




MACBIO

Marine and Coastal Biodiversity Management



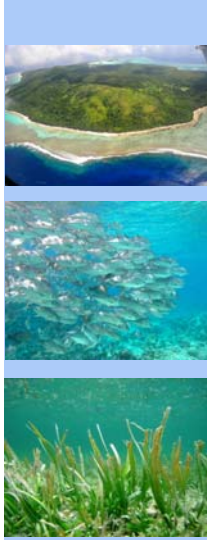



On behalf of:
 Federal Ministry
 for the Environment, Nature Conservation,
 Building and Nuclear Safety
 of the Federal Republic of Germany



Python Geoprocessing Modules

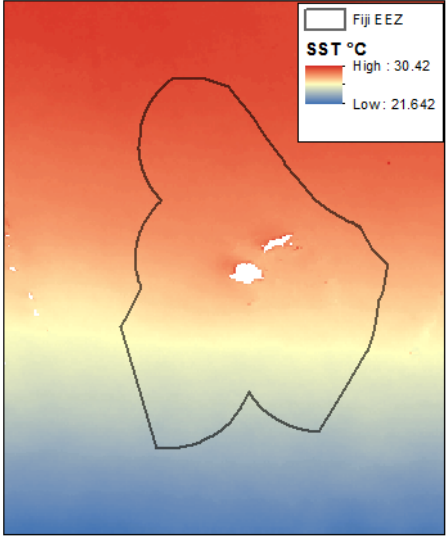
- **arcpy** raster/vector/crs processing module
 - ArcGIS (**closed** source software)
- **gdal/ogr/osr** raster/vector/crs processing modules
 - QGIS (**free open** source software)




17.11.2015

MACBIO

Task: Clip Raster with Polygon

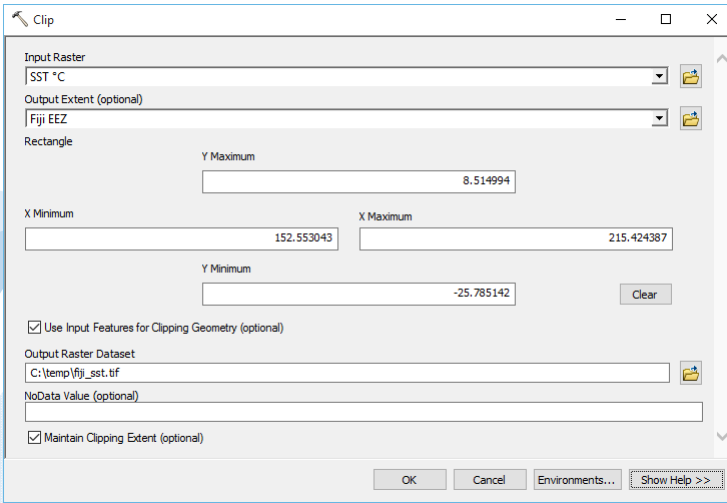


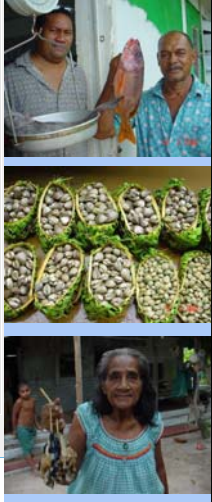


EEZ: <http://www.marineregions.org/>
 SST: <http://www.oracle.ugent.be/>

MACBIO

Task: Clip Raster with Polygon [ArcMap GUI]

