AI-Powered
Financial Modelling

Tools, Architecture & What's Now Possible

A Reference Guide for Finance Professionals

February 2026

Prepared by Steve Parton (SJP Consulting / CBL Analytics)

in conversation with Claude (Anthropic)

A guide for finance and analytics professionals exploring AI tools for Excel, advanced analytics, and modern modelling workflows.

Contents

1. Introduction

The landscape of AI-assisted financial modelling tools is evolving rapidly. For finance and analytics professionals, the building blocks now available — from Claude's financial plugins to Copilot Agent Mode to open source tools like xlwings and Quarto — open up workflow approaches that were not practical even six months ago. This document provides an objective guide to making best use of what now exists: the AI tools for spreadsheet work, the complementary Python/R toolset for advanced analytics and reporting, and a practical architecture that brings them together.

This guide covers three areas: the current AI tools that integrate with or generate spreadsheets, a workflow architecture that combines Excel (for stakeholder-facing models) with Python (for advanced analytics, simulation, and governance), and a practical getting-started path.

What's New This Week (February 24–25, 2026)

As this guide was being finalised, Anthropic announced a significant set of updates at its Enterprise Agents event on February 24, 2026:

- Financial services plugins — five new open source plugins (financial analysis, investment banking, equity research, private equity, wealth management) released on GitHub under Apache 2.0. 41 skills, 38 commands, 11 MCP integrations.
- Partner plugins — LSEG (fixed income, FX, equities, macro analysis) and S&P; Global (Capital IQ tearsheets, earnings previews).
- Cross-application workflows — Claude can now carry context between Excel and PowerPoint in a single session.
- New MCP connectors — FactSet and MSCI join S&P; Global, LSEG, Daloopa, Morningstar, Moody's, PitchBook, Aiera, Chronograph, Egnyte.
- Cowork enterprise features — private plugin marketplaces, per-user provisioning, auto-install.

These plugins are squarely aimed at corporate finance — valuations of listed companies, deal workflows, equity research, and portfolio management using public market data. They do not address project finance, infrastructure feasibility, or other specialist modelling domains (see Section 7). The architecture in Sections 4–6 complements these tools — adding governance, simulation, and reporting layers.

2. AI Tools for Spreadsheet Work

The following section compares five distinct Claude surfaces and two major competitors, focusing on formula handling, VBA support, named ranges, data connectors, and automation depth.

2.1 Claude Surfaces

2.1.1 Claude Chat (claude.ai / Desktop App)

The standard web and desktop chat interface with built-in Python code execution. Creates .xlsx files from scratch with formulas, formatting, charts, and multiple tabs.

Strengths: Good for generating new workbooks from natural language. Can write VBA code, Power Query M code, and Office Scripts as text. Uploads up to 30MB. Strong reasoning.

Limitations: Reads cell values only from uploaded workbooks — does not reconstruct formula logic. Cannot embed VBA into .xlsm files. No live editing.

2.1.2 Claude Cowork (Desktop Agent)

Sandboxed Linux VM with full Python access. Designed for multi-step, iterative workflows. Strengths: Multi-tab models with working formulas, todo tracking, verification steps, Word/PPT/PDF output. Limitations: Same openpyxl engine as Chat. No COM-based automation.

2.1.3 Claude Code (CLI Tool)

Terminal-based agentic coding tool with full local file system access. Strengths: The most powerful option — if xlwings or win32com is installed, can automate Excel via COM including VBA injection, named ranges, and full object model access. Limitations: Requires terminal comfort and a local Python environment.

2.1.4 Claude in Excel (Add-in)

Sidebar add-in running inside Excel itself. Strengths: Reads actual formulas and cell dependencies. Edits cells, builds pivots, creates charts, applies conditional formatting. Supports MCP connectors (S&P;, FactSet, LSEG, Moody's). Limitations: Cannot read/write/execute VBA. No named ranges. No data tables. No saved chat history.

