**GUIDANCE MANUAL for the**

**MATCH MAP STYLE TOOLBOX**

Aug 2021

Aijing Li

**Contents**

Section 1. Background Information .………………………..…………...……….………….. p.1

Section 2. Setup …………………………………..………………………………..………….. p.3

Section 3. Interface & Code …………………………………………………...….………….. p.5

Section 4. Pre-Execution Checklist ……...…………...……………………………..……….. p.7

Section 5. Running the Toolbox & Generating Outputs ………..………………………….. p.8

Section 6. Errors, Warnings & Cautions …………………..…………………..…………….. p.11

Section 7. Post-Execution Adjustments ………….……..……………………….….……….. p.13

Section 8. Adding New Templates & Updating Packages …….……………….………….. p.14

**Introduction**

This manual is a comprehensive guide to the Match Map Style Toolbox, which was produced for the City Resilience Program (CRP) at the World Bank Group’s Global Facility for Disaster Reduction and Recovery. The Toolbox automates the process of producing and symbolizing the maps used for CRP’s City Scan product from raw data inputs. It has been implemented in Python 2.7 with the ArcPy package under ArcMap 10.8, as well as the NumPy and pandas packages.

This manual encompasses eight Sections. Section 1 includes an overview and details on the Toolbox’s Python code structure. Sections 2, 3, and 4 describe the setup and preparation steps needed before running the Toolbox, especially for users new to it. Sections 5, 6, and 7 explain how to run the Toolbox, troubleshoot possible issues afterwards, and make adjustments to the outputs. Section 8 contains recommendations on updating the Toolbox’s Python scripts to incorporate new map templates that may be added in the future.

1. **Background Information**

**1.1 Objective**

The objective of the Match Map Style Toolbox is to generate a set of specific maps and accompanying spatial analyses for a specific AOI based on predefined raw data, symbology, and styles. It relies on templates and raw data from a “source city” and raw data from a “new city” to generate new maps using predermined styles for the new city.

**1.2 Mechanism of the Toolbox**

In general, when importing data into ArcMap, a layer with a link to the data source and associated symbology settings is generated. This Toolbox can be considered as asset of “find-and-replace” operations that exploits these two layer properties. Specifically, it iterates through all the layers inside a map document (.mxd), finds layers linked to a data file, and replaces the original link with a new link to the new city data folder. For example, consider a layer linked to a population raster (.tif):

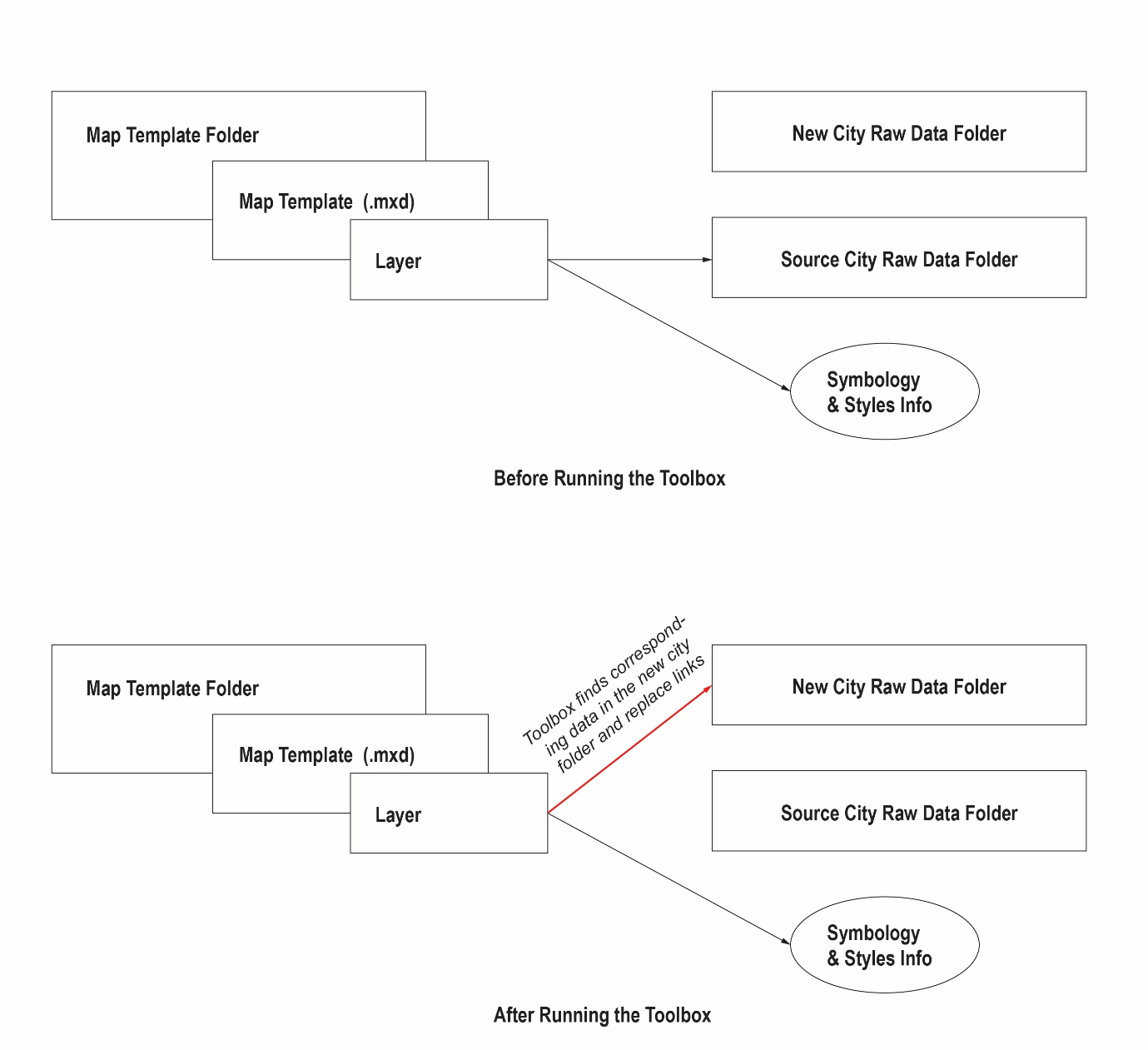
C:\....\**sourcecityfolder**\... \01\_population\_**Sourcecity**\_AOI.tif

