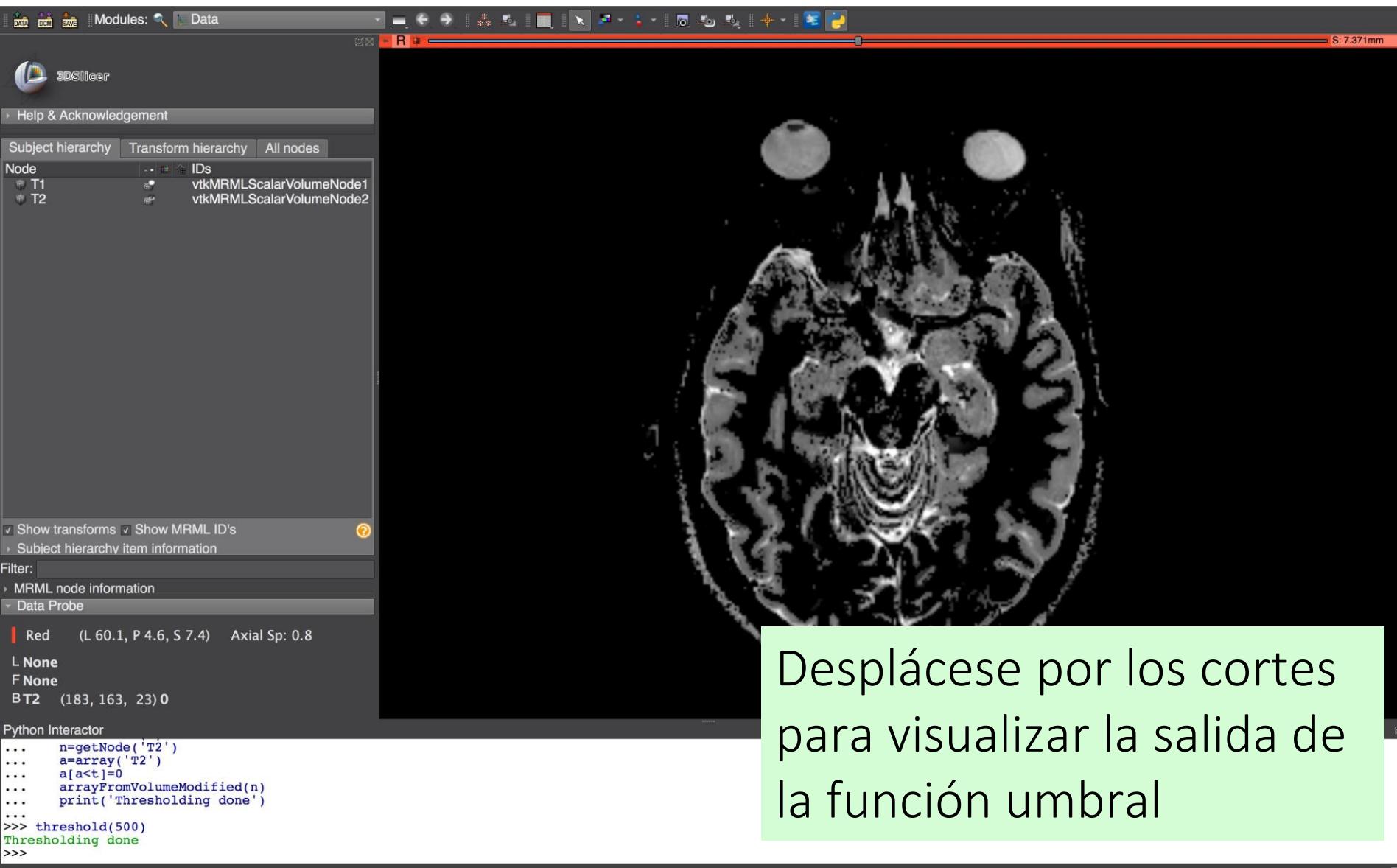
Python Interactor

```
>>>  
>>>  
>>> def threshold(t):  
...     n=getNode('T2')  
...     a=array('T2')  
...     a[a<t]=0  
...     arrayFromVolumeModified(n)  
...     print('Thresholding done')  
...
```

# Función Python: umbral



# Función Python: umbral



Desplácese por los cortes  
para visualizar la salida de  
la función umbral

# Panorama General

- Slicer facilita el acceso para analizar y modificar tipos de datos complejos
- Slicer es compatible con una amplia gama de paquetes informáticos científicos de Python.
- Slicer es un entorno de investigación para realizar experimentos con imágenes médicas

## Parte 3

Familiarizarse con Qt en Slicer

# Qt & PyQt

- Qt es la herramienta principal de Slicer para crear widgets, cuadros de diálogo, entradas de texto, etc.
- PyQt expone la mayoría de las funcionalidades de Qt y es accesible a través del interactuador Python en Slicer.
- Las interfaces de usuario se pueden crear sobre la marcha para una rápida creación de prototipos y depuración.