2.1.5 Claude Surface Comparison Matrix

Capability	Chat	Cowork	Code (CLI)	In Excel
Create new .xlsx	Yes (Python)	Yes (Python)	Yes (any lib)	Yes (in-place)
Read existing formulas	Values only	Values only	Full (COM)	Full
Edit workbooks in-place	No	No	Yes (COM)	Yes
VBA: generate as text	Yes	Yes	Yes	No
VBA: embed/execute	No	No	Yes (COM)	No
Named ranges	openpyxl	openpyxl	Full (COM)	No
Conditional formatting	openpyxl	openpyxl	Full (COM)	Yes
Pivot tables	Limited	Limited	Full (COM)	Yes
Charts	Basic	Basic	Full (COM)	Yes

Financial data (MCP)	No	No	No	Yes
Multi-step workflows	Limited	Strong	Strong	Moderate

2.2 Microsoft Copilot (Agent Mode in Excel)

Agent Mode reached GA in January 2026. The most Excel-native AI assistant — built on the Excel calculation engine. Generates formulas, creates pivot tables, generates named ranges, builds charts. Users can switch between GPT 5.2 and Claude Opus 4.5. Can generate VBA code but cannot execute it. Regional note: Personal/Family subscriptions not available in EU or UK.

2.3 Google Gemini (in Google Sheets)

Tightly integrated into Sheets for formula suggestions, profiling, charting, and cleaning. Key limitation: No VBA support (uses Apps Script). Struggles with multi-sheet relationships. Best suited for teams already in Google Workspace doing lightweight collaborative analysis.

2.4 Head-to-Head: AI Excel Tools

Capability	Copilot Agent	Claude in Excel	Gemini
Named ranges	Yes (generates)	No	N/A (Sheets)
VBA execution	No (text only)	No	N/A
Formula depth	Strong (native)	Strong (reads)	Moderate
Pivot tables	Yes	Yes	Sheets-native
Data connectors	M365 ecosystem	MCP (S&P, FactSet...)	Workspace
Model switching	GPT 5.2 / Opus 4.5	Sonnet / Opus	Gemini only
Regional restrictions	Not EU/UK (consumer)	None known	None known

3. Python in Excel (Microsoft Built-in)

3.1 What It Does

Users type Python code directly into cells using the `=PY()` function. Code executes in a sandboxed Azure container with pandas, NumPy, matplotlib, scikit-learn, and NLTK pre-installed. Results display in the worksheet grid.

3.2 What It Does Not Do

Python in Excel is positioned as an alternative to the formula language, not to VBA. Constraints:

- Cannot access local files, devices, or the user's computer
- Cannot call web APIs or install custom packages
- Cannot interact with VBA macros or modify workbook properties
- Requires internet connection (cloud execution only)
- Subject to monthly compute quotas (`#BLOCKED` error when exceeded)
- No parallel execution — runs left-to-right, top-to-bottom

In essence, Python in Excel is a two-dimensional Jupyter notebook embedded in the grid — excellent for analytics but cannot automate workbook operations.

3.3 Office Scripts (TypeScript-based Automation)

Microsoft's cloud-first automation language using TypeScript. Deep integration with Power Automate for scheduled/event-driven flows. Strategic outlook: Microsoft has stated VBA will not be enhanced. All development investment goes into Office Scripts and Power Platform. Retain VBA for legacy; adopt Office Scripts for new cloud-based workflows.

4. Modern Workflow Architectures

4.1 xlwings: Replacing VBA with Python

xlwings is an open-source Python library (BSD licence) providing full programmatic control over Excel via COM (Windows) or application scripting bridge (Mac). Syntax deliberately close to VBA — the lowest-friction migration path for VBA-experienced professionals.