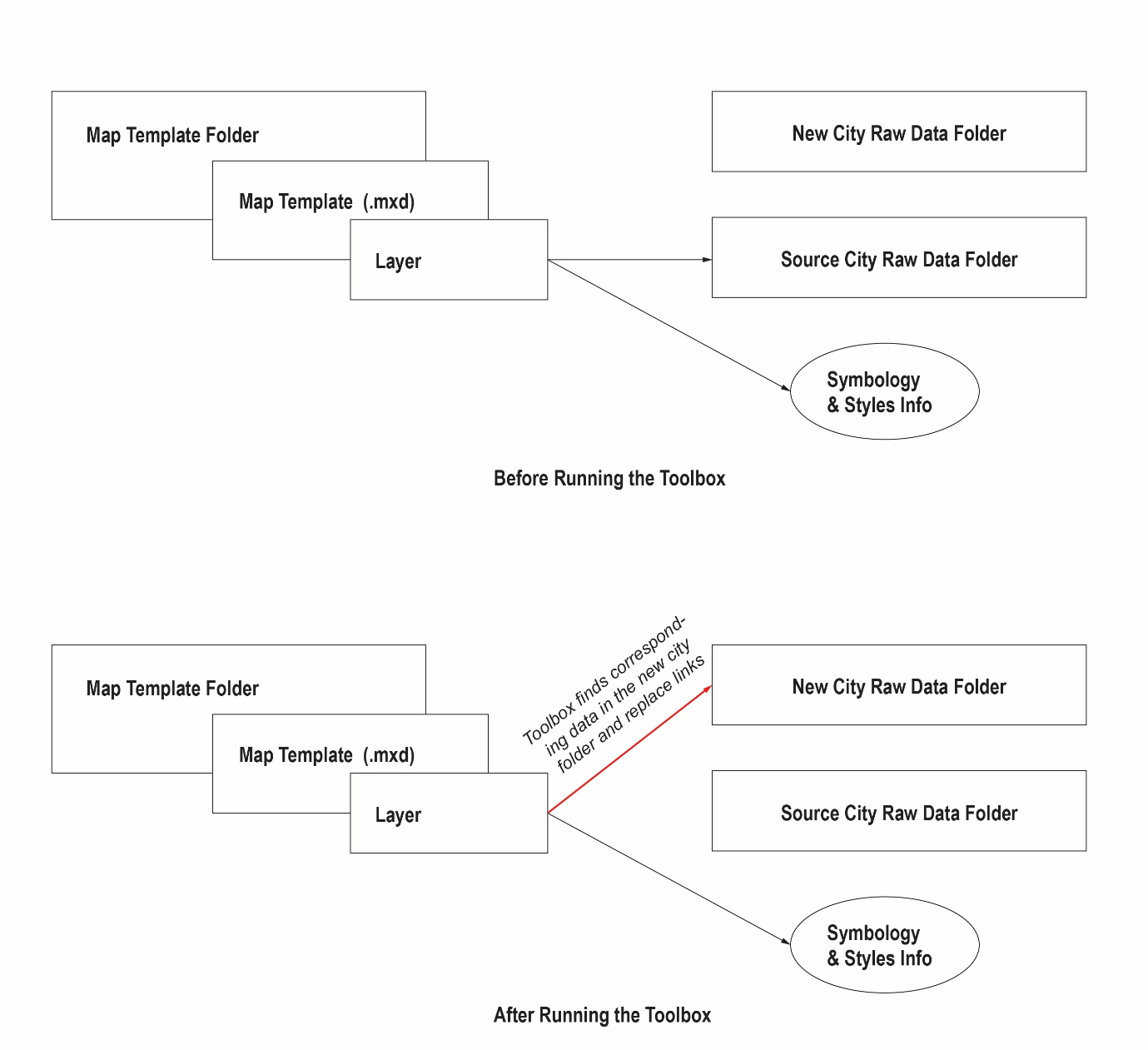
The toolbox searches in the new city folder until it finds a file called:

C:\....\**newcityfolder**\...\01\_population\_**Newcity**\_AOI.tif

and changes the link accordingly.

Meanwhile, the style of this population layer is stored separately from the data link and remains unchanged. Therefore, the new data file will inherit the style (symbology, layout text, graphic elements, etc.) of the previous map template.

Fig. 1.1 Mechanism of the Match Map Style Toolbox: Before

Fig. 1.2 Mechanism of the Match Map Style Toolbox: After

After switching the data link from the source city to the new city, the Toolbox updates the scale, extent, and legend labels based on the data values of the new city. The Toolbox supports updating symbology for feature classes with a single symbol, graduated colors, unique values, and rasters with classified, stretched, or unique value colormaps. If the symbology uses class breaks, the class breaks will be recalculated using equal intervals based on the number of intervals in the map template. Other elements that are not linked to the new data, such as static text elements, will remain unchanged.

The table below details how the Toolbox updates legend class breaks and labels for classified symbologies. Data with a maximum value larger than or equal to 100 will be rounded up to 10s and displayed as whole numbers; data with a maximum value larger than or equal to 25 will be rounded up to 5s and displayed as whole numbers; and data with a maximum value larger than or equal to 5 will be rounded up to 0.5s and displayed with one decimal place. Otherwise, data will be displayed with two decimal places and no rounding. These rules apply to classified symbology only, i.e. not to stretched colormaps or maps with unique values.

|  |  |  |
| --- | --- | --- |
| Maximum Value in Data | Class Break Round-up | Number of Decimal Places |
| ≥ 100 | 10 | 0 |
| ≥ 25 | 5 | 0 |
| ≥ 5 | 0.5 | 1 |
| < 5 | none | 2 |

1. **Setup**

The Toolbox requires the user to input links to three folders:

* A folder of complete map templates documents (.mxd format) with defined symbology and styles
* A folder of raw data from the “preexisting” source city AOI used in the map documents
* A folder of raw data from another, new city that will be used to generate new maps for it based on the predetermined styles in the map templates

This Section includes rules and guidance on folder management. In general, the three folders need to be independent of each other: one can not be a subfolder of the other. The Toolbox does not require that the folders be stored in any specific location on a device.