# Función Python: toggle



Crea una función toggle() en el interactor Python:

```
def toggle():
    n=getNode("T1")
    a=array("T1")
    a[a<0] = 0
    a[a>1000] = 700
    a[:]=a.max()-a
    arrayFromVolumeModified(n)
```

# Creación de un botón Qt

The screenshot shows the 3DSlicer interface with the Python Interactor tab selected. On the left, the 3D Slicer node browser displays nodes T1 and T2 under the Subject hierarchy. The main area contains a code snippet for creating a QtPushButton:

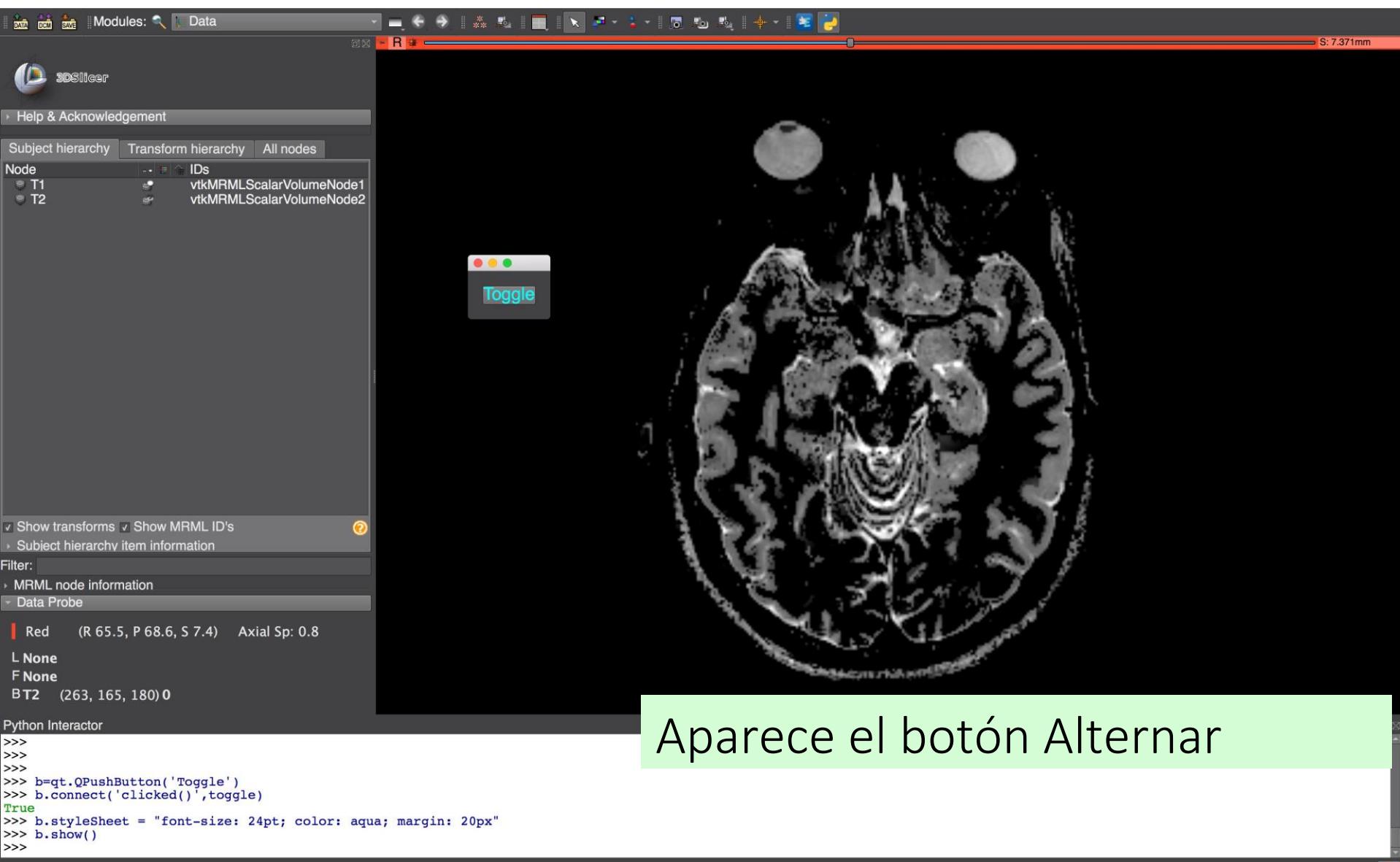
```
b=qt.QPushButton('Toggle')
b.connect('clicked()',toggle)
b.setStyleSheet = "font-size: 24pt;
color: aqua; margin: 20px"
b.show()
```

A callout box highlights the `styleSheet` line with the text "styleSheet es css".