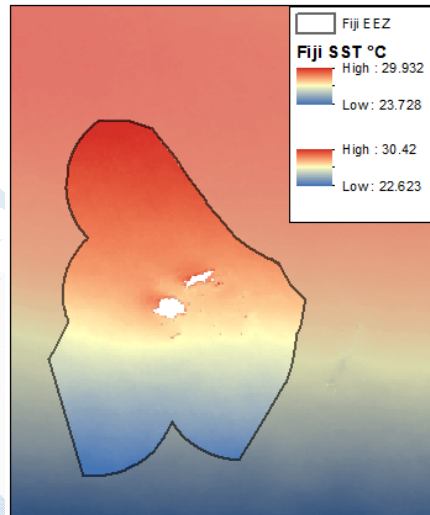




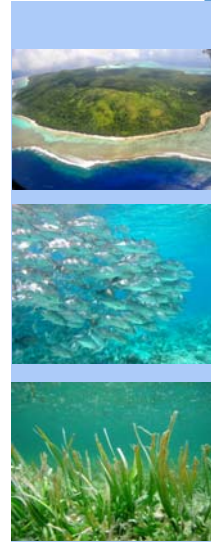
17.11.2015

MACBIO

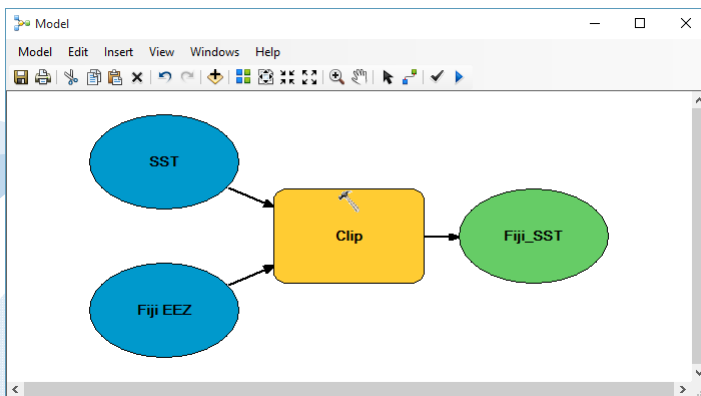
Output from ArcMap GUI



17.11.2015



ArcGIS Model Builder: design process

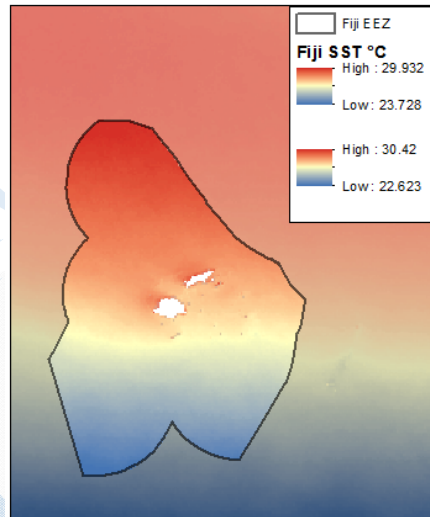


17.11.2015

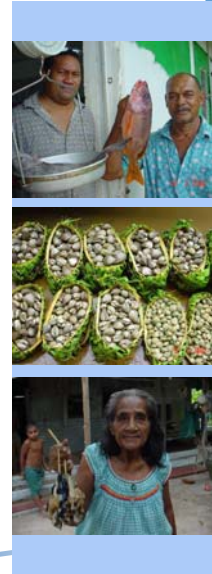
MACBIO



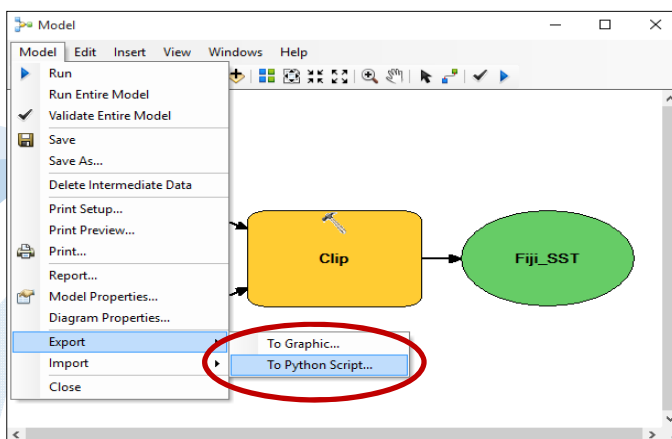
Output from ArcGIS Model Builder



17.11.2015

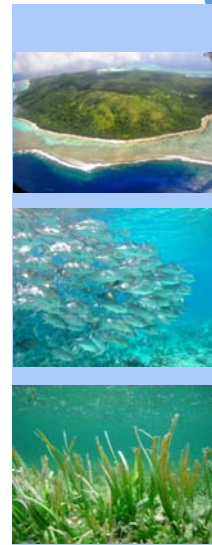


ArcGIS Model Builder: export python script

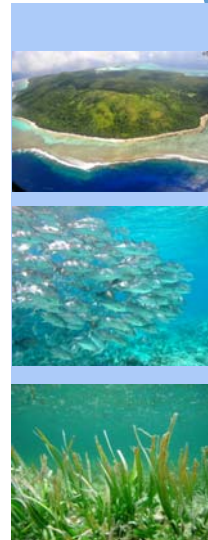
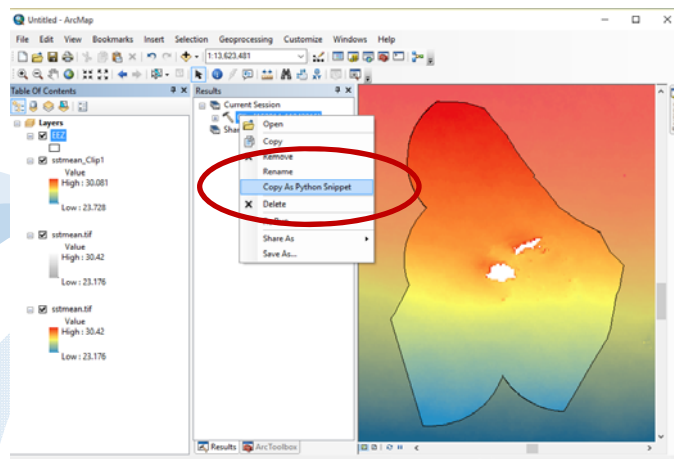


17.11.2015

MACBIO



