



Paul Cézanne, *Moulin sur la Coulevre à Pontoise, 1881*,  
Staatliche Museen zu Berlin, Nationalgalerie

Programming into  
Slicer3:

The “Hello Python”  
Tutorial

Sonia Pujol, Ph.D.

Surgical Planning Laboratory  
Harvard University

---



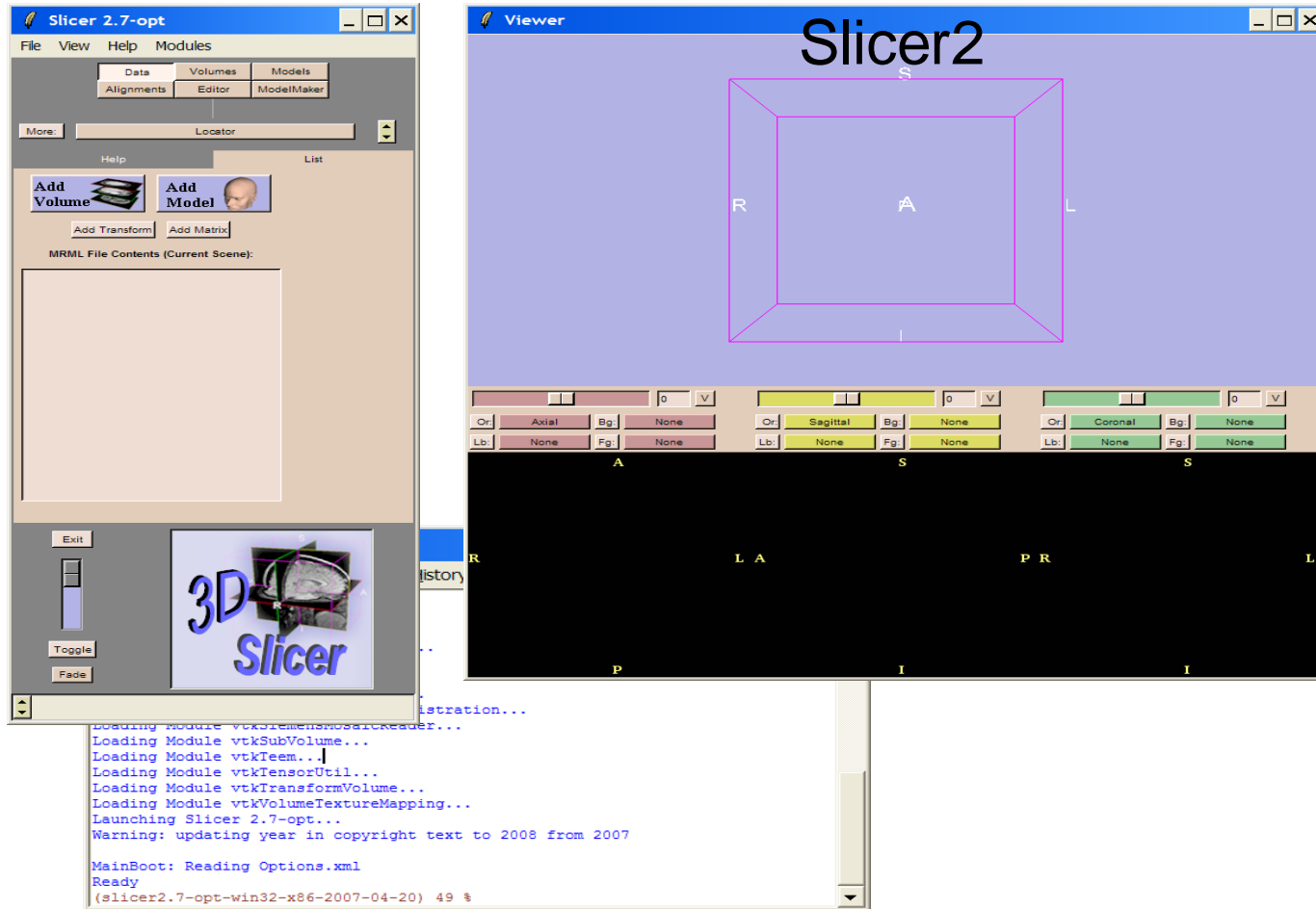
# The NA-MIC Kit





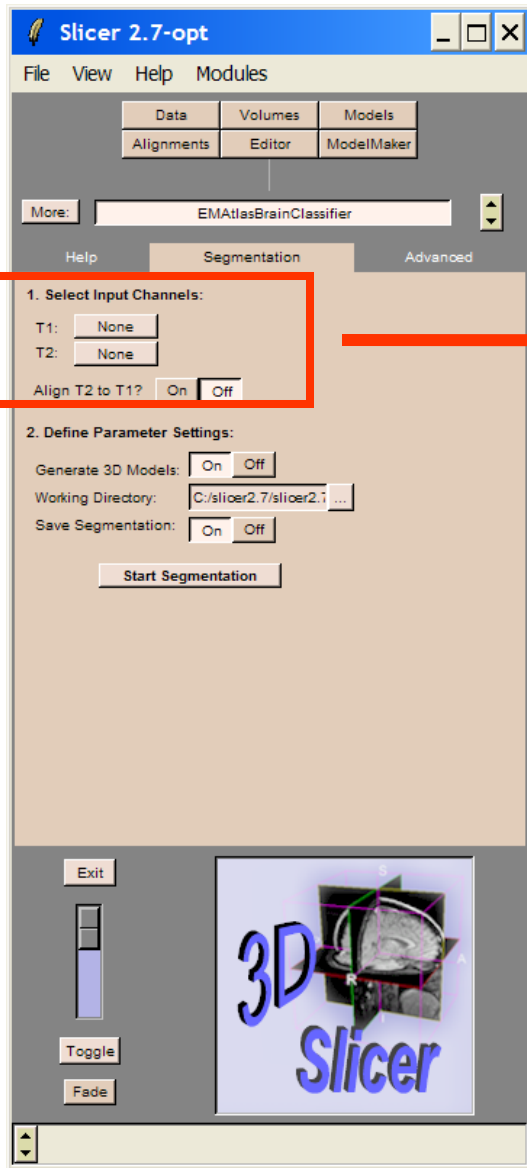
- An **end-user application** for image analysis
- An **open-source environment** for software development
- A software platform that is both **easy to use** for clinical researchers and **easy to extend** for programmers

# Before Slicer3



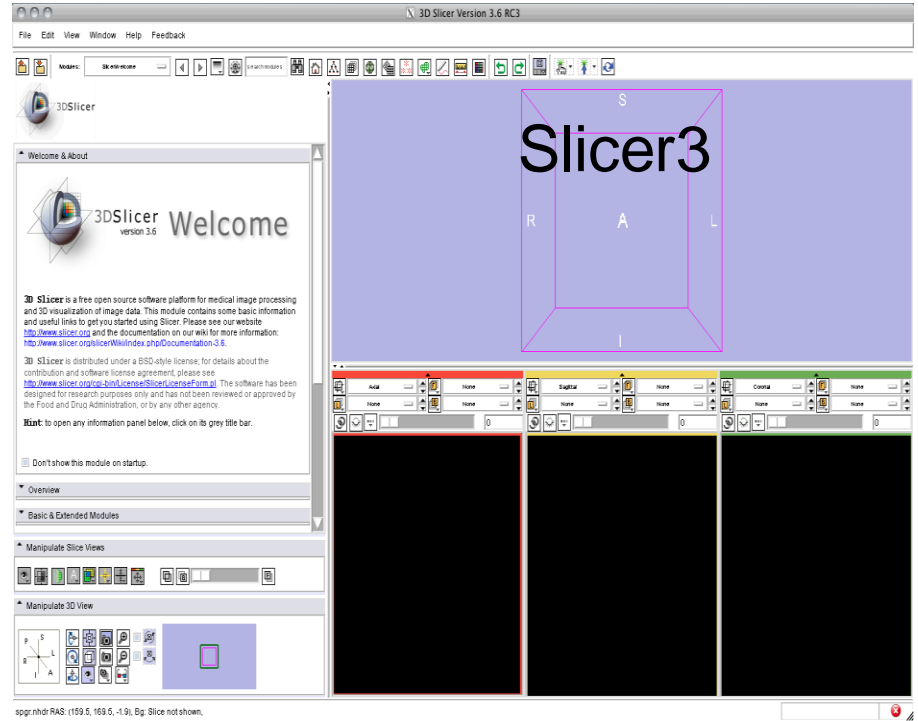
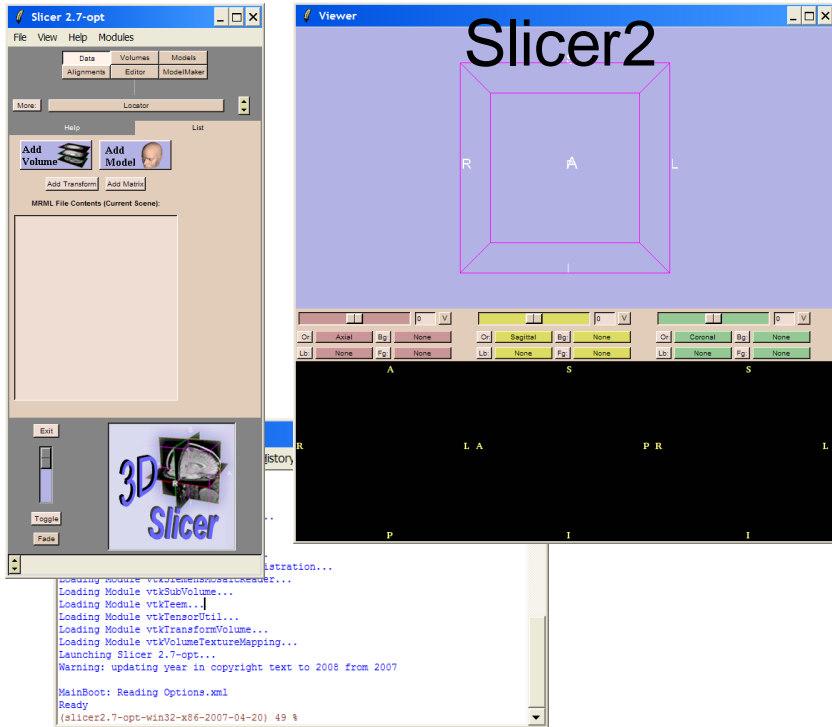


# Programming into Slicer2



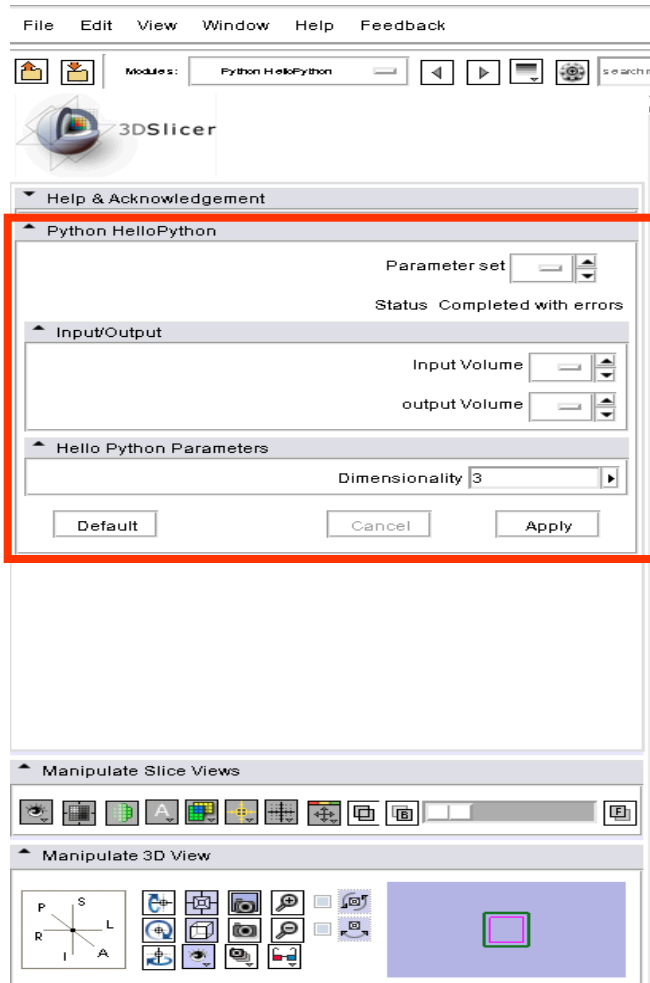
```
#-----
# 1. Step
#-----
set f $fSeg.fStep1
DevAddLabel $f.ITitle "1. Select Input Channels: " WTA
pack $f.ITitle -side top -padx $Gui(pad) -pady 1 -anchor w
frame $f.fInput -bg $Gui(activeWorkspace)
pack $f.fInput -side top -padx 0 -pady 0 -anchor w
foreach frame "Left Right" {
    frame $f.fInput.f$frame -bg $Gui(activeWorkspace)
    pack $f.fInput.f$frame -side left -padx 0 -pady $Gui(pad) }
foreach LABEL "T1 T2" Input "SPGR T2W" {
    DevAddLabel $f.fInput.fLeft.I$Input " $(LABEL):"
    pack $f.fInput.fLeft.I$Input -side top -padx $Gui(pad) -pady 1 -anchor w
    set menubutton $f.fInput.fRight.m${Input}Select
    set menu $f.fInput.fRight.m${Input}Select.m
eval {menubutton $menubutton -text [Volume$(EMAtlasBrainClassifier(Volume,${Input}),node) GetName] -
relief raised -bd 2 -width 9 -menu $menu} $Gui(WMBA)
    eval {menu $menu} $Gui(WMA)
    TooltipAdd $menubutton "Select Volume defining ${Input}"
    set EMAtlasBrainClassifier(mbSeg-${Input}Select) $menubutton
    set EMAtlasBrainClassifier(mSeg-${Input}Select) $menu
    # Have to update at UpdateMRML too
    DevUpdateNodeSelectButton Volume EMAtlasBrainClassifier Seg-${Input}Select Volume,$Input
    pack $menubutton -side top -padx $Gui(pad) -pady 1 -anchor w }
frame $f.fAlign -bg $Gui(activeWorkspace)
TooltipAdd $f.fAlign "If the input T1 and T2 are not aligned with each other set flag here"
pack $f.fAlign -side top -padx 0 -pady 2 -padx $Gui(pad) -anchor w
DevAddLabel $f.fAlign.I$Align "Align T2 to T1? "
pack $f.fAlign.I$Align -side left -padx $Gui(pad) -pady 1 -anchor w
foreach value "1 0" text "On Off" width "4 4" {
    eval {radiobutton $f.fAlign.r$value -width $width -indicatoron 0
-text "$text" -value "$value" -variable EMAtlasBrainClassifier(AlignInput)} $Gui(WCA)
    pack $f.fAlign.r$value -side left -padx 0 -pady 0 }
```

# From Slicer2 to Slicer3





# The New Execution Model



```
<?xml version="1.0" encoding="utf-8"?>
<executable>
  <category> Demonstration </category>
  <title> Python HelloPython </title>
  <description> Slicer Developer Course </description>
  <version> 1.0 </version>
  <documentation-url> </documentation-url>
  <license></license>
  <contributor>
    Sonia Pujol, Ph.D., Surgical Planning Laboratory, Harvard Medical School
  </contributor>
  <acknowledgements> National Alliance for Medical Image Computing (NAMIC), Grant
  U54 EB005149. </acknowledgements>
  <parameters>
    <label>Input/Output</label>
    <description>Input/output parameters</description>
    <image>
      <name>helloPython</name>
      <label>Input Volume</label>
      <channel>input</channel>
      <index>0</index>
      <default>None</default>
      <description>Input volume</description>
    </image>
    <image>
      <name>helloPythonOutputVolume</name>
      <label>Output Volume</label>
      <channel>output</channel>
      <index>1</index>
      <default>None</default>
      <description>Output filtered</description>
    </image>
  </parameters>
</executable>
```



# Slicer3 Execution Model

---

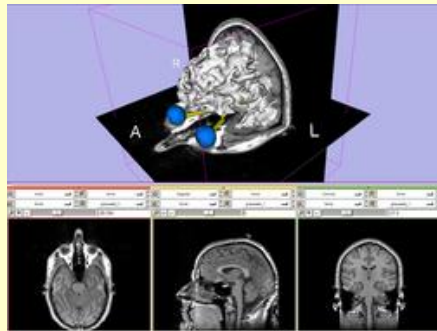
- This course is based on the [Execution Model](#) which provides a mechanism for incorporating command line programs as Slicer modules.
- Jim Miller, Dan Blezek, Bill Lorensen (GE)
- This course uses the Python interpreter that has been integrated to Slicer.