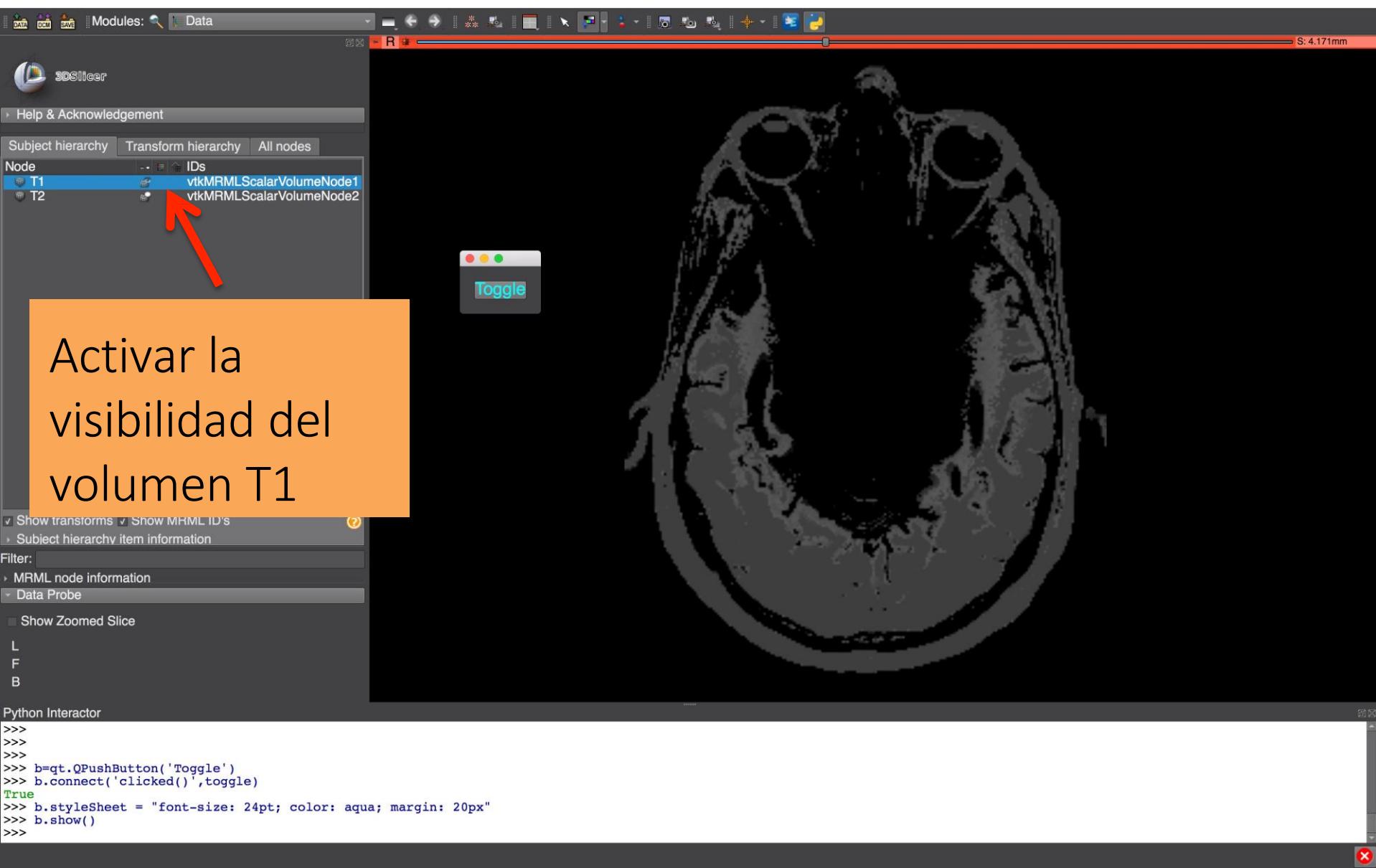
Python Interactor history:

```
>>>
>>>
>>>
>>>
>>> b=qt.QPushButton('Toggle')
>>> b.connect('clicked()',toggle)
True
>>> b.setStyleSheet = "font-size: 24pt; color: aqua; margin: 20px"
>>> b.show()
```