ArcGIS Model Builder: export python script



17.11.2015

MACBIO

Python script from ArcGIS Model Builder

Import arcpy module
import arcpy

Import Module

Local variables:

inFiji_EEZ = r"C:\temp\FIJI_EEZ.shp"

inSST = r"C:\temp\SST.tif"

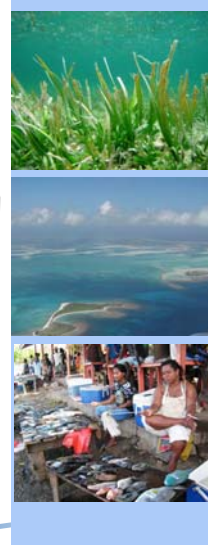
outFiji_SST = r"C:\temp\FIJI_SST.tif"

Assign input/output

Process: Clip

arcpy.Clip_management(SST, "#", Fiji_SST, Fiji_EEZ, "#", "ClippingGeometry")

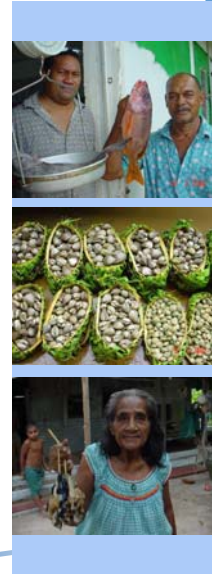
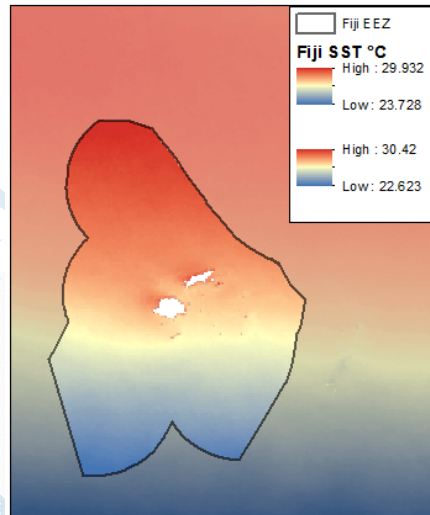
Process Data



17.11.2015

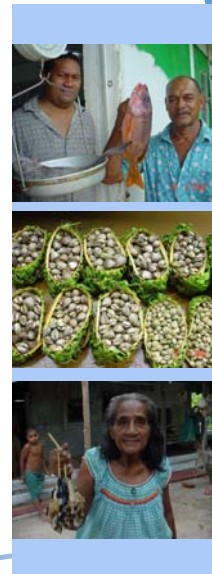
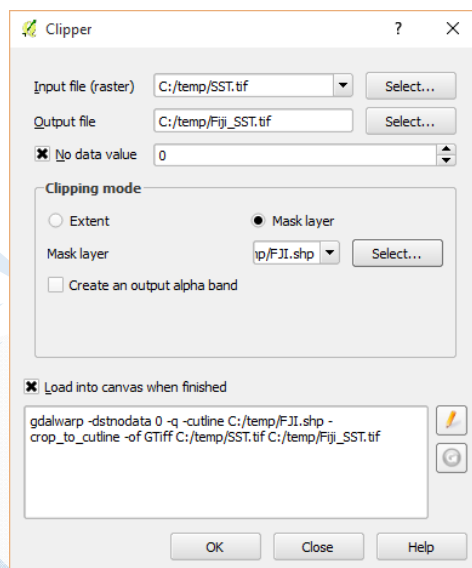
MACBIO

