
PyConUS
2025
Pittsburgh

# Snakes in a Grid:

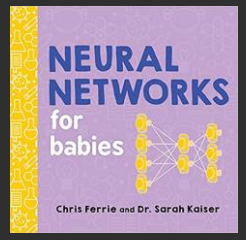
## Working with spreadsheets with Python + Python in Excel

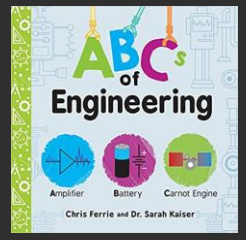
Sarah Kaiser, Sr. Dev Advocate @ Microsoft
@crazy4pi314 | sckaiser.com | aka.ms/grid-snakes

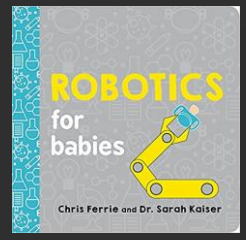
PyConUS 2025

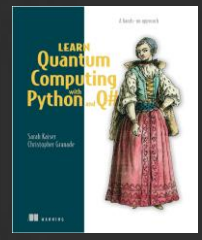
# Intro

- 🥑 Dev Advocate @ MSFT
- 📡 Quantum space tech dev
- 📚 Author for all ages
- ⚔️ Final Fantasy gamer
- 🚢 Seattle boater



# Agenda

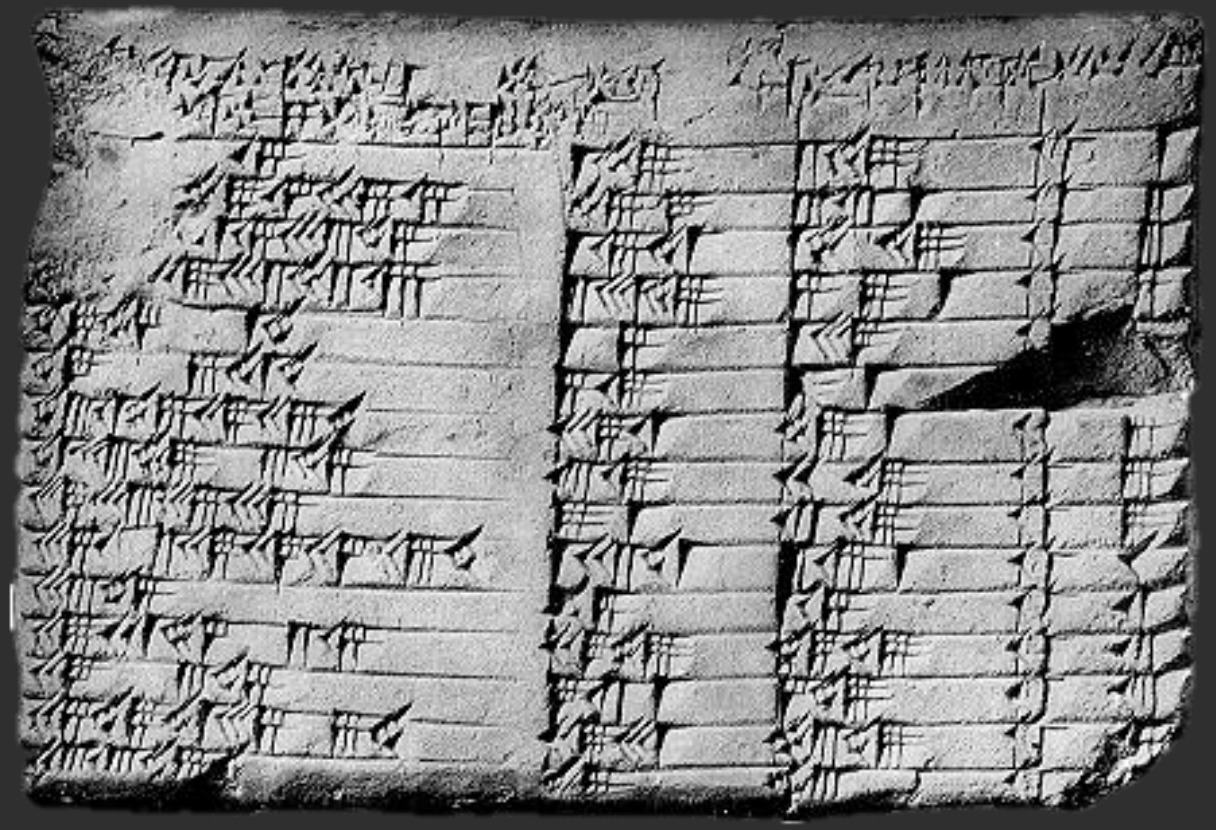
1. What is a spreadsheet?
2. DataFrames ↔ sheets w/ **`pandas`**
3. Customize sheets w/ **`openpyxl`**
4. Use Python directly in spreadsheets
5. GAME TIME

# You will learn how to..

- ❑ Load and work with spreadsheets in Python
- ❑ Create + customize spreadsheets from Python
- ❑ Run Python directly in your spreadsheets

# 1.
# what is a spreadsheet *really*?

# Tab(let)ular data



"Plimpton 322". *personal.math.ubc.ca*. Retrieved 2023-05-20.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | sepal_length | sepal_width | petal_length | petal_width | species |
| 2 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 3 | 4.9 | 3 | 1.4 | 0.2 | setosa |
| 4 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 5 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |

data.csv

```
Year,Make,Model,Length
1997,Ford,E350,2.35
2000,Mercury,Cougar,2.38
```

# Tabular text formats

- csv, tsv, or other text delineated files
  - **Lowest common denominator format**
  - Human readable
  - Efficient storage
- Mainly used for storage, need other tools to analyze/plot

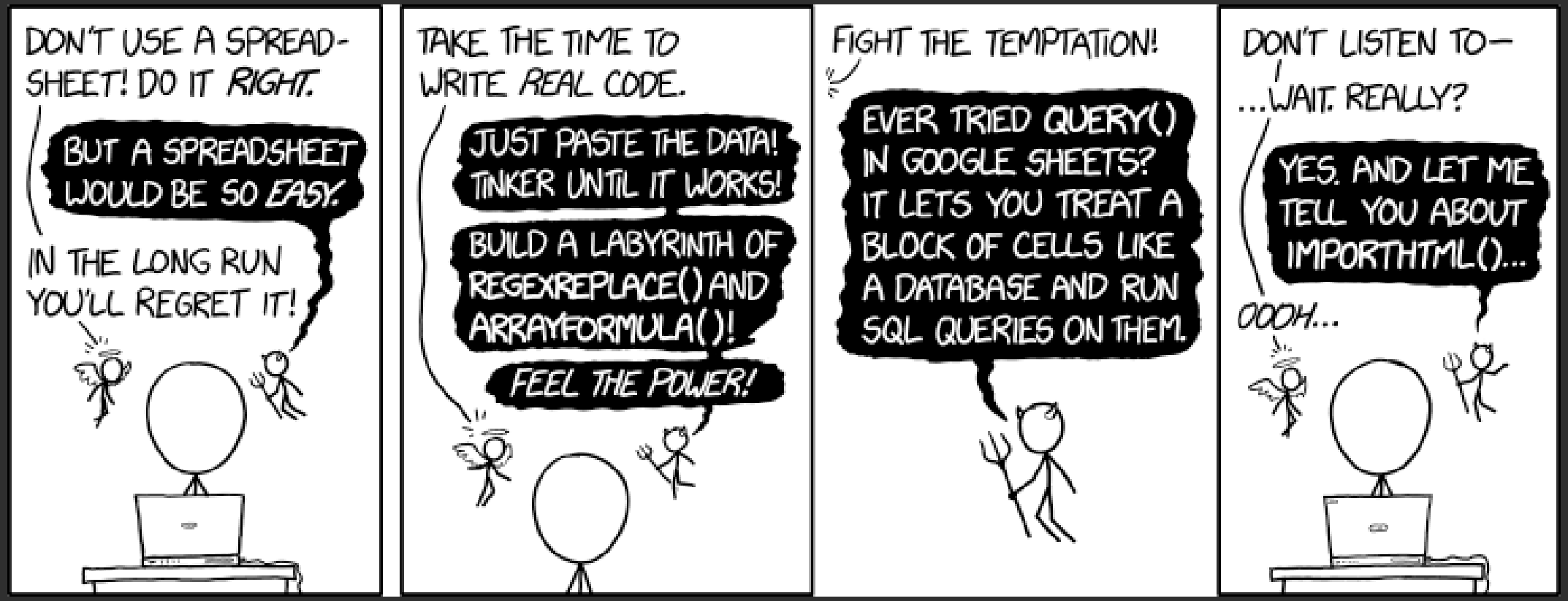
# Spreadsheets: The future?