Core Capabilities:

- Scripting: Automate Excel using VBA-like syntax (`wb.sheets['Sheet1'].range('A1').value`)
- Macros: Replace VBA macros with Python, triggered from buttons or ribbon commands
- UDFs: Custom Excel functions in Python (Windows only) with NumPy/pandas support
- Named ranges: Full read/write access — a critical gap in most AI tools
- Deployment: xlwings Server (centralised) or xlwings Lite (WebAssembly, no install needed)

4.2 Monte Carlo Simulation

The Excel-Python Hybrid Approach: Keep the financial model in Excel (stakeholder-facing) and use Python for the simulation engine. xlwings provides the bridge:

1. Build the financial model in Excel with clearly identified input assumptions
2. Use xlwings to connect Python to the workbook
3. Run Monte Carlo in Python using NumPy for random sampling across distributions
4. For each iteration: write sampled inputs → trigger recalculation → capture outputs
5. Analyse results: histograms, percentile tables, tornado charts, threshold probabilities
6. Write summary results back to Excel or render via Quarto

4.3 Quarto: Automated, Reproducible Reporting

Open-source publishing system from Posit. Takes markdown + executable Python/R code → polished PDFs, Word documents, HTML pages, presentations, dashboards, and books. All outputs are reproducible.

Why Quarto replaces VBA-based reporting:

- Parameterisation: Same template, different values → customised reports per business unit/period
- Data integration: Read from Excel, SQL, APIs, or CSV directly
- Professional formatting: Publication-quality output from templates
- Version control: Plain text .qmd files work with Git
- Batch rendering: One command renders all entities, months, or scenarios

5. Dual-Model Reconciliation

Excel as Source of Truth

5.1 The Concept

Build the financial model simultaneously in both Python and Excel, reconcile them, then use as the foundation for all downstream analytics. Once reconciled:

- Excel = sole source of truth for assumptions (stakeholder-facing)
- Python = analytical engine (reads from Excel, runs simulation, sensitivity, statistics)
- Quarto = reporting layer (pulls from both Excel and Python)

5.2 Why This Architecture Works Now

- AI makes dual-model construction feasible: AI generates both Excel formulas and equivalent Python logic from a single specification.
- Current AI tools cannot replace VBA: Separating into Excel assumptions + Python computation eliminates the need for VBA entirely.
- Python in Excel is not enough alone: =PY() cannot access files, APIs, or custom packages. External Python removes these limits.
- Reconciliation = built-in model audit: The dual-model approach produces a verified model as a byproduct of the build process.
- Quarto unifies reporting: Reports show assumptions as stakeholders entered them alongside analytical depth Excel cannot produce.

5.3 Formula Change Detection

Value reconciliation detects output differences but not necessarily why. The formula snapshot approach addresses this:

- Baseline capture: Export every formula string (not values) as a JSON file keyed by sheet/cell.
- Formula diff on failure: Compare current formulas to baseline to identify changed cells.
- PR workflow: Updated snapshots committed to Git → pull requests show readable formula diffs.

This two-layer approach — value reconciliation plus formula snapshot — provides comprehensive change detection that approximates formal model audit, automated and re-runnable on every iteration.

5.4 Version Control: Excel as Artifact, YAML as Source

The architecture treats .xlsx files as build artifacts rather than source files. Version-controlled source consists of text files: assumptions (YAML), formula map (YAML/JSON), structure definition (YAML), and Python model code. Governance happens in Git; the user experience stays in Excel.

6. Reference Architecture

6.1 The Stack

Layer	Tool	Role
Front end	Microsoft Excel	Assumption input, stakeholder review
Automation bridge	xlwings	Replace VBA: read/write Excel, named ranges
Analytics engine	Python / R	Monte Carlo, statistical analysis, transformation
In-grid analytics	Python in Excel	Embedded analytics within the workbook
Cloud automation	Office Scripts + Power Automate	Scheduled, event-driven automation
AI assistance	Claude in Excel / Copilot	Formula generation, model review
Reporting	Quarto	Parameterised PDF/Word/HTML reports
Version control	Git	Track changes to scripts, templates, config

6.2 Getting Started

Each layer can be adopted independently:

- Start with AI in-Excel: Claude in Excel or Copilot for formula generation and exploration.
- Add Python alongside Excel: xlwings to read/write existing workbooks.
- Build the dual-model approach: Generate both Excel and Python versions from one spec.
- Introduce Quarto reporting: Move one report process to Quarto as proof of concept.
- Add Monte Carlo: Simulation in Python/R, connected to Excel via xlwings.