* 1. **Map Templates**

Map templates should have their data layers properly linked to the source city data folder, as well as their respective styles, symbology, and layouts ready.

* 1. **Source and New City Raw Data**

There is no requirement on where within the source and new city data folders the individual files are placed. However, there are three rules on file naming:

1. **If the file name does not contain the name of the city, the file names should be the same across the source city and new city, e.g.:**

C:\....\tillaberi\_niger\...\OSM\_Roads.shp

C:\....\zinder\_niger\...\OSM\_Roads.shp

1. **If the file name contains the name of the city, everything other than the city name should be the same, e.g.:**

Source city:

C:\....\tillaberi\_niger\...\01\_population\_tillaberi\_AOI.tif

New city:

C:\....\zinder\...\01\_population\_zinder\_AOI.tif ✔

C:\....\zinder\...\01\_population\_zinder\_AOI\_4326.tif **✘**

C:\....\zinder\... \01\_population\_AOI\_zinder.tif **✘**

**Note that the city names are case sensitive and must match between the source city dataset and new city dataset. e.g.:**

Source city:

01\_population\_Tillaberi\_AOI.tif

New city:

01\_population\_Zinder\_AOI.tif ✔

01\_population\_zinder\_AOI.tif **✘**

Source city:

01\_population\_TILLABERI\_AOI.tif

New city:

01\_population\_ZINDER\_AOI.tif ✔

01\_population\_Zinder\_AOI.tif **✘**

Source city:

01\_population\_tillaberi\_AOI.tif

New city:

01\_population\_zinder\_AOI.tif ✔

01\_population\_Zinder\_AOI.tif **✘**

**c. For maps that require pre-processing, certain keywords must be used in the file names of their datasets in order to be processed properly.** The table below lists these keywords:

|  |  |  |
| --- | --- | --- |
| **Map theme** | **Keyword in Raw Data File Name (Case sensitive)** | **Related Processing** |
| Administrative boundary | “admin” | Apply the regional scale |
| AOI | “AOI” | Used to set scale and extent for all new maps |
| Raster for fluvial flood 1 in every x years | “FU\_1inx.tif”  (for x in 5, 10, 20, …, 1000) | Identify and locate flood data |
| Raster for pluvial flood 1 in every x years | “P\_1inx.tif”  (for x in 5, 10, 20, …, 1000) | Identify and locate flood data |
| Population raster | “population” | Tabulate area subject to each type of flood |
| WSF reclassed land-use class raster | “WSF\_reclass” | Tabulate area subject to each type of flood |
| Openstreetmap road centrality edges | “edge” | Tabulate percentage of major road subject to each type of flood |
| Openstreetmap fire stations shapefile | “fire” | Tabulate percentage of infrastructure subject to each type of flood |
| Openstreetmap health facilities shapefile | “health”  (and no “isochrone” in order to differentiate from the accessibility data) | Tabulate percentage of infrastructure subject to each type of flood |
| Openstreetmap police stations shapefile | “police” | Tabulate percentage of infrastructure subject to each type of flood |
| Openstreetmap schools shapefile | “schools”  (and no “isochrone” in order to differentiate from the accessibility data) | Tabulate percentage of infrastructure subject to each type of flood |
| Elevation raster | “elevation” | Generate derived slope raster |
| Monthly SOL | “SOL” | Get min and max across serial rasters for uniform legend labels |
| Air pollution (PM 2.5) raster | “air” | Apply global legend class breaks |
| Surface imperviousness | Do not include city name and use exactly the same across two cities, e.g. “PIS1985-2015\_30m.tif” or “VIIRS\_2019.tif” |  |
| Accessibility isochrones | “isochrone” | To differentiate from AOI, school and hospital shapefiles |

1. **Interface & Code**

**Interface in ArcMap 10.x**

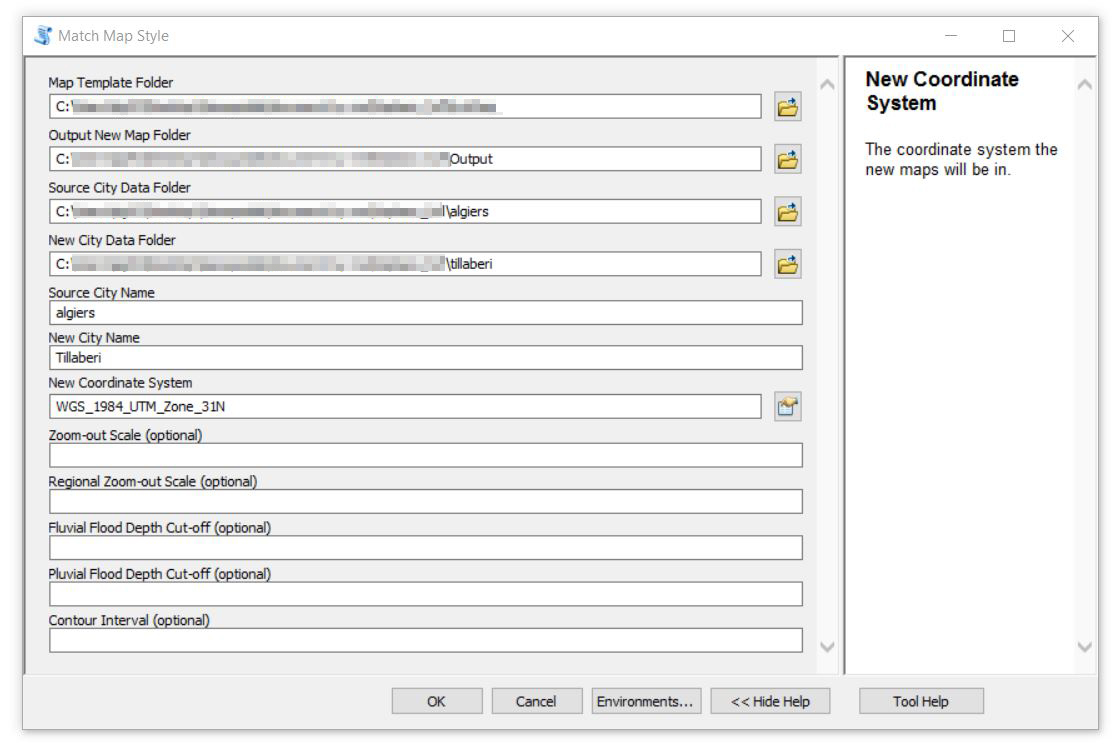
****

Fig 3.1. interface of the toolbox in ArcMap 10.8

**Required Inputs to the Toolbox**

***Map Template Folder****:* The folder of complete map documents with predetermined styles. All data used in these map documents should be stored in the *Source City Data Folder.*

***Source City Data Folder****:* The folder of raw data of the source city, used in the map templates in *Map Template Folder.*

***New City Data Folder****:* The folder of raw data of the new city. The names of the files inside the folder should match with those in the *Source City Data Folder*.