- Applications that allow storing and manipulation of tabular data
  - ex. VisiCalc, Lotus 1-2-3, Excel, LibreOffice Calc, PySpread
- Automatic/real-time calculation alongside data
- “What-if” analysis

# Accessible data science

| Type | Color | Location |
|---|---|---|
| Basalt | Dark grey | Hawaii |
| Shale | Light grey | Riverbank |
| Quartz | Pink | Playground |
| Marble | Dalmatian | Northern MN |
| Sandstone | Tan | Lake superior |

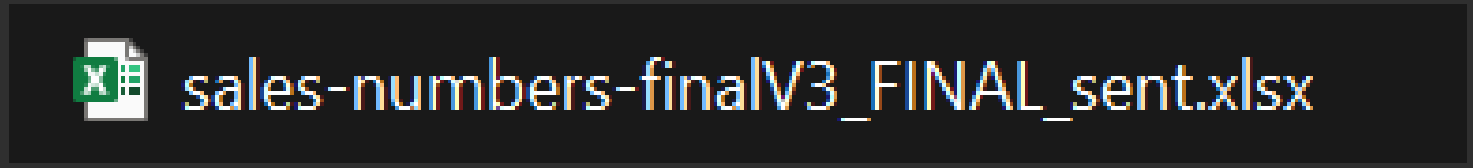
https://xkcd.com/2180/




My brother once asked me if there was a function to produce a calendar grid from a list of dates in Google Sheets. I replied with a single-cell formula that took in a list of dates and outputted a calendar. It used SEQUENCE(), REGEXMATCH(), and a double-nested ARRAYFORMULA(), and it locked up the browser for 15 seconds every time it ran. I think he learned a lot about asking me things.

# Spreadsheets: Limitations

- Harder to audit/version control
  - Can be harder to collaborate
- Potentially lots of function repetition
- More limited data sizes



# File extension history

## xls

- Binary exchange file format
- Supported macros
- Larger file sized
- Limited selection of functions and plotting

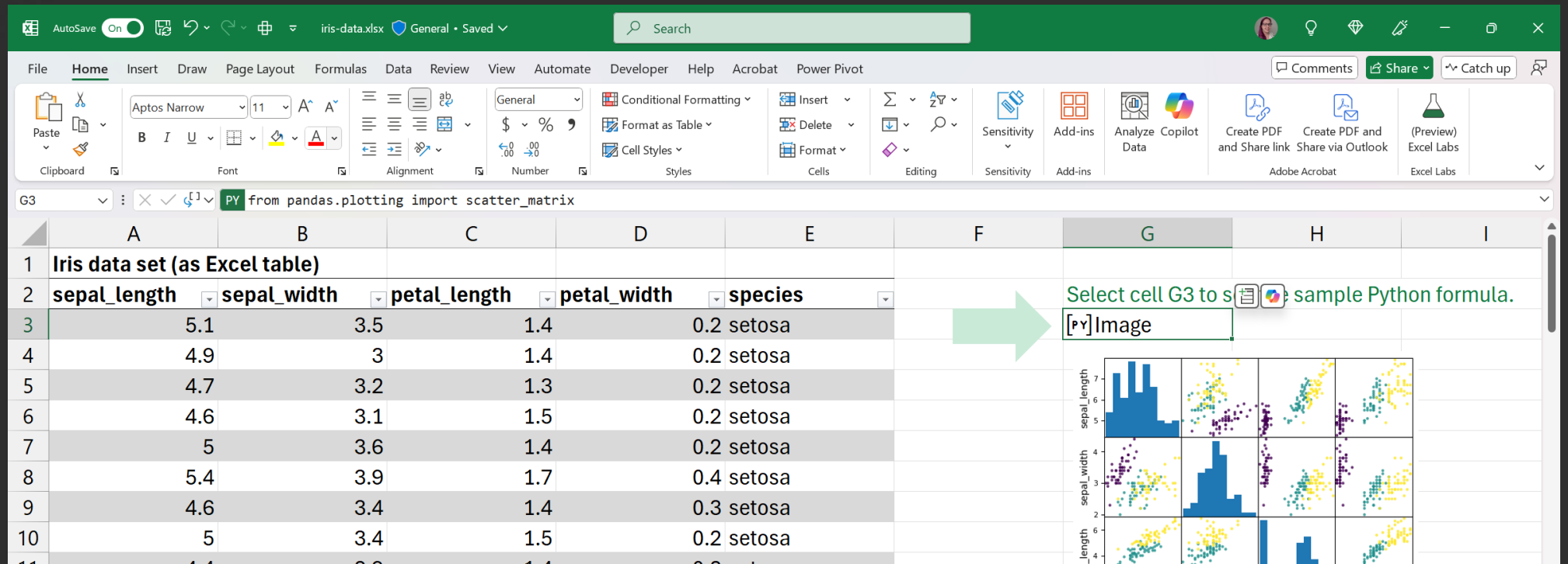
## xlsx (~2007)

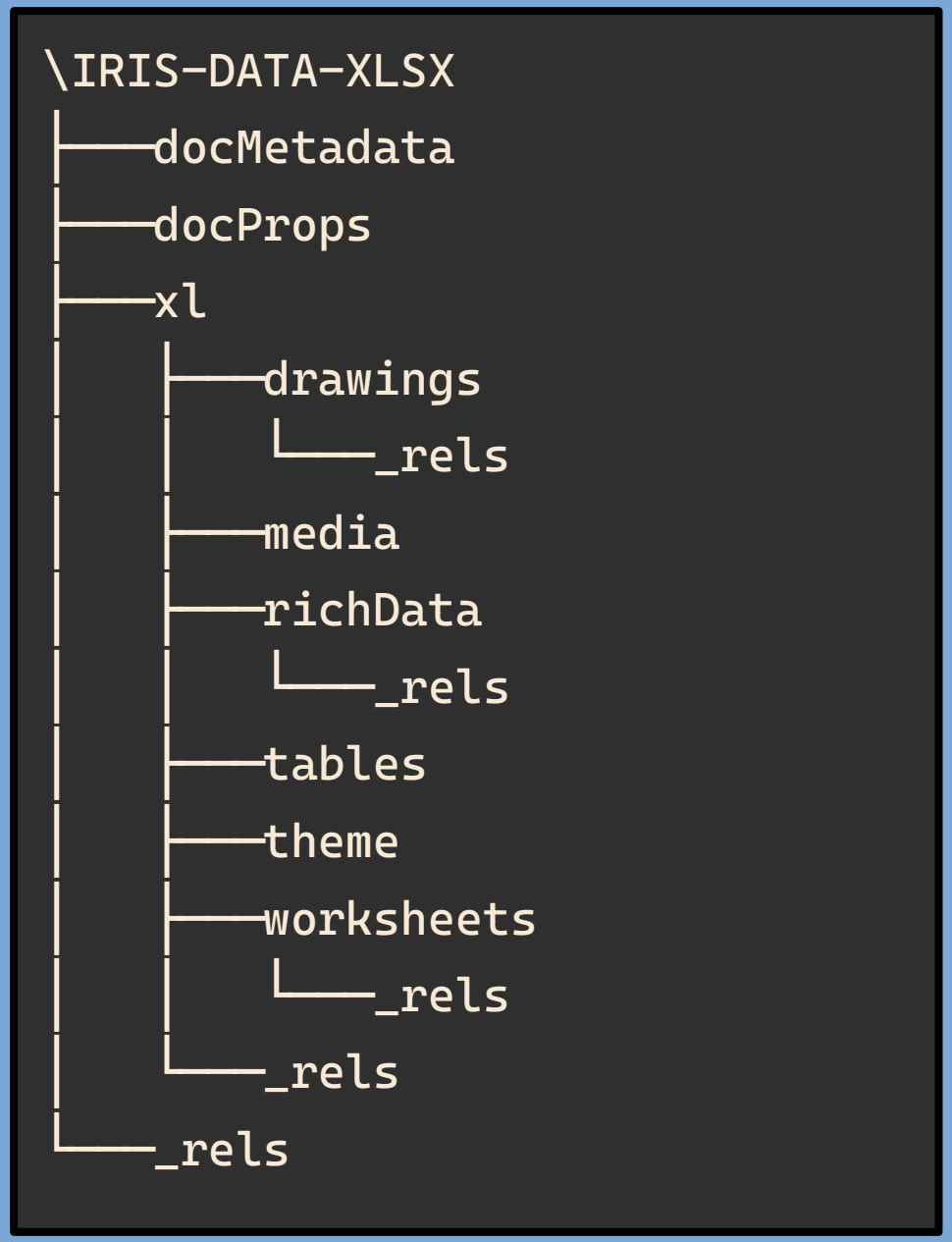
- XML based file format
- xlsm for macro enabled, better security controls
- Better storage efficiency
- Large and extendable function and plotting tools

