

Hemanta Pokharel

Dr. Lei Wang

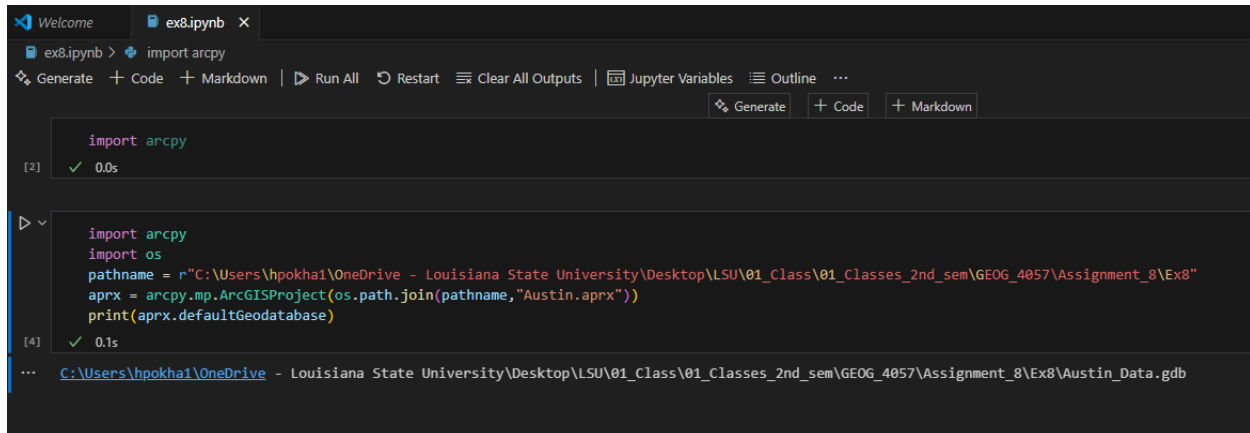
GEOG 4057

April 17, 2025

Work with maps

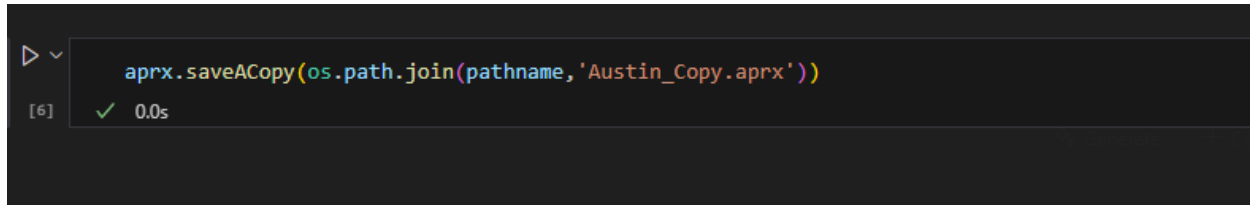
Manipulate the project in VS Code

1. Open the project and print some information



The screenshot shows a Jupyter Notebook interface in VS Code. The notebook has two cells. The first cell contains the code `import arcpy` and has been executed successfully, as indicated by a green checkmark and the output `[2] ✓ 0.0s`. The second cell contains the following code: `import arcpy`, `import os`, `pathname = r"C:\Users\hpokha1\OneDrive - Louisiana State University\Desktop\LSU\01_Class\01_Classes_2nd_sem\GEOG_4057\Assignment_8\Ex8"`, `aprx = arcpy.mp.ArcGISProject(os.path.join(pathname, "Austin.aprx"))`, and `print(aprx.defaultGeodatabase)`. This cell has also been executed successfully, with a green checkmark and the output `[4] ✓ 0.1s`. Below the code editor, the output of the second cell is displayed as a file path: `C:\Users\hpokha1\OneDrive - Louisiana State University\Desktop\LSU\01_Class\01_Classes_2nd_sem\GEOG_4057\Assignment_8\Ex8\Austin_Data.gdb`.