***Output New Map Folder****:* The folder where new files will be outputed to.

***Source City Name***: The name of the source city*,* case insensitive, i.e., for the city of Tillaberi, Tillaberi, TILLABERI, tillaberi, or TillAbeRI are all acceptable.

***New City Name****:* The name of the new city*,* case insensitive, i.e., for the city of Tillaberi, Tillaberi, TILLABERI, tillaberi, or TillAbeRI are all acceptable.

***New Coordinate System****:* The correct coordinate system for the new city. It is recommend to use projected coordinate systems. If the user chooses a coordinate system in degrees, the output area tables will be in degrees.

**Optional Inputs to the Toolbox**

***Zoom-Out Scale****:* It is visually preferable to create maps are are slightly zoomed out from the exact extent of an AOI. Therefore, the Toolbox automatically moves the current view of the maps to a default of 20% zoom-out of the AOI scale (i.e. the map scale is 1:1.2\*the scale of the AOI’s exact extent). The optional input here is the greater or lesser zoom-out from the AOI extent.

***Regional Zoom-Out Scale****:* Similar to the above, the default regional zoom-out, applied solely to the administrative boundary map, is 40%, i.e. the map scale is 1:1.40\*the scale of the exact extent of AOI.

***Fluvial Flood Depth Cut-Off****:* The minimum threshold value for fluvial flood water depth in meters. The default is 0.15m.

***Pluvial Flood Depth Cut-off****:* The minimum threshold value for pluvial flood water depth in meters. The default is 0.15m.

***Contour Interval***: This option defines the interval, in meters, for the countours derived from the elevation raster. If no input is entered, the Toolbox will generate an interval based on the difference between the max and min values of the raster.

1. **Pre-Execution Checklist**

* Pandas and Numpy packages for Python 2.7 installed (included in the Python 2.7 interpreter within Arcmap).
* License for the Spatial Analyst extension.
* Individual data files named according to the rules in Section 2.
* A folder of complete map templates (.mxd documents).
* A folder to store data for the source city used in map templates.
* A folder to store data for the new city.
* All customized symbology (colormaps, icons etc.) stored locally or imported from a style file. The lack of such a file will not affect the map generation itself; yet any customization will be lost if manual adjustments are made after new maps are generated.

1. **Running the Toolbox & Generating Outputs**

For a first-time user, first import the Toolbox as below in order to run it. Open the ArcToolbox pane by clicking on the highlighted icon in the default toolbar.

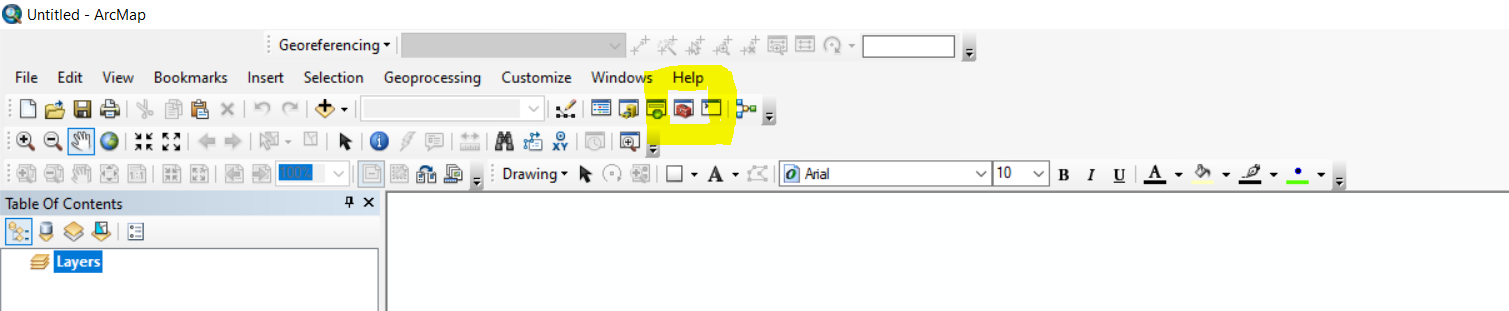


Fig.5.1 Step 1 open ArcToolbox

In the newly opened pane, right-click on ArcToolbox and select “Add Toolbox…”

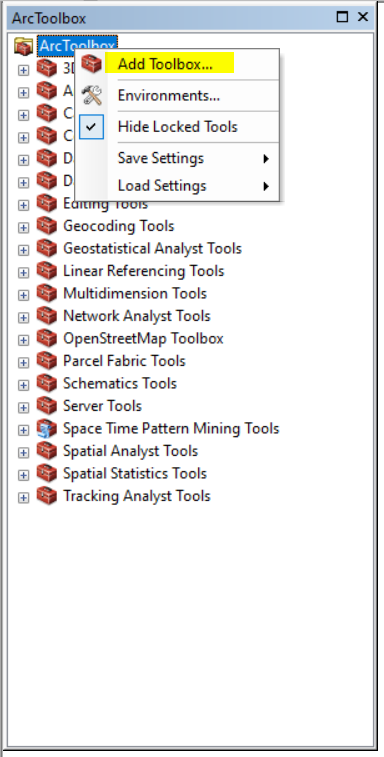


Fig.5.2 Step 2 Add new toolbox

Navigate to the location of the toolbox and select the corresponding .tbx file, and wait for the new toolbox to be loaded into the pane.