Output from python script



17.11.2015

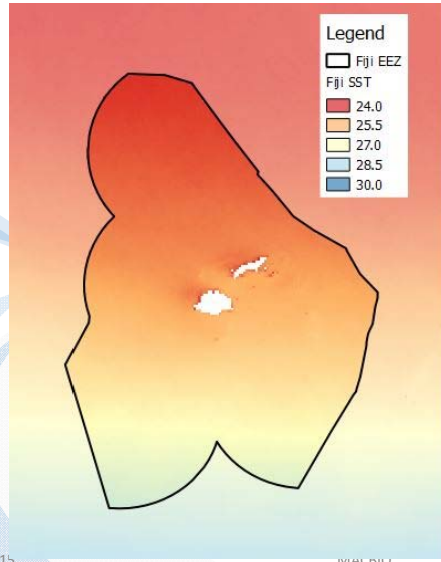
Task: Clip Raster with Polygon [QGIS GUI]



17.11.2015

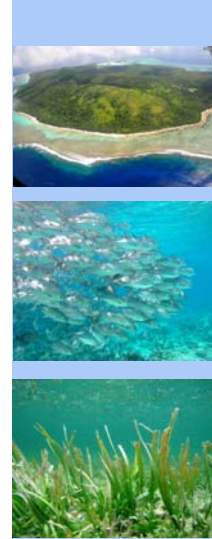
IVTACBIO

Output from QGIS GUI



17.11.2015

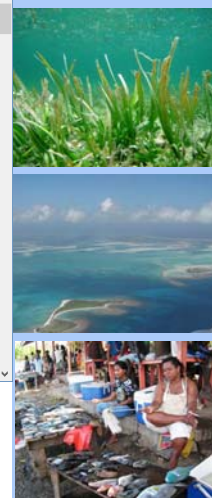
MACBIO



Clip Raster with Polygon [QGIS]

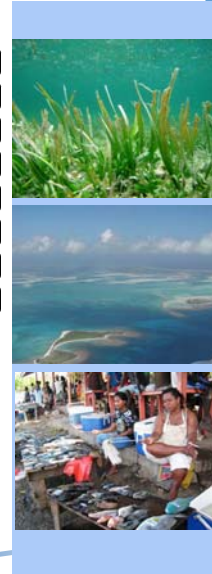
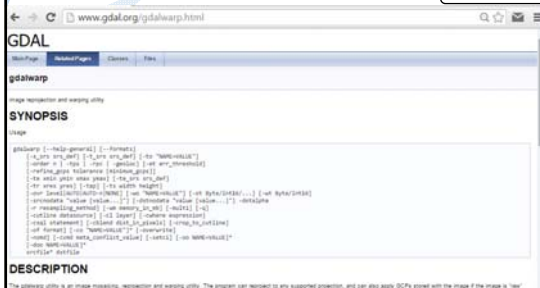
The screenshot shows the QGIS Clipper dialog box and a console window. The Clipper dialog has the following settings: Input file (raster) is 'C:/temp/SST.tif', Output file is 'C:/temp/Fiji_SST.tif', No data value is '0', Clipping mode is 'Mask layer', and Mask layer is 'fp/FJI.shp'. The console window shows the command: `gdalwarp -dstnodata 0 -q -cutline C:/temp/FJI.shp -crop_to_cutline -of GTiff C:/temp/SST.tif C:/temp/Fiji_SST.tif`. A white arrow points from the console command to the Mask layer dropdown in the Clipper dialog. A white circle highlights the console command, with a label 'Console command generated by QGIS' pointing to it.

MACBIO



Clip Raster with Polygon [GDAL]

gdalwarp.exe Call executable program
 -dstnodata 0 Assign nodata value
 -q Quiet (no progress reports)
 -cutline C:/temp/EEZs/FJI.shp Assign clip geometry
 -crop_to_cutline Write nodata outside clip
 -of GTiff Assign output format
 C:/temp/SST.tif Assign input file
 C:/temp/EEZs/FJI_SST.tif Assign output file



