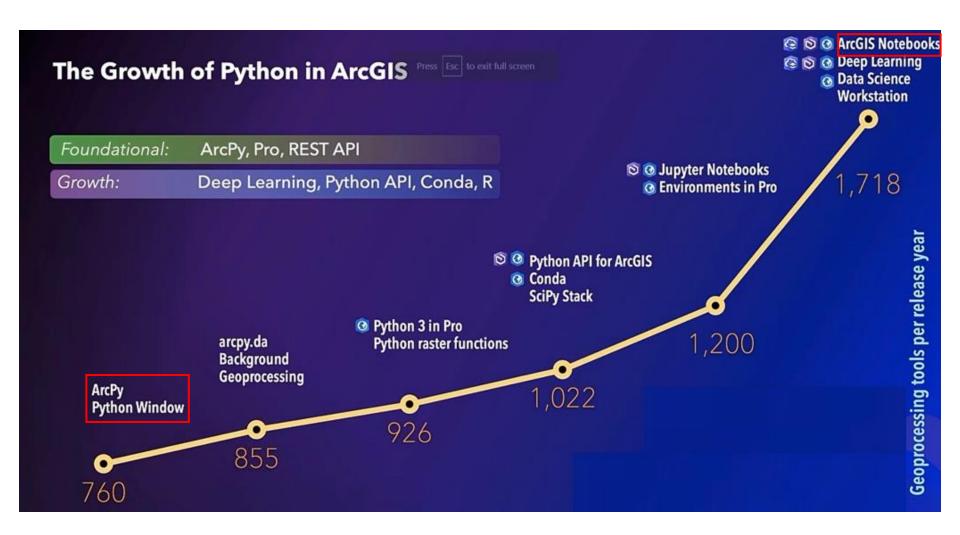
CRP 5680 Urban Spatial Data Analytics Lab 9 – Integrating ArcGIS Pro with Python notebook

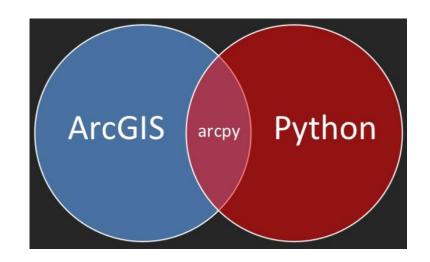
Why ArcGIS Pro + Python?

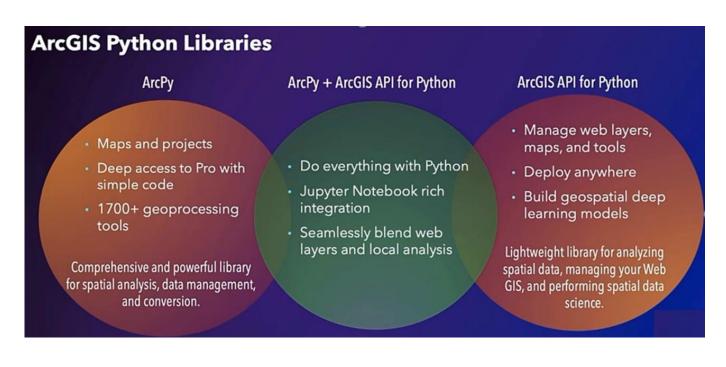


ArcGIS-based Python libraries (packages):

ArcPy (for ArcGIS desktop): a site package of Python that is created to perform geographic data analysis, data conversion, data management, and map automation with Python.

ArcGIS API for Python (for cloud-based/online services, the ESRI Enterprise platforms): A powerful Python library for mapping, spatial analysis, data science, geospatial AI and automation.