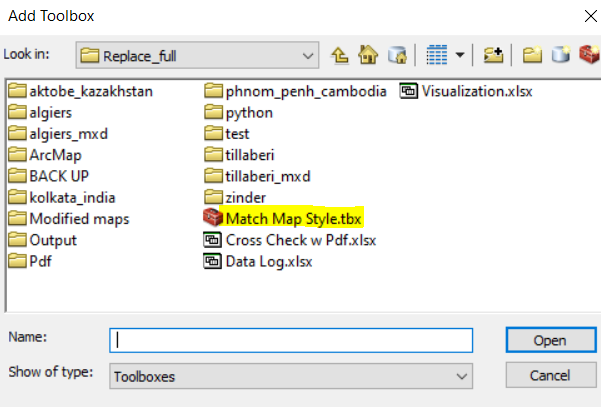


Fig.5.3 Step 3 Add new toolbox

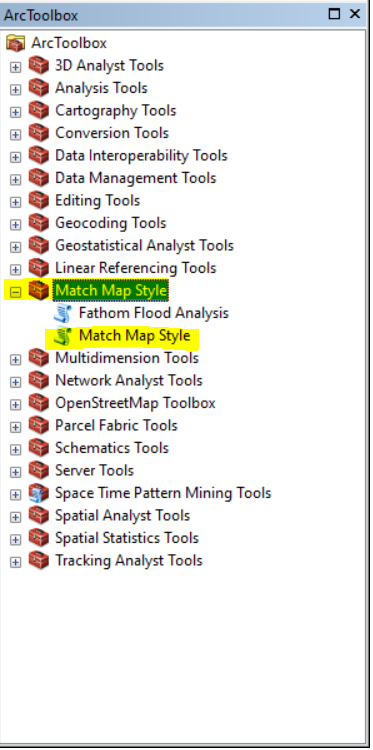


Fig.5.4 Step 4 Import complete

Now double-click the Match Map Style tool and call out the interface window. Details on the interface can be found in Section 3.

Copy the Toolbox and paste it into “My Toolboxes’ in the Catalog so that in the future it can be directly accessed from there.

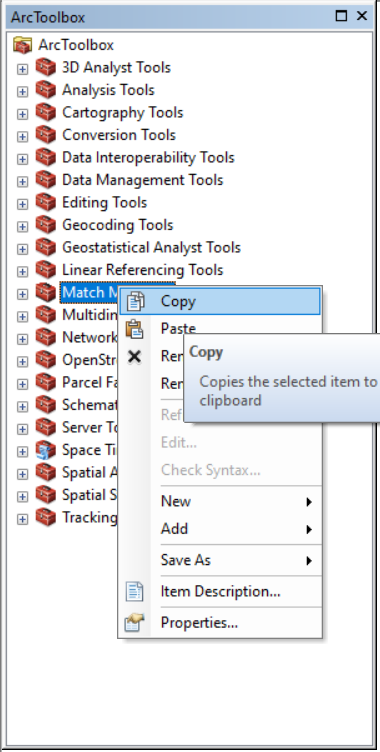
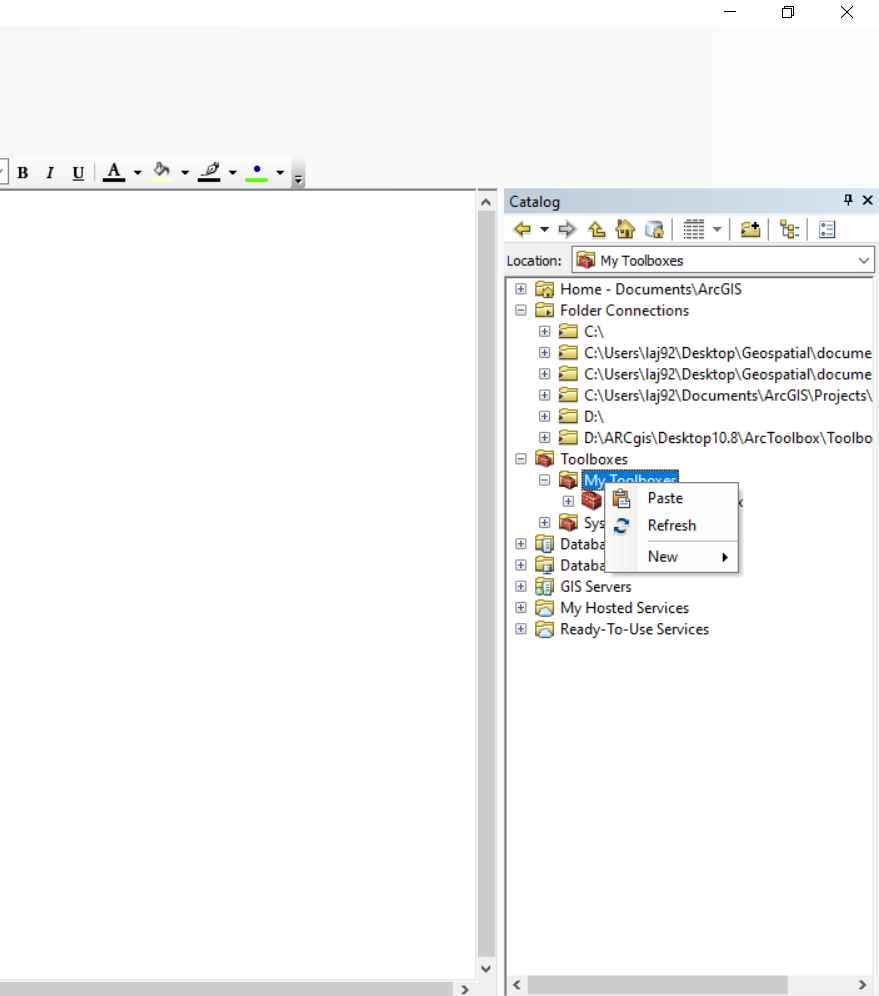
 

Fig.5.5 Step 5 Paste the toolbox for future use

The Toolbox will generate the following files in the designated output folder:

* Updated map documents (.mxd)
* Execution Log & Replacement Dictionary (.txt)
* Excel tables for flood area analysis (.exsl)
* PNGs (300dpi) of updated maps

Warnings will appear in the dialogue box in ArcMap after executing the Toolbox. More details on the execution results can be found in the *Execution\_Log.txt*. See Section 6 for dealing with warnings.

