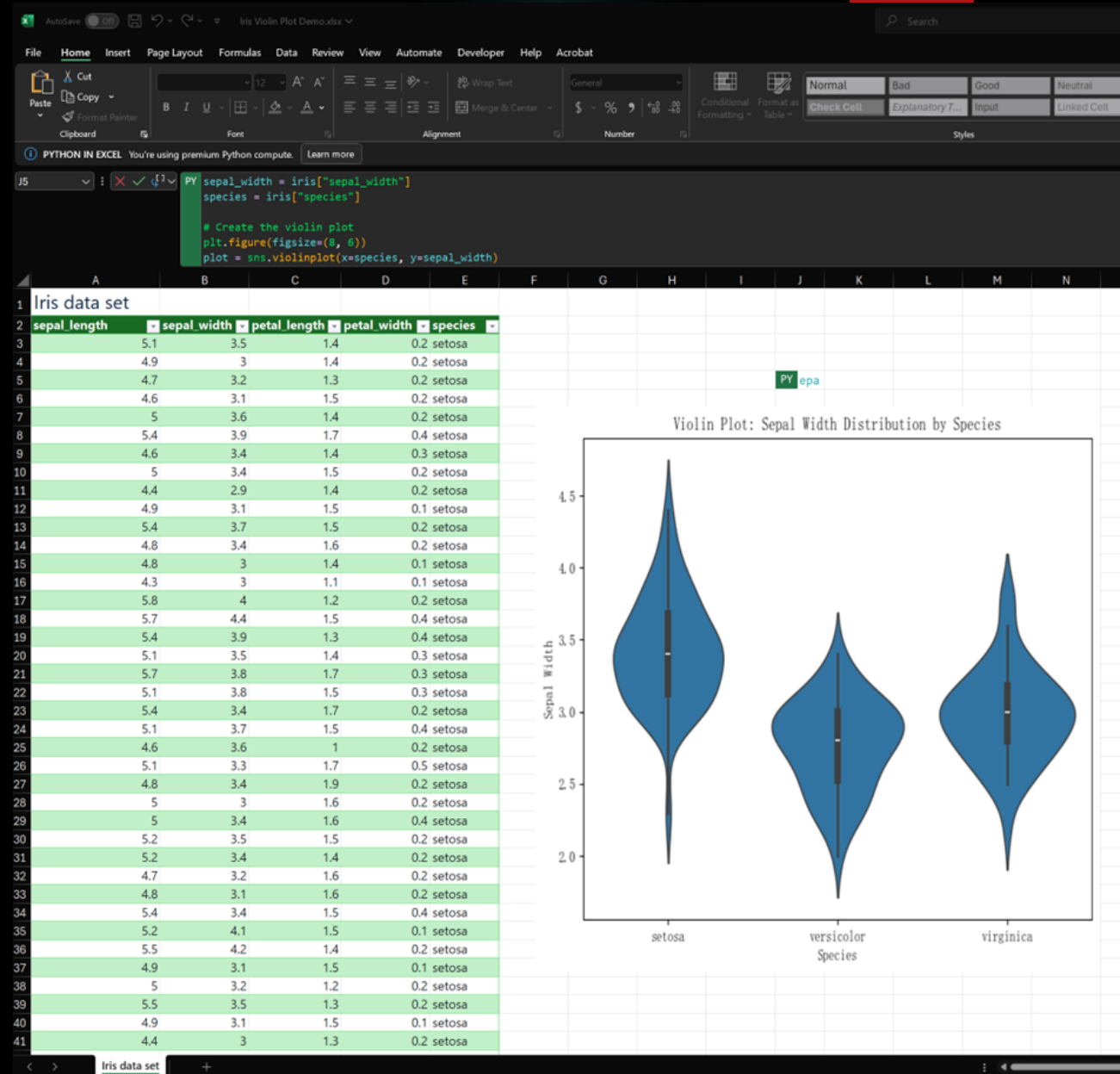
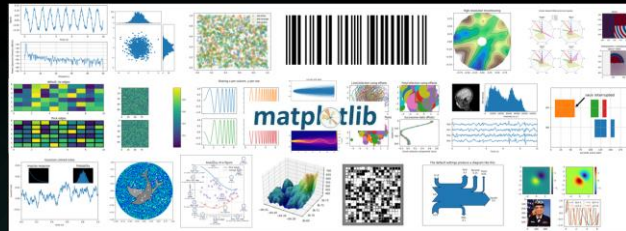