# Pre-requisite

- This course supposes that you have taken the following tutorial:



*Slicer3 Data Loading and Visualization, Sonia Pujol*

- The tutorial is available on the Slicer3.6 101 compendium:  
[http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training#Software\\_tutorials](http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training#Software_tutorials)

This course requires the following material

- Slicer3-3.6.1 release version

<http://www.slicer.org/pages/Special:SlicerDownloads>

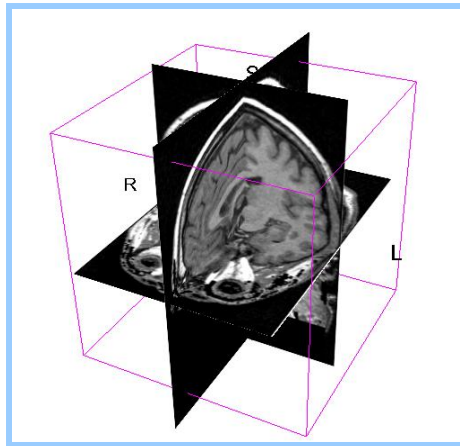
- HelloPython.zip

[http://www.slicer.org/slicerWiki/index.php/Slicer3.6:TrainingSoftware\\_tutorials](http://www.slicer.org/slicerWiki/index.php/Slicer3.6:TrainingSoftware_tutorials)

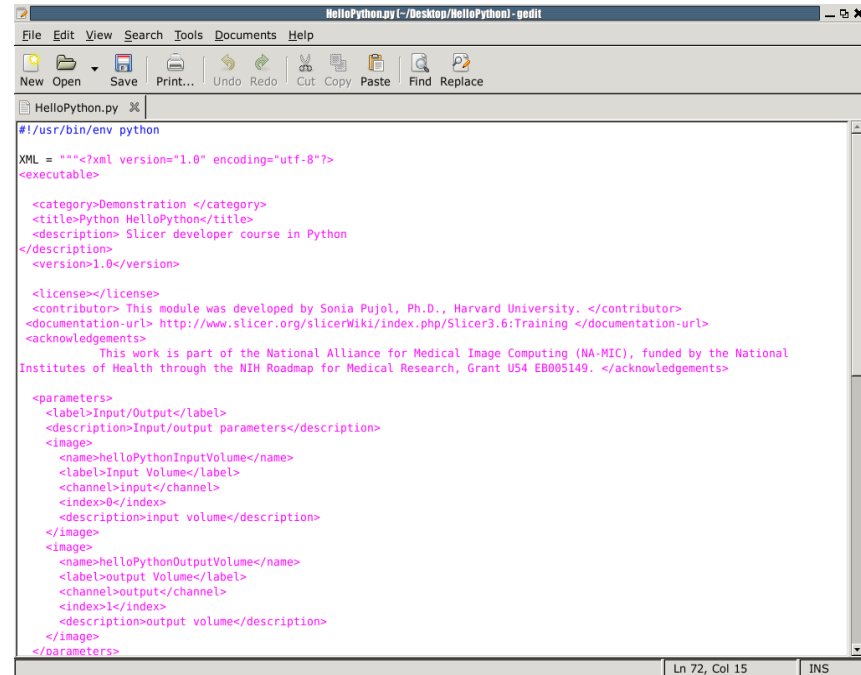
## **Disclaimer**

It is the responsibility of the user of 3DSlicer to comply with both the terms of the license and with the applicable laws, regulations and rules.

## Unzip the HelloPython.zip archive



spgr.nhdr spgr.raw.gz  
(124 SPGR images)



```
#!/usr/bin/env python

XML = """<?xml version="1.0" encoding="utf-8"?>
<executable>

  <category>Demonstration </category>
  <title>Python HelloPython</title>
  <description> Slicer developer course in Python
</description>
  <version>1.0</version>

  <license></license>
  <contributor> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
  <documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
  <acknowledgements>
    This work is part of the National Alliance for Medical Image Computing (NA-MIC), funded by the National
    Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. </acknowledgements>

  <parameters>
    <label>Input/Output</label>
    <description>Input/output parameters</description>
    <image>
      <name>helloPythonInputVolume</name>
      <label>Input Volume</label>
      <channel>input</channel>
      <index>0</index>
      <description>input volume</description>
    </image>
    <image>
      <name>helloPythonOutputVolume</name>
      <label>output Volume</label>
      <channel>output</channel>
      <index>1</index>
      <description>output volume</description>
    </image>
  </parameters>

"""
```

HelloPython.py



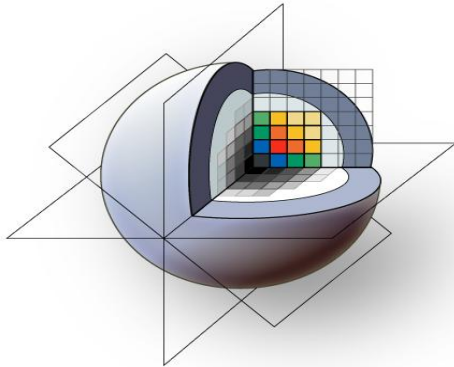
# Overview

---

- Part A: Integration of the HelloPython.py program into Slicer3
- Part B: Implementation of the Laplace operator in the HelloPython module
- Part C: Image Sharpening using the Laplace operator



3DSlicer

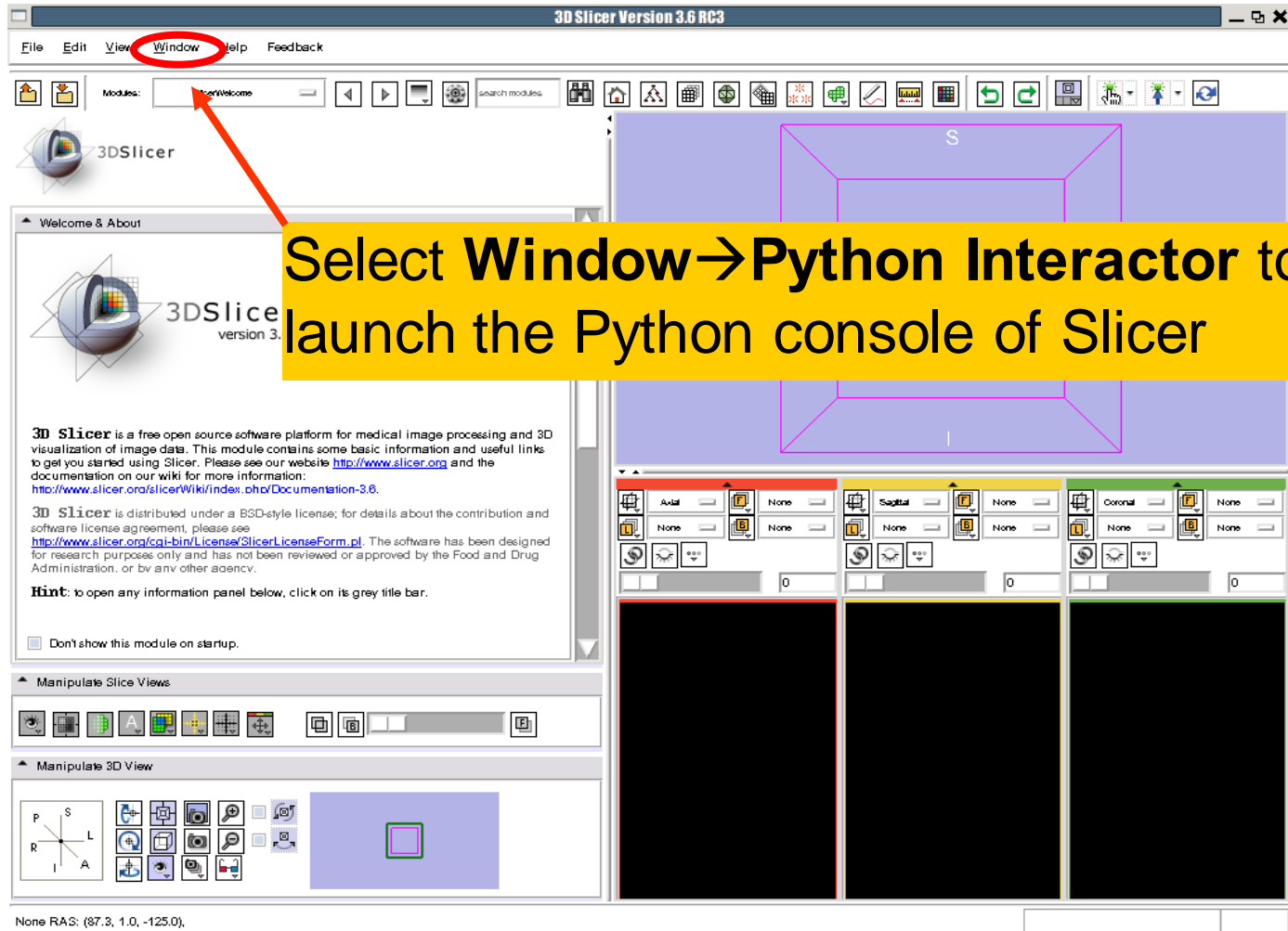


# 3DSlicer

## Part A: Integrating HelloPython into Slicer3

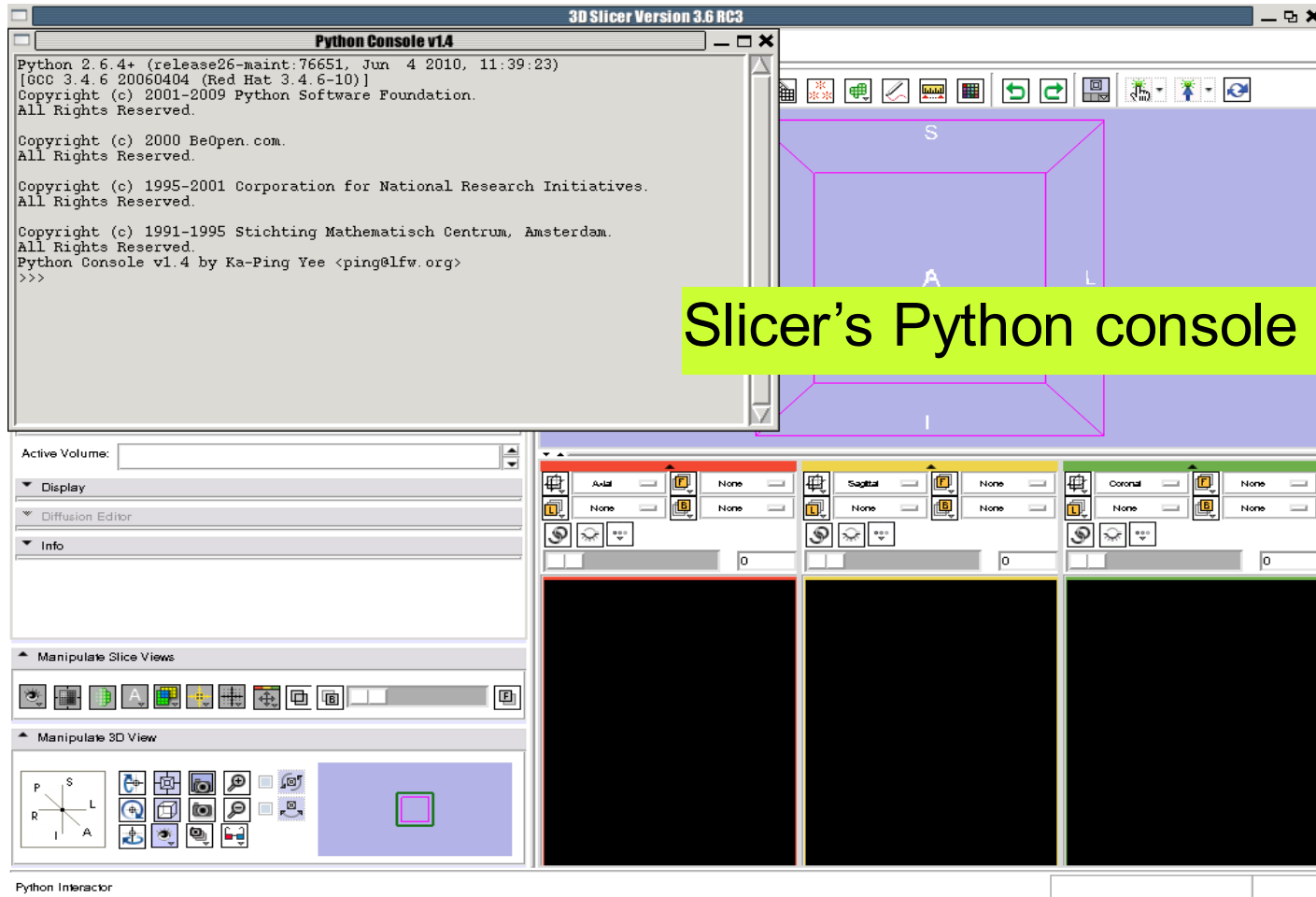
```
>HelloPython.py [C:/Desktop/HelloPython]-gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
HelloPython.py
#!/usr/bin/env python
XML - ***?xml version="1.0" encoding="utf-8"?
<executable>
<category>Demonstration </category>
<title>Python HelloPython</title>
<description> Slicer developer course in Python
</description>
<version>1.0</version>
<license></license>
<contributor> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
<documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
<acknowledgements>
This work is part of the National Alliance for Medical Image Computing (NA-MIC), funded by the National
Institutes of Health through the NIH Roadmap for Medical Research, Grant US4 EB005149. </acknowledgements>
<parameters>
<label>Input/Output</label>
<description>Input/output parameters</description>
<image>
<name>helloPythonInputVolume</name>
<label>Input Volume</label>
<channel>input</channel>
<index>0</index>
<description>input volume</description>
</image>
<image>
<name>helloPythonOutputVolume</name>
<label>output Volume</label>
<channel>output</channel>
<index>1</index>
<description>output volume</description>
</image>
</parameters>
```

# Python Console



The screenshot shows the 3D Slicer Version 3.6 RC3 interface. The 'Window' menu is circled in red, and a red arrow points to the 'Python Interactor' option. A yellow callout box contains the text: **Select Window → Python Interactor to launch the Python console of Slicer**. The interface includes a menu bar (File, Edit, View, Window, Help, Feedback), a toolbar, and a main workspace with three viewports (Axial, Sagittal, Coronal) and a 3D view. The 'Welcome & About' module is open, displaying information about 3D Slicer.