1. **Errors, Warnings & Cautions**
2. ***“WARNING: Didn't find file ### in the source city data folder"***

This error occurs when the Toolbox cannot find the data linked to the layer in the dictionary obtained by walking through the source city folder. The data linked to the layer might not be in the source city folder, i.e. there is a broken link in the map templates. If not, migrate the data to the source city folder and relink it to the layer.

1. ***“WARNING: Fail to replace file link for map %s, layer %s might be unmatched/invalid, please check the database"***

This error is most likely due to the corresponding data missing in the new city folder.

1. ***“WARNING: fail to access values of the raster for map.”***

This warning occurs when the Toolbox fails to rewrite the symbology break classes for classified rasters, or the Toolbox cannot filter out None values to obtain the min and max values for recalculating the class breaks. If this error occurs, it is necessary to manually update legend class breaks.

1. ***“WARNING: failed to update legend labels for this classified raster.”***

This warning occurs when the toolbox tries to re-write the symbology labels for classified rasters. If this error occurs, it is necessary to manually update map legend labels.

1. ***“WARNING: OSM SHP for ### is missing, calculate flood area skipped"***

This warning occurs when calculating the flood overlap area tables and the Toolbox fails to find OSM shapefiles named with the proper keywords. If data is available for the facilities in question, check the raw data folder to see if its naming contains the keyword. Note that the keywords are case-sensitive. For details, see Section 2.2c.

1. ***“WARNING: One of the fathom raw raster for 1 in 5, 10, 20, ...1000 might be missing. Flood classification fails.”***

The condition statement in the flood classify function needs raw rasters for fluvial and pluvial flooding from 1 in 5 years all the way to 1 in 1000. This warning occurs when one of the fo;es is missing. If this warning occurs, the toolbox will skip all the flood-related analysis, and the updated flood maps will be empty.

1. ***"WARNING: Monthly SOL might be missing. Calculate min and max for serial maps skipped"***

This warning appears when the toolbox tries to obtain consistent max and min values for serial maps (e.g. Monthly SOL) but fails to find this data in the new city raw data folder.

1. **Caution of overwriting the output**

The overwrite right is on when running the Toolbox. This will affect map documents and tables in the output folder and derived flood data in the new city data folder if files with duplicated names exist in the destination folder when outputting results. Make sure to back up important files.

1. **Some classes symbolized by unique value are missing**

For maps with unique values, only values appearing in the original template will show up in new maps. For example, if there are only areas with low and medium landslide risk in the source city's data, only low and medium values will appear in the new city’s map even if there are areas with medium and high risk in the new city. Therefore, when making templates, it is recommended to use a city whose data includes all potential values. The user can also manually add back missing classes for the new city by right-clicking on the layer “Properties-Symbology-Add Value.”

1. **The legend labels of classes are in reverse order to what is shown in the map**

The Toolbox requires classes to be sorted in increasing order. In the original map template, right click on layer to see in the “Properties-Symbology” tab if the classes are listed in increasing order (with the minimum value in the top row). If not, right-click and choose “reverse sorting.”

1. **Post-Execution Adjustments**

Some maps might require additional adjustments that the Toolbox does not automatically implement. These are summarized below:

|  |  |  |  |
| --- | --- | --- | --- |
| **MXD Map Document** | **Location in City Scan** | **Issue** | **Adjustment Needed** |
| criticality | Road network criticality | Symbology class break uses natural breaks. | 1. Choose natural breaks Jenks classification in the *Symbology* tab  2. Copy the edge layer, define query for major roads, limiting the highway field to “primary,” “trunk,” and “motorway”  3. Adjust colors for smallest classes |
| Impervious | Surface imperviousness (built-up density) | There can be two types of data used for this map (2019 RGB composite or 1985-2015). The user needs to prepare map templates for both. | 1. Include map templates for both data types (the templates can be from two cities as long as the file names do not contain cities)  2. After running the toolbox, delete the one with missing data |
| LC | Land cover | Classes not appearing in the template  won’t be shown in the new city. | Right click on land cover layer–Properties–Symbology–Add Values to add missing land cover types |

Note that all customized symbology like colormaps must be stored locally or imported from a style file. Otherwise, when manual adjustments are made to the new maps, ArcMap will reset the colormaps to one of its default colormaps.

1. **Adding New Templates & Updating Packages** 
   1. **Code Structure**

The code for the Toolbox can be divided into four parts: set up, helper functions, pre-processing, and for-loop. The first three parts are linear, and the main loop is a loop applied to all user input map templates.

***set up***: This part of the code imports relevant packages, checks licenses for extensions, gets the inputs from the user, and creates log files.