Enhance Excel with Python: Create a Violin Plot from the Iris Dataset

A STEP-BY-STEP GUIDE FOR
BUSINESS INTELLIGENCE AND
DATA SCIENCE PROFESSIONALS





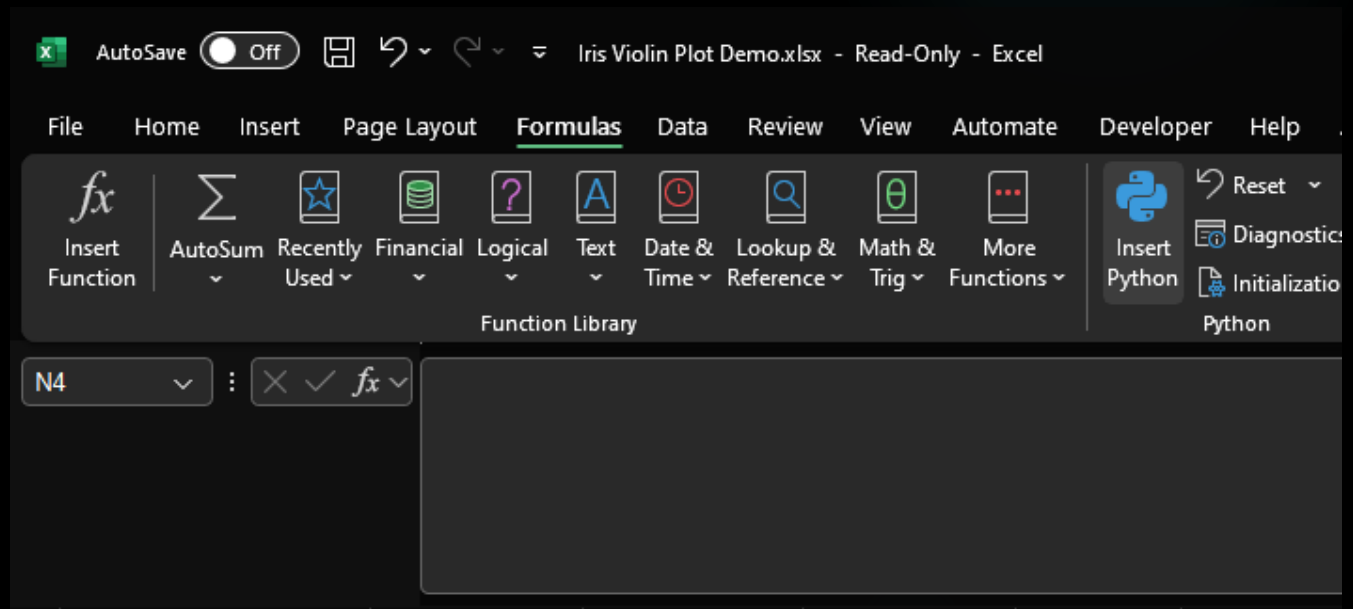
Why Use Python in Excel?

- ▶ Excel is powerful but limited in advanced visualizations.
- ▶ With Python's libraries like Matplotlib, Plotly, and Bokeh, you can extend Excel's capabilities beyond its native limits.

• Example: Create a **Violin Plot** to visualize data distribution—something not natively available in Excel.