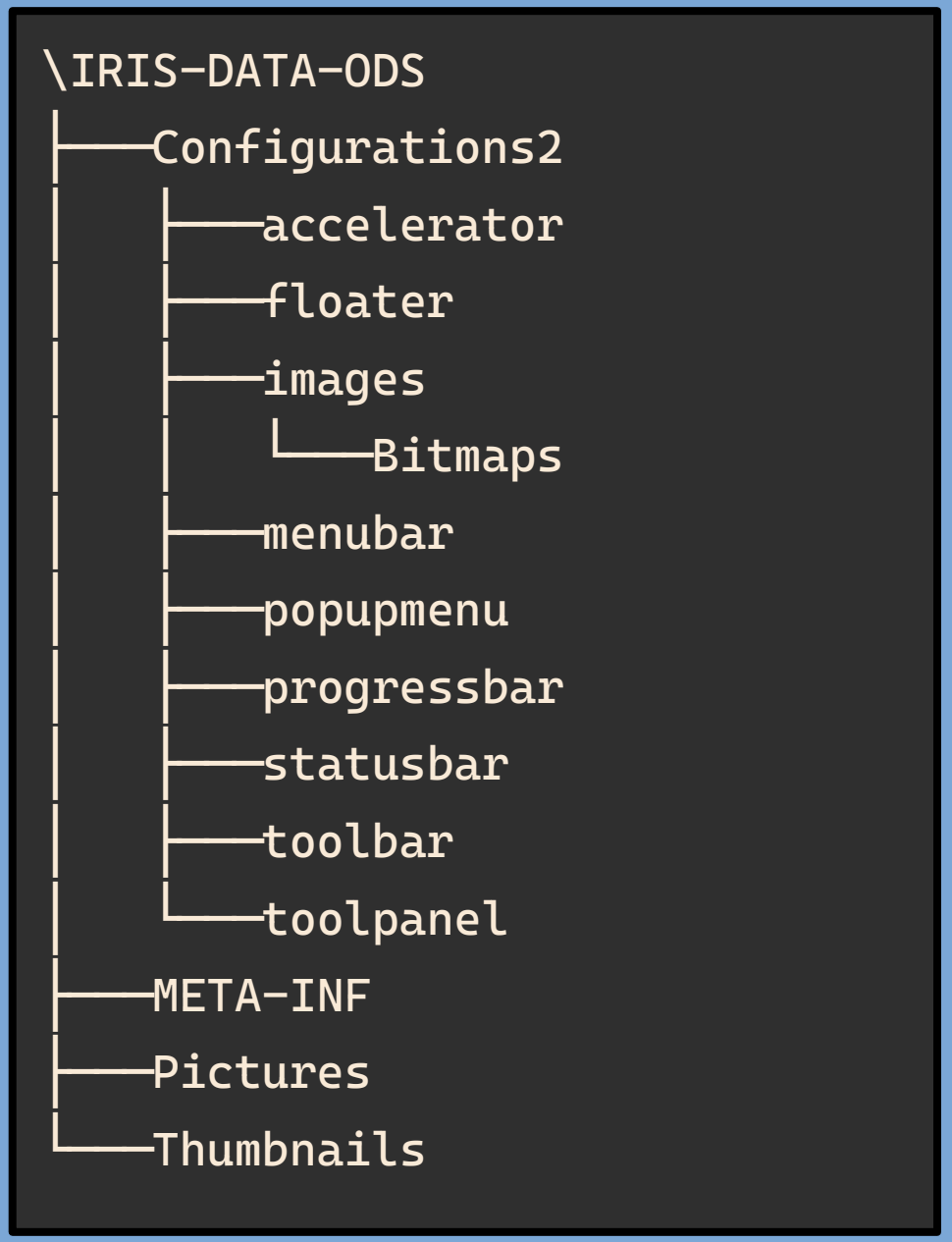
Differences between the OpenDocument Spreadsheet (.ods) format and the Excel for Windows (.xlsx) format

# Unwrapping an xlsx file

- Change extensions to *.zip and unpack an xlsx/odf spreadsheet




\IRIS-DATA-XLSX
docMetadata
docProps
xl
drawings
_rels
media
richData
_rels
tables
theme
worksheets
_rels
_rels
_rels


\IRIS-DATA-ODS
Configurations2
accelerator
floater
images
Bitmaps
menubar
popupmenu
progressbar
statusbar
toolbar
toolpanel
META-INF
Pictures
Thumbnails

| | A | B | C |
|---|---|---|---|
| 1 | sepal length | sepal width | species |
| 2 | 5.1 | 3.5 | setosa |
| 3 | 4.9 | 3 | setosa |
| 4 | 4.7 | 3.2 | setosa |
| 5 | 4.6 | 3.1 | setosa |

sheet1

# Cell reference syntax

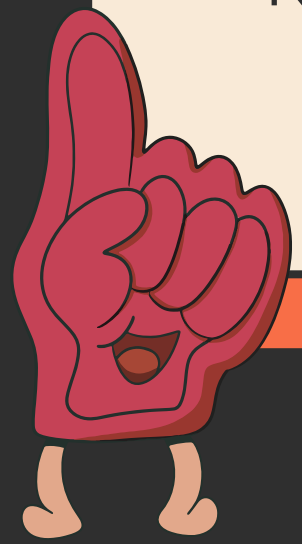
- B2
- A3:C3
- sheet1!C5
- $C$1
  - $ pins a reference, can do column, row or both

# Dates are hard...

- Excel incorrectly assumes that the year 1900 is a leap year - Microsoft 365 Apps | Microsoft Learn
  - Dates and Times — openpyxl 3.1.3 documentation
- ISO 8601 is best pratice
  - 2025-05-12T23:38:55Z or 20250512T233855Z
- **tl;dr** Be cautious if you are using dates prior to 1900!

# Tip: Manage your metadata

- If you have your own way of keeping track of data that's great!
- If not, try using the first (or last) sheet of a workbook or python packages like **`pandas`** or **`xarray`** to help keep information about data stored with the data!



| | A | B | C | D |
|---|---|---|---|---|
| 1 | **What's in this Dataset?** | | | |
| 2 | **Name** | Summer Sports Experience and "Kids in Motion" Programming: 2022 to current (2025) | | |
| 3 | **URL** | https://data.cityofnewyork.us/Recreation/Summer-Sports-Experience-and-Kids-in-Motion-Progra/4pta-f4ca/about_data | | |
| 4 | **Description** | The Kids in Motion (KIM) program provides free activities in NYC's playgrounds, including organized sports, games, fitness demos, and board games. The Summer Sports Experience (SSE) program, which provides sports instruction to children ages 8 to 14. This dataset contains the weekly attendance for each of these programs at a specific site. The programs run from April to October each year. | | |
| 5 | **Rows** | 12.3K | | |
| 6 | **Columns** | 7.00 | | |
| 7 | **Each row is a...** | Each record represents weekly attendance at at specific site. | | |
| 8 | | | | |
| 9 | **Schema** | | | |
| 10 | *Column Name* | *Description* | *API Field Name* | *Data Type* |
| 11 | **Borough** | Borough in which Kids in Motion or Summer Sports Experience program occurred | borough | Text |
| 12 | **ParkorPlayground** | Name of park, playground, or recreation center where Kids in Motion or Summer Sports Experience program occurred | parkorplayground | Text |
| 13 | **Date** | Date class occurred | date | Floating Timestamp |
| 14 | **SportsPlayed** | Name of sport played for Summer Sports Experience | sportsplayed | Text |
| 15 | **KIMorSSE** | Is this class part of Kids in Motion or Summer Sports Experience | kimorsse | Text |
| 16 | **Attendance** | Attendance for the week | attendance | Number |
| 17 | | | | |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |
| 21 | | | | |
| 22 | | | | |
| 23 | | | | |

README | data

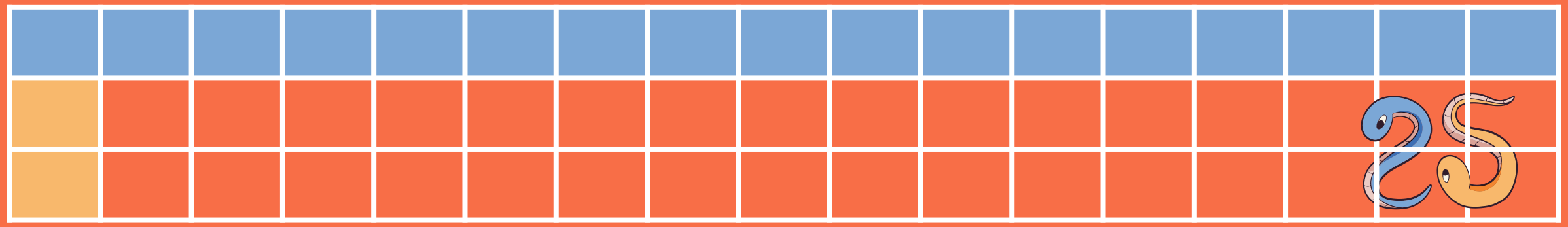
Summer Sports Experience and "Kids in Motion" Programming: 2022 to current | NYC Open Data

# Exercise: Introduce your data

From the tutorial repo, select either animal crossing or board game sample data and with your spreadsheet tool of choice, make a "README" sheet to document the data set origin and assumptions you might have about the data.

Find someone sitting nearby and then introduce your data set and see if they have any questions or assumptions that could be documented as well.