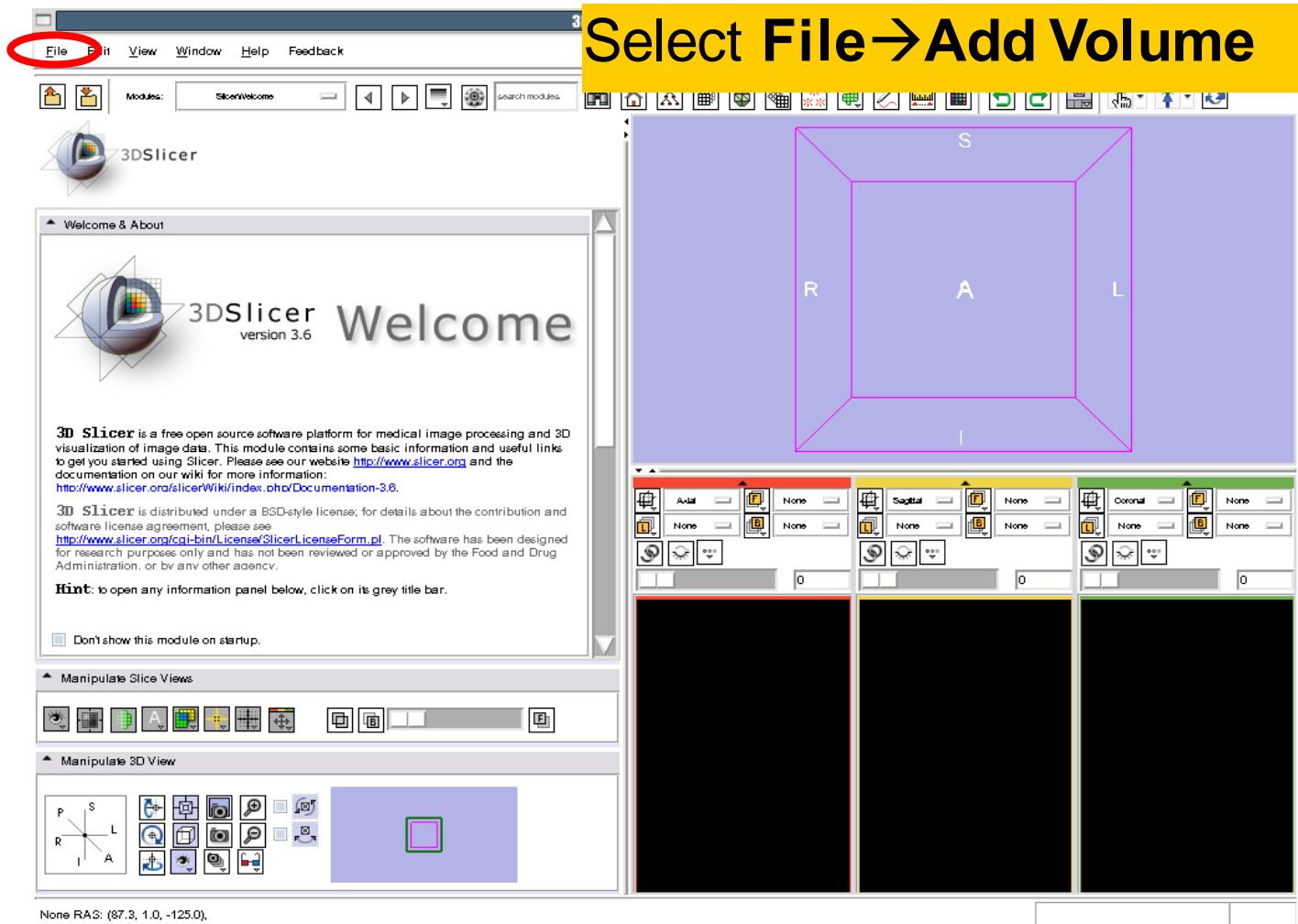
# Python Console



Slicer's Python console

# Python Console

**Select File → Add Volume**

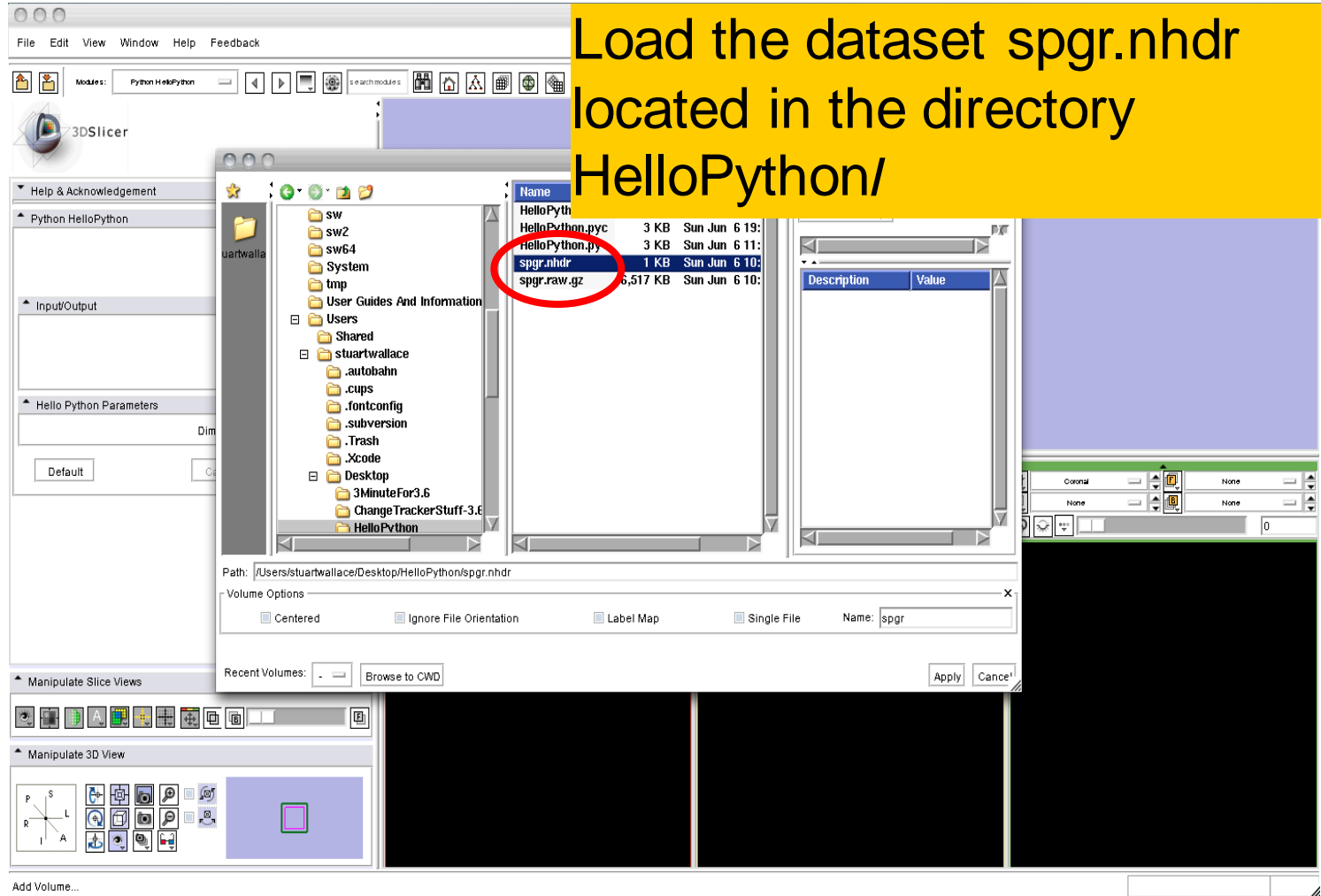


None RAS: (87.3, 1.0, -125.0)



# Python Console

Load the dataset spgr.nhdr located in the directory HelloPython/



Name	Size	Modified
HelloPyth		
HelloPython.pyc	3 KB	Sun Jun 6 19:
HelloPython.py	3 KB	Sun Jun 6 11:
spgr.nhdr	1 KB	Sun Jun 6 10:
spgr.raw.gz	6,517 KB	Sun Jun 6 10:

Path: /Users/stuartwallace/Desktop/HelloPython/spgr.nhdr

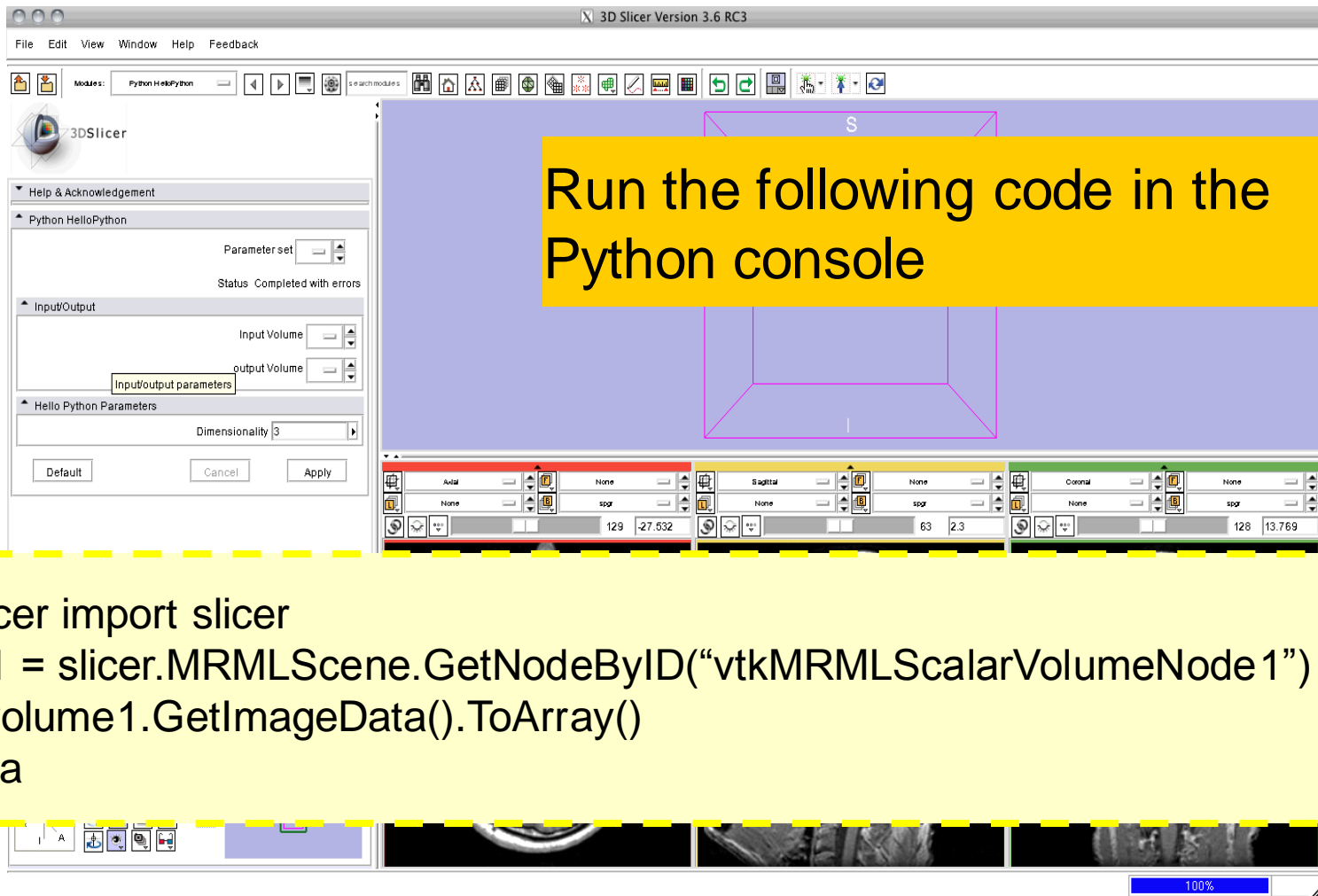
Volume Options

Centered  Ignore File Orientation  Label Map  Single File Name: spgr

Recent Volumes: - Browse to CWD

Apply Cancel

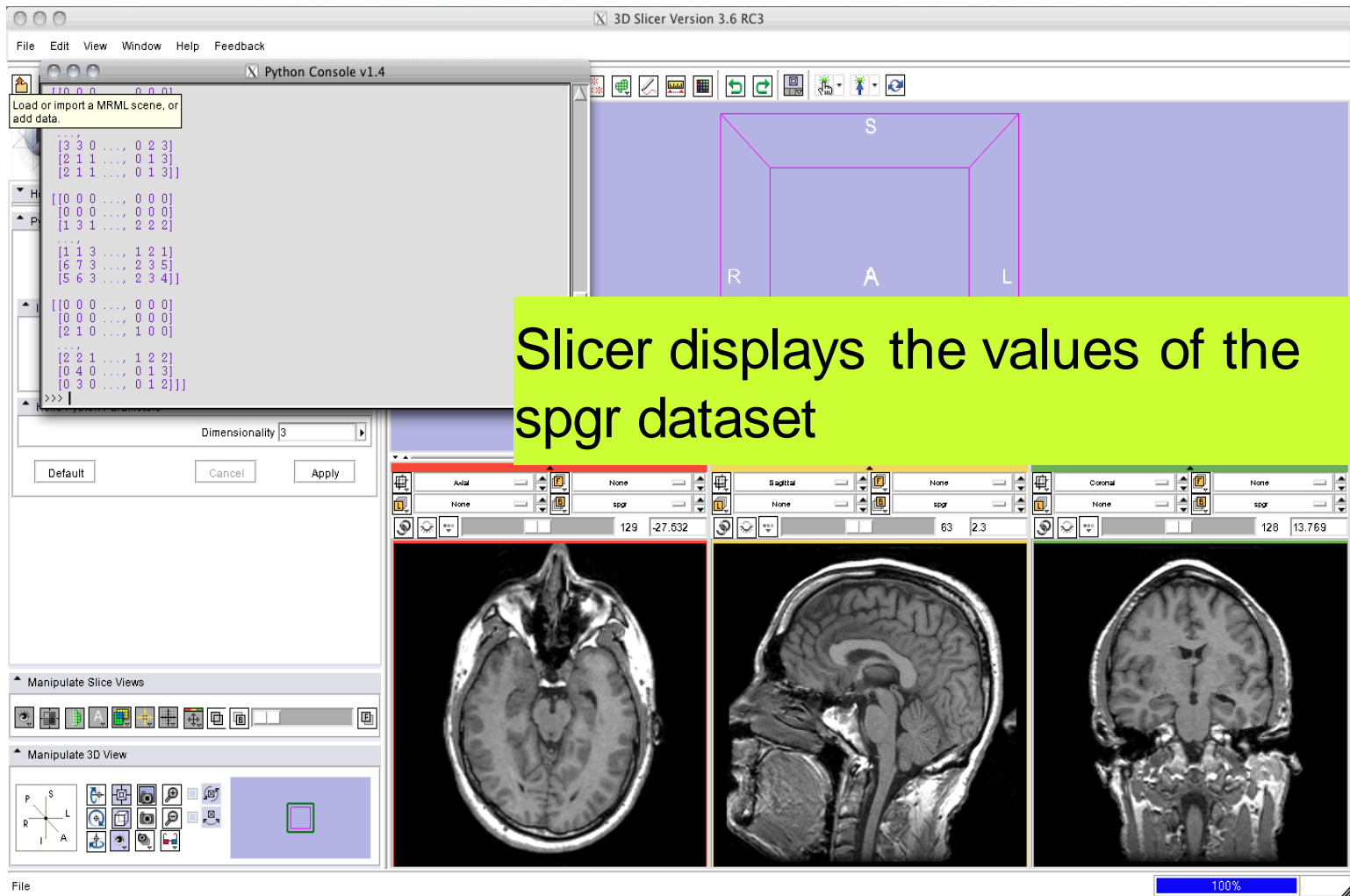
# Python Console



The screenshot shows the 3D Slicer 3.6 RC3 interface. The main window displays a 3D view of a volume with a purple wireframe box. A yellow callout box is overlaid on the 3D view, containing the text: "Run the following code in the Python console". The left sidebar shows the "Python HelloPython" module with various parameters. The bottom of the interface shows the "Slices" panel with three columns for Axial, Sagittal, and Coronal views, each with a "None" button and a "sqr" button. The status bar at the bottom right shows "100%".

```
from Slicer import slicer
volume1 = slicer.MRMLScene.GetNodeByID("vtkMRMLScalarVolumeNode 1")
data = volume1.GetImageData().ToArray()
print data
```

# Python Console



The screenshot displays the 3D Slicer 3.6 RC3 interface. The Python Console (v1.4) is open, showing a list of matrices. The 3D view shows a brain slice with a purple bounding box labeled with 'S' (Superior), 'R' (Right), 'A' (Anterior), and 'L' (Left). The console output includes the following matrices:

```

[[0 0 0 0 0 0]
 [3 3 0 0 0 2 3]
 [2 1 1 0 0 1 3]
 [2 1 1 0 0 1 3]]

[[0 0 0 0 0 0]
 [0 0 0 0 0 0]
 [1 3 1 0 0 2 2]]

[[1 1 3 0 0 1 2 1]
 [6 7 3 0 0 2 3 5]
 [5 6 3 0 0 2 3 4]]


[[0 0 0 0 0 0]
 [0 0 0 0 0 0]
 [2 1 0 0 0 1 0 0]]

[[2 2 1 0 0 1 2 2]
 [0 4 0 0 0 1 3]
 [0 3 0 0 0 1 2]]

```

A yellow text box overlaid on the console area reads: "Slicer displays the values of the spgr dataset". Below the console, the 'Manipulate Slice Views' and 'Manipulate 3D View' panels are visible, along with three orthogonal slice views (Axial, Sagittal, Coronal) showing the brain slice. The status bar at the bottom indicates 100% zoom.