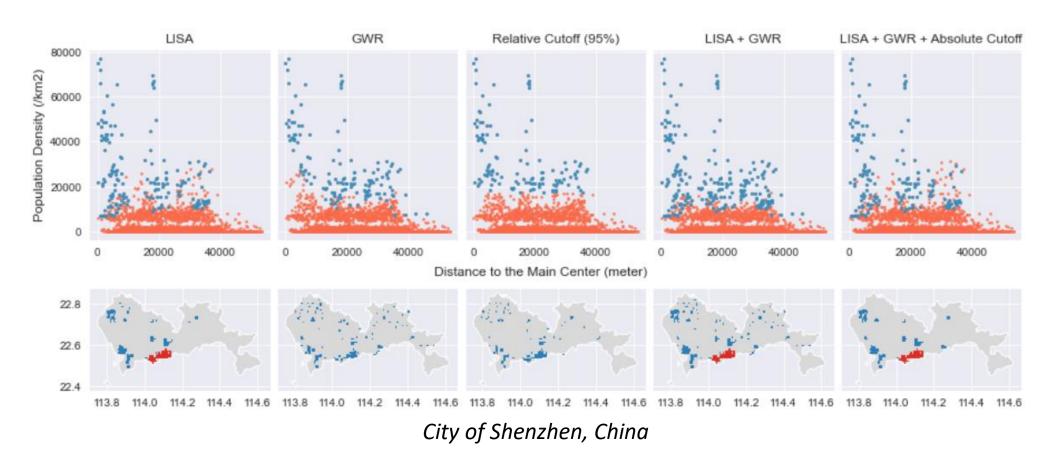


Using ArcPy combined with the Python ecosystem, we can:

- Automate repetitive tasks
 - Through loop and for statement
- Invoke ArcGIS Pro tools:
 - geoprocessing tools
 - raster functions
 - Python Toolbox for Script Customization
- incorporate multiple data sources in your analysis
 - .shp, .csv, Excel, .tif, Stata...
- Conduct aspatial data collection, data management, data visualization, and spatial analysis in a single platform
 - Pandas, numpy, matplotlib, etc...

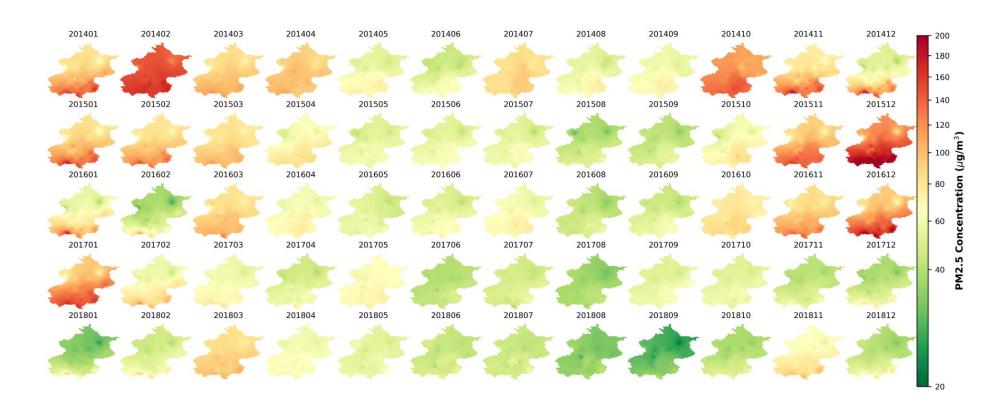
Applications of Combining ArcGIS Pro and Python

- Using ArcGIS Pro and Arcpy to identify city centers using ESDA methods
- Packages used: Arcpy, matplotlib, seaborn, geopandas



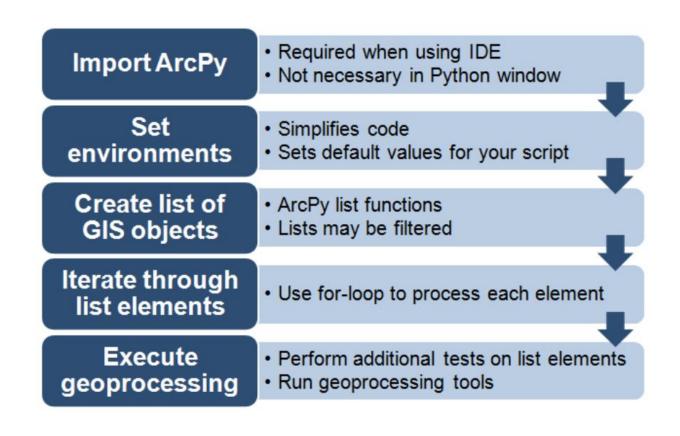
Applications of Combining ArcGIS Pro and Python

- Automate the spatial interpolation tools to generate the monthly trend of PM2.5 in Beijing
- Packages: Arcpy, rasterio, matplotlib



Common scripting workflow

Each Python script that you create will differ depending on the script's purpose. However, many scripts that you write, will use the following workflow.