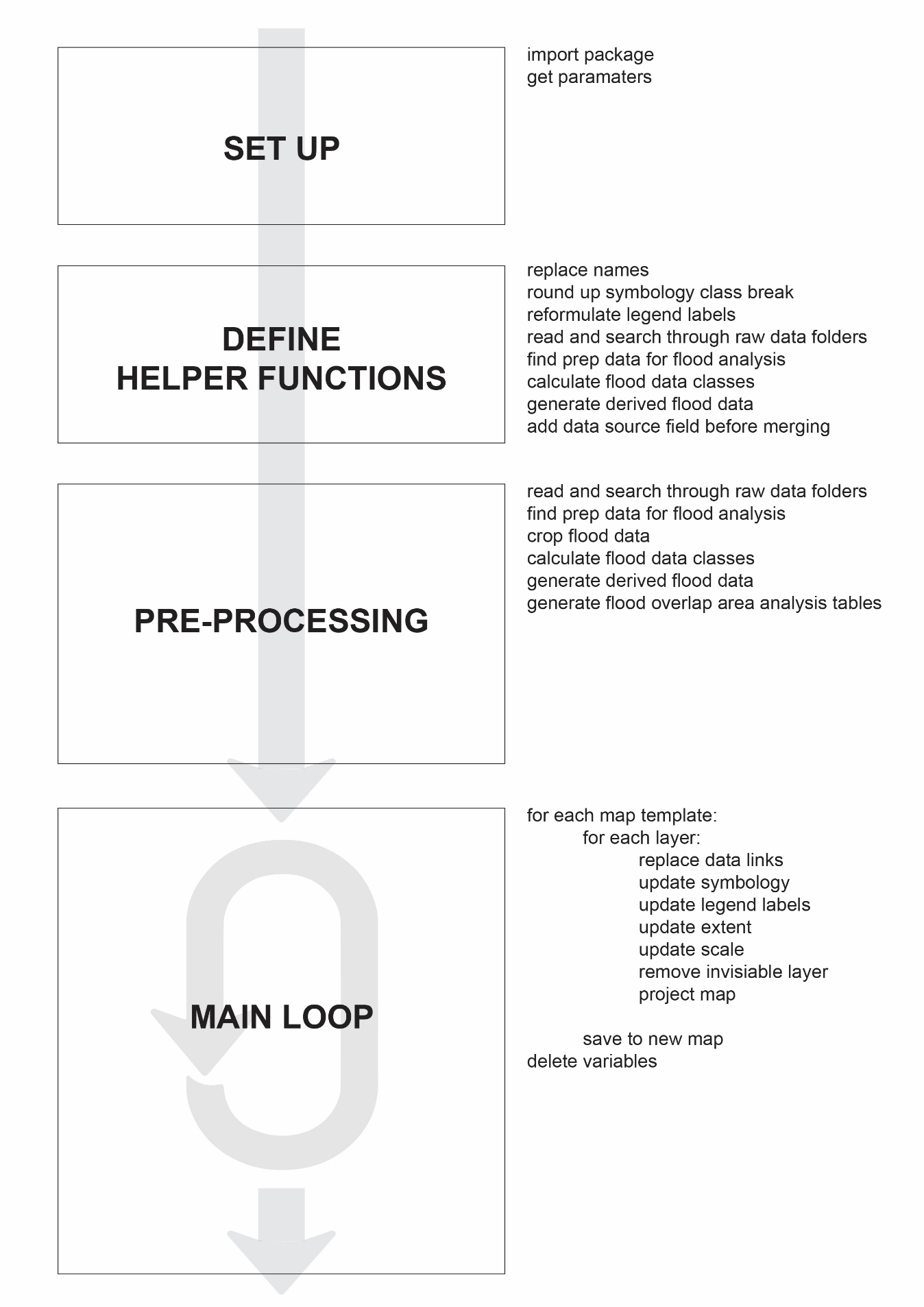
***helper functions***: This part of the code hosts helper functions used in later parts. Helper functions include both universal and map-specific functions. A list is shown in Fig.8.1.

***pre-processing***: Some maps require special processing, including additional data derived from the raw data, special rules for generating legend breaks, calculating overlapping areas with another dataset, etc. This section is used to execute this pre-processing before the main loop. The code uses the find\_prep\_data(new\_city\_path) function to find file paths for data that need pre-processing. Note that the general function used to obtain a full dictionary of both raw and derived data for the new city is also located in this section (the line: dic, dic\_newpath = record\_file\_dictionary(source\_city\_path, new\_city\_path, source\_city, new\_city)*).* This line should be called before the flood area overlap analysis.

***main-loop***: This part contains a for-loop that passes through each of the map documents (.mxd) and applies the universal functions: replace the data link of the source city to the one for the new city, and update legends, extent, and scale for each map.

The code is designed to put as much separate/customized processing as possible into the *pre-processing* part, while in the *main loop* there are mostly universal functions and some minor branches for special cases.

Fuctions included:



import package

get user input paramaters

replace names

round up symbology class break

reformulate legend labels

read and search through raw data folders

find prep data for flood analysis

calculate flood data classes

generate derived flood data

add data source field before merging

get min and max value for SOL serial maps

generate slope and contours

get file paths for pre-processing data

find prep data for flood analysis

crop flood data

calculate flood data classes

generate derived flood data

get a full dictionary of raw and derived data for the new city

generate flood overlap area analysis tables

get min and max value for SOL serial maps

generate slope and contours

for each map template:

for each layer:

replace data links

update symbology

update legend labels

update extent

update scale

remove invisiable layer

project map

save to new map

delete variables

(general functions)

(functions for selected maps)

Fig 8.1. Structure of Code

For developers, the code is divided by comments into the same structure as in Fig 8.1. See Fig.8.2.