2. Save a copy of the project (10 pnts)

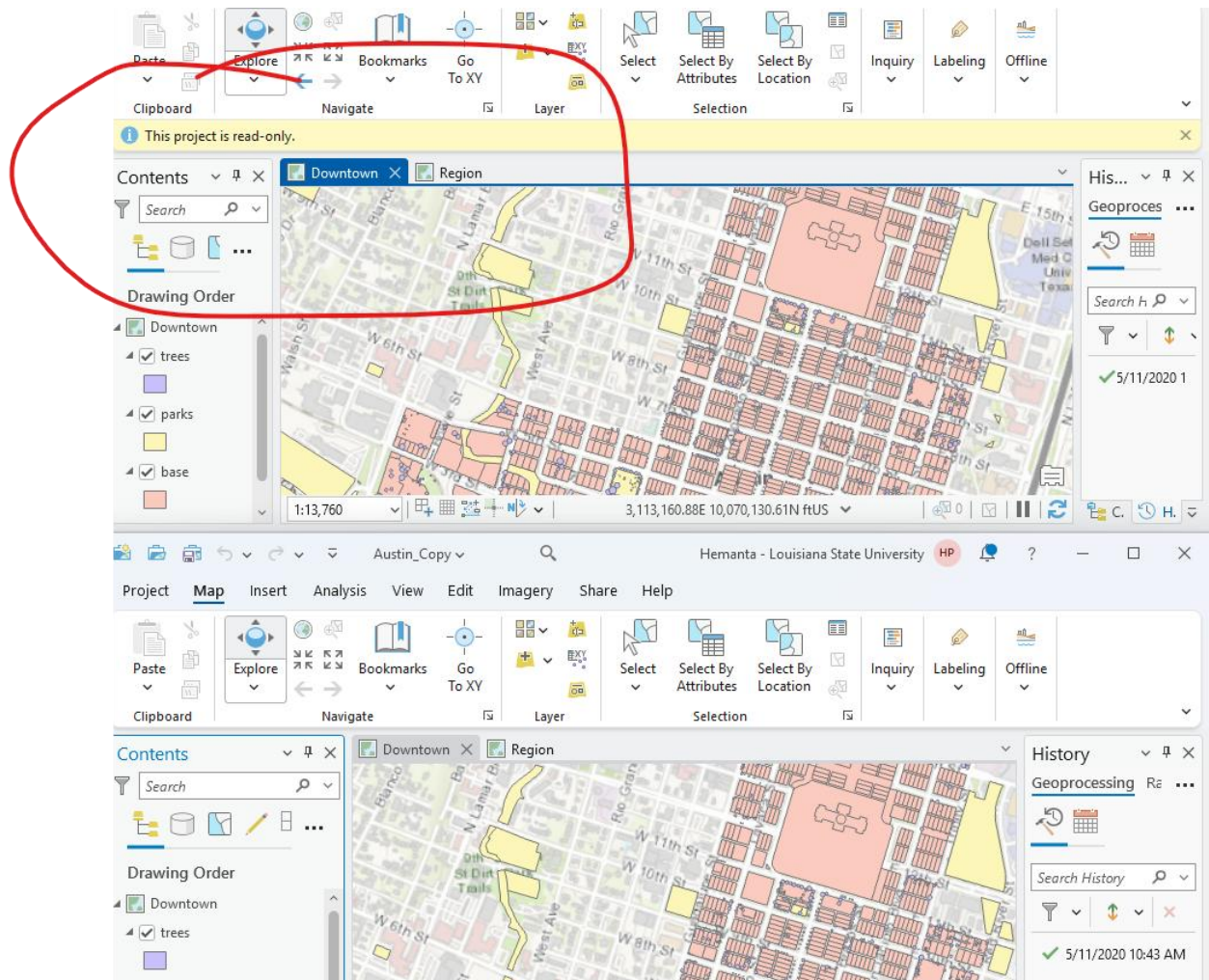


The screenshot shows a Jupyter Notebook interface in VS Code. The notebook has one cell containing the code `aprx.saveACopy(os.path.join(pathname, 'Austin_Copy.aprx'))`. This cell has been executed successfully, as indicated by a green checkmark and the output `[6] ✓ 0.0s`.

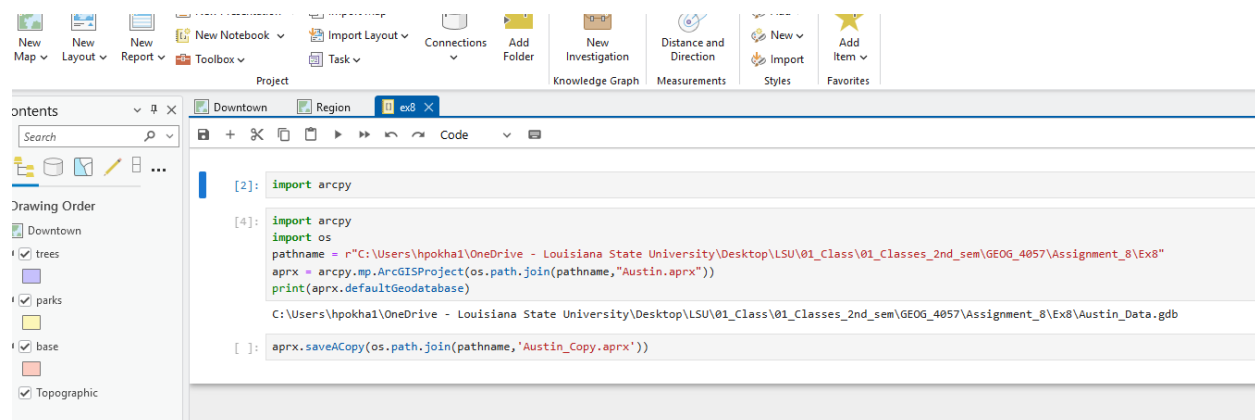
3. Examine the new project file (10 pnts)

Is everything the same as the original project? (Compare the difference in the file sizes) and why?

Ans: The original file is read only and the new file is editable. For the original read only file, ArcGIS Pro might have store extra internal metadata or safety buffers to protect the file from accidental edits.