# HelloPython.py



## Module Description

## Module Parameters

## Execute function

```
#!/usr/bin/env python
XML = """<?xml version="1.0" encoding="utf-8"?>
<executable>
  <category>Demonstration </category>
  <title>Python HelloPython</title>
  <description> Slicer developer course in Python
</description>
  <version>1.0</version>

  <license></license>
  <contributor> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
  <documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
  <acknowledgements>
    This work is part of the National Alliance for Medical Image Computing (NA-MIC), funded by the National Institutes of
    Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. </acknowledgements>

  <parameters>
    <label>Input/output</label>
    <description>Input/output parameters</description>
    <image>
      <name>helloPythonInputVolume</name>
      <label>Input Volume</label>
      <channel>input</channel>
      <index>0</index>
      <description>input volume</description>
    </image>
    <image>
      <name>helloPythonOutputVolume</name>
      <label>output Volume</label>
      <channel>output</channel>
      <index>1</index>
      <description>output volume</description>
    </image>
  </parameters>
  <parameters>
    <label>Hello Python Parameters</label>
    <description>Parameters of the Python Hello Python module </description>
    <integer>
      <name>dimensionality</name>
      <longflag>dimensionality</longflag>
      <description>Dimensionality of the Laplace operator</description>
      <label>Dimensionality</label>
      <default>3</default>
      <constraints>
        <minimum>2</minimum>
        <maximum>3</maximum>
      </constraints>
    </integer>
  </parameters>

</executable>
"""

def Execute ():
    slicer = __import__("Slicer")
    slicer = slicer.slicer
    scene = slicer.HIFILScene

    return
```



# Module Description

```
#!/usr/bin/env python
XML = """<?xml version="1.0" encoding="utf-8"?>
<executable>
<category>Demonstration </category>
<title>Python HelloPython</title>
<description> Slicer developer course in Python </description>
<version>1.0</version>
<license></license>
<contributor> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
<documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
<acknowledgements>
    This work is part of the National Alliance for Medical Image Computing (NA-MIC),
    funded by the National Institutes of Health through the NIH Roadmap for Medical Research,
    Grant U54 EB005149.
</acknowledgements>
```

# Module Parameters

```
<parameters>  
  <label>Input/Output</label>  
  <description>Input/output parameters</description>
```

Input  
Volume

```
<image>  
  <name>HelloPythonInputVolume</name>  
  <label>Input Volume</label>  
  <channel>input</channel>  
  <index>0</index>  
  <description>input volume</description>  
</image>
```

A file that  
specifies  
the image

Output  
Volume

```
<image>  
  <name>HelloPythonOutputVolume</name>  
  <label>Output Volume</label>  
  <channel>output</channel>  
  <index>1</index>  
  <description>output volume</description>  
</image>  
</parameters>
```



# Execute Function

---

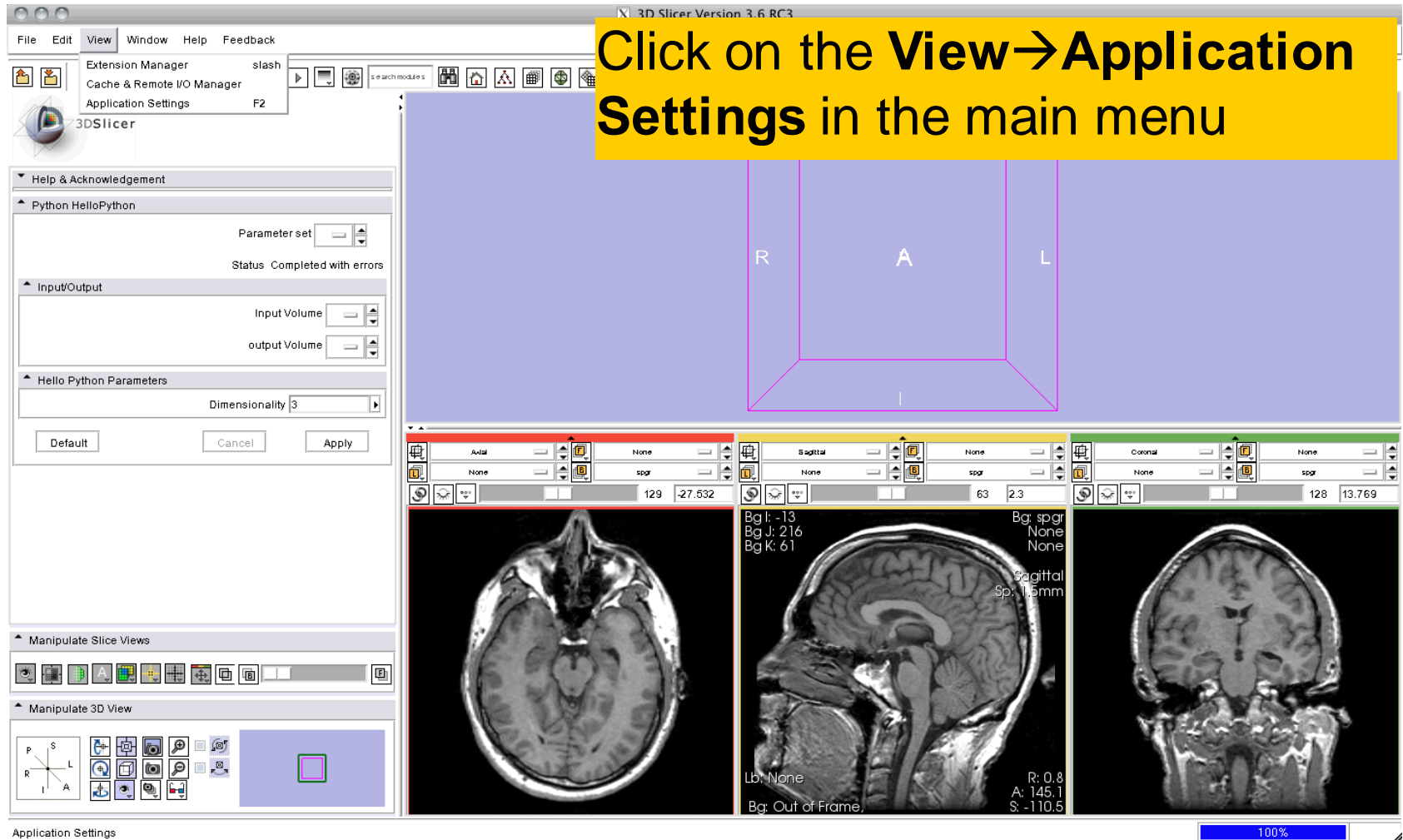
```
def Execute ():  
  
    Slicer = __import__("Slicer")  
    slicer = Slicer.slicer  
    scene = slicer.MRMLScene  
  
    return
```



# Integrating HelloPython to Slicer3

3D Slicer Version 3.6 RC3

**Click on the **View** → **Application Settings** in the main menu**

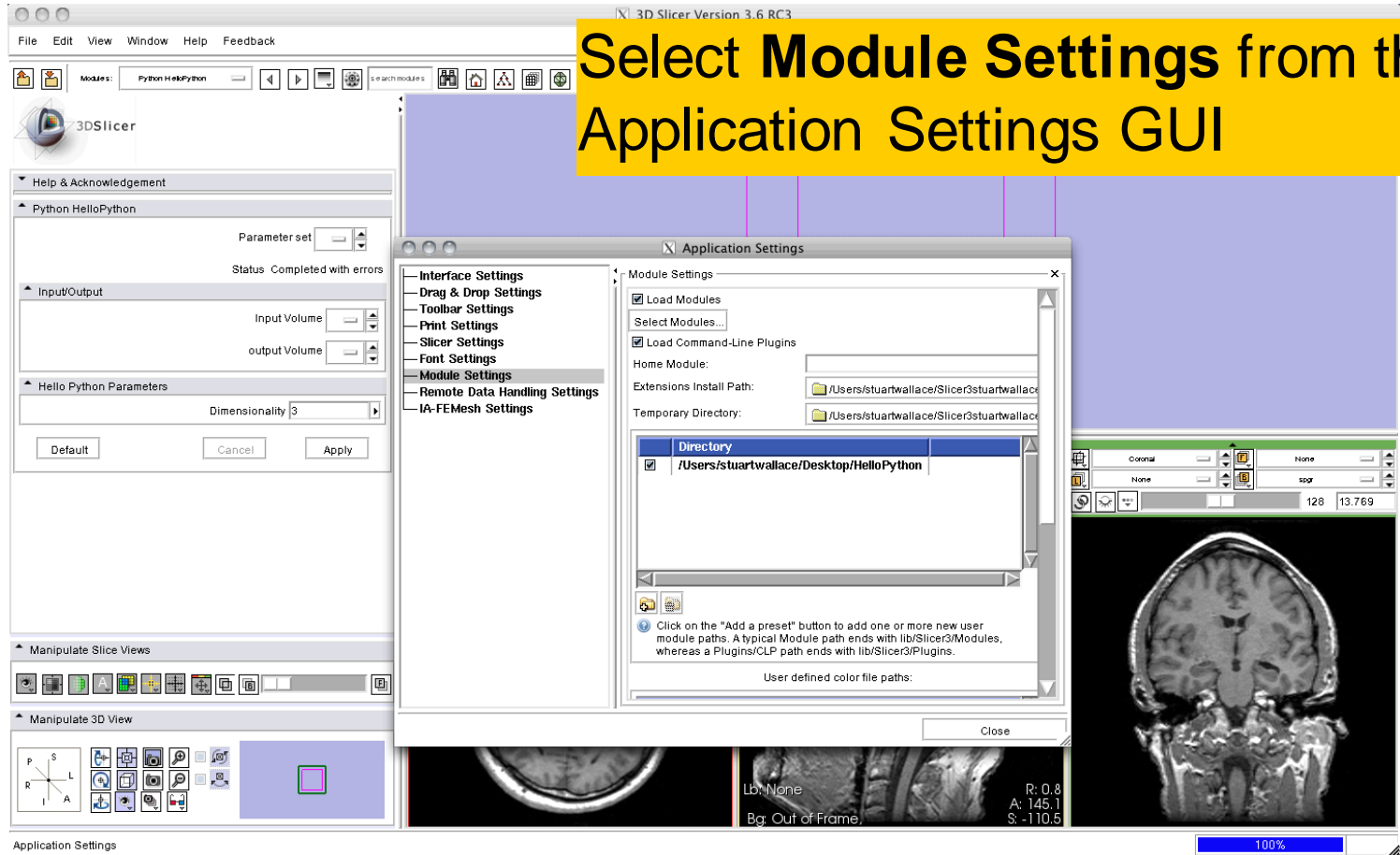


The screenshot shows the 3D Slicer 3.6 RC3 interface. The main window displays three orthogonal MRI slices (Axial, Sagittal, Coronal) and a 3D view of the brain volume. The Application Settings dialog for Python HelloPython is open, showing various parameters and options. The dialog is titled "Application Settings" and has a "View" menu item highlighted. The "View" menu is open, showing "Application Settings" as the selected option. The "Application Settings" dialog has a "View" menu item highlighted. The "View" menu is open, showing "Application Settings" as the selected option. The "Application Settings" dialog has a "View" menu item highlighted. The "View" menu is open, showing "Application Settings" as the selected option.

Application Settings

100%

# Integrating HelloPython to Slicer3



**Select Module Settings from the Application Settings GUI**

3D Slicer Version 3.6 RC3

File Edit View Window Help Feedback

Modules: Python HelloPython

3DSlicer

Help & Acknowledgement

Python HelloPython

Parameter set: [dropdown]

Status: Completed with errors

Input/Output

Input Volume: [dropdown]

output Volume: [dropdown]

Hello Python Parameters

Dimensionality: 3

Default Cancel Apply

Manipulate Slice Views

Manipulate 3D View

Application Settings

Interface Settings

Drag & Drop Settings

Toolbar Settings

Print Settings

Slicer Settings

Font Settings

Module Settings

Remote Data Handling Settings

IA-FEMesh Settings

Module Settings

Load Modules

Select Modules...

Load Command-Line Plugins

Home Module:

Extensions Install Path: /Users/stuartwallace/Slicer3stuartwallace

Temporary Directory: /Users/stuartwallace/Slicer3stuartwallace

Directory	
<input checked="" type="checkbox"/>	/Users/stuartwallace/Desktop/HelloPython

Click on the "Add a preset" button to add one or more new user module paths. A typical Module path ends with lib/Slicer3/Modules, whereas a Plugins/CLP path ends with lib/Slicer3/Plugins.

User defined color file paths:

Close

Coronal None

None spgr

128 13.769

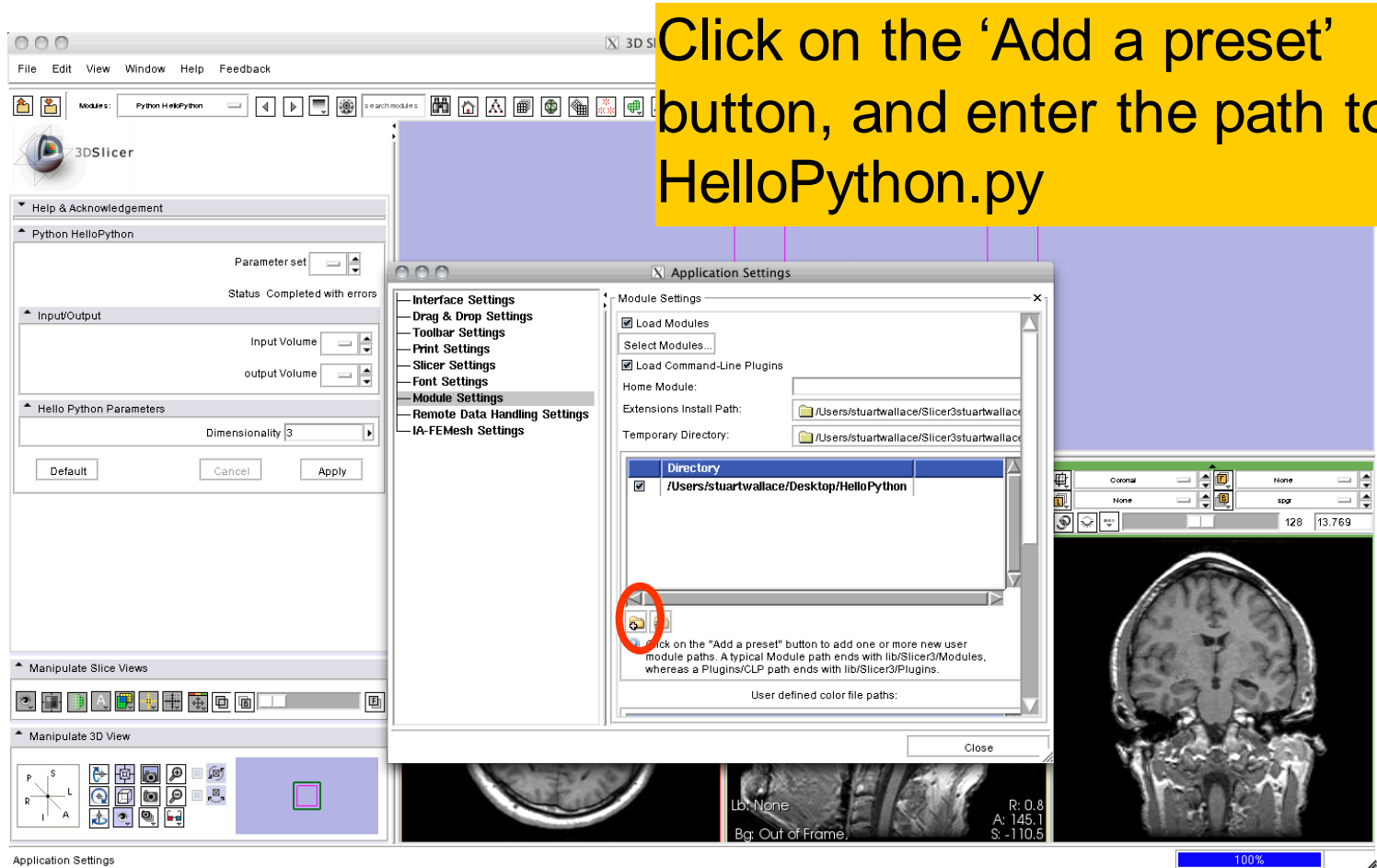
R: 0.8  
A: 145.1  
S: -110.5

Lb: None  
Bg: Out of Frame

100%

# Integrating HelloPython to Slicer3

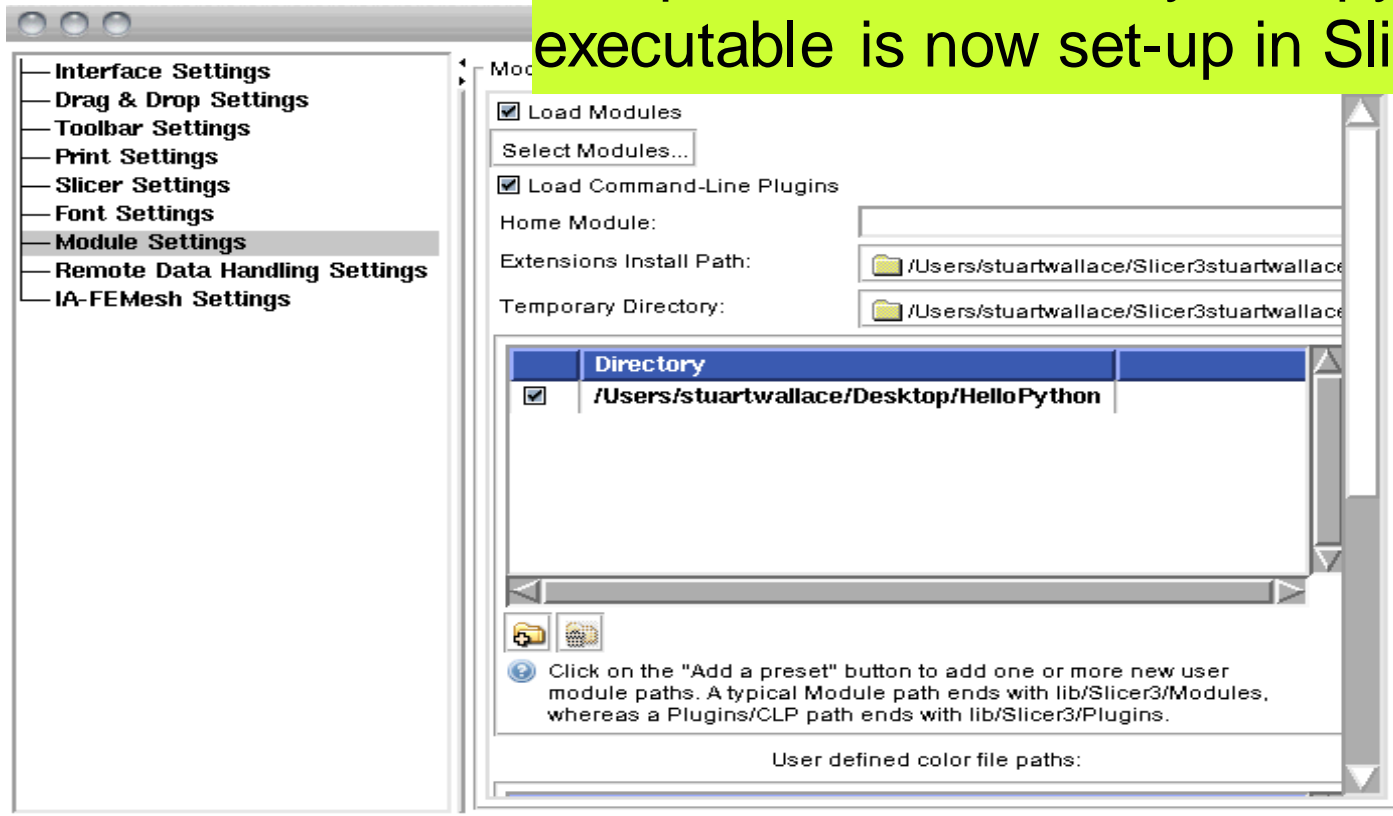
Click on the 'Add a preset' button, and enter the path to HelloPython.py



The screenshot shows the 3DSlicer application window with the 'Application Settings' dialog box open. The 'Module Settings' tab is selected, and the 'Add a preset' button is circled in red. The 'Directory' list shows the path `/Users/stuartwallace/Desktop/HelloPython`. The 'Add a preset' button is located at the bottom left of the dialog box. The background shows the 3DSlicer interface with a brain MRI slice and various toolbars.