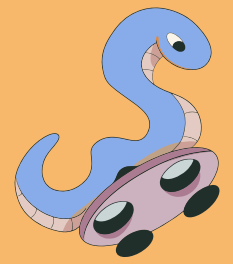
aka.ms/grid-snakes



# 2. dataframes ↔ spreadsheets with pandas

# Pandas <-> Spreadsheet terms

| pandas | Spreadsheet |
|---|---|
| DataFrame | worksheet |
| Series | column |
| Index | row headings |
| row | row |
| NaN | empty cell |



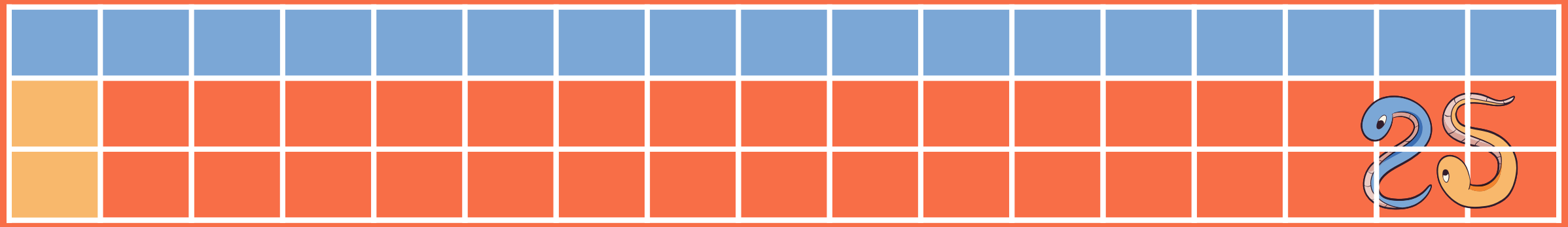
# Let’s look at some code!

# Demo: Now you give it a go!

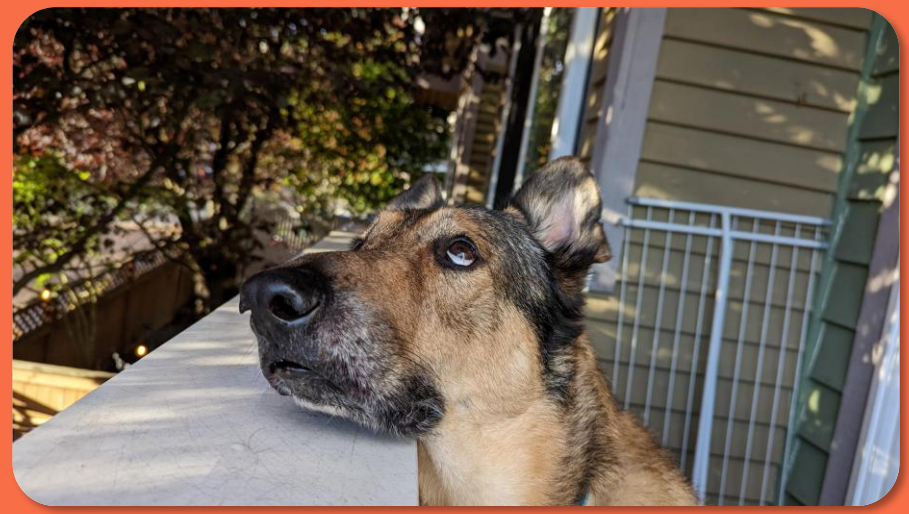
Take your selected sample data and use **`pandas`** to load into Python, clean it up how you might think would be helpful (maybe combine sheets, parse out columns, etc.), and export back to a spreadsheet.

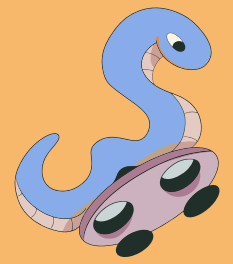
Document any assumptions you make in the cleaning in the README.

`aka.ms/grid-snakes`



# 3. Build spreadsheets with openpyxl



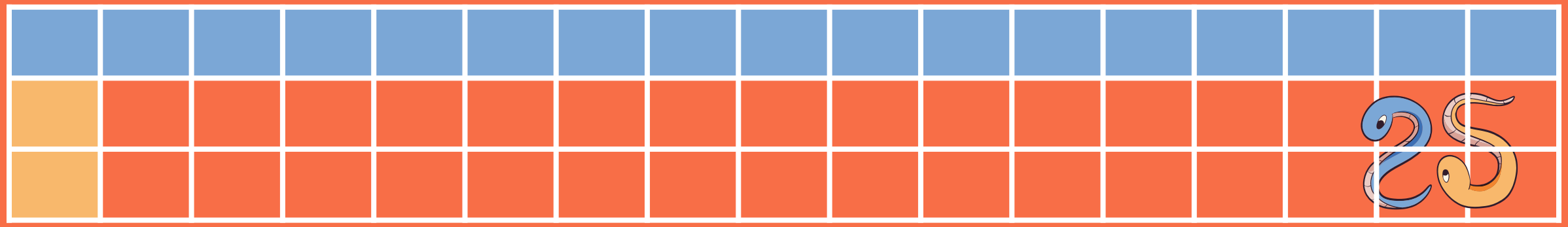


# Let’s look at some code!

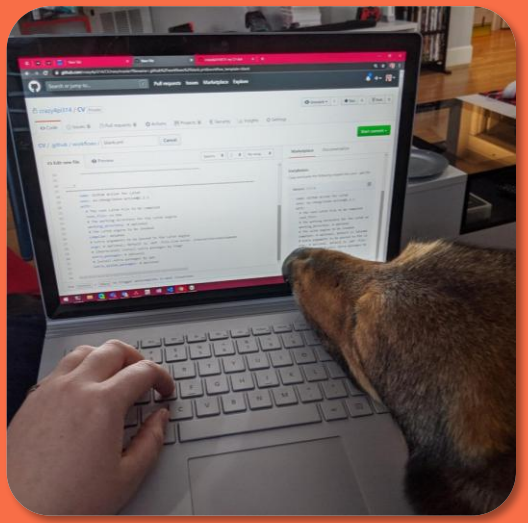
# Demo: Now you give it a go!

You will use your spreadsheet viewer of choice to add a plot to your sample data and then use `openpyxl` to grab that image and then load your data to generate a comparable plot in Python.