- Leave the ArcGIS Pro and the project opened because the following instructions will be based on this project
- Add the notebook you created for this exercise into the project - notebooks
- Open the added notebook



Work with maps

1. print a list of maps in the project

```
[3]: aprx = arcpy.mp.ArcGISProject('CURRENT')
     maps = aprx.listMaps()
     for m in maps:
         print(m.name)
         print(m.mapUnits)
     del aprx
```

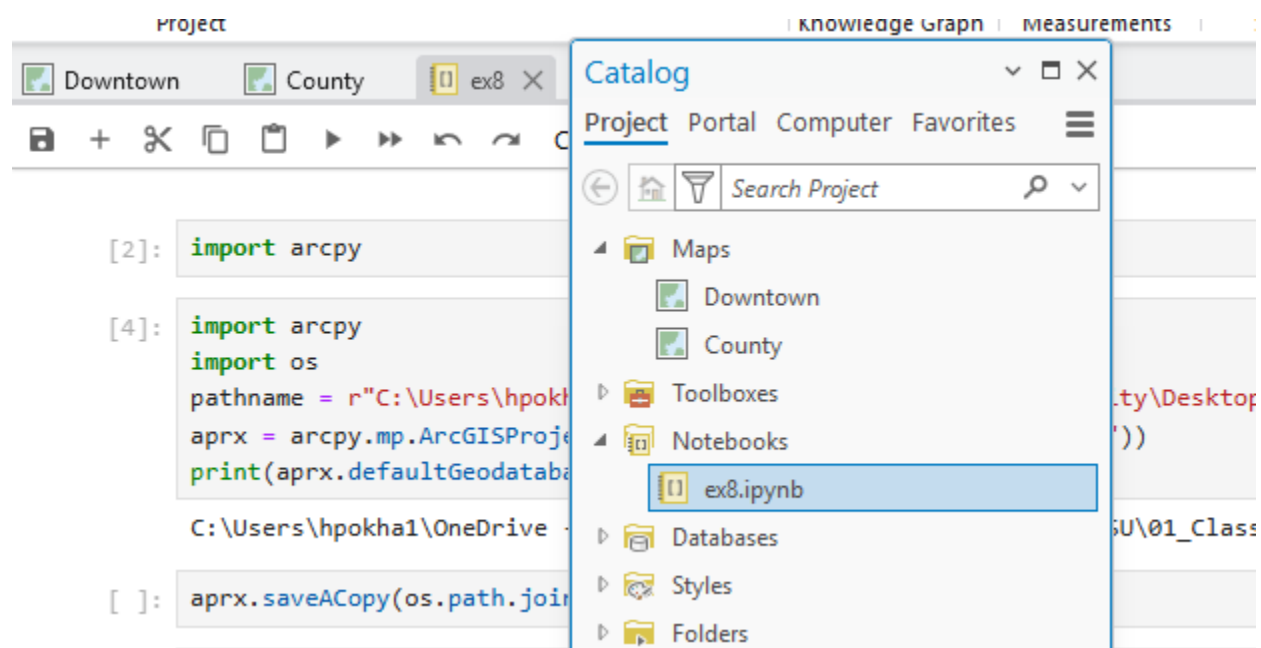
```
Downtown
Foot_US
Region
Foot_US
```

What does del do in the last line? Was the project file deleted?

Ans: Del line removes the objects and make the memory free after using the project and done with it. No, the project file wasn't deleted.

2. change the name of a map

In the catalog pane the name of the map “region” is changed to “county”

