

PYTHON IN EXCEL

5 Excel features that make Python in Excel click.

What to know before you write a single PY() formula.

George Mount · stringfestanalytics.com

THE REAL PREREQUISITE

Python in Excel assumes you know Excel.

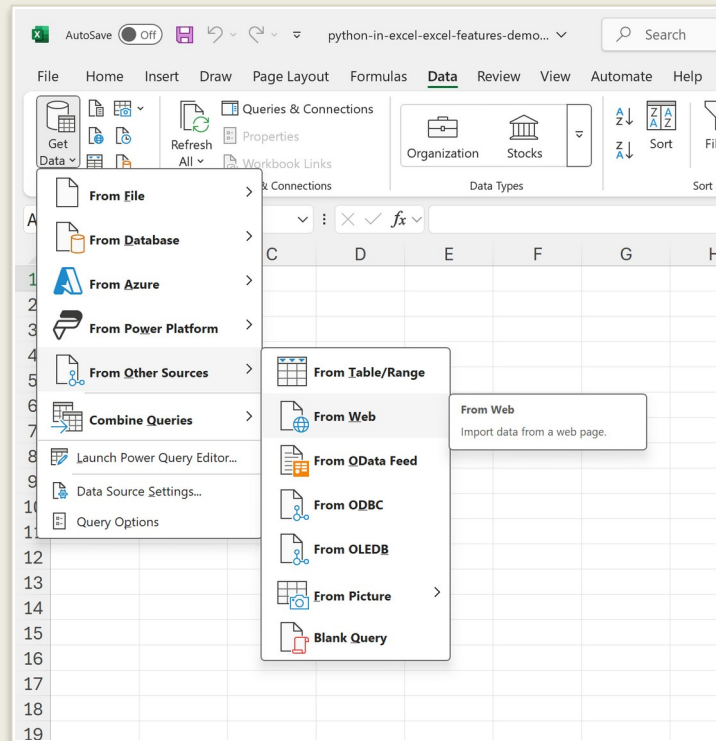
When folks ask me about getting started with Python in Excel, I typically recommend a baseline of Python knowledge first. But the integration also assumes a solid understanding of Excel itself, since it leverages an impressive set of existing features.

Here are five worth getting comfortable with.

1

POWER QUERY

Python in Excel operates in a closed system.

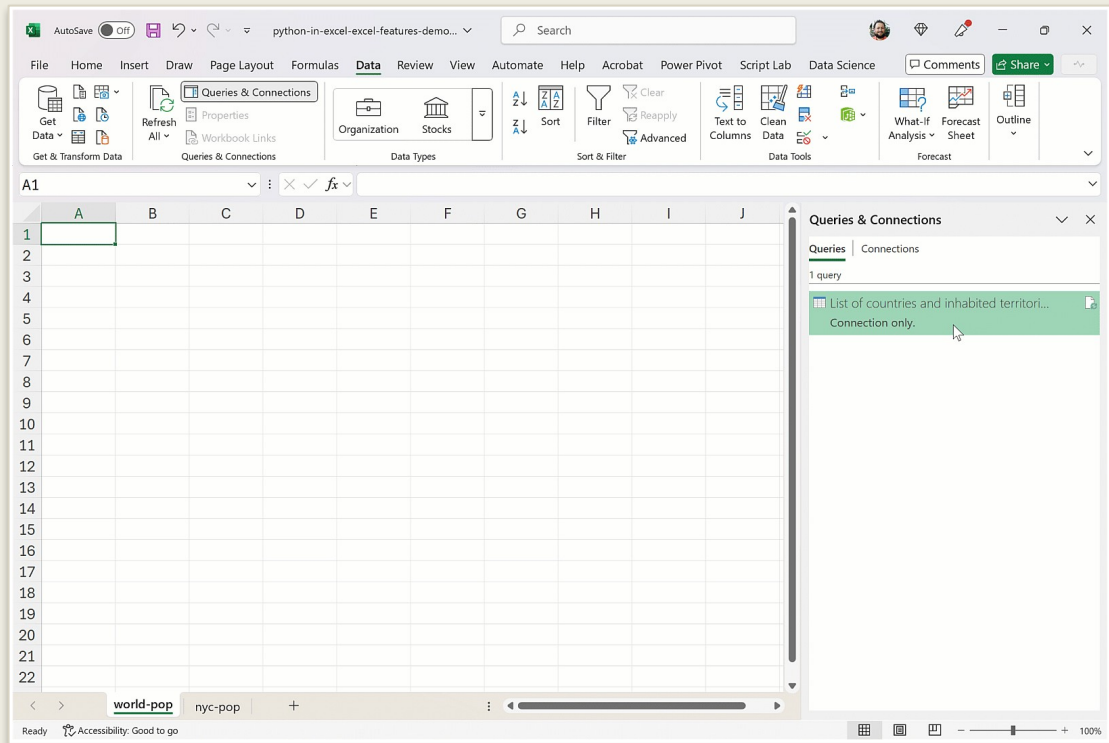


Which means you can't use Python to pull in external data through flat files, APIs, web scraping, and so forth. That task still falls to Power Query, which is highly capable in its own right. Start with Data > Get Data > From Web.