# Integrating HelloPython to Slicer3

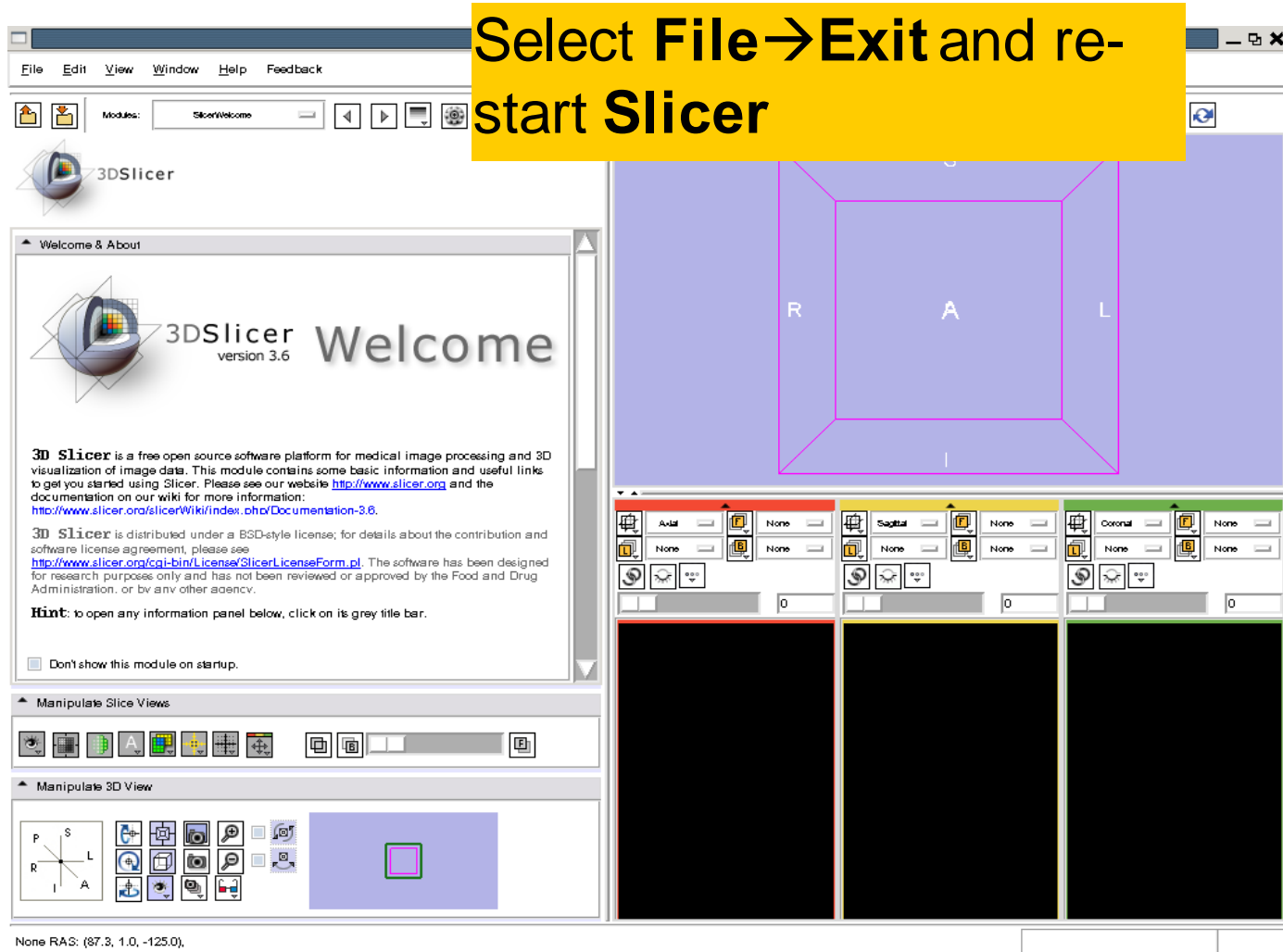
The path to the HelloPython.py executable is now set-up in Slicer3.



Click on the **Close** to exit the Application Settings window.

# Integrating HelloPython to Slicer3

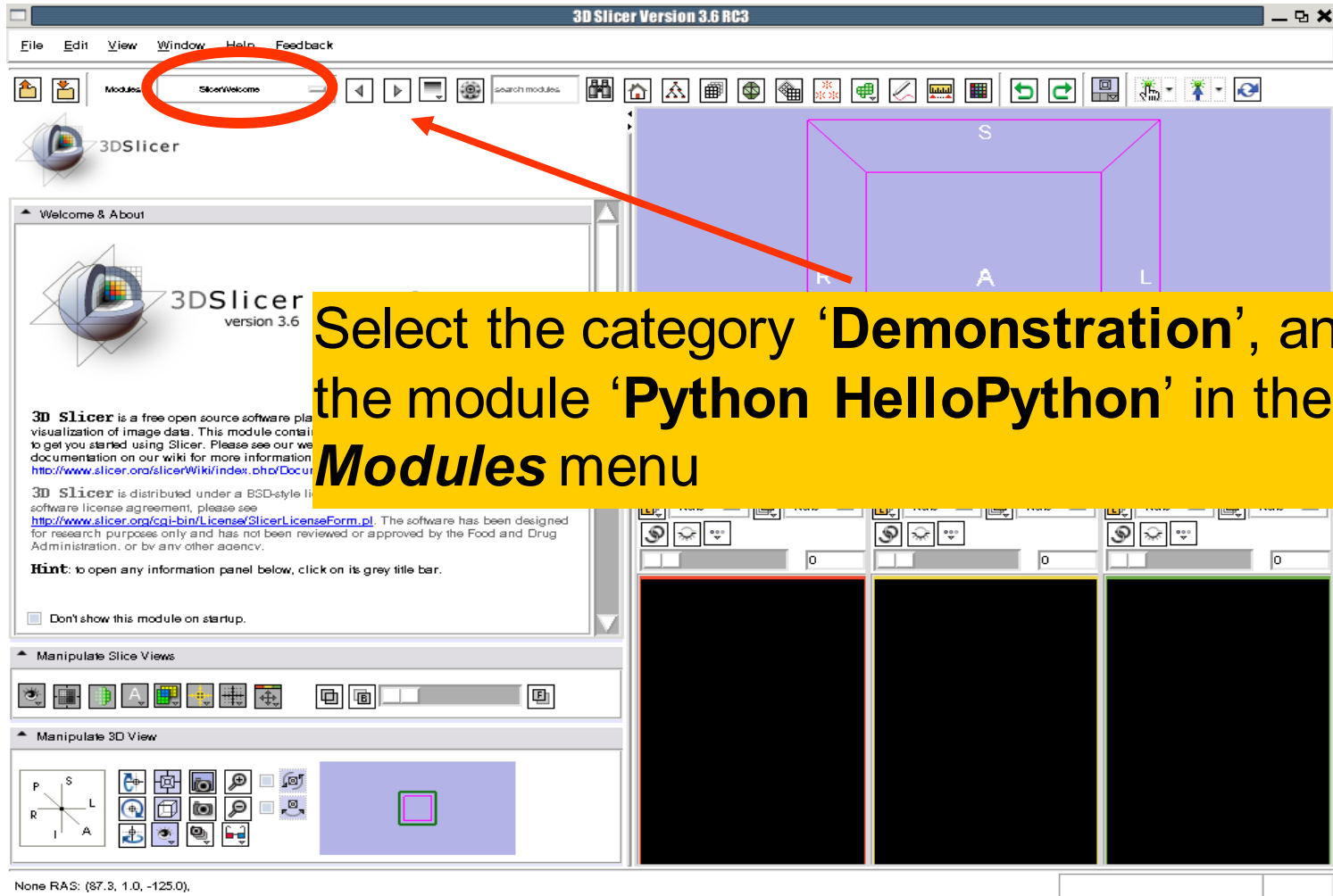
Select File → Exit and re-start Slicer



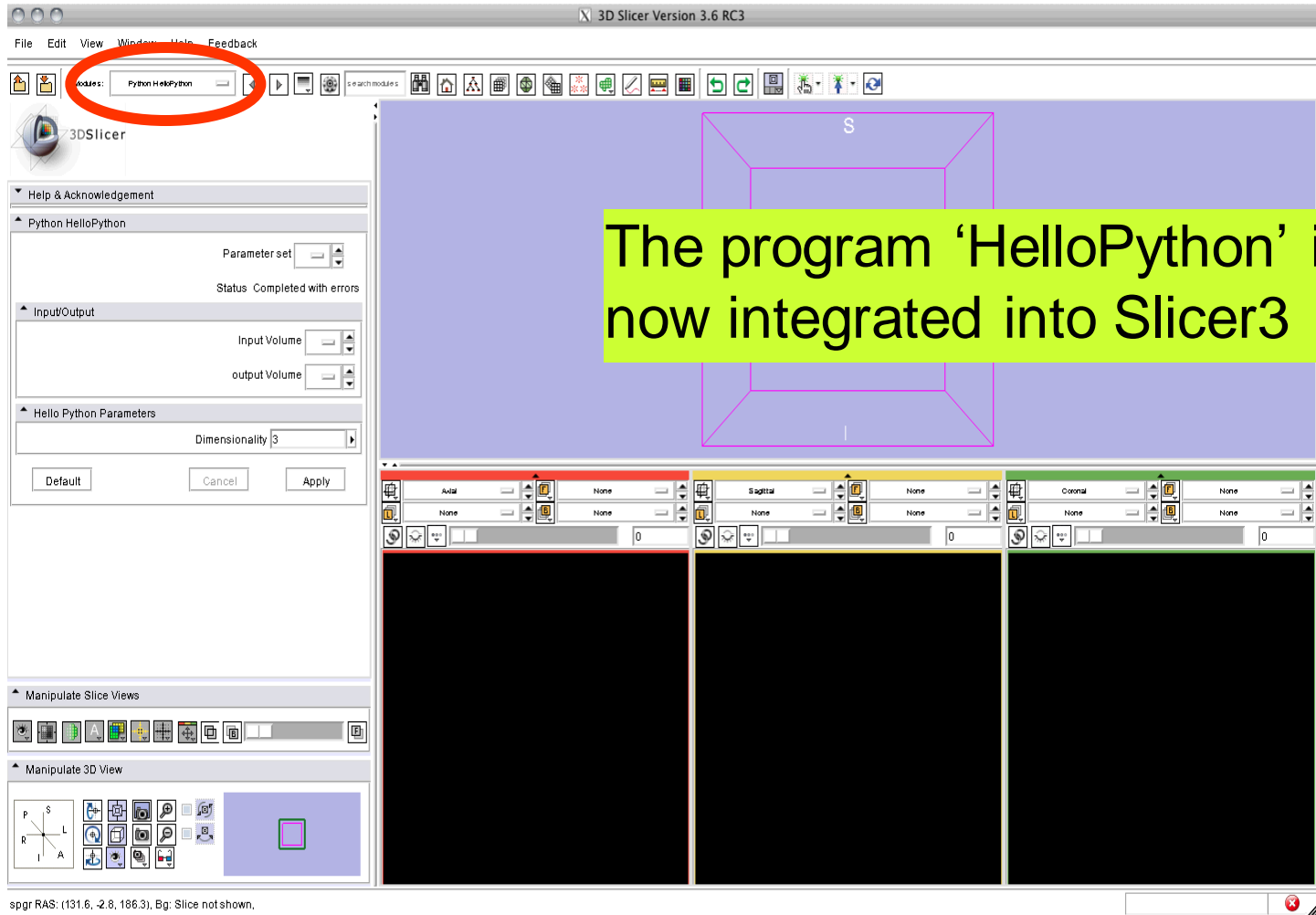
None RAS: (87.3, 1.0, -125.0),



# HelloPython module

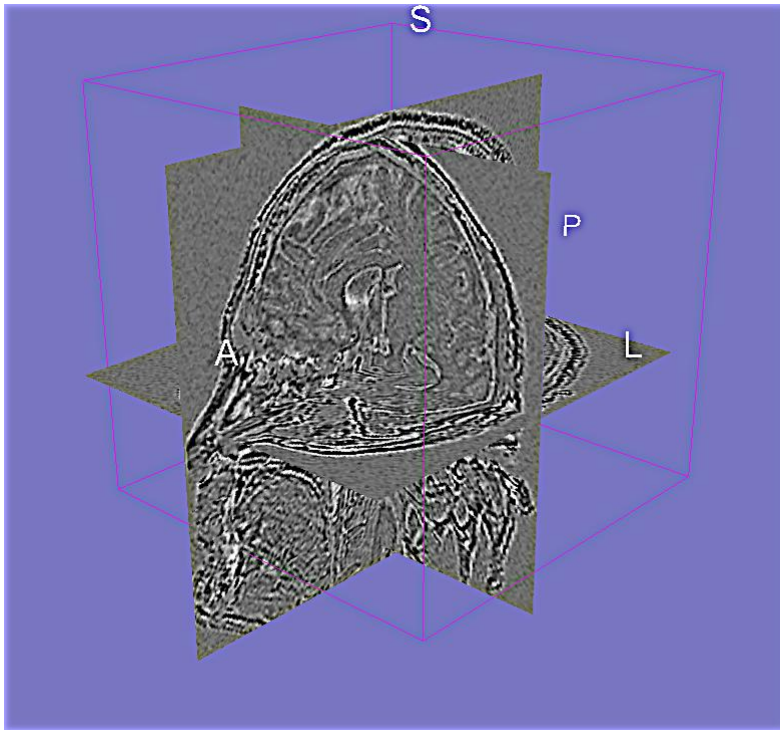


# HelloPython Module



The screenshot shows the 3D Slicer 3.6 RC3 interface. The 'Modules' dropdown menu is circled in red, showing 'Python HelloPython' selected. The 'Python HelloPython' panel is open, displaying 'Parameter set' and 'Status: Completed with errors'. The 'Input/Output' section has 'Input Volume' and 'Output Volume' dropdowns. The 'Hello Python Parameters' section has a 'Dimensionality' dropdown set to '3'. The 'Manipulate Slice Views' and 'Manipulate 3D View' sections are also visible. The main 3D view shows a purple volume with a pink wireframe box labeled 'S'. The status bar at the bottom indicates 'spgr RAS: (131.6, -2.8, 186.3), Bg: Slice not shown.'