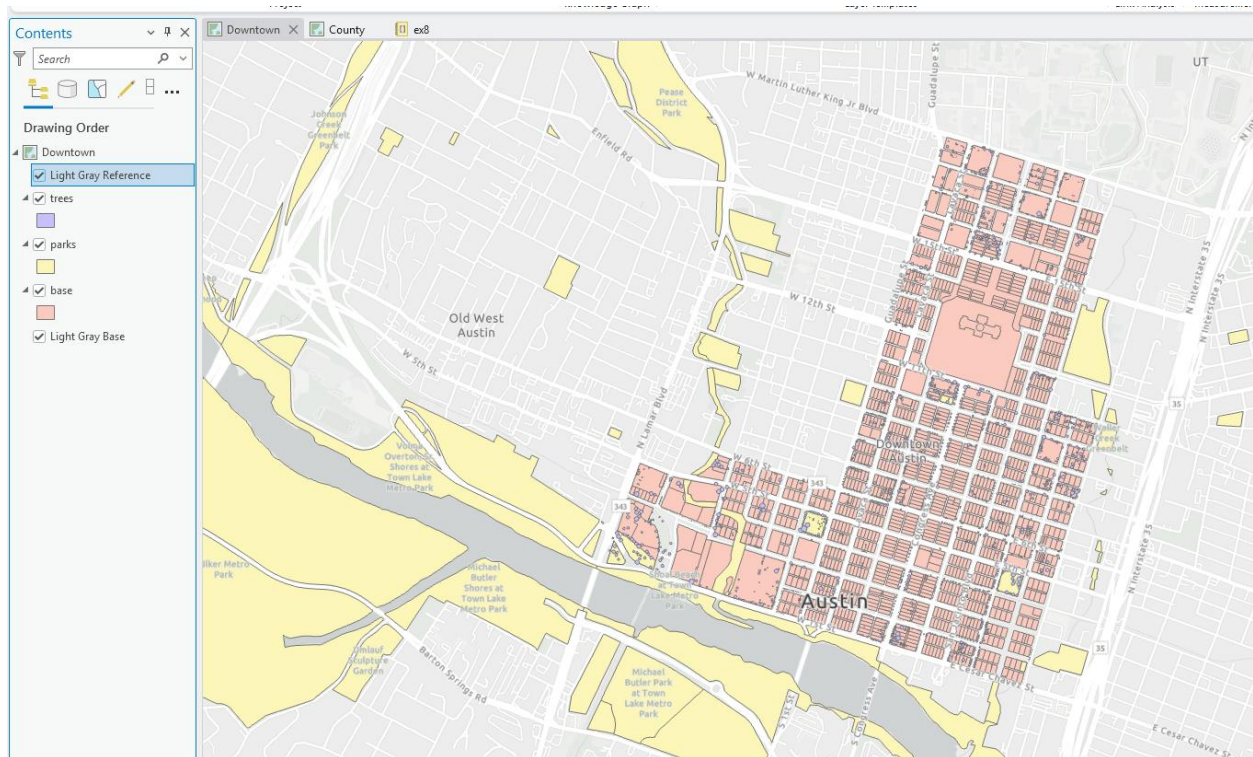


3. list the layers in a map

```
[5]: aprx = arcpy.mp.ArcGISProject('CURRENT')
maps = aprx.listMaps()
for m in maps:
    print("Map: " + m.name)
    lyrs = m.listLayers()
    for lyr in lyrs:
        print(lyr.name)
del aprx
```

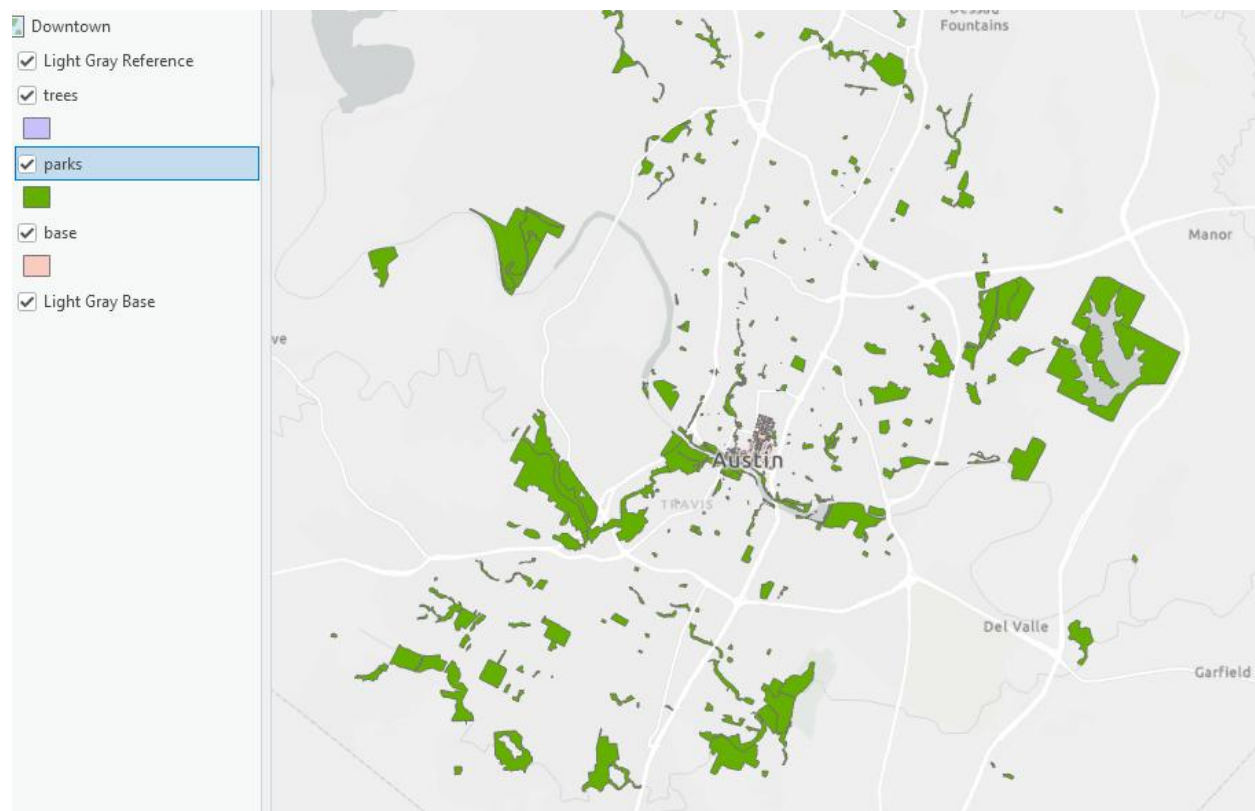
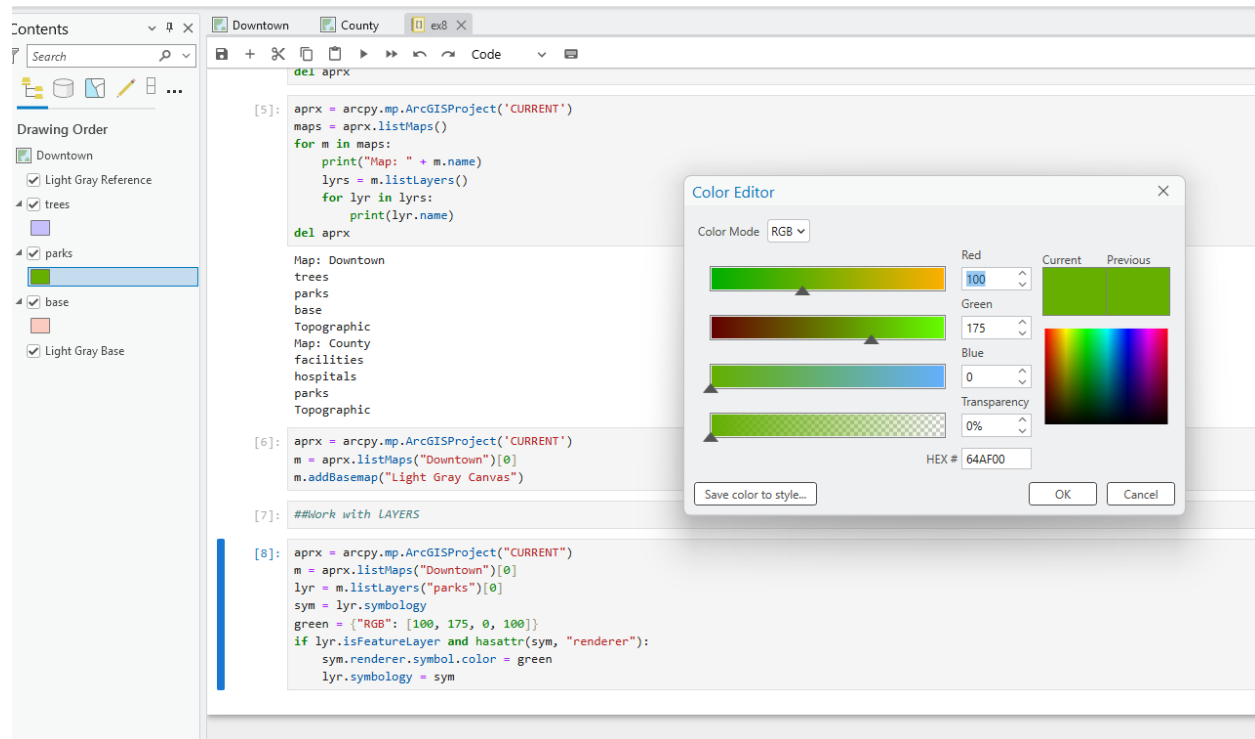
```
Map: Downtown
trees
parks
base
Topographic
Map: County
facilities
hospitals
parks
Topographic
```