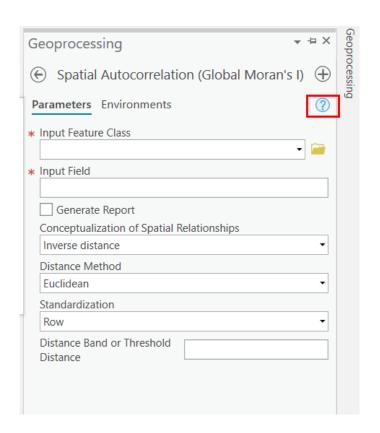
Import the package and set the environment

```
In [6]: import arcpy

# Set geoprocessing environments
workspace = r"D:\TA_CRP5680_UrbanDataSci\Lab9\Lab9_data_demo"
arcpy.env.workspace = workspace
arcpy.env.overwriteOutput = True
```

- The overwriteOutput parameter controls whether tools will automatically overwrite any existing output when your script is run.
 - True: tools will execute and overwrite the output dataset.
 - False: existing outputs will not be overwritten, and the tool will return an error.

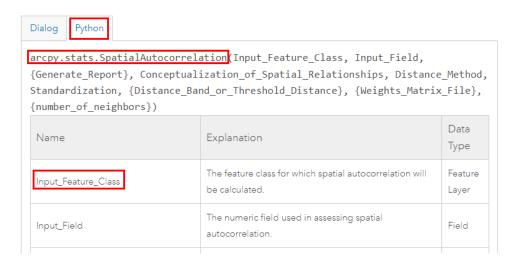
Example of geoprocessing: Moran's I Index





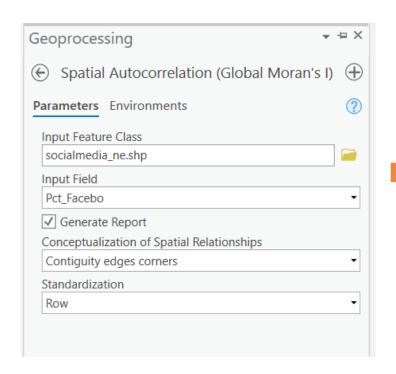
Tool reference

Parameters





Global Moran's I: from user interface to python



Example of OLS regressions: Arcpy + for-loop

Step 1: read the shapefile as spatially enabled dataframe (SeDF)

```
# read shapefile using .spatial.from_featureclass()
gdf = pd.DataFrame.spatial.from_featureclass(workspace + "/housing/Beijing_housing2016.shp")
gdf.head(3)
```

Step 2: data management using Pandas: categorize the DF into two parts.

```
#### Categorize the data into two parts: the urban core and the suburban area.
urban_core_lst = ["Dongcheng", "Xicheng", "Chaoyang", "Haidian", "Fengtai", "Shijingshan"]

gdf_core = gdf.loc[(gdf["disname_en"].isin(urban_core_lst)), :]
gdf_core["hid"] = gdf.loc[: , "FID"]

gdf_sub = gdf.loc[~(gdf["disname_en"].isin(urban_core_lst)), :]
gdf_sub["hid"] = gdf.loc[: , "FID"]
```

Example of OLS regressions: Arcpy + for-loop

Step 3: write (export) the SeDF to shapefile

```
gdf_core.spatial.to_featureclass(workspace + "/Lab9_data_output/housing_core.shp")
gdf_sub.spatial.to_featureclass(workspace + "/Lab9_data_output/housing_sub.shp")
```

Step 4: OLS regressions using the function – OrdinaryLeastSquare_stats:

A Brief Introduction of ArcGIS API Python

- For cloud-based/online services, the ESRI Enterprise platforms, and ArcGIS Online
- Provide APIs for common geospatial functions: distance matrix, network analysis, and geocoding services (but need credits!)
- Provide convenient visualization tools and rich data sources. Here is an example of where to eat in Ithaca.