Python script from GDAL Command

```
# Import arcpy module
import os

# Local variables:
inFiji_EEZ = r"C:\temp\FJI_EEZ.shp"
inSST = r"C:\temp\SST.tif"
outFiji_SST = r"C:\temp\FJI_SST.tif"

# Process: Clip
command = ['gdalwarp',
           '-dstnodata 0',
           '-cutline',
           inFiji_EEZ,
           '-crop_to_cutline',
           '-of GTiff',
           inSST,
           outFiji_SST]
os.system(' '.join(command))
```

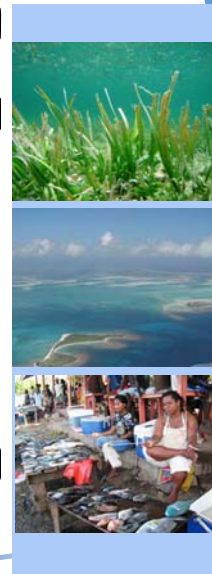
Import Module

Assign input/output

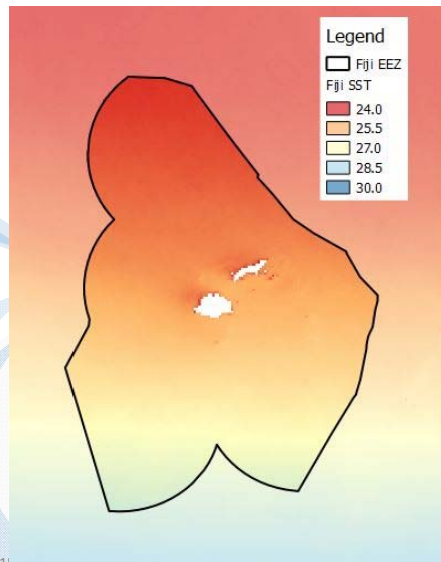
Process Data

17.11.2015

MACBIO

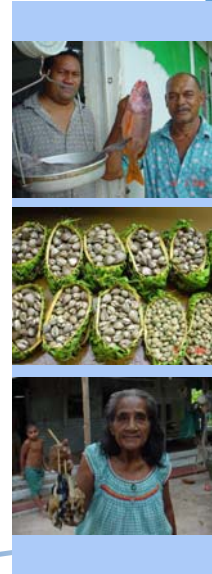


Output from command line



17.11.2015

MACBIO



Automation of arcpy using python script

```

# Import modules
import os
# Local variables:
shapeFolder = r"C:\temp\EEZs"
sstRaster = r"C:\temp\SST.tif"
# Set workspace
arcpy.env.workspace = shapeFolder
# Process: Clip
for ds in arcpy.ListFeatureClasses():
    outRaster = ds.split('.')[0] + "_SST.tif"
    arcpy.Clip_management(sstRaster, "#", outRaster, ds, "#",
        "ClippingGeometry")
    
```

Import Module

Assign input

Process list of datasets

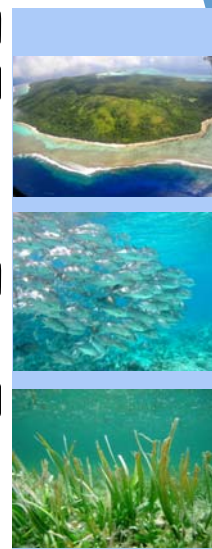
Assign output

Process Data

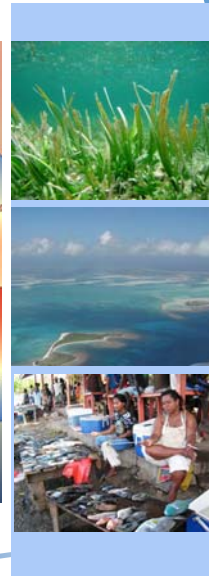
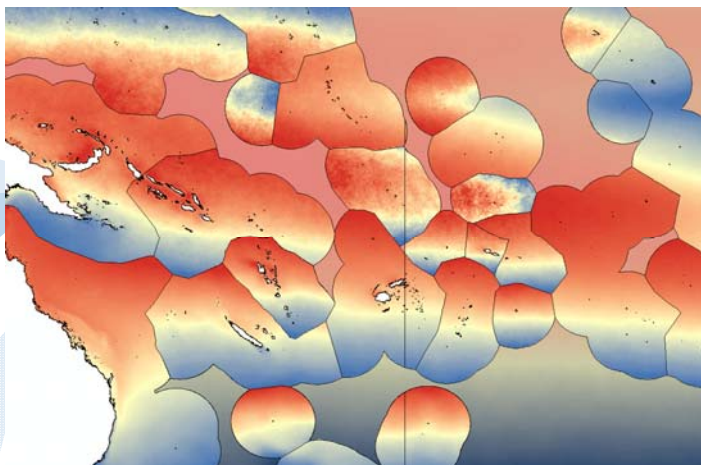
Execution time: 115.5 seconds