4. Change the basemap



Work with layers

1. change the symbology of layers

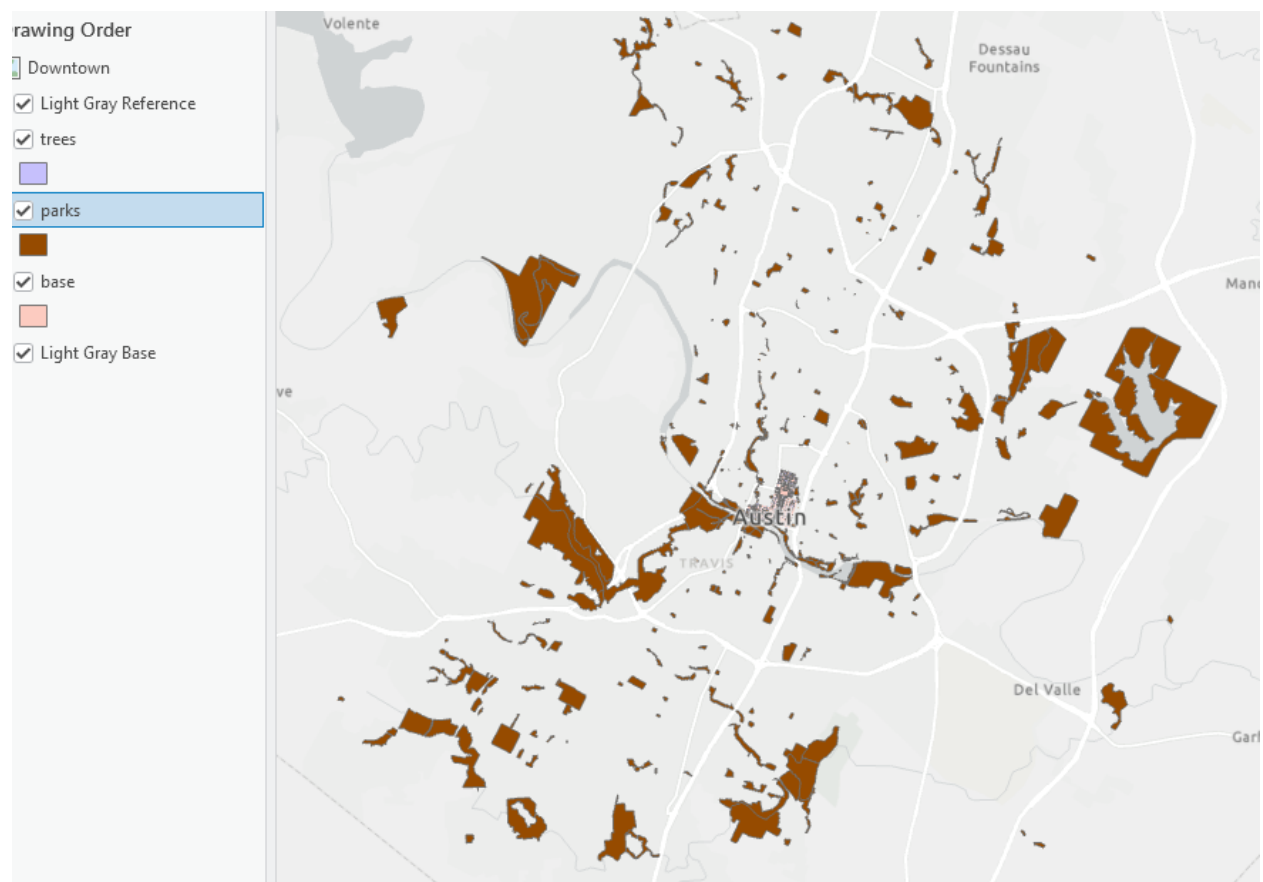


2. What type is the 'green' variable? What do the numbers in the list mean? If you want to show a brown color, what numbers you would fill it with? Change it in the block and confirm the color is changed to brown.

It is a dictionary and “RGB” is key. The numbers are value for RED, GREEN, BLUE, and TRANSPARENCY. In this case it is greenish color with zero transparency (opaque).

It can be made brown with the code `brown = {"RGB": [150, 75, 0, 100]}`

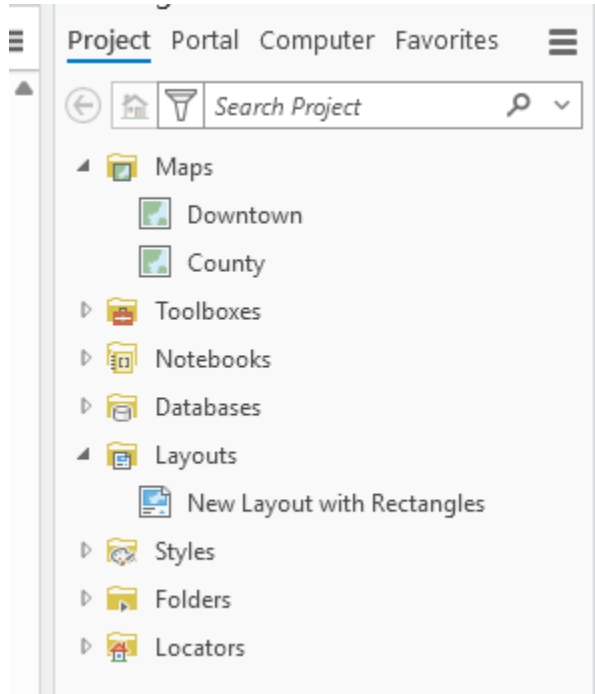
```
[11]: aprx = arcpy.mp.ArcGISProject("CURRENT")
      m = aprx.listMaps("Downtown")[0]
      lyr = m.listLayers("parks")[0]
      sym = lyr.symbology
      brown = {"RGB": [150, 75, 0, 100]} # Brown color
      if lyr.isFeatureLayer and hasattr(sym, "renderer"):
          sym.renderer.symbol.color = brown
          lyr.symbology = sym
```