1

POWER QUERY, CONTINUED

Then pull the data into Python with xl().

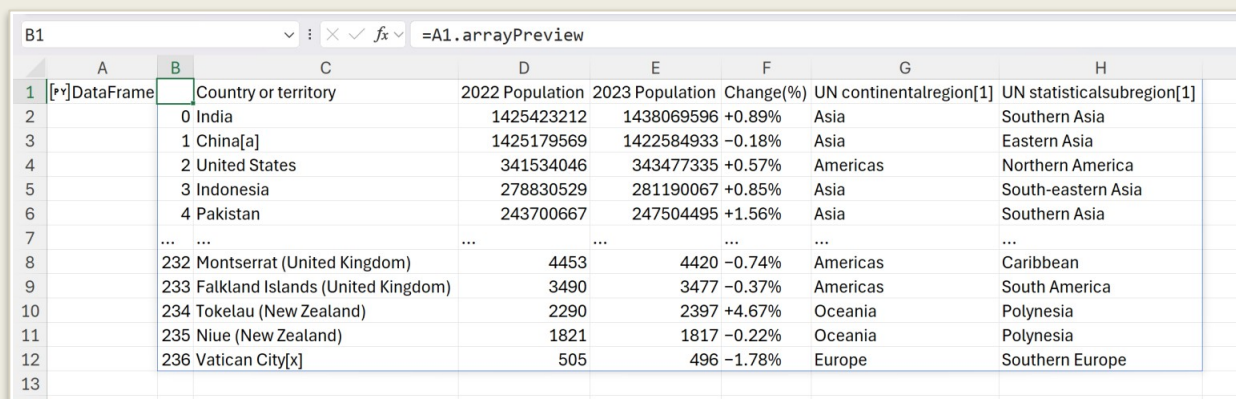


Once your query is loaded as Connection Only, head to the Queries & Connections pane on the right, right-click your query, and rename it to something code-friendly like `world_pop`. From there, you can reference it from Python with `xl("world_pop")`.

2

LINKED DATA TYPES

Inspect Python objects without flooding the grid.



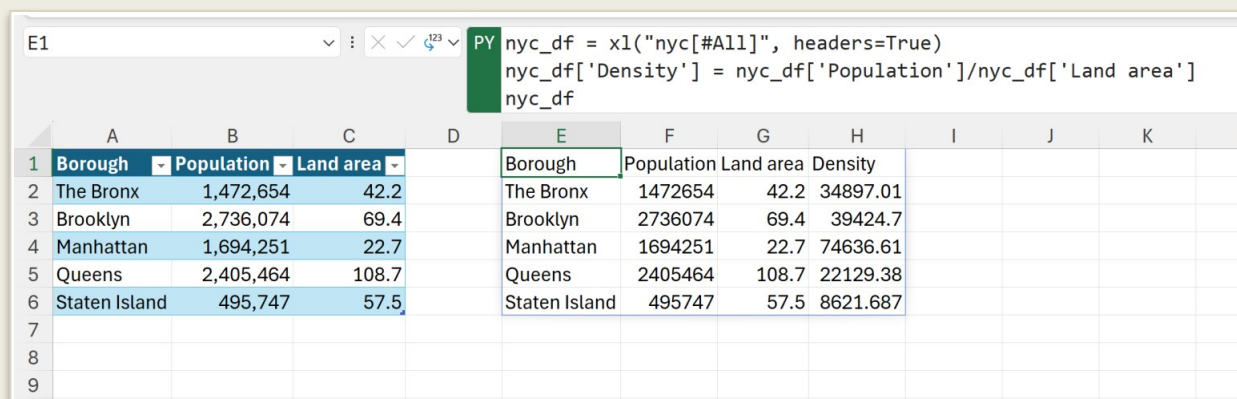
	A	B	C	D	E	F	G	H
1	[*] DataFrame		Country or territory	2022 Population	2023 Population	Change(%)	UN continentalregion[1]	UN statisticalsubregion[1]
2		0	India	1425423212	1438069596	+0.89%	Asia	Southern Asia
3		1	China[a]	1425179569	1422584933	-0.18%	Asia	Eastern Asia
4		2	United States	341534046	343477335	+0.57%	Americas	Northern America
5		3	Indonesia	278830529	281190067	+0.85%	Asia	South-eastern Asia
6		4	Pakistan	243700667	247504495	+1.56%	Asia	Southern Asia
7	
8		232	Montserrat (United Kingdom)	4453	4420	-0.74%	Americas	Caribbean
9		233	Falkland Islands (United Kingdom)	3490	3477	-0.37%	Americas	South America
10		234	Tokelau (New Zealand)	2290	2397	+4.67%	Oceania	Polynesia
11		235	Niue (New Zealand)	1821	1817	-0.22%	Oceania	Polynesia
12		236	Vatican City[x]	505	496	-1.78%	Europe	Southern Europe
13								

Staying in Python object mode keeps things streamlined because you don't need to display every value in the dataset. To get a look at what's inside, click the table-like icon next to the cell and select the arrayPreview field for a compact view of the data.