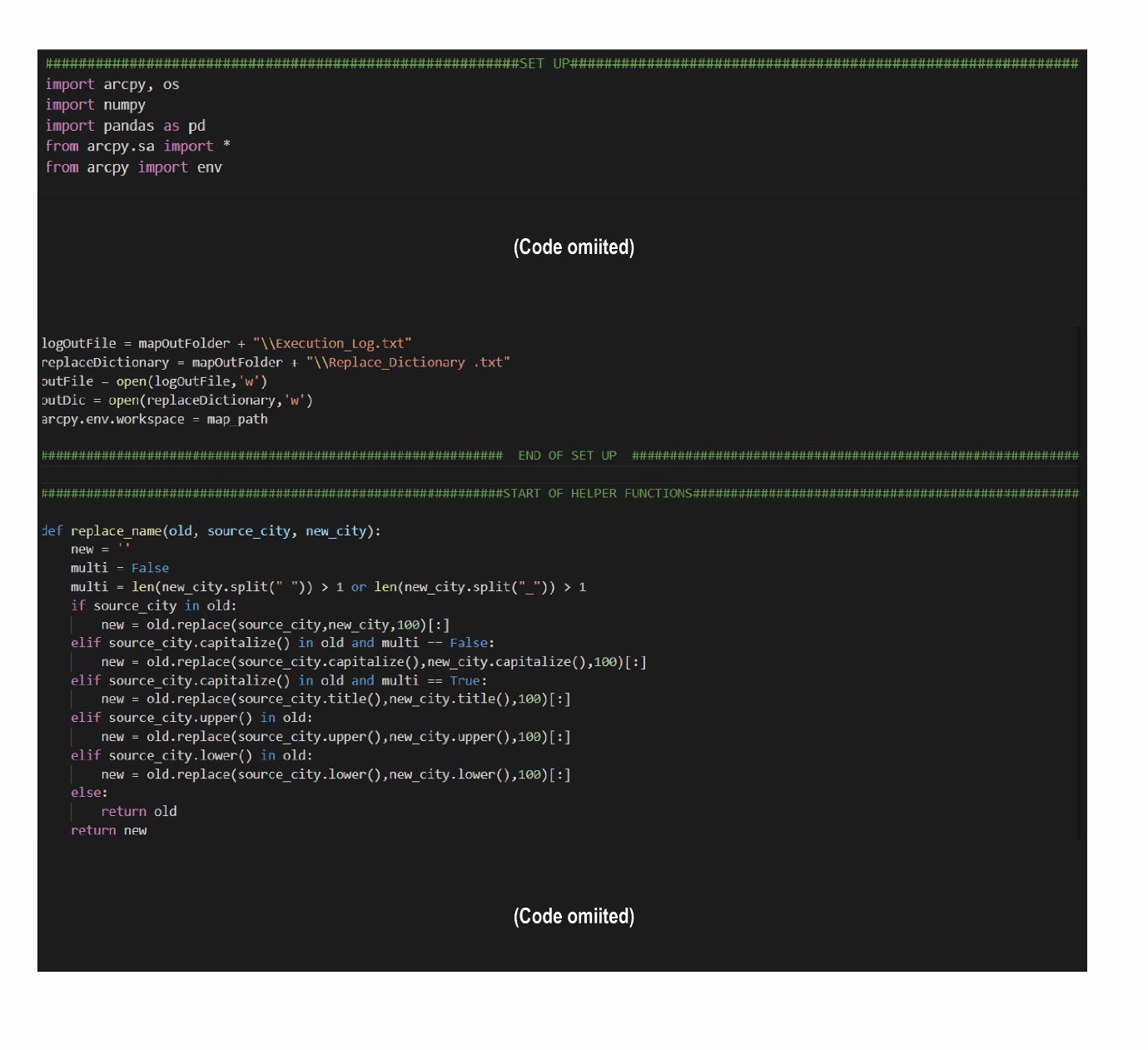


Fig 8.2. Example of Structure Hint inside the Code

* 1. **Migrating to Other Platforms**

**From ArcMap 10.x to ArcGIS 2.x**

The Toolbox is developed with the Arcpy package under ArcMap 10.8 and Python 2.7, but it can be adapted to ArcGIS Pro 2.x and Python 3.7. Following are the major changes that would be needed for such a migration:

* **Change Python syntax**: standard syntax change from Python 2 to 3, mostly the print statement
* **Change names of some ArcPy functions**: ESRI keeps most ArcPy functions under ArcMap 10.x in the new ArcPy under ArcGIS Pro 2.7. But there might be some naming changes, e.g. arcpy.CalculateStatistics\_management is now arcpy.management.CalculateStatistics. Most Python IDEs offers debugging checks to help locate undefined functions.
* **Change file management from mxd to arxp**: many functions in arcpy.mp in ArcGIS Pro needs to reference a project file (.aprx) rather than a map document (.mxd). For details, see the official guidance: [*https://pro.arcgis.com/en/pro-app/latest/arcpy/mapping/migratingfrom10xarcpymapping.htm*](https://pro.arcgis.com/en/pro-app/latest/arcpy/mapping/migratingfrom10xarcpymapping.htm)*.*

As a result, the developer will need to put all map templates into an .arxp project.

In ArcMap, one can export directly from a Map object, so the Toolbox updates the extent, scale and legends only for the Map object. But in ArcGIS, Map objects and Layout objects are separate. Therefore, the developer will need additional lines of code to update Map extent and scale in Layout objects as well.

* **Modify the main loop with the new replace data link function**: While lyr.replaceDataSource is used for ArcMap 10.8 to replace the data links, the corresponding new function in ArcGIS Pro 2.x is updateConnectionProperties. It is recommended to use the dictionary generated by the toolbox together with lyr.updateConnectionProperties for ArcGIS Pro 2.x. For details, see “Using the connectionProperties dictionary” Section at [*https://pro.arcgis.com/en/pro-app/latest/arcpy/mapping/updatingandfixingdatasources.htm*](https://pro.arcgis.com/en/pro-app/latest/arcpy/mapping/updatingandfixingdatasources.htm).

**From ArcMap 10.x to QGIS**

QGIS’s plugins and packages are different from those of the ArcPy package developed by ESRI. To migrate the toolbox to QGIS, the developer will need to replace the functions from the ArcPy package with the equivalent in QGIS. Below are the major changes needed:

* **Change Python syntax**: standard syntax change from Python 2 to 3, mostly the print statement
* **Correspond the Main Loop & Replace Data Link**

It is possible to set the data source for a layer while keeping related settings and options (graphic styles, labels etc.) in QGIS. In ArcMap, this is completed by the replaceDataSource function. In QGIS, the closest functions that performs a similar task are QgsVectorLayer.writeLayerXML and QgsVectorLayer.readLayerXML, modifying the DOM document on the fly and reloading the layer. See this post for details and example code: *https://geogear.wordpress.com/2015/05/15/changing-qgis-layer-datasource-with-python/*.

* **Correspond Helper Functions**

The Toolbox’s helper functions use NumPy, pandas and common ArcPy GIS operations. NumPy and pandas can be imported into QGIS. But the developer will need to find equivalent functions in QGIS for everything else.

**8.3 Developing New Templates**

New maps may be created for the City Scan and added to the Toolbox for processing. To modify the Python script accordingly, the developer will first need to configure the Python IDE by changing the interpreter to the Python interpreter that came with ArcMap 10.x to import the ArcPy package. By default, this is located at

C:\Python27\ArcGIS10.8\python.exe

If the map does not require any pre-processing of its raw data, the developer only needs to make a new template with the determined symbology and layout and add that template into the map template folder.

If the map requires some pre-processing of the raw data, the following workflow is recommended:

* set up a map template with processed data from the source city and save it to the map template folder
* create new functions needed for the derived data and execute them before the Main Loop section of the code (see Section 8.1 and Fig 8.1)

The find\_prep\_data(new\_city\_path) function can find the raw data the developer needs with the arcpy.da.walk module. After obtaining the file path of raw data, the developer can use keywords in the filename of the raw data to trigger the targeted pre-processing functions for the new map. Be sure to record any new keywords needed in an updated keywords table in Section 2.2.