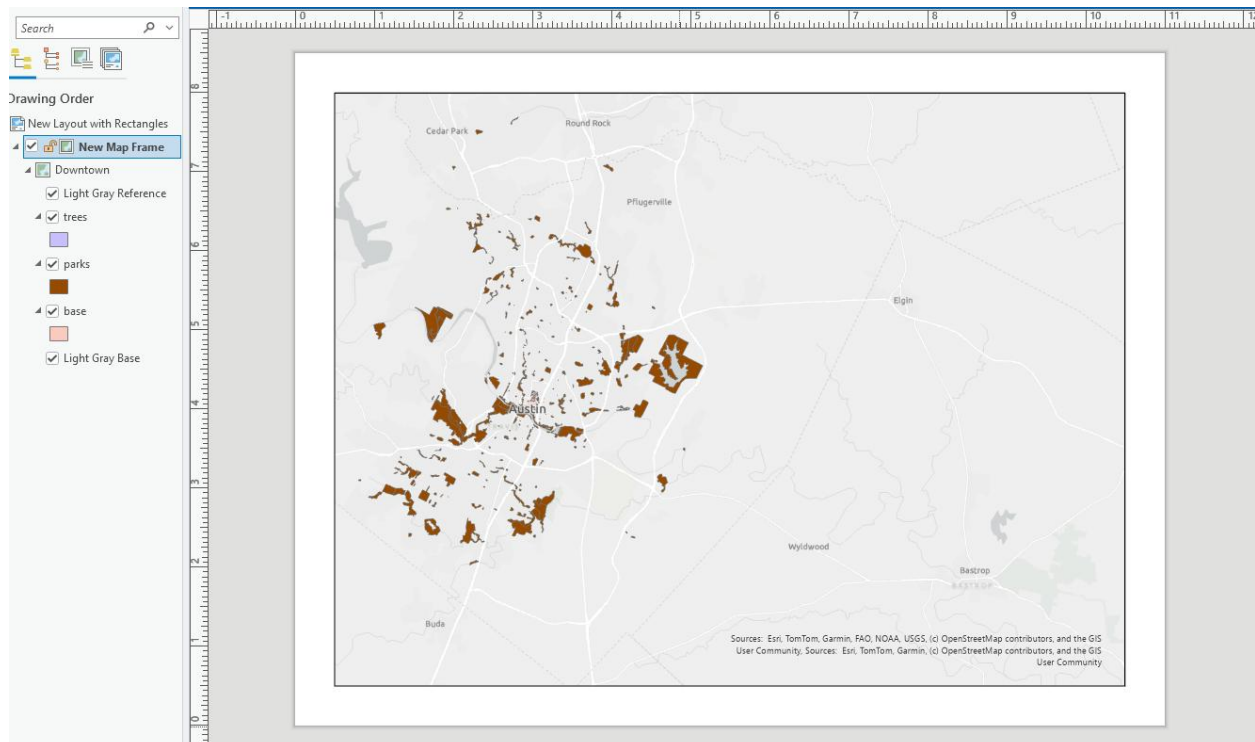
Work with a layout

1. Add a layout to the project

Create a layout and insert the map "Downtown" in the layout



The size of the layout is 11 inches x 8.5 inches which is landscape US Letter size.

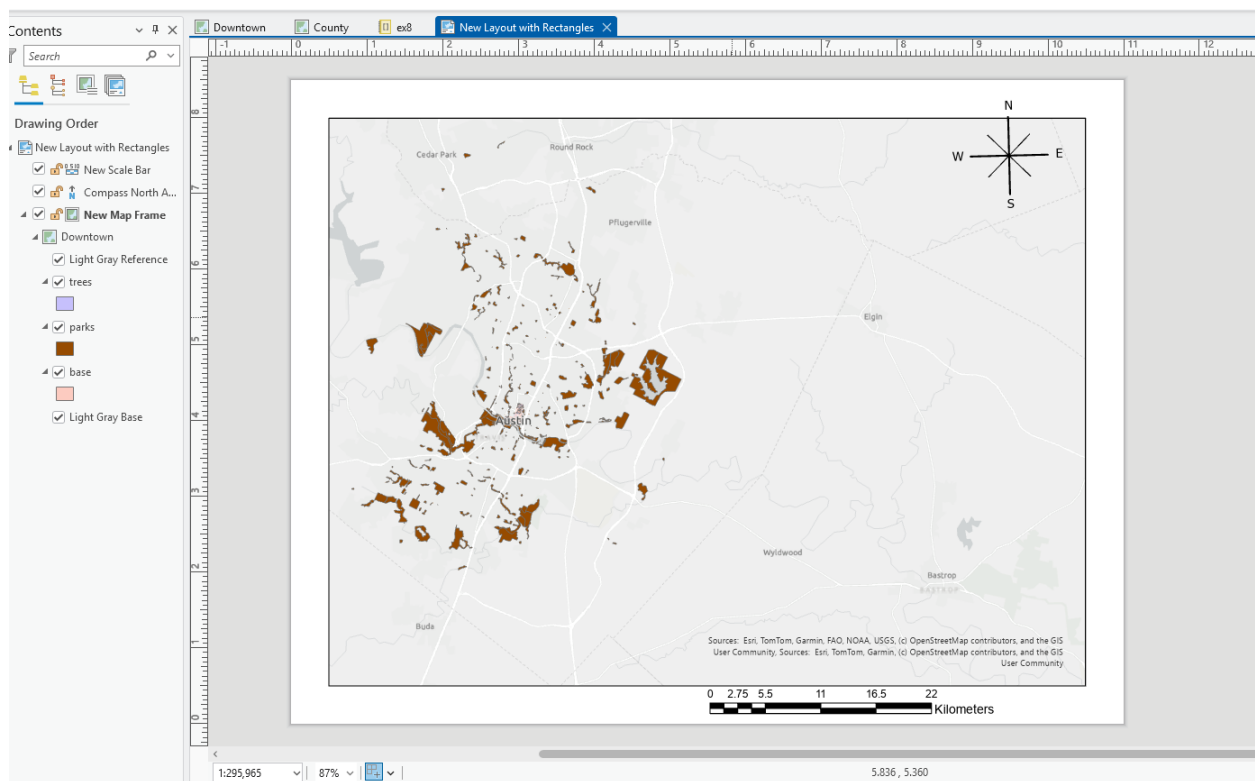


The size of the map frame is 10 inches x 7.5 inches. It start at the position lower left at (0.5,0.5) that means 0.5 inch margin all around.