3

EXCEL TABLES

Tables give you a dependable dynamic data source.



The screenshot shows an Excel spreadsheet with a Python code snippet in cell E1 and a data table below it. The code snippet is:

```
nyc_df = xl("nyc[#All]", headers=True)
nyc_df['Density'] = nyc_df['Population']/nyc_df['Land area']
nyc_df
```

The data table has the following columns: Borough, Population, Land area, and Density. The rows represent the five boroughs of NYC: The Bronx, Brooklyn, Manhattan, Queens, and Staten Island.

Borough	Population	Land area	Density
The Bronx	1,472,654	42.2	34897.01
Brooklyn	2,736,074	69.4	39424.7
Manhattan	1,694,251	22.7	74636.61
Queens	2,405,464	108.7	22129.38
Staten Island	495,747	57.5	8621.687

Importing your data into Python in Excel as a table, rather than using a static cell reference, keeps it tied to the table name. If the data's shape, size, or position changes, the reference adapts automatically. Add a sixth borough to NYC and the code keeps working.

4

DYNAMIC ARRAYS & THE SPILL OPERATOR

Spill ranges open up dynamic calculations.

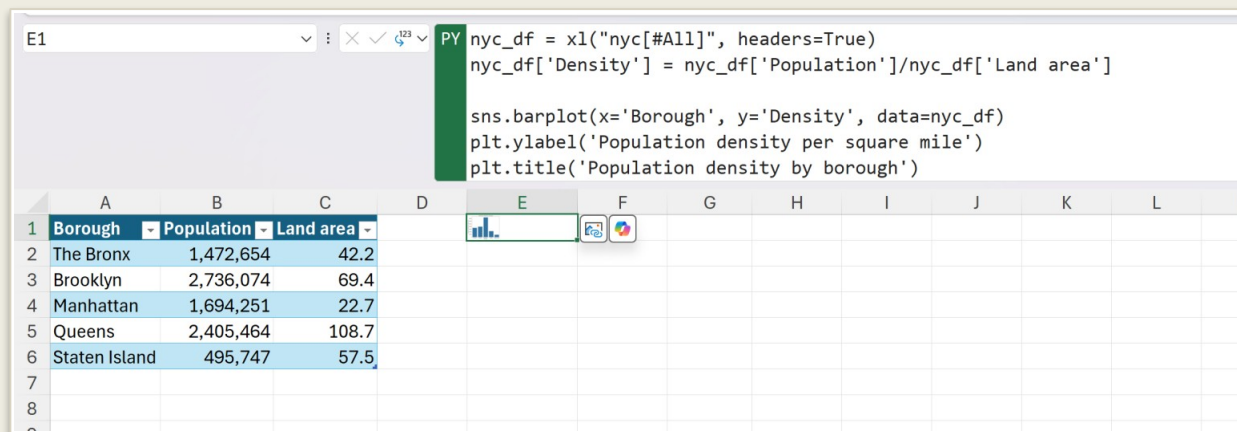
	A	B	C	D	E	F	G	H	I
1	Borough	Population	Land area		Borough	Population	Land area	Density	
2	The Bronx	1,472,654	42.2		The Bronx	1472654	42.2	34897.01	
3	Brooklyn	2,736,074	69.4		Brooklyn	2736074	69.4	39424.7	
4	Manhattan	1,694,251	22.7		Manhattan	1694251	22.7	74636.61	
5	Queens	2,405,464	108.7		Queens	2405464	108.7	22129.38	
6	Staten Island	495,747	57.5		Staten Island	495747	57.5	8621.687	
7									
8					Count rows	6 =ROWS(E1#)			
9					Sum of population (Column 2)	8,804,190	=SUM(CHOOSECOLS(E1#,2))		
10									

Python output spills across cells as a dynamic array, which means you can reference the whole range with the # operator. From there, ROWS(E1#) counts the rows and SUM(CHOOSECOLS(E1#,2)) sums a specific column, giving you live Excel formulas on top of Python output.

5

IMAGES STORED IN CELLS

Python charts come back anchored to a cell.



Storing the plot directly in a cell has clear benefits: it's neatly anchored for easier handling and programmatic use, like treating images as input values for functions such as VLOOKUP(). The trade-off is they render tiny by default, so click the image-link icon to send a resizable copy out onto the worksheet.

PUTTING IT TOGETHER

Worlds collide, and that's the point.

You can leverage Python for tasks that are tricky in Excel, push the results back into Excel, and then use Excel for things that are harder in Python.

Mastering Python in Excel takes a solid understanding of both halves of the equation. It's a skillset I expect to grow more valuable as AI continues to transform how we work with data.

Read the full post at stringfestanalytics.com