6.3 Key Gaps to Watch

- VBA execution: No AI tool can execute VBA automatically.
- Named ranges: Only Copilot Agent Mode generates them.
- Data tables (what-if): Not supported by any AI tool in-place.
- Copilot regional availability: Not in EU/UK for consumer plans.
- Python in Excel compute limits: Monthly quotas disrupt large simulations.
- Office Scripts maturity: Feature coverage doesn't yet match VBA.

7. Skill Libraries

From Existing Models to Reusable AI Workflows

The financial plugins are excellent at corporate finance — a `/dcf Apple` command builds a standard DCF from public market data in minutes. But finance covers much broader territory.

Consider a renewables project finance model: generation engine (capacity × load hours × monthly profile × availability × degradation), PPA waterfall, FAST-compliant corkscrew balances, and sector-specific integrity checks. None of this exists in any plugin. Custom Claude skills fill the gap.

7.1 Building Skills from Existing Excel Models

1. Feed the existing model to Claude Code — reads formulas, structure, and logic via COM libraries. Other surfaces read values only.
2. Claude generates the skill — a SKILL.md capturing methodology, build sequence, compliance rules, and verification checks.
3. Pre-populate the formula YAML — extracted directly from the existing model.
4. Add sector reference documents — domain-specific assumptions, cost ranges, reasonableness checks.

The old model becomes the training data; the skill becomes the reusable institutional knowledge.

7.2 Where Plugins and Custom Skills Fit Together

Domain	Plugin?	Custom skill?
Corporate DCF (listed companies)	Yes	No (plugin sufficient)
Comparable company analysis	Yes	No
LBO / leveraged buyout	Yes	No
3-statement models (SEC)	Yes	No
Renewables project finance	No	Yes — generation, PPA, FAST
Toll road / transport	No	Yes — traffic, concession
Mining / resources	No	Yes — reserves, commodity
Real estate development	No	Yes — staging, absorption
Data centre / infrastructure	No	Yes — power, redundancy

7.3 Studying the Plugin Skill Files

The plugin skill files are worth studying regardless. They are markdown files in a public GitHub repo (anthropics/financial-services-plugins), Apache 2.0 licensed. Reading how Anthropic structured their DCF and deal workflow skills provides a practical template for your own domain-specific skills.

8. Conclusion

The AI tools have largely solved the model build problem. Claude's financial plugins, Copilot Agent Mode, and Claude in Excel make it remarkably fast to go from a specification to a finished, formatted workbook.

The question for finance professionals is how to make best use of all these building blocks together. The architecture described here — Excel as the stakeholder-facing model, Python for advanced analytics and simulation, xlwings as the bridge, Quarto for reproducible reporting, and Git for governance — wraps around the AI-generated outputs to deliver reuse, auditability, and analytical depth.

Each layer can be adopted independently, and the benefits compound as more of the stack comes together. The building blocks are already in place.

Sources & Further Reading

- Microsoft: Python in Excel Availability
- Microsoft: Agent Mode in Excel
- Microsoft: Differences between Office Scripts and VBA
- Anthropic: Use Claude in Excel
- xlwings Documentation
- xlwings: My Thoughts on Python in Excel
- Quarto: Parameterized Reports with Python
- Posit: Building a Reporting Infrastructure with Quarto
- Analytics Vidhya: Monte Carlo Simulation in Excel with Python
- PyXLL: Monte Carlo Simulations in Excel with Python
- FM Magazine: Introduction to Python in Excel