17.11.2015

MACBIO



Output from automated arcpy python script



17.11.2015

MACBIO

Automation of GDAL using python script

```
# Import modules
import os
# Local variables:
shapeFolder = r"C:\temp\EEZs"
sstRaster = r"C:\temp\SST.tif"
# Process: Clip
for ds in os.listdir(shapeFolder):
    if ds.endswith('.shp'):
        outRaster = ds.split('.')[0] + "_SST.tif"
        command = ['gdalwarp',
                  '-dstnodata 0',
                  '-cutline',
                  inFiji_EEZ,
                  '-crop_to_cutline',
                  '-of GTiff',
                  inSST,
                  outFiji_SST]
        os.system(' '.join(command))
```

Import Module

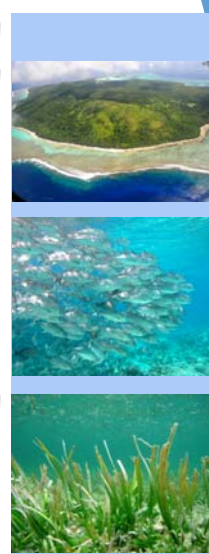
Assign input

Process list of datasets

Assign output

Process Data

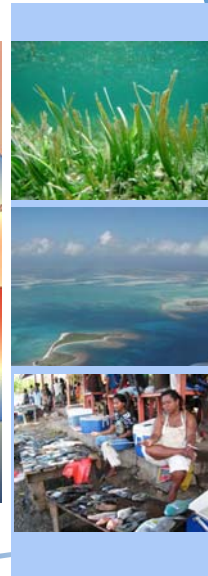
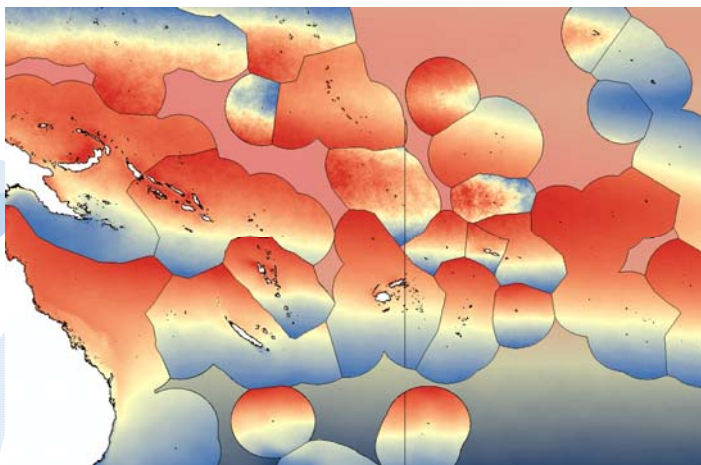
Execution time: 20.4 seconds



17.11.2015

MACBIO

Output from automated gdal python script



17.11.2015

MACBIO

Further visualization of spatial data

```

# import modules
import arcpy
import matplotlib.pyplot as plt
import numpy as np
# set local variables
shapeFolder = r"C:\temp\EEZs"
sstRaster = r"C:\temp\SST.tif"
rasterList = []
countryList = []
# process each shapefile
arcpy.env.workspace = shapeFolder
for ds in arcpy.ListFeatureClasses():
    outRaster = ds.split('.')[0] + "_SST.tif"
    arcpy.Clip_management(sstRaster, "#", raster, ds, "#", "ClippingGeometry")
    array = arcpy.RasterToNumPyArray(outRaster)
    rasterList.append(np.ma.compressed(np.ma.masked_less_equal(array, 0)))
    countryList.append(ds.split('.')[0])
# create boxplots
fig, ax = plt.subplots()
ax.set_ylabel("degrees C", )
bp = plt.boxplot(rasterList)
plt.xticks(np.arange(1, len(countryList)+1), countryList, rotation='vertical')
plt.title('SST values for each country')
plt.show()
    
```

Import Modules

Assign input

Create storage in memory

Process list of datasets

Create output rasters

Write data to memory

Create boxplot object

Create y-axis label

Add data to boxplots

Add x-axis label

Add title

17.11.2015

MACBIO

Output of python script