aka.ms/grid-snakes

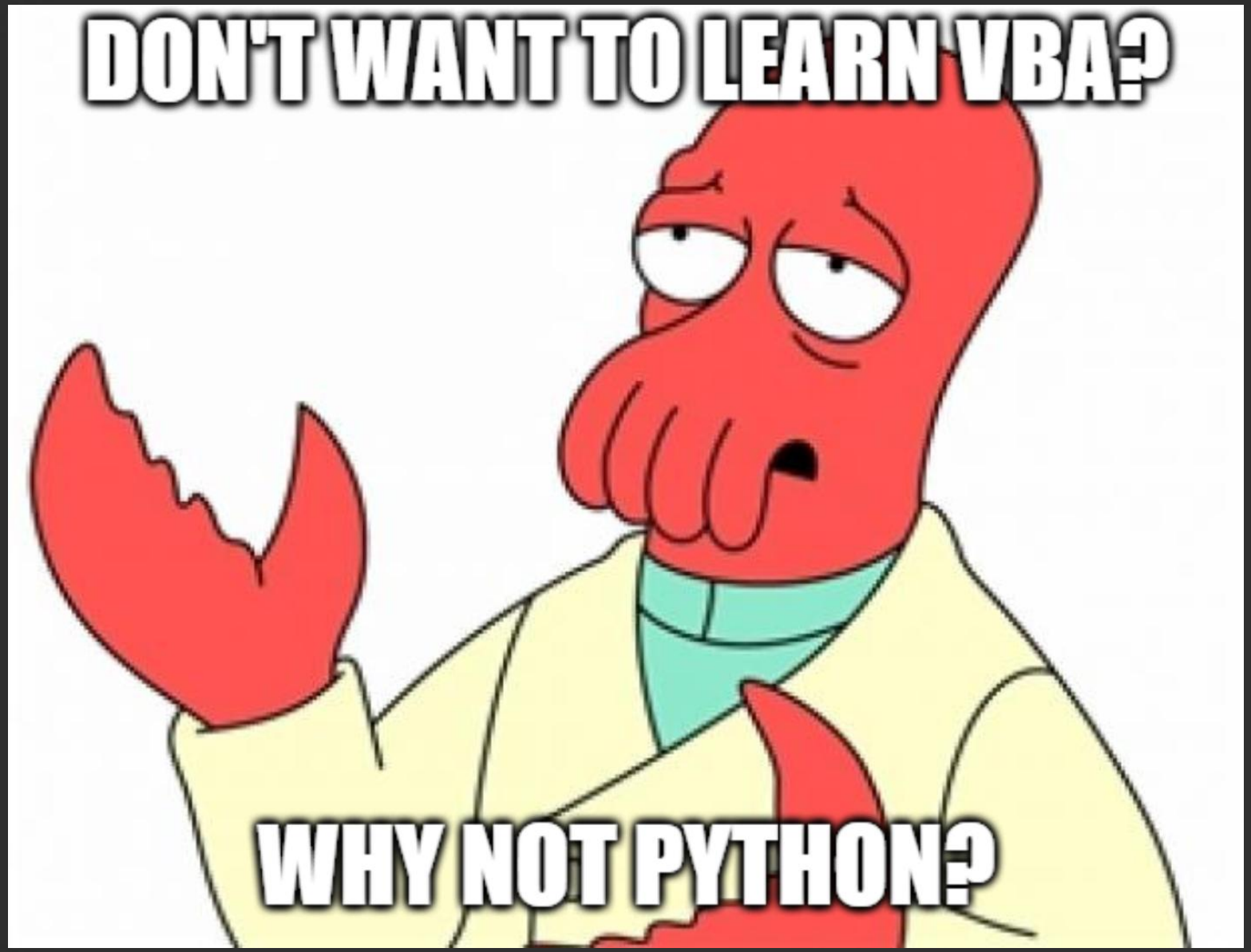



# 4. Use Python directly in spreadsheets



# Macros refine data

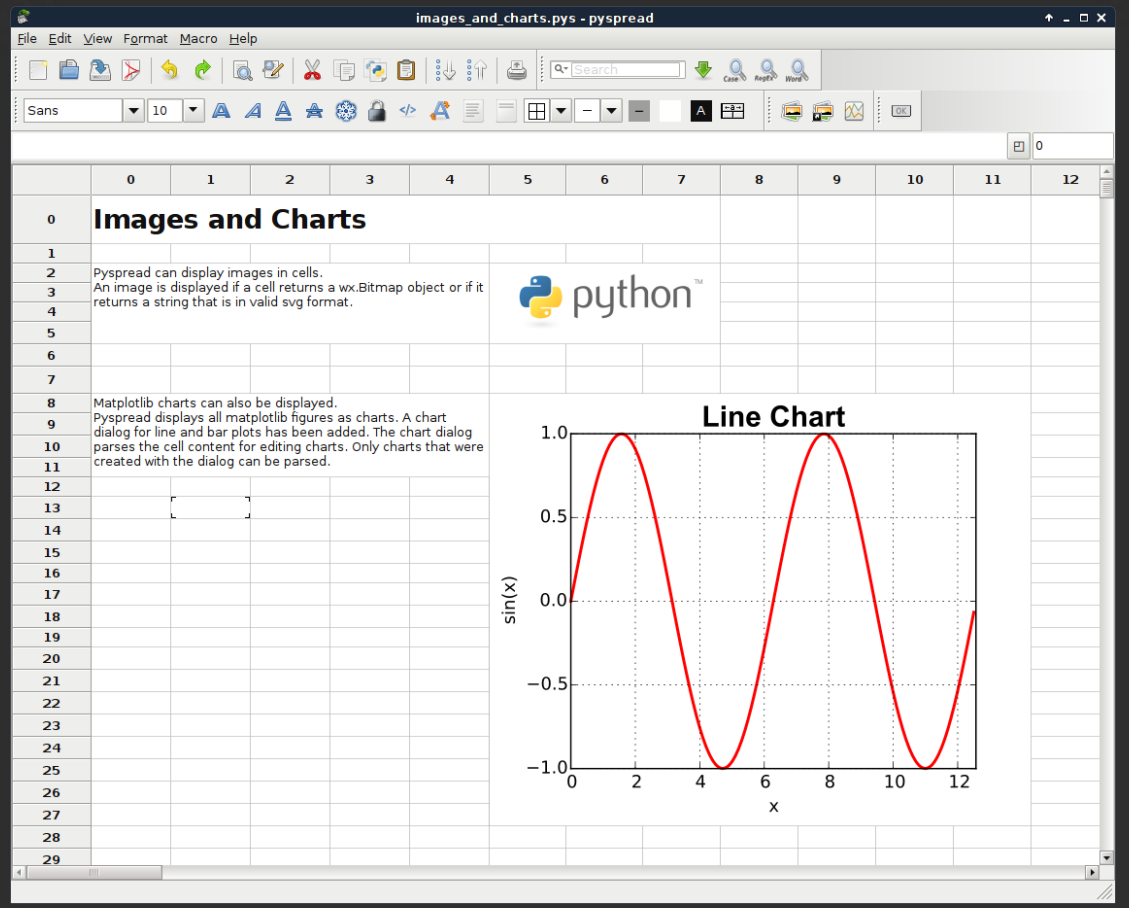
- Spreadsheets already had basic calculator functions
- Macros were a way to record or script advanced actions and functions in a spreadsheet
- Can have some security risks
- VBA or Visual Basic for applications was all there was for a long time


DON'T WANT TO LEARN VBA?
WHY NOT PYTHON?

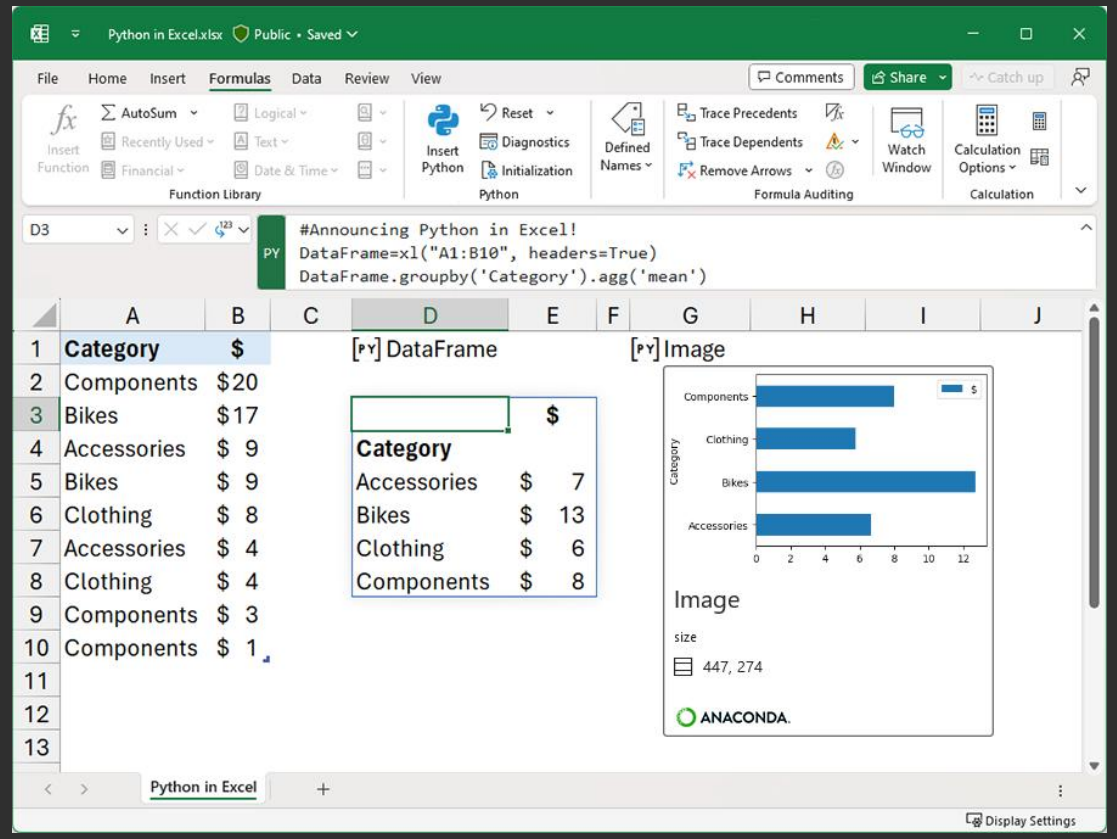
# Some options:

## Pyspread

- Full PyQT6 spreadsheet editor built in Python, using Python for the formulas
- Works with csv, but uses own spreadsheet format (*.psy)



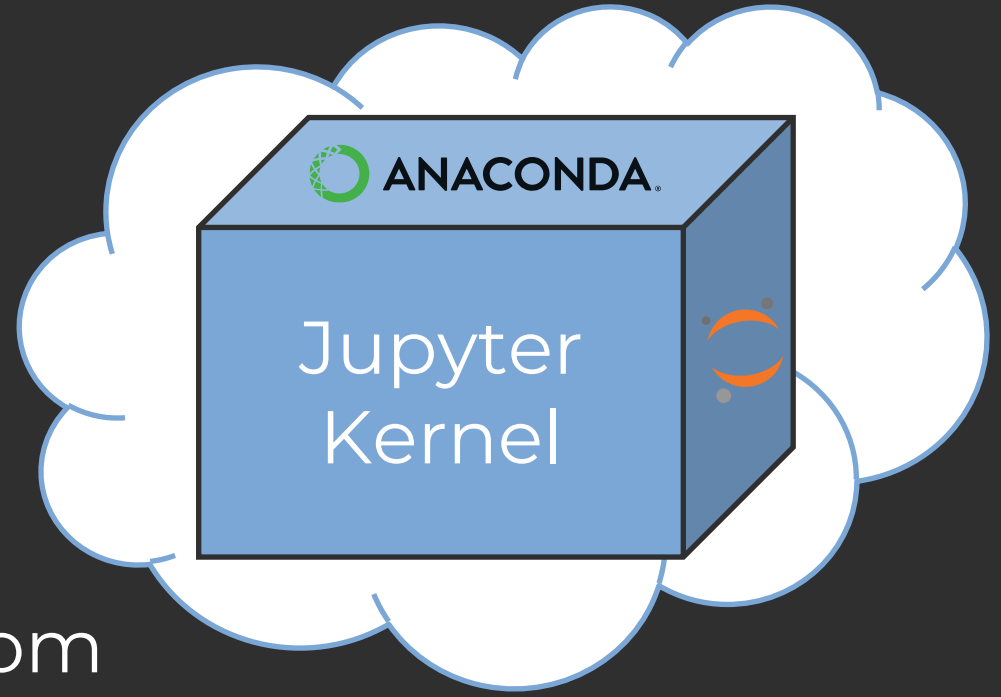
# Some options:



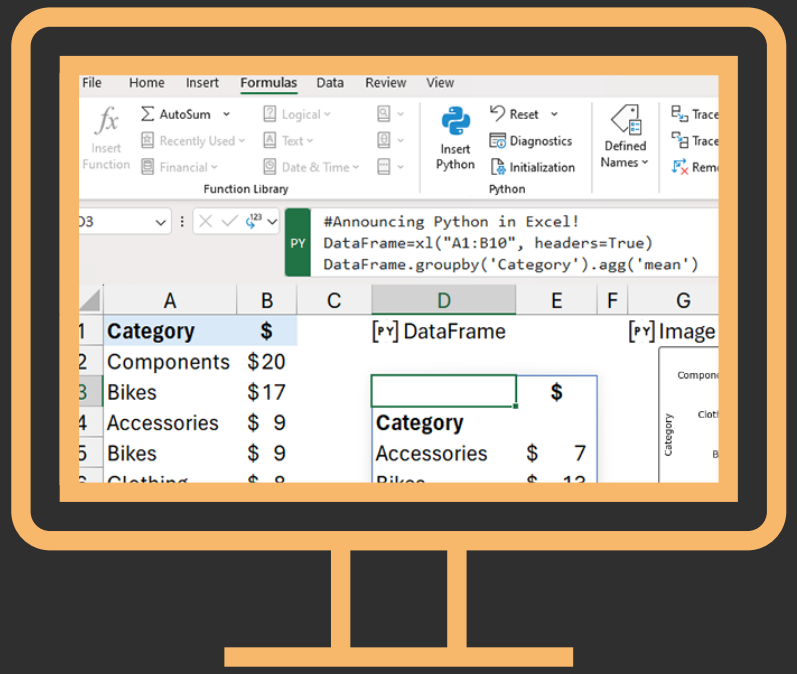

## Python in Excel

- Add-on feature of Excel that allows you run Python directly in the cell
- Easy to just add Python to existing workflows and analyses without creating new “projects”

The containers stay online as long as the workbook is open or until a timeout occurs.




Excel shuttles data to and from python code via **`xl`** package




Returns a limited set of values to Excel (ex. strings, numbers, images)

Python code runs on hypervisor isolated containers built on Azure Container Instances.

**Remote Execution**

The container has Python and a curated set of secured libraries provided by Anaconda.

**Trusted Packaging**

No local disk access, user token, or network access.

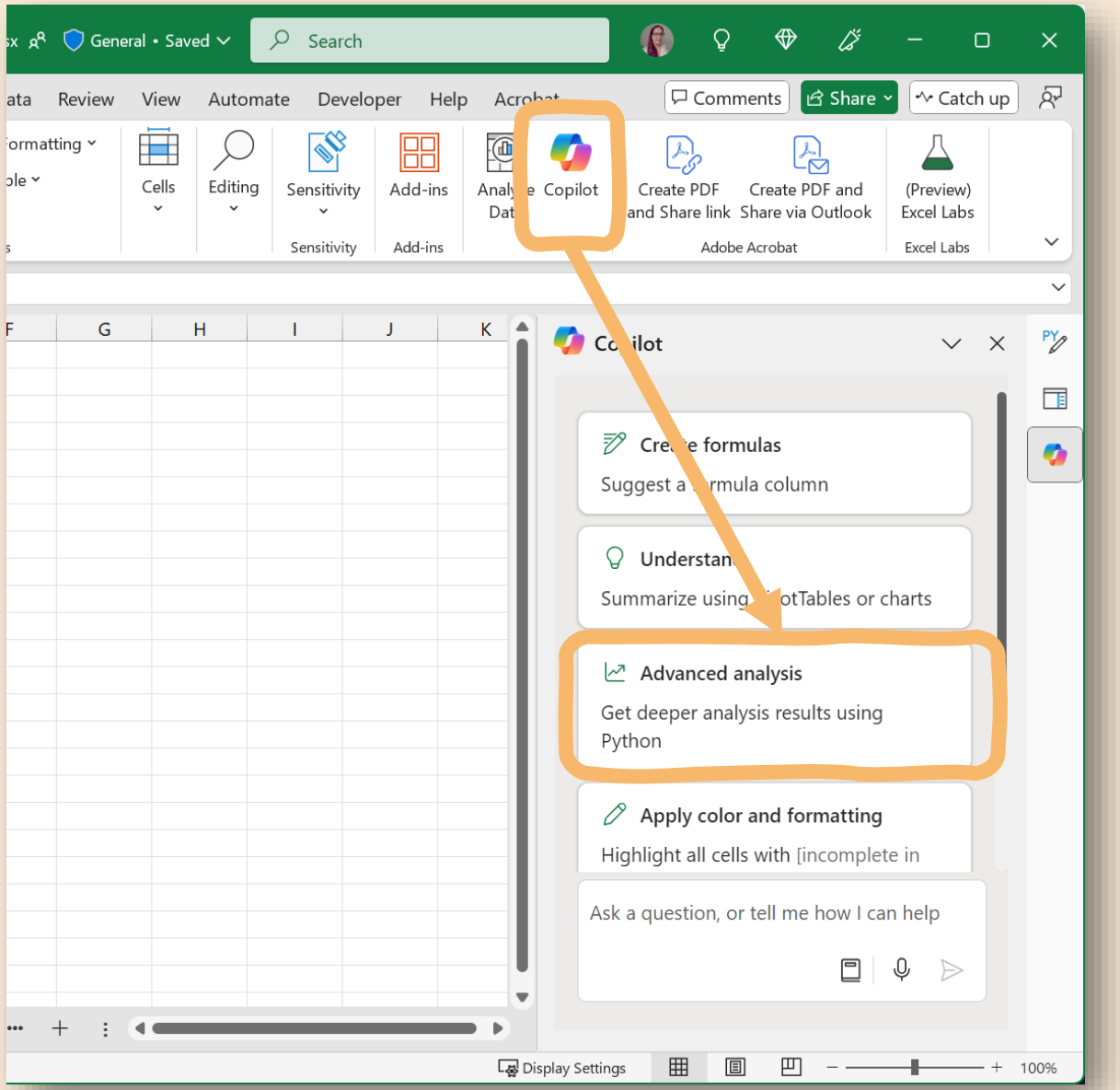
**Secure Devices**

Connected to other tools like OneDrive, Advanced Analysis, sharing links, etc.