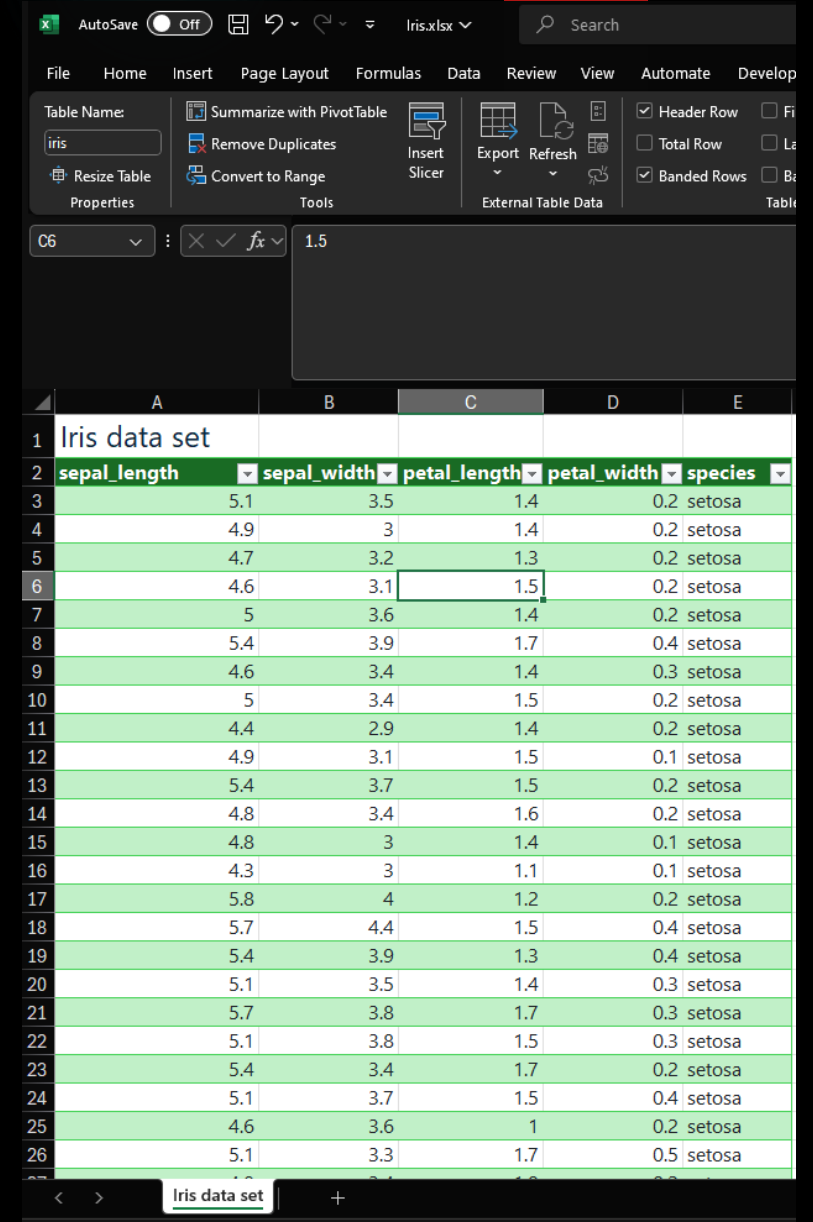
How to Get Started with Python in Excel

- ▶ Availability:
 - ▶ Python in Excel is available for Enterprise and Business users running Microsoft 365.
- ▶ To enable Python in Excel:
 - ▶ Use the Insert Python option in the Formulas tab
 - ▶ Alternatively, simply type =PY in a cell.



Download the Iris Dataset

- ▶ Download the Iris dataset
 - ▶ https://github.com/dallasdigitaldata/linkedin_excel_functions/blob/main/PythonInExcel/iris/Iris.xlsx
 - ▶ Click the download button once you access the GitHub page.