The program 'HelloPython' is now integrated into Slicer3



# Part B: Implementing the Laplace\* Operator

\*named after Pierre-Simon, Marquis de Laplace (1749-1827)





# Execute Function

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume):
```

Add the I/O code

```
Slicer = __import__("Slicer")
```

```
slicer = Slicer.slicer
```

```
scene = slicer.MRMLScene
```

```
inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
```

```
outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
```

```
return
```



# Laplace Operator

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume):
```

```
    Slicer = __import__("Slicer")
```

```
    slicer = Slicer.slicer
```

```
    scene = slicer.MRMLScene
```

```
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
```

```
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
```

```
    laplacian = slicer.vtkImageLaplacian()
```

```
    laplacian.SetInput(inputVolume.GetImageData())
```

```
    return
```

Add the Laplace operator

# Laplace Operator

```
<parameters>
  <label>Input/Output</label>
  <description>Input/output parameters</description>
  <image>
    <name>HelloPythonInputVolume</name>
    <label>Input Volume</label>
    <channel>input</channel>
    <index>0</index>
    <description>input volume</description>
  </image>
  <image>
    <name>HelloPythonOutputVolume</name>
    <label>Output Volume</label>
    <channel>output</channel>
    <index>1</index>
    <description>output volume</description>
  </image>
</parameters>
```

```
<parameters>
  <label>Hello Python Parameters</label>
  <description> Parameters of the Python Hello Python module </description>
</parameters>
```

Add a new parameter group for the Laplace operator



# Laplace Operator

```
<parameters>
  <label>Hello Python Parameters</label>
  <description>Parameters of the Python Hello Python module </description>
  <integer>
    <name>dimensionality</name>
    <longflag>dimensionality</longflag>
    <description>Dimensionality of the Laplace operator</description>
    <label>Dimensionality</label>
    <default>3</default>
    <constraints>
      <minimum>2</minimum>
      <maximum>3</maximum>
    </constraints>
  </integer>
</parameters>
```

Add the Laplace  
operator's  
dimensionality

# Laplace Operator

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,
```

```
dimensionality=3):
```

```
Slicer = __import__("Slicer")
```

```
slicer = Slicer.slicer
```

```
scene = slicer.MRMLScene
```

```
inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
```

```
outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
```

```
laplacian = slicer.vtkImageLaplacian()
```

```
laplacian.SetInput(inputVolume.GetImageData())
```

```
laplacian.SetDimensionality(dimensionality)
```

```
return
```

Set-up the corresponding dimensionality parameter in the Python code



# Laplace Operator

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,  
dimensionality=3):
```

```
Slicer = __import__("Slicer")
```

```
 slicer = Slicer.slicer
```

```
 scene = slicer.MRMLScene
```

```
 inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
```

```
 outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
```

```
 laplacian = slicer.vtkImageLaplacian()
```

```
 laplacian.SetInput(inputVolume.GetImageData())
```

```
 laplacian.SetDimensionality(dimensionality)
```

```
 laplacian.Update()
```

```
 outputVolume.SetAndObserveImageData(laplacian.GetOutput())
```

```
 return
```

Add code to get the output  
of the Laplace operator



# Laplace Operator

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,  
dimensionality=3):
```

```
    Slicer = __import__("Slicer")
```

```
    slicer = Slicer.slicer
```

```
    scene = slicer.MRMLScene
```

```
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
```

```
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
```

```
    laplacian = slicer.vtkImageLaplacian()
```

```
    laplacian.SetInput(inputVolume.GetImageData())
```

```
    laplacian.SetDimensionality(dimensionality)
```

```
    laplacian.Update()
```

```
    outputVolume.SetAndObserveImageData(laplacian.GetOutput())
```

```
    matrix = slicer.vtkMatrix4x4()
```

```
    inputVolume.GetIJKToRASMatrix(matrix)
```

```
    outputVolume.SetIJKToRASMatrix(matrix)
```

```
    return
```

Place back the Laplacian of the image in the RAS reference system.

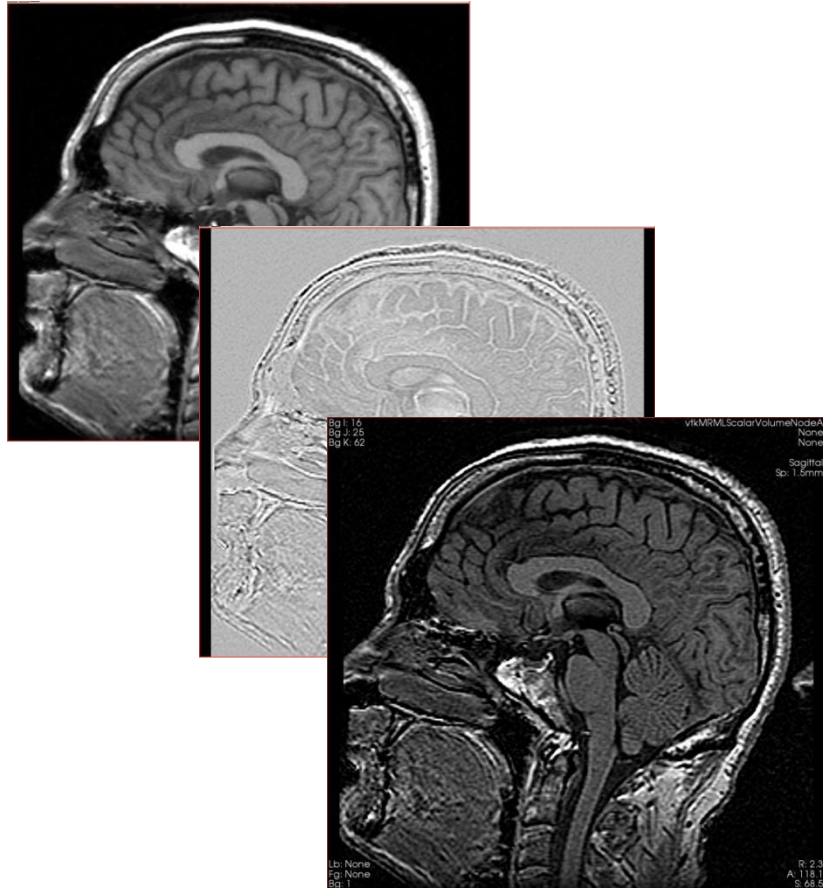


# Integrating HelloPython to Slicer3

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,
dimensionality=3):
    Slicer = __import__("Slicer")
    slicer = Slicer.slicer
    scene = slicer.MRMLScene
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
    laplacian = slicer.vtkImageLaplacian()
    laplacian.SetInput(inputVolume.GetImageData())
    laplacian.SetDimensionality(dimensionality)
    laplacian.Update()
    outputVolume.SetAndObserveImageData(laplacian.GetOutput())
    matrix = slicer.vtkMatrix4x4()
    inputVolume.GetIJKToRASMatrix(matrix)
    outputVolume.SetIJKToRASMatrix(matrix)
    return
```

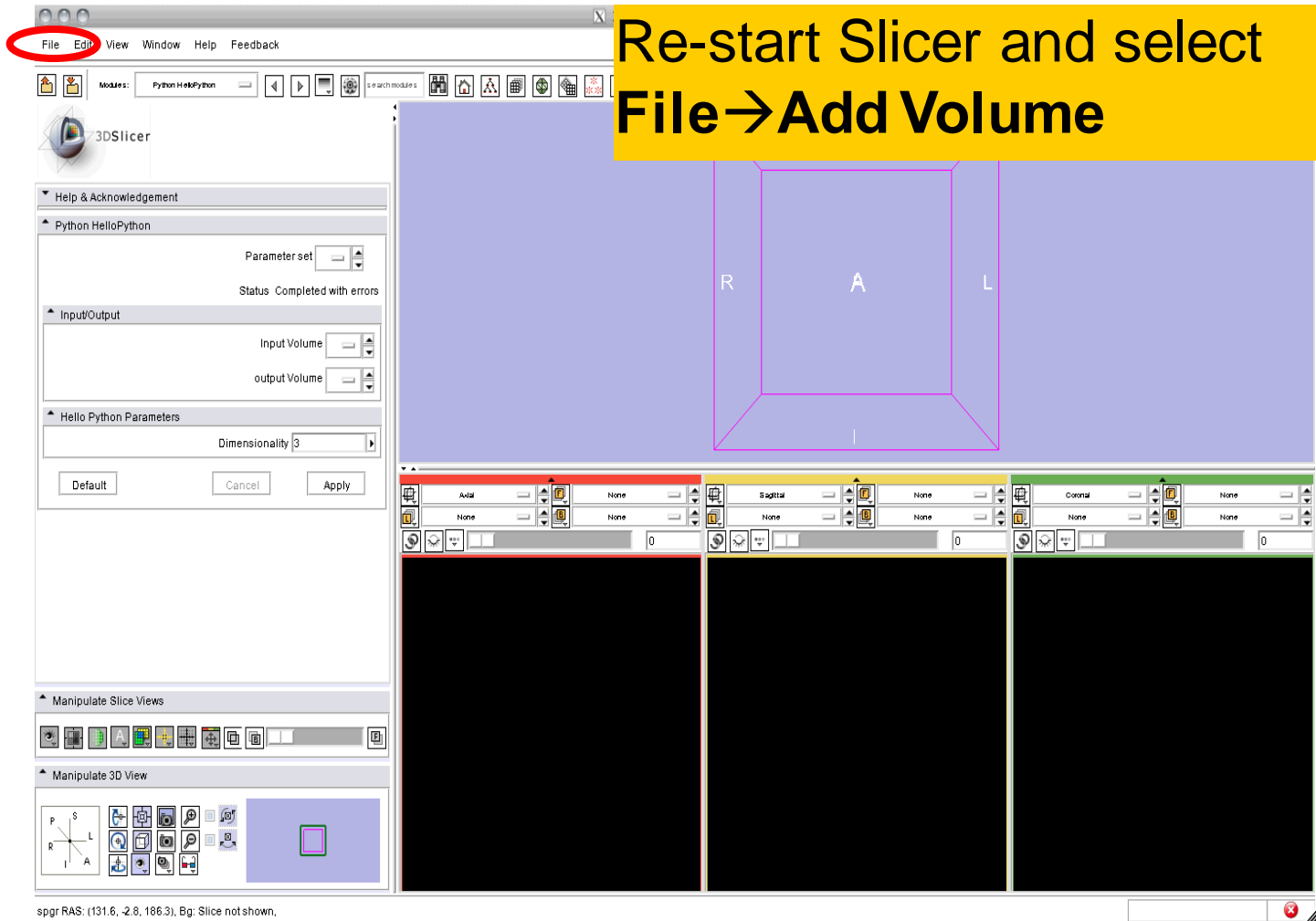
**Save the HelloPython.py file and exit Slicer.**





# Part C: Image Sharpening with the Laplace Operator

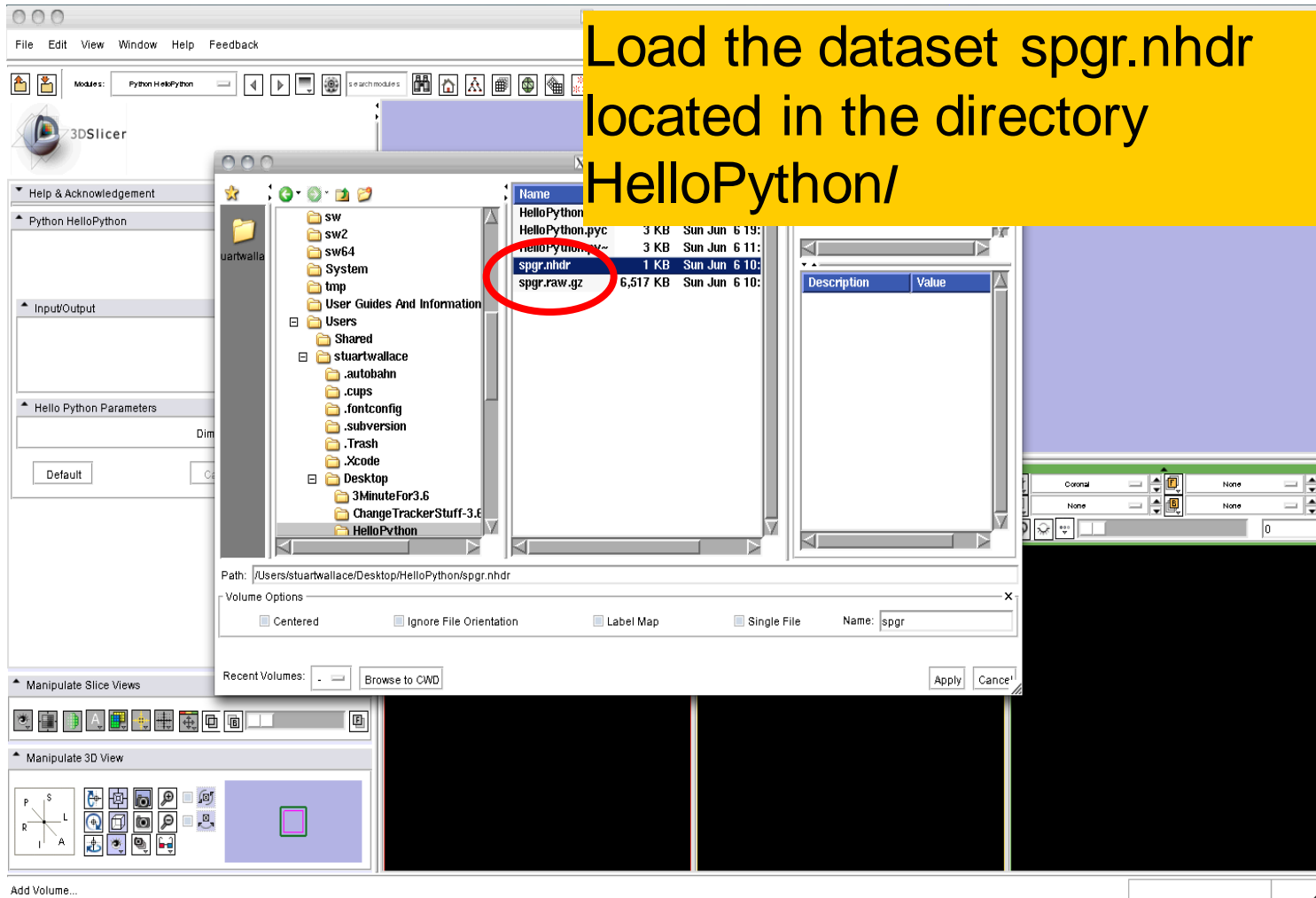
# Running the Laplace Operator



The screenshot shows the 3DSlicer application window. The 'File' menu is circled in red. A yellow callout box contains the text: "Re-start Slicer and select File → Add Volume". The main 3D view area shows a purple rectangular volume with axes labeled R (Right), A (Anterior), L (Left), and I (Inferior). Below the 3D view are three slice view panels: Axial, Sagittal, and Coronal. The status bar at the bottom indicates "spgr RAS: (131.6, -2.8, 186.3), Bg: Slice not shown."

