

FOR EXCEL ANALYSTS

# There's more to Python and Excel than Python *in Excel.*

Why that matters more in a gen AI world.

by George Mount, Stringfest Analytics

THE GOOD NEWS

# Python in Excel is a great gateway.

You can run pandas, matplotlib, and statsmodels right inside a workbook. For analysts who've never written a line of Python before, that's a real win.

### THE CATCH

**But it's a slice  
of what Python  
can do.**

Python in Excel is sandboxed. It runs on a fixed set of packages, on Microsoft's cloud, with no escape hatch. Most of what Python is famous for sits outside that box.

# What it covers.

The Python in Excel library leans heavily on analysis:

- Data wrangling with pandas
- Charts with matplotlib and seaborn
- Stats with statsmodels and scikit-learn
- Calculations that would be miserable in formulas

# What it leaves out.

Python is also good at the things Excel users actually live with:

- Building workbooks from scratch
- Splitting, combining, and reformatting files
- Pulling data from APIs and databases
- Running batch jobs across hundreds of files
- Wiring Excel into the rest of your stack

THE TOOLKIT

# Python *for Excel.*

None of these come with Python in Excel:

- openpyxl: read and write .xlsx files
- xlwings: drive Excel from Python
- xlsxwriter: produce polished Excel reports
- pyxll: build Excel add-ins in Python

They live outside the sandbox. They're where most of the automation work actually happens.

# Now add gen AI.

What's kept Excel folks away from openpyxl or xlwings is the syntax. Picking up Python on top of a day job was a tall order. Then AI showed up. I describe what I need, Claude drafts the code, I review and run it. The thinking part is still on you, and that part was always the real work.

### WHY IT MATTERS NOW

# Python is the connective tissue.

Pulling data from an API, calling an LLM, transforming the response, writing the result back to a workbook, scheduling the whole thing to run weekly. Python is what stitches those steps together. Excel alone can't.

### WHERE TO START

# A practical path.

- Run Python in a notebook (Jupyter, Anaconda).
- Try openpyxl first. It's the gentlest entry point.
- Have an AI draft the code, then read it carefully.
- Treat Python in Excel as one tool in a bigger kit.

**STRINGFEST ANALYTICS**

**ONE LAST THING**

# **Want more Excel + AI thinking?**

I write and teach about modern Excel, Python, and AI for finance and operations teams.

**[stringfestanalytics.com](https://stringfestanalytics.com)**

Follow for more.