The screenshot shows a Microsoft Excel spreadsheet titled 'Iris.xlsx'. The 'Table Name' is 'iris'. The ribbon includes 'Table' and 'Tools' tabs. The 'Table' tab shows options like 'Summarize with PivotTable', 'Remove Duplicates', 'Convert to Range', 'Insert Slicer', 'Export', 'Refresh', and 'External Table Data'. The 'Tools' tab shows 'Header Row', 'Total Row', 'Banded Rows', and 'Table'. The spreadsheet displays the Iris dataset with columns: 'sepal_length', 'sepal_width', 'petal_length', 'petal_width', and 'species'. The data is organized into a table with 26 rows and 5 columns. The first row is the header, and the subsequent rows contain numerical values for sepal and petal measurements, along with the species name 'setosa'.

	A	B	C	D	E
1	Iris data set				
2	sepal_length	sepal_width	petal_length	petal_width	species
3	5.1	3.5	1.4	0.2	setosa
4	4.9	3	1.4	0.2	setosa
5	4.7	3.2	1.3	0.2	setosa
6	4.6	3.1	1.5	0.2	setosa
7	5	3.6	1.4	0.2	setosa
8	5.4	3.9	1.7	0.4	setosa
9	4.6	3.4	1.4	0.3	setosa
10	5	3.4	1.5	0.2	setosa
11	4.4	2.9	1.4	0.2	setosa
12	4.9	3.1	1.5	0.1	setosa
13	5.4	3.7	1.5	0.2	setosa
14	4.8	3.4	1.6	0.2	setosa
15	4.8	3	1.4	0.1	setosa
16	4.3	3	1.1	0.1	setosa
17	5.8	4	1.2	0.2	setosa
18	5.7	4.4	1.5	0.4	setosa
19	5.4	3.9	1.3	0.4	setosa
20	5.1	3.5	1.4	0.3	setosa
21	5.7	3.8	1.7	0.3	setosa
22	5.1	3.8	1.5	0.3	setosa
23	5.4	3.4	1.7	0.2	setosa
24	5.1	3.7	1.5	0.4	setosa
25	4.6	3.6	1	0.2	setosa
26	5.1	3.3	1.7	0.5	setosa

Insert Python Code

- ▶ In cell I3, type =PY() and paste the Python code from the example.
- ▶ See below how it changes to Python Entry mode.

The screenshot shows the Microsoft Excel interface with the 'Formulas' tab selected. The formula bar is in 'Python Entry' mode, indicated by a green vertical bar on the left and the text 'PY' in the input field. A tooltip above the formula bar reads 'PY(Use Ctrl+Enter to commit Python code)'. The worksheet contains a table of Iris data with columns: sepal_length, sepal_width, petal_length, petal_width, and species. The data is as follows:

	sepal_length	sepal_width	petal_length	petal_width	species
1	Iris data set				
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
6	5	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa

Insert Python Code

- ▶ Tip: You can type the code manually if you want to practice coding in Excel.
- ▶ Use Ctrl + Enter to run the cell

```
[ ] PY import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Load data from the Iris dataset in the Excel worksheet using xl() function
iris = xl("iris[#All]", headers=True)

# Select the sepal width and species columns for the violin plot
sepal_width = iris["sepal_width"]
species = iris["species"]

# Create the violin plot
plt.figure(figsize=(8, 6))
plot = sns.violinplot(x=species, y=sepal_width)

# Add title and labels
plt.title('Violin Plot: Sepal Width Distribution by Species')
plt.xlabel('Species')
plt.ylabel('Sepal Width')

# Show the plot
plt.show()
```

PY(Use Ctrl+Enter to commit Python code)