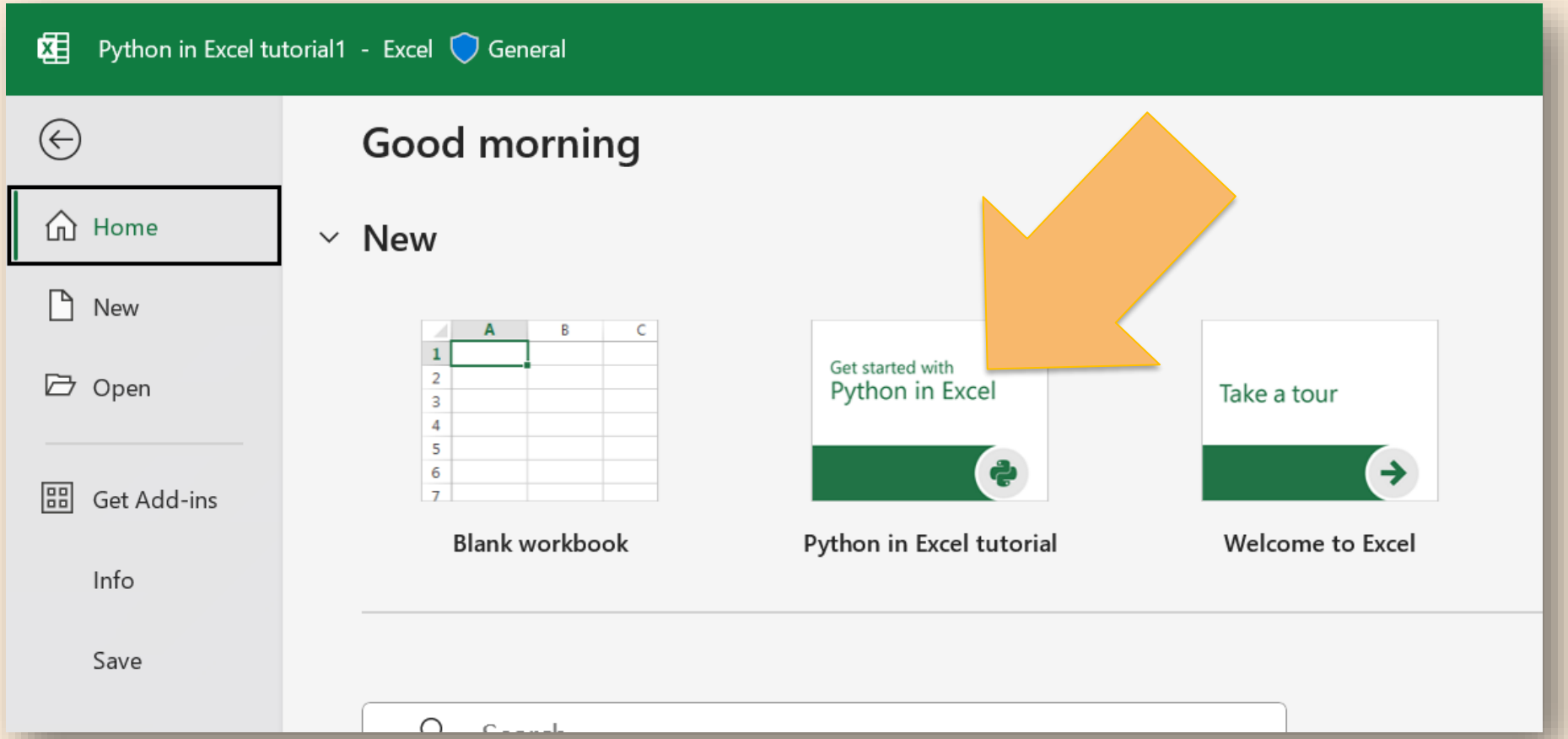
**Easy Collaboration**

# You can pair program with Copilot

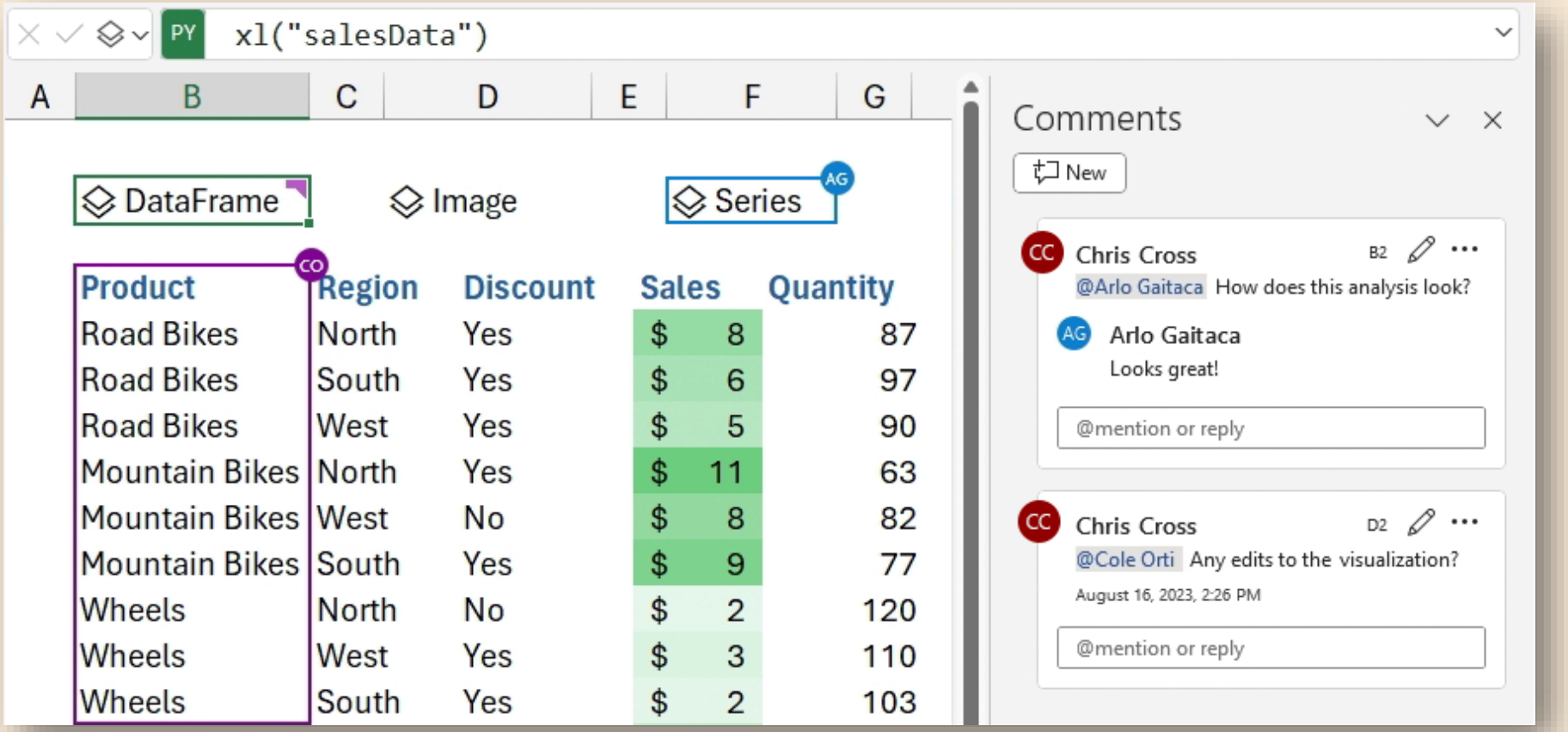
Use Advanced Analysis mode in Copilot which can generate Python code for you to start with!



# Fantastic reference for all things Python in Excel



# Easy collaboration, no setup



```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels as sm
```

# More great packages!

| | | | | |
|---|---|---|---|---|
| Astropy | mlxtend | plotnine | SciPy | tabulate |
| beautifulsoup4 | NetworkX | Prince | seaborn | TheFuzz |
| Faker | NLTK | PyTables | snowballstemmer | wordcloud |
| imbalanced-learn | NumPy | PyWavelets | squarify | |
| IPython | pandas | qrcode | statsmodels | |
| Matplotlib | Pillow | scikit-learn | SymPy | |

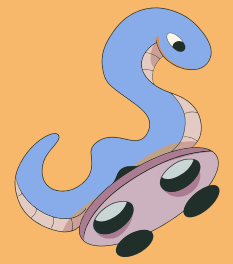
# Python in Excel tradeoffs

## Advantages

- Highly vetted, secure environments to run code on the cloud
- Interoperates with other 365 experience tools like live editing, share links, and comments
- Copilot integration

## Limitations

- Need network access to use Python and can't access internet in cloud container to get data
- Supports a small subset of the extensive Python package ecosystem
- Current data size limit (~100MB)



# Let’s look at some code!

# Setup your own Excel

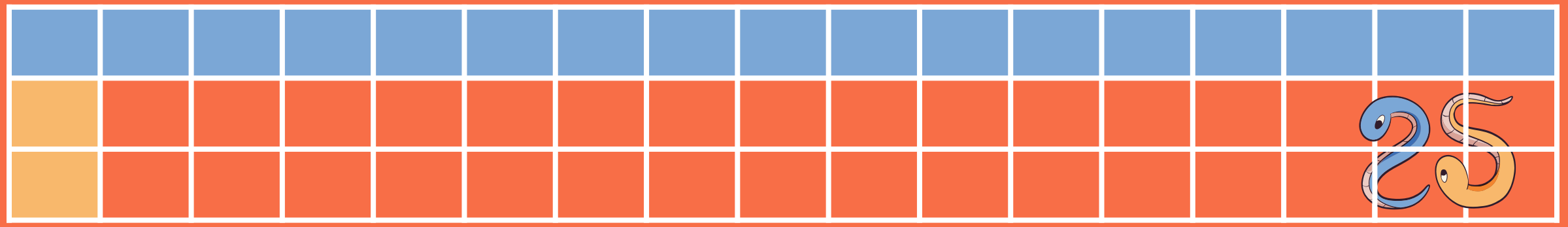
- ❑ Look at the sheet and claim a login
  - aka.ms/grid-snakes-login
- ❑ Go to https://excel.cloud.microsoft/ and login
  - Use an authenticator app to setup account, you can delete later

# Exercise: Packages in Python in Excel

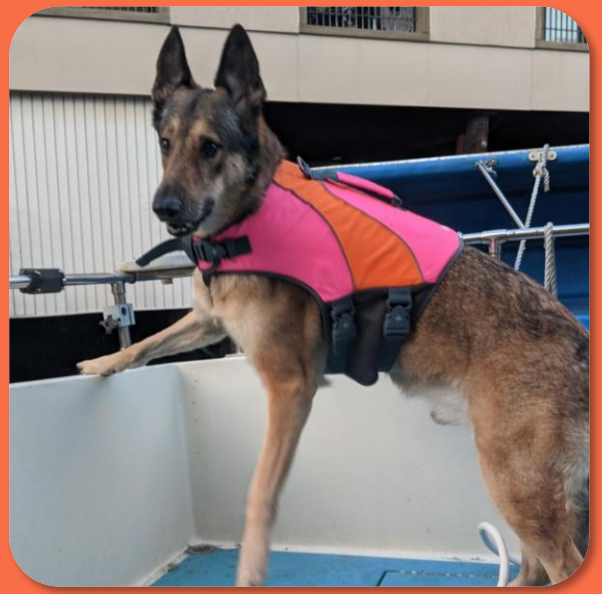
In the tutorial repo, download then upload to the excel online you just logged into `4_pyex-packages.xlsx` which has a list of the packages in the current Python in Excel environment.