The advantages of using cod cover the manual layout setup are:

- More precise control: color code can help have the exact color if we want.
- Automates repetitive tasks
- Saves time and reduces human error
- Great for bulk or dynamic map production

2. Add layout elements



3. Run the follow code to create a legend

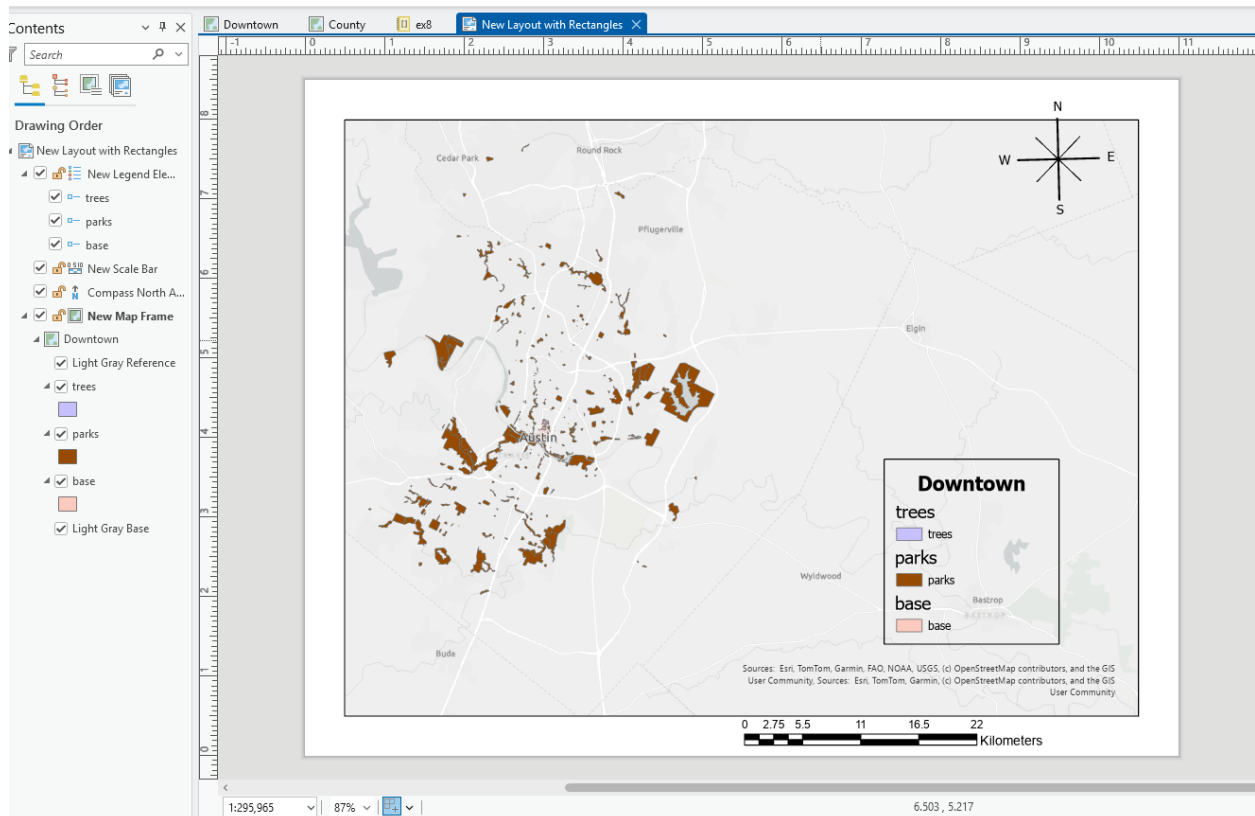
```
mt = lyt.createMapFrame(makeRec_LL(0.5,0.5,10,7.5), m, "New Map Frame")

[15]: #Create a north arrow
naStyle = aprx.listStyleItems('ArcGIS 2D', 'North_Arrow', 'Compass North 1')[0]
na = lyt.createMapSurroundElement(arcpy.Point(9.5,7.5), 'North_Arrow', mf,
                                naStyle, "Compass North Arrow")

na.elementWidth = 0.5

#Create a scale bar
sbName = 'Double Alternating Scale Bar 1 Metric'
sbStyle = aprx.listStyleItems('ArcGIS 2D', 'Scale_bar', sbName)[0]
sbEnv = MakeRec_LL(5.5, 0.1, 4, 0.5)
sb = lyt.createMapSurroundElement(sbEnv, 'Scale_bar', mf, sbStyle, 'New Scale Bar')

[18]: legSi = aprx.listStyleItems('ArcGIS 2D', 'LEGEND', 'Legend 3')[0]
leg = lyt.createMapSurroundElement(arcpy.Point(1,7), 'LEGEND', mf, legSi, 'New Legend Element')
leg.elementWidth = 3
leg.elementHeight = 3
leg.fittingStrategy = 'AdjustFontSize'
leg.columnCount = 1
leg.title = 'Downtown'
```



4. Export the layout to a pdf

```
[22]: pathname = r"C:\Users\hpokha1\OneDrive - Louisiana State University\Desktop\LSU\01_Class\01_Classes_2nd_sem\GEOG_4057\Assignment_8\Ex8"  
      lyt.exportToPDF(os.path.join(pathname, 'downtown.pdf'))  
  
[22]: 'C:\Users\hpokha1\OneDrive - Louisiana State University\Desktop\LSU\01_Class\01_Classes_2nd_sem\GEOG_4057\Assignment_8\Ex8\downtown.pdf'
```