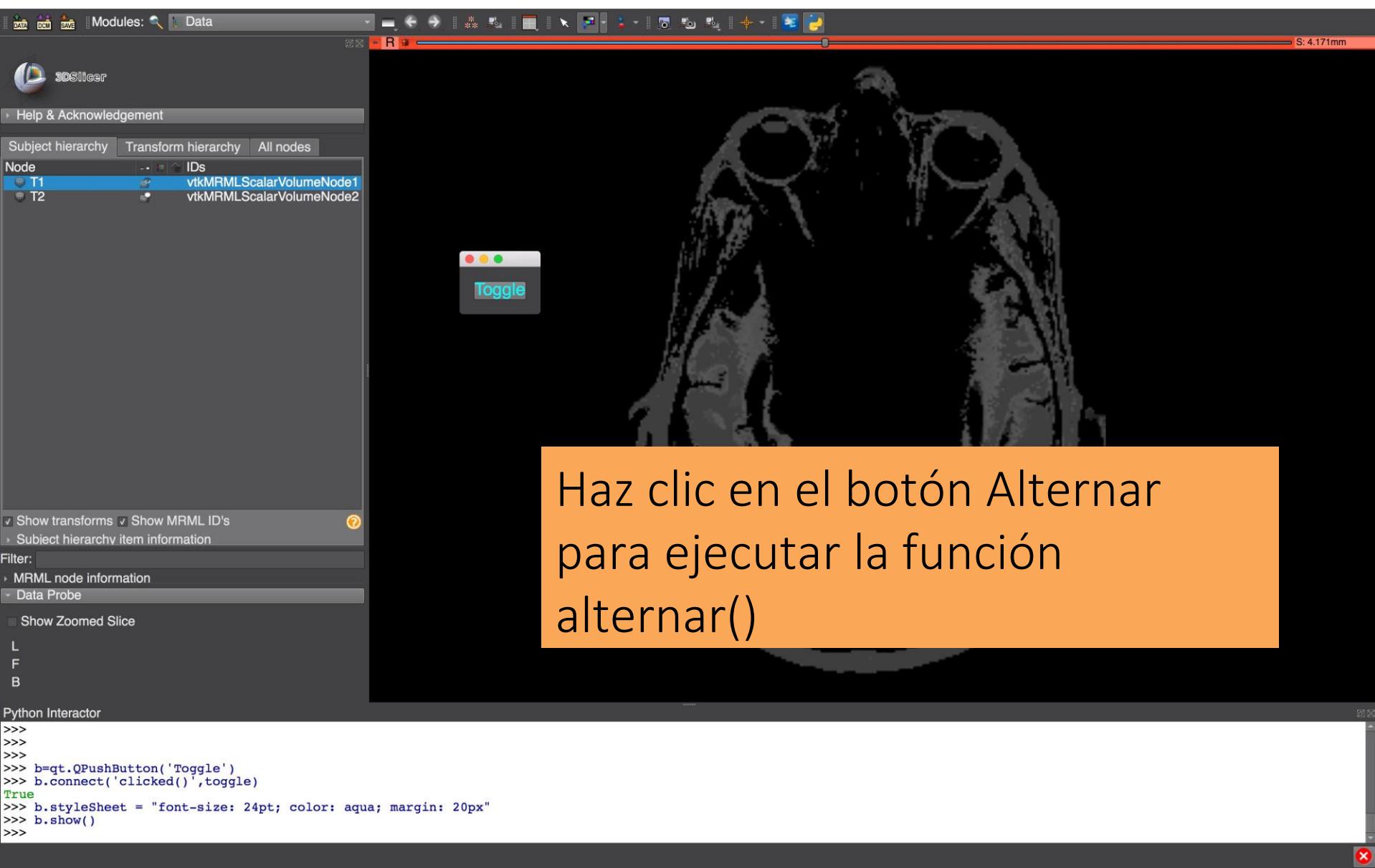
# Creación de un botón Qt



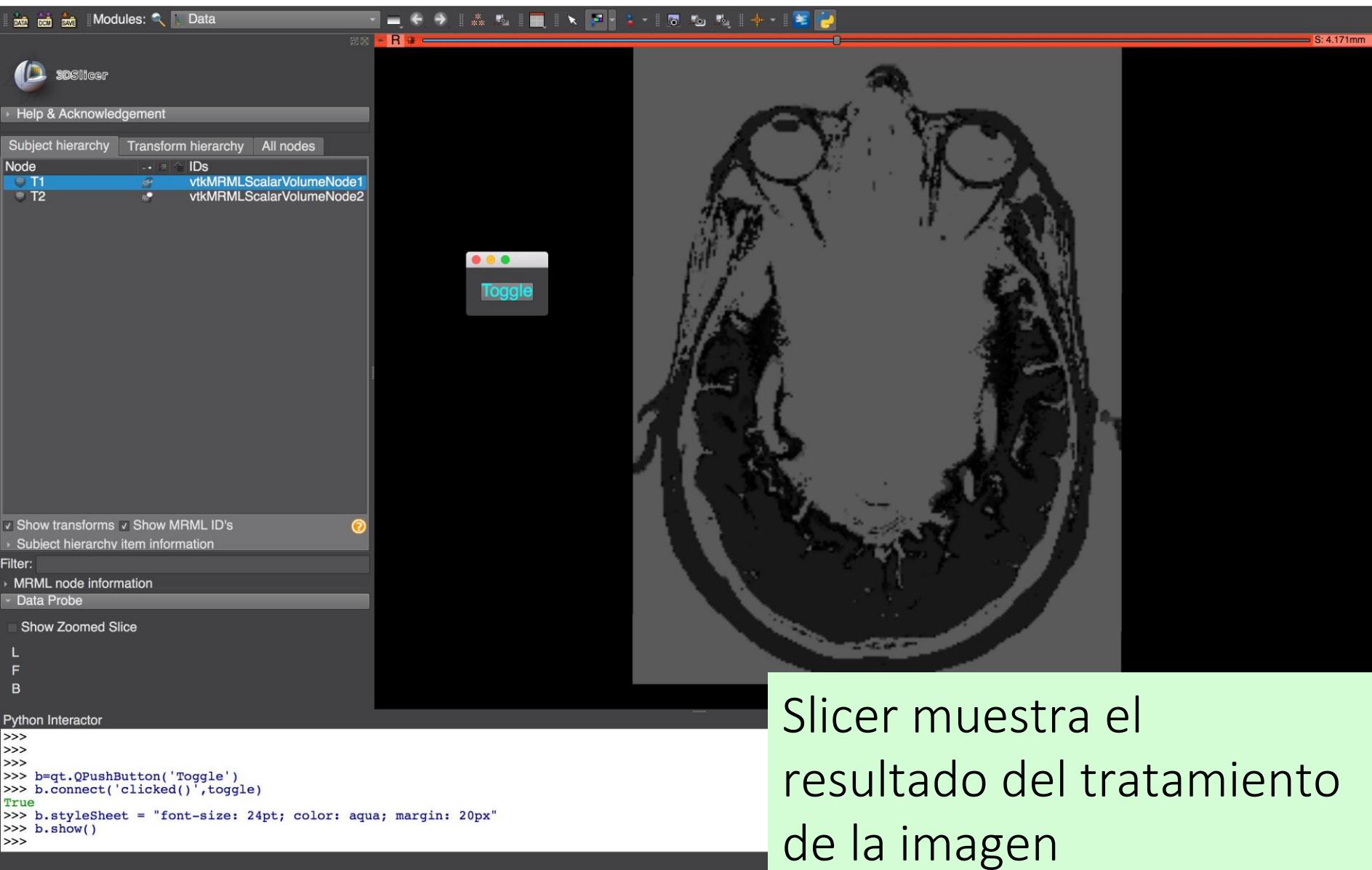
# Creación de un botón Qt



# Creación de un botón Qt



# Creación de un botón Qt



# Ejemplos de módulos con secuencias de comandos

- El tutorial muestra cómo crear una interfaz sencilla en Python.
- Slicer integra muchos módulos de secuencias de comandos sofisticados, como estadísticas de segmentos, datos de muestras, módulo de endoscopia, etc.
- Para más información, consulte el repositorio de secuencias de comandos de Slicer:

<https://www.slicer.org/wiki/Documentation/Nightly/ScriptRepository>

# Conclusión

- Slicer permite a los desarrolladores crear interfaces complejas y ágiles para los usuarios
- La plataforma de software ofrece posibilidades de personalización ilimitadas
- Slicer da acceso a bibliotecas subyacentes avanzadas a través de un paquete multiplataforma que es fácil de desplegar a los usuarios

# Agradecimientos



Neuroimage Analysis Center  
(NIBIB P41 EB015902)

**PNL** Sylvain Bouix, Ph.D.  
Psychiatry Neuroimaging Laboratory