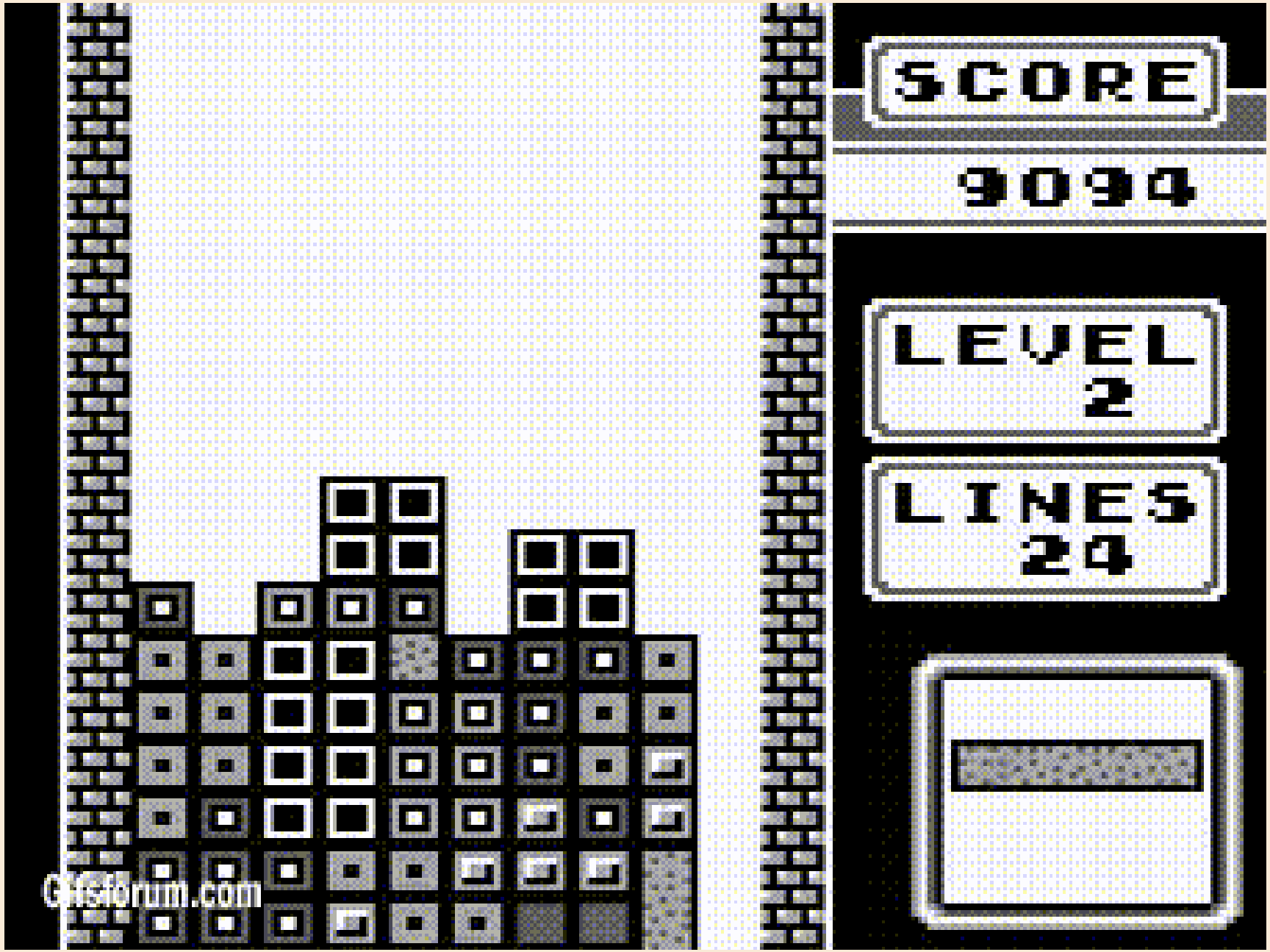
Pick one of the non-core libraries and create a sample of using that library in Excel. In 15 min find a friend or small group and share your sample!

aka.ms/grid-snakes




# =SUM(everything)




SCORE
9094
LEVEL
2
LINES
24

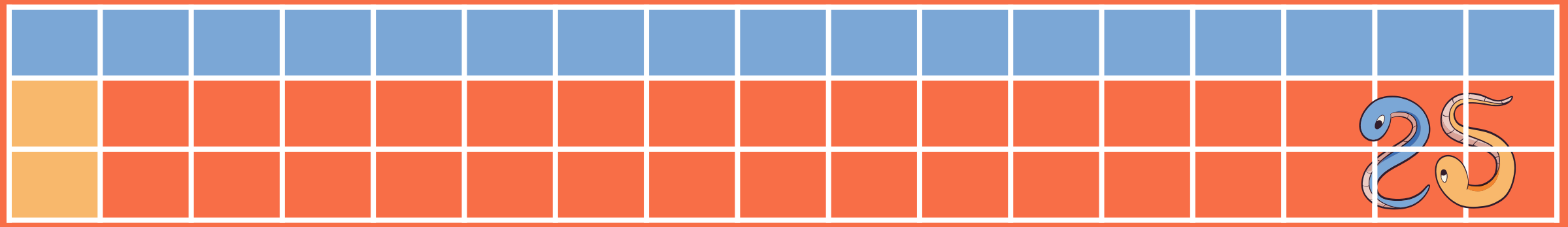

SCORE
12
EXCEL ESPORTS.
YES IT'S A THING

# Exercise: Excel E-Sports challenge

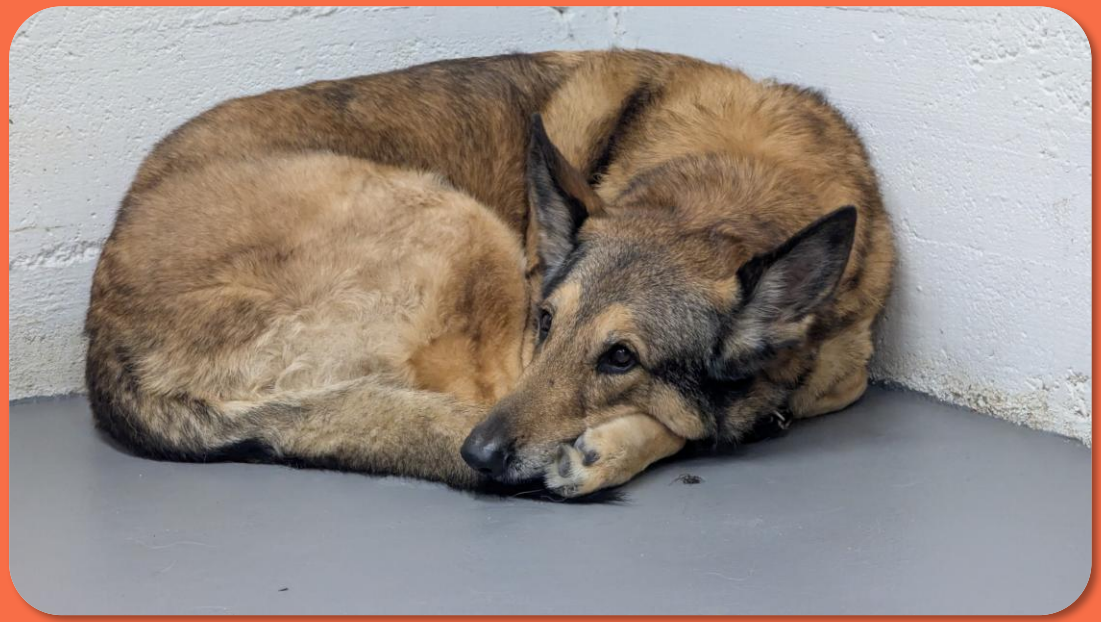
In the tutorial repo, download/open the file **`4_excel-esports-tetris.xlsx`** for instructions.

You can either work in Python and export your answers in a spreadsheet or use Python in Excel to solve it.  Highest score in the next 30 min gets a special edition prize ☺

`aka.ms/grid-snakes`




# Wrap-up



# What did we just do?

- ❑ Loaded and exported tabular data with **`pandas`**
- ❑ Created and customized spreadsheets from Python with **`openpyxl`**
- ❑ Used Python directly in a sheet with Python in Excel or Pysheets

# Thank You!

Say hi at the Microsoft booth at PyCon!
aka.ms/pycon

@crazy4pi314 | sckaiser.com