# Running the Laplace Operator

Load the dataset spgr.nhdr located in the directory HelloPython/



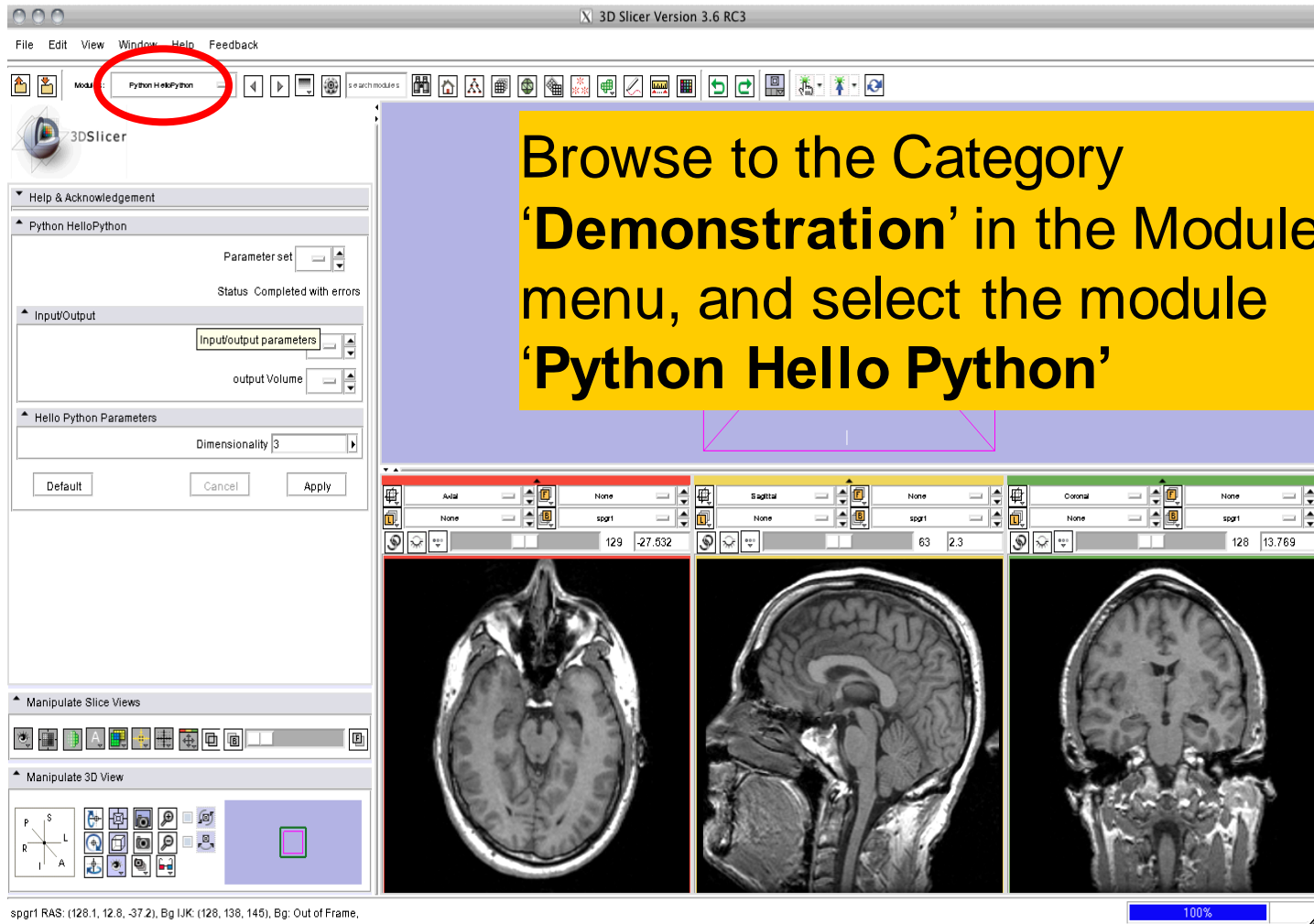
Path: /Users/stuartwallace/Desktop/HelloPython/spgr.nhdr

Volume Options

Centered  Ignore File Orientation  Label Map  Single File Name: spgr

Recent Volumes:

# Running the Laplace Operator

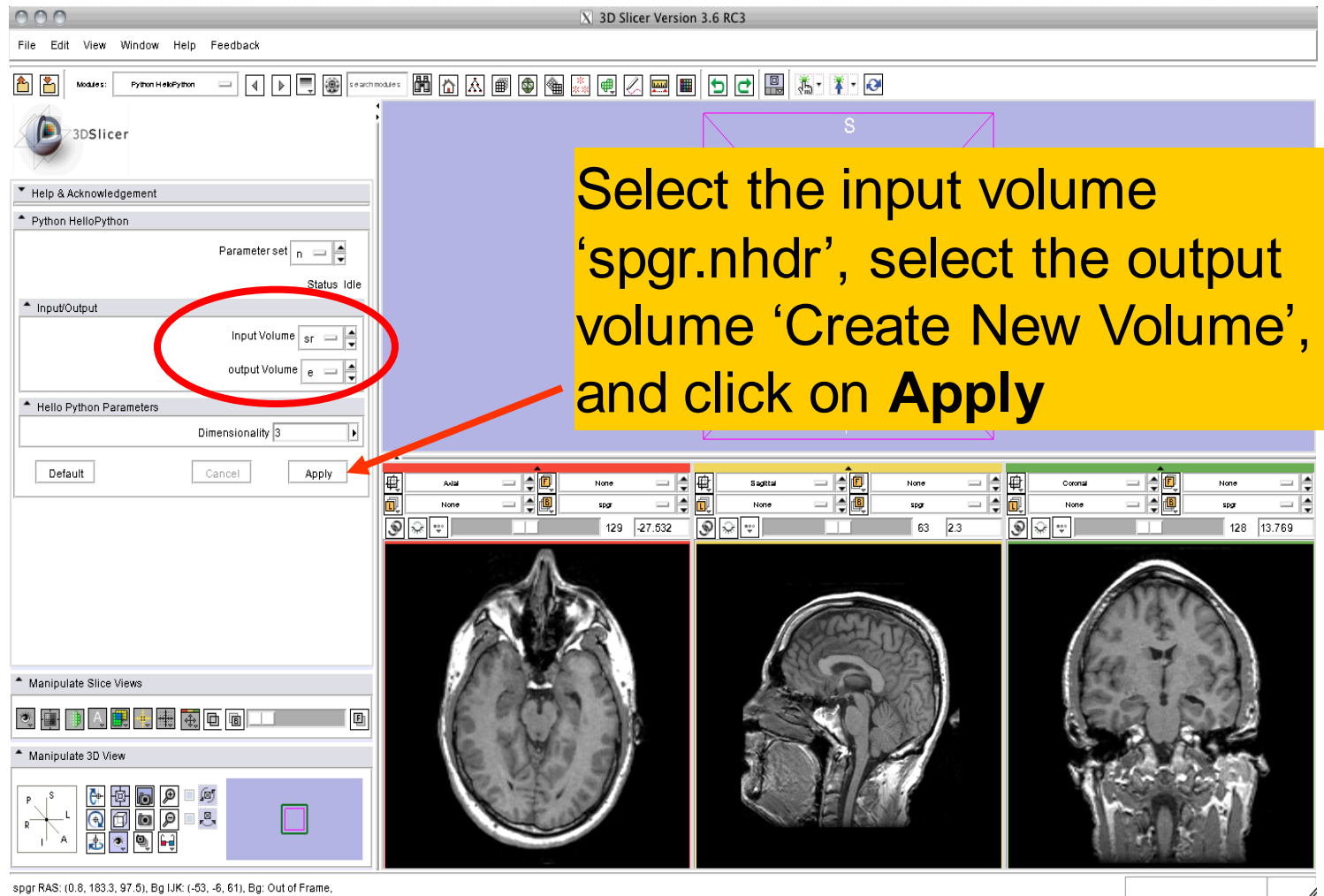


The screenshot shows the 3D Slicer 3.6 RC3 interface. The 'Modules' menu is open, and 'Python Hello Python' is selected, highlighted with a red circle. A yellow text box with black text is overlaid on the right side of the interface, containing the following instructions:

Browse to the Category  
**'Demonstration'** in the Modules  
menu, and select the module  
**'Python Hello Python'**

The interface also shows the 'Python Hello Python' module's parameter set, status, and input/output fields. The bottom of the interface displays three slice views (Axial, Sagittal, Coronal) and a 3D view of a brain MRI scan.

# Running the Laplace Operator



3D Slicer Version 3.6 RC3

File Edit View Window Help Feedback

Modules: Python HelloPython

3DSlicer

Help & Acknowledgement

Python HelloPython

Parameter set: n

Status: Idle

Input/Output

Input Volume: sr

Output Volume: e

Hello Python Parameters

Dimensionality: 3

Default Cancel Apply

Manipulate Slice Views

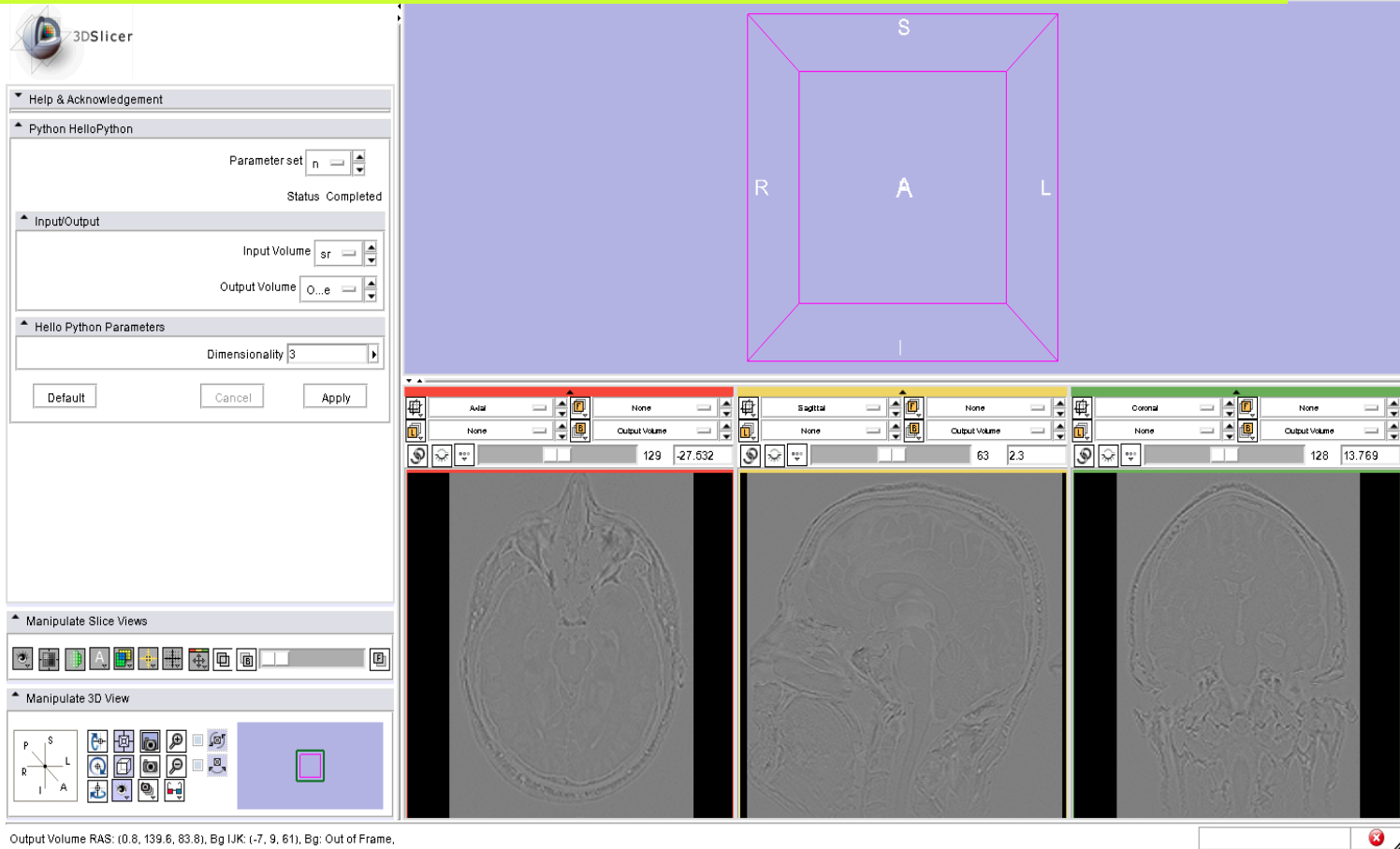
Manipulate 3D View

spgr RAS: (0.8, 183.3, 97.5), Bg IJK: (-63, -6, 61), Bg: Out of Frame.

**Select the input volume 'spgr.nhdr', select the output volume 'Create New Volume', and click on **Apply****

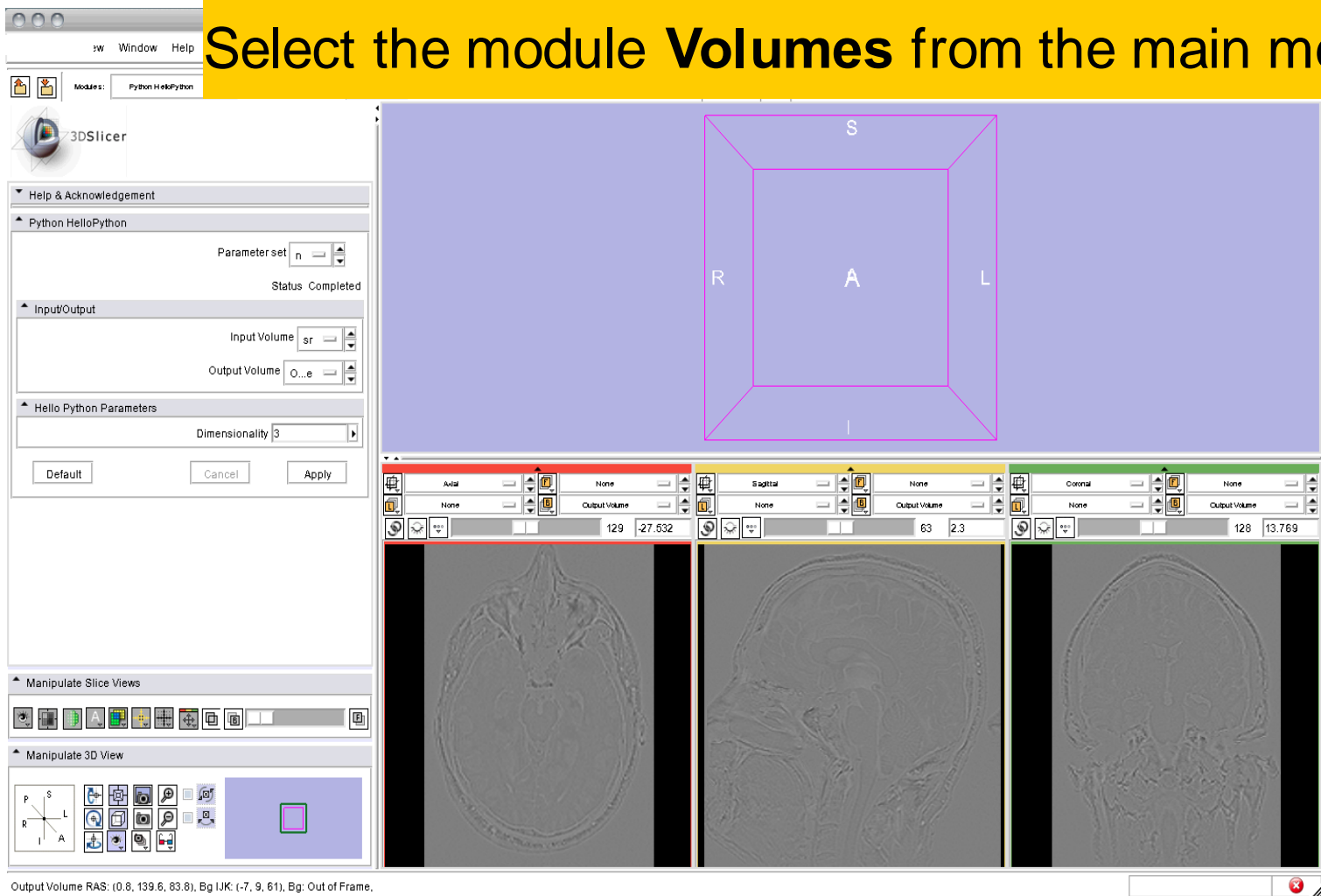
# Running the Laplace Operator

Slicer displays the Laplacian of the spgr image.