Display the Python Plot

- ▶ Right click on the new python image and click Display Plot over Cells

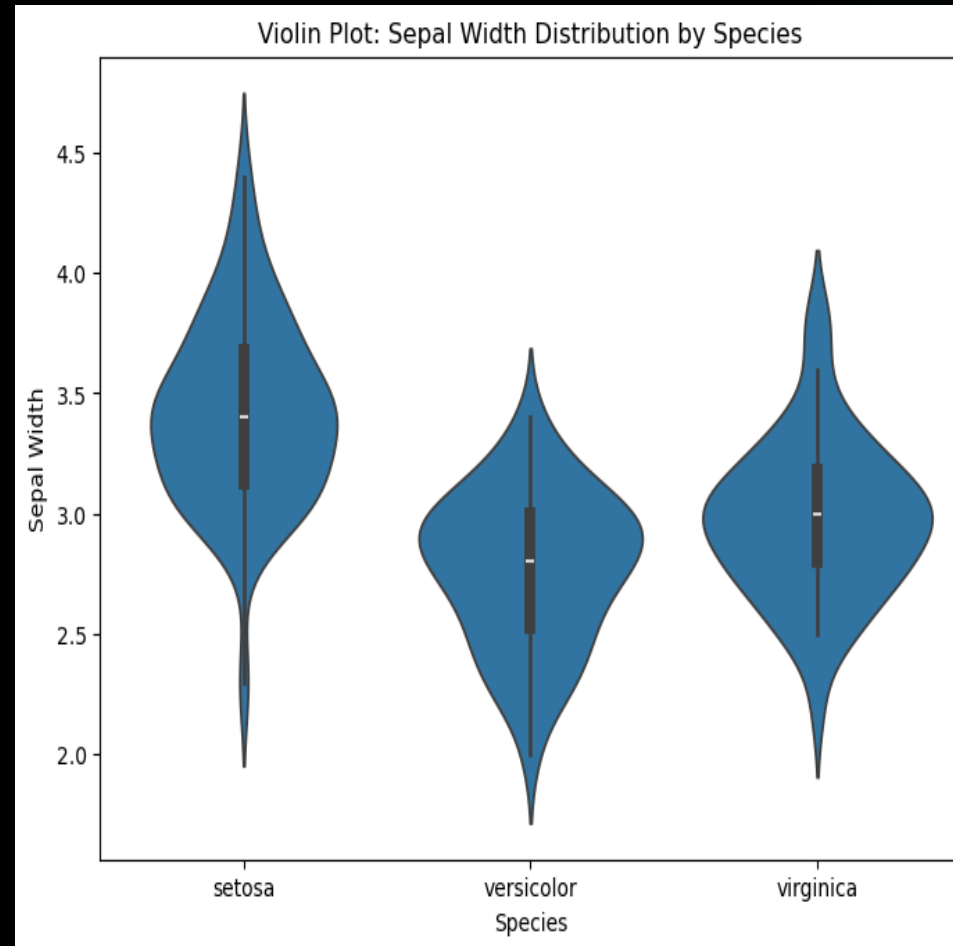
The screenshot shows the Microsoft Excel interface with a Python script in the formula bar. The script imports matplotlib.pyplot as plt, seaborn as sns, and pandas as pd. It then loads the Iris dataset into an Excel worksheet using the xl() function. The worksheet contains the following data:

	sepal_length	sepal_width	petal_length	petal_width	species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3	1.4	0.1	setosa
14	4.3	3	1.1	0.1	setosa
15	5.8	4	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa
20	5.1	3.8	1.5	0.3	setosa
21	5.4	3.4	1.7	0.2	setosa
22	5.1	3.7	1.5	0.4	setosa
23	4.6	3.6	1	0.2	setosa
24	5.1	3.3	1.7	0.5	setosa
25	4.8	3.4	1.9	0.2	setosa

The right-click context menu is open, showing options like Cut, Copy, Paste Options, Smart Lookup, Show Data Type Card, Picture in Cell, Insert..., Delete..., Clear Contents, Python Output, Display Plot over Cells (highlighted), Show Diagnostics, Quick Analysis, Filter, Sort, Get Data from Table/Range..., New Comment, New Note, Format Cells..., Pick From Drop-down List..., Define Name..., Link, Open Hyperlink, and Link to this Range.

Plot in all its Glory!

- Now we can see our Violin Plot



Conclusion and Next Steps

- ▶ By integrating Python into Excel, you open up a range of new possibilities for data analysis and visualization.
- ▶ Download the Iris dataset and try more plots like boxplots or histograms.
- ▶ Explore other Python libraries (e.g., plotly) for interactive visualizations. Experiment with different datasets to unlock Excel's full potential with Python.