# Laplacian of the image

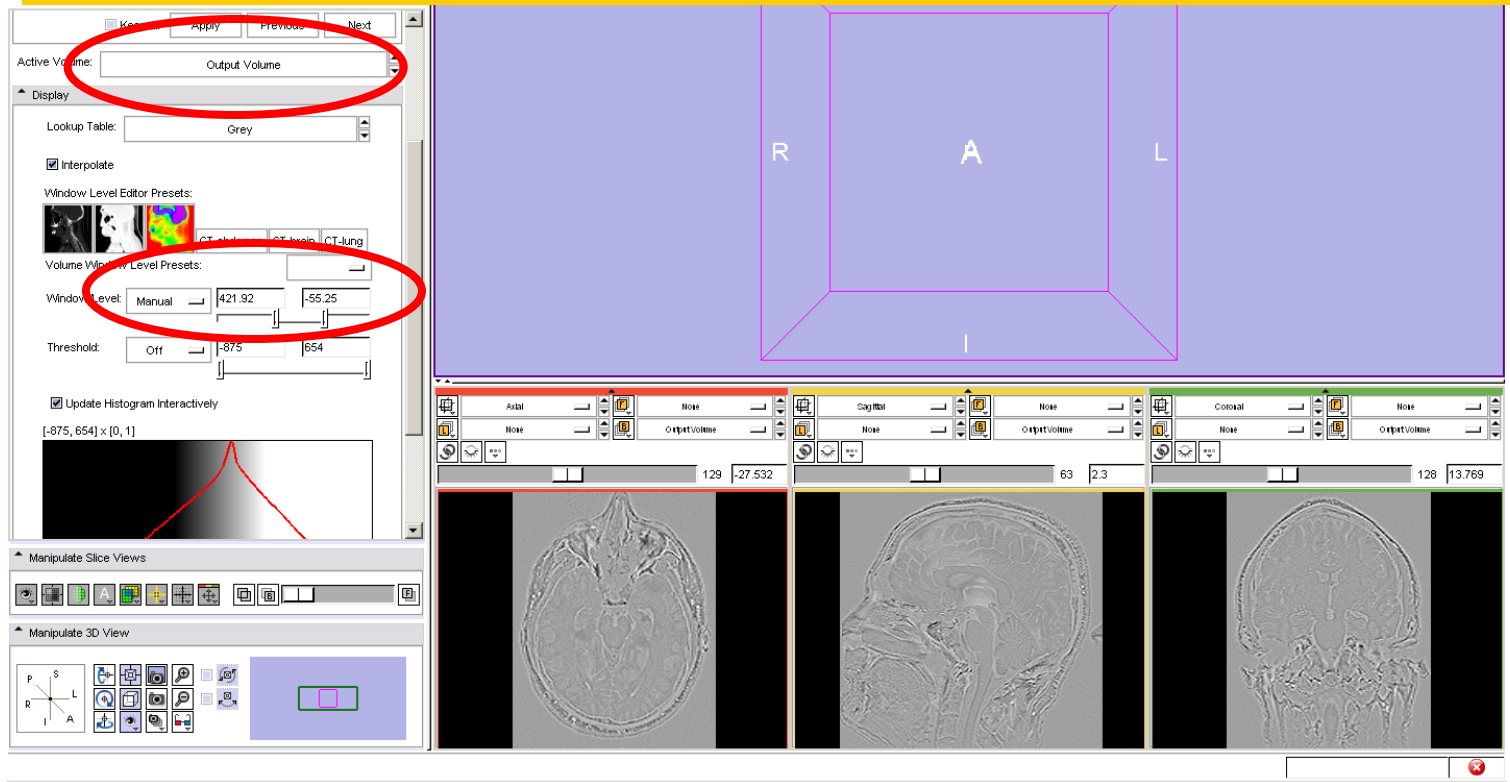
Select the module **Volumes** from the main menu



The screenshot shows the 3DSlicer interface. On the left, the 'Python HelloPython' module is active, showing parameters like 'Parameter set: n', 'Status: Completed', 'Input Volume: sr', and 'Output Volume: O...e'. Below this are sections for 'Hello Python Parameters' (Dimensionality: 3) and 'Manipulate Slice Views' (Axial, Sagittal, Coronal). The main 3D view shows a brain slice with a purple bounding box labeled with axes S (Superior), R (Right), A (Anterior), L (Left), and I (Inferior). Below the 3D view are three slice view windows: Axial, Sagittal, and Coronal, each showing a corresponding slice of the brain. The status bar at the bottom indicates 'Output Volume RAS: (0.8, 139.6, 83.8), Bg IJK: (-7, 9, 61), Bg: Out of Frame.'

# Laplacian of the image

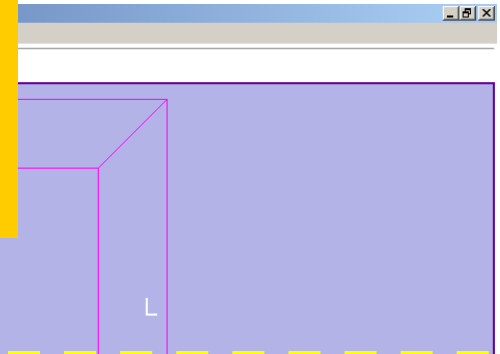
Set the Active Volume to **Output Volume** and adjust the Window/Level parameters



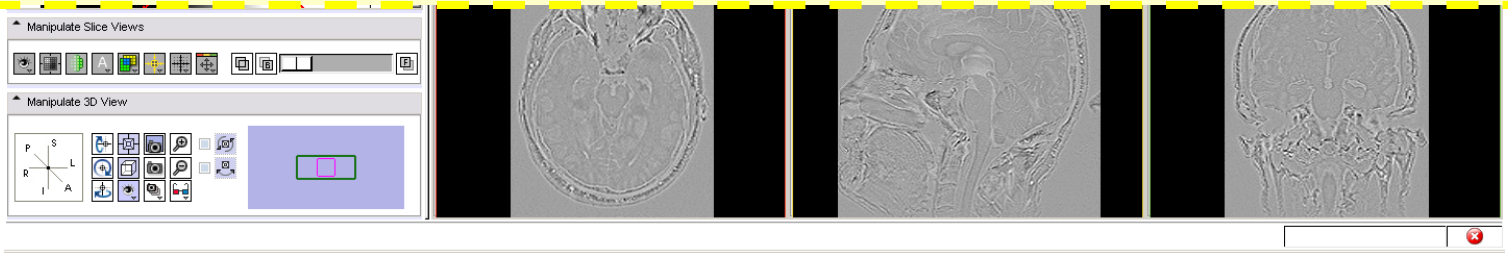


# Image Sharpening

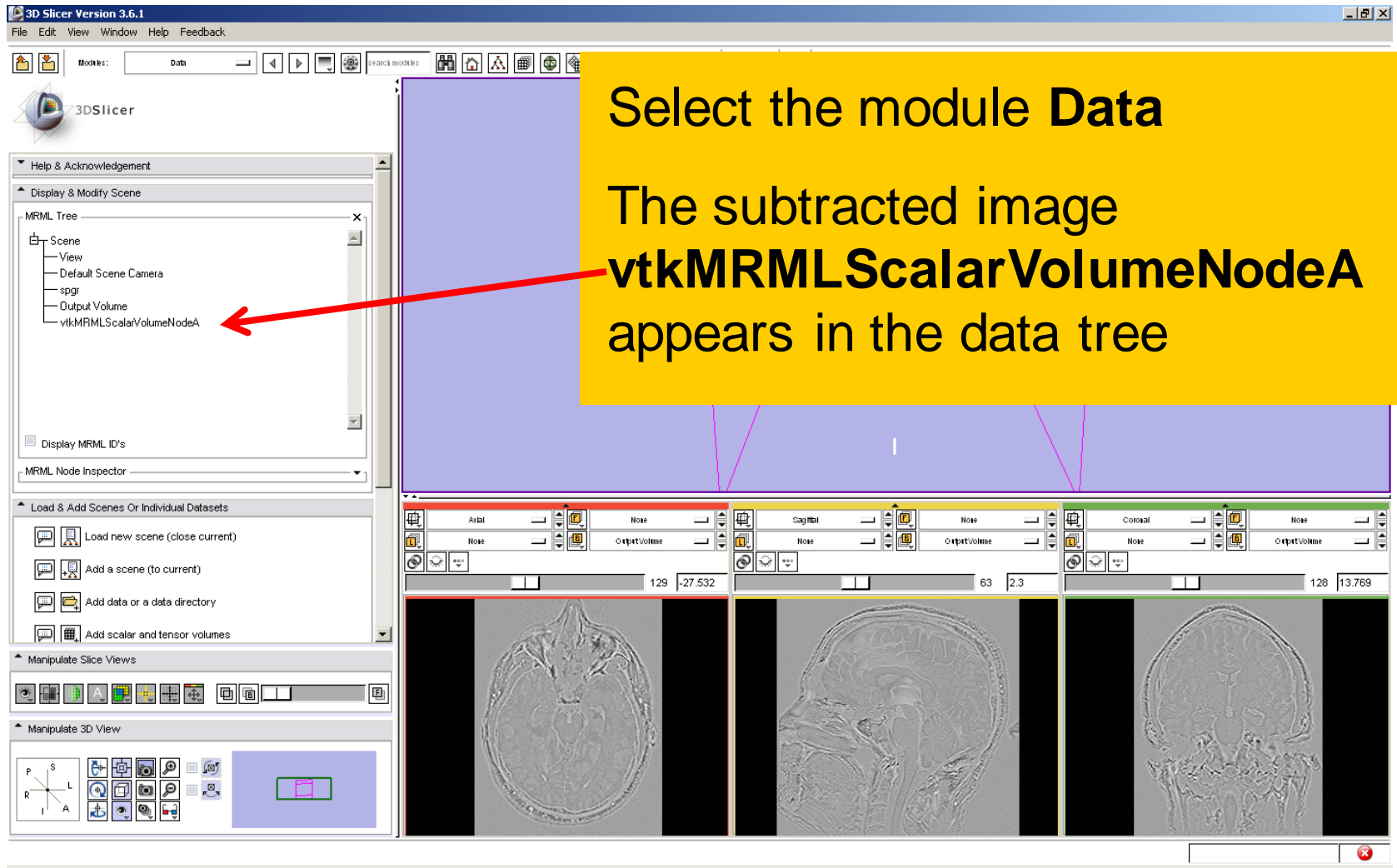
Run the following code in the Python console to subtract the Laplacian of the image to the original image



```
import Slicer
volume1 = Slicer.slicer.MRMLScene.GetNodeByID("vtkMRMLScalarVolumeNode1")
volume2 = Slicer.slicer.MRMLScene.GetNodeByID("vtkMRMLScalarVolumeNode2")
plugin = Slicer.Plugin("Subtract Images")
plugin.Execute(volume1,volume2)
```



# Image Sharpening

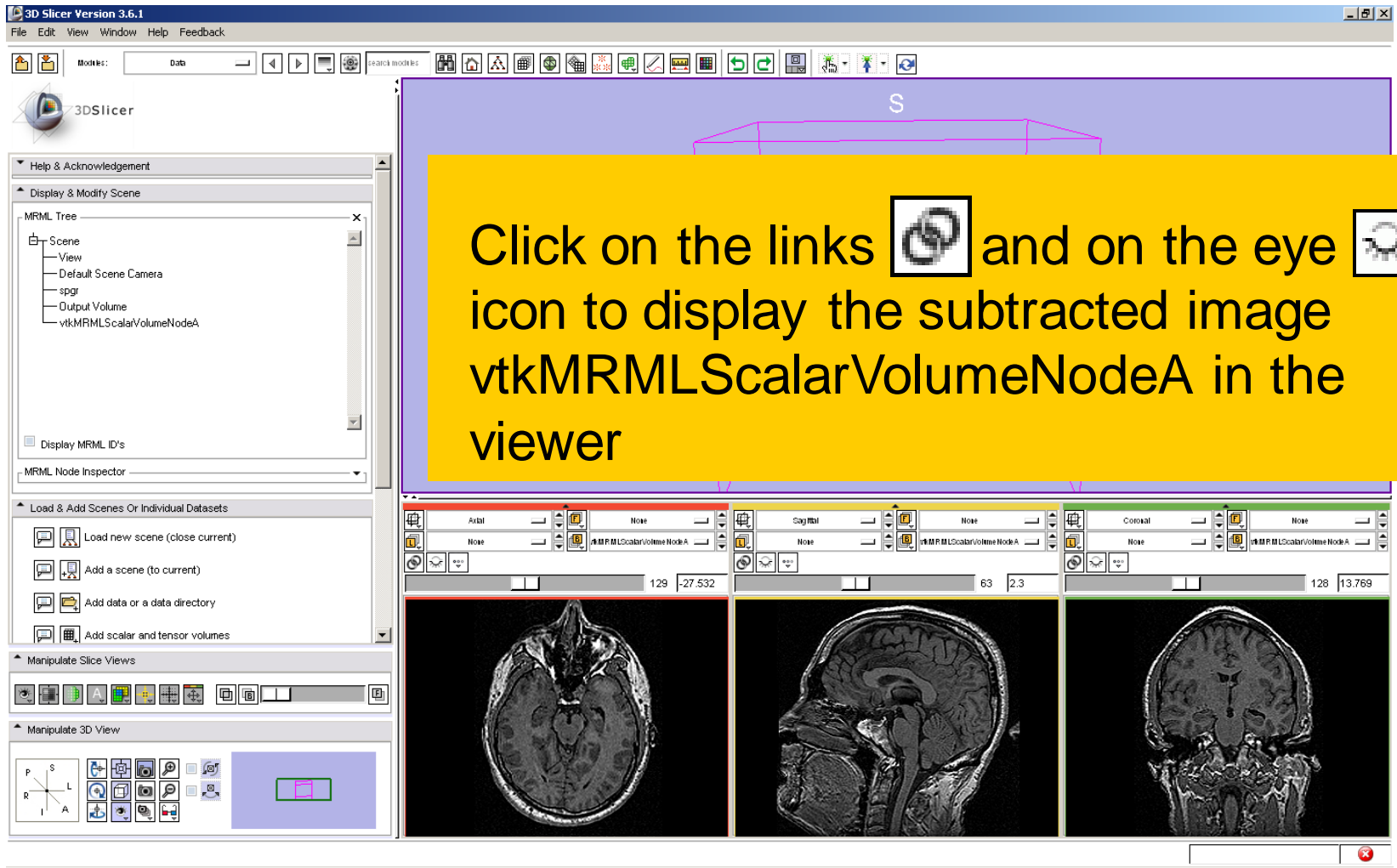


The screenshot shows the 3D Slicer 3.6.1 interface. The MRML Tree on the left lists the following nodes: Scene, View, Default Scene Camera, spgr, Output Volume, and **vtkMRMLScalarVolumeNodeA**. A red arrow points to this node. The main 3D view area shows three orthogonal slice views: Axial, Sagittal, and Coronal. Each view has a corresponding control panel with 'None' and 'Output Volume' buttons. The slice views display a grayscale image of a brain scan.



Select the module **Data**

The subtracted image **vtkMRMLScalarVolumeNodeA** appears in the data tree

# Image Sharpening



The screenshot shows the 3D Slicer 3.6.1 interface. On the left, the MRML Tree panel is visible, showing a hierarchy: Scene > View > Default Scene Camera > spgr > Output Volume > vtkMRMLScalarVolumeNodeA. Below this is the MRML Node Inspector. The main view area displays three orthogonal MRI slices: Axial, Sagittal, and Coronal. A yellow box is overlaid on the main view area, containing the following text and icons:

Click on the links  and on the eye  icon to display the subtracted image vtkMRMLScalarVolumeNodeA in the viewer

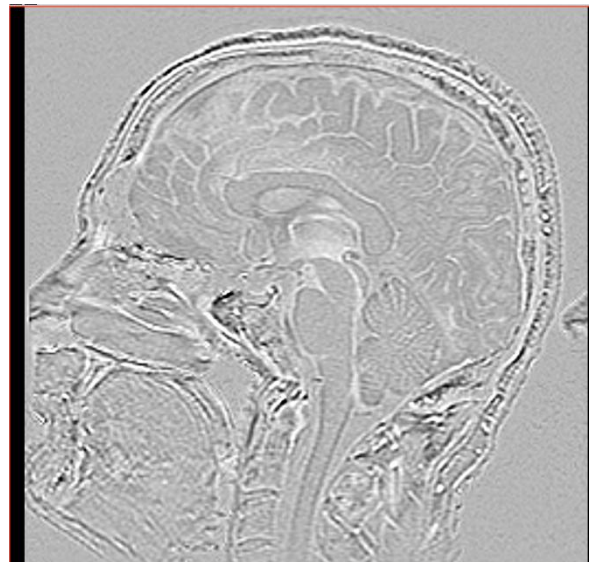
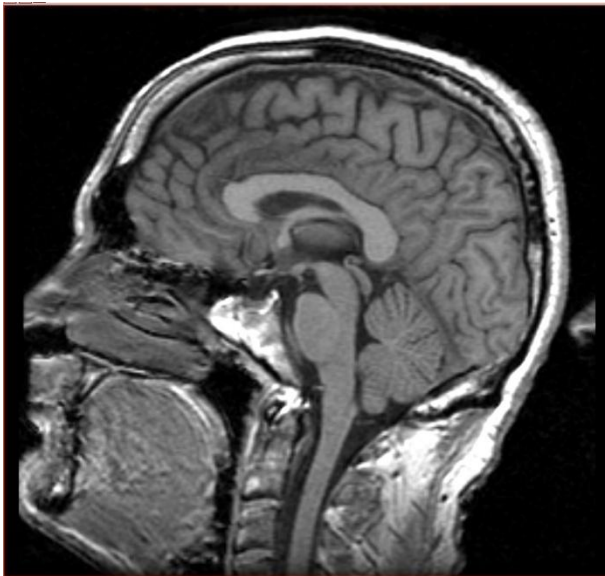
The interface also includes a top menu bar (File, Edit, View, Window, Help, Feedback), a toolbar with various icons, and a 3D view manipulator at the bottom left.

# Image Sharpening

original

Laplacian

Laplacian filtered





# Conclusion

---

- This course demonstrates how to integrate an external program in Python within Slicer3
- The **Execution Model** of Slicer3 provides a simple mechanism for incorporating command line programs as Slicer modules in Python.





# Acknowledgments

---



**National Alliance for Medical Image Computing**

NIH U54EB005149



**Neuroimage Analysis Center**

NIH P41RR013218