
Synthetic data in Excel

A walkthrough using Faker
in Python in Excel.

Three situations where this comes up.

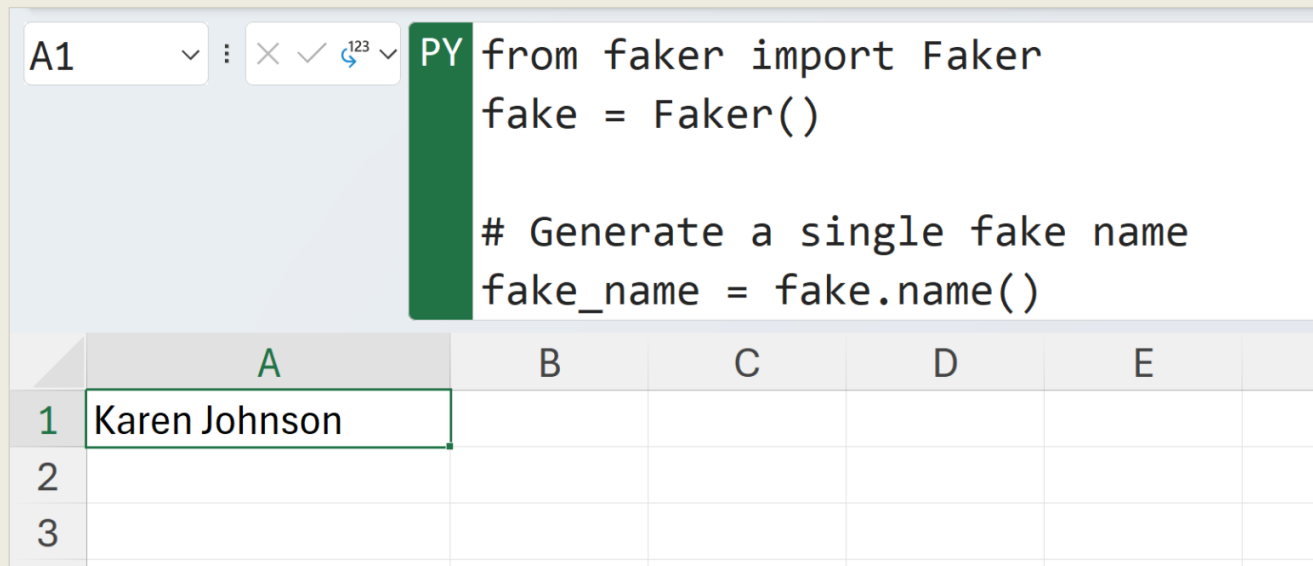
1. Building a demo or training file before you have access to the real data.
2. Stress-testing a workbook to see what breaks once row counts get large.
3. Feeding AI tools (Copilot, Claude, anything) without pasting real customer or employee information into a prompt.

Faker, the Python library, solves all three. It ships inside Python in Excel, so you don't need to install anything.

STEP 1 OF 5

Step 1. Import Faker and create an instance.

The first line brings the Faker class into your session. The second creates an instance you'll call methods on. `fake.name()` returns one randomly-generated name.



```
A1 PY from faker import Faker
fake = Faker()

# Generate a single fake name
fake_name = fake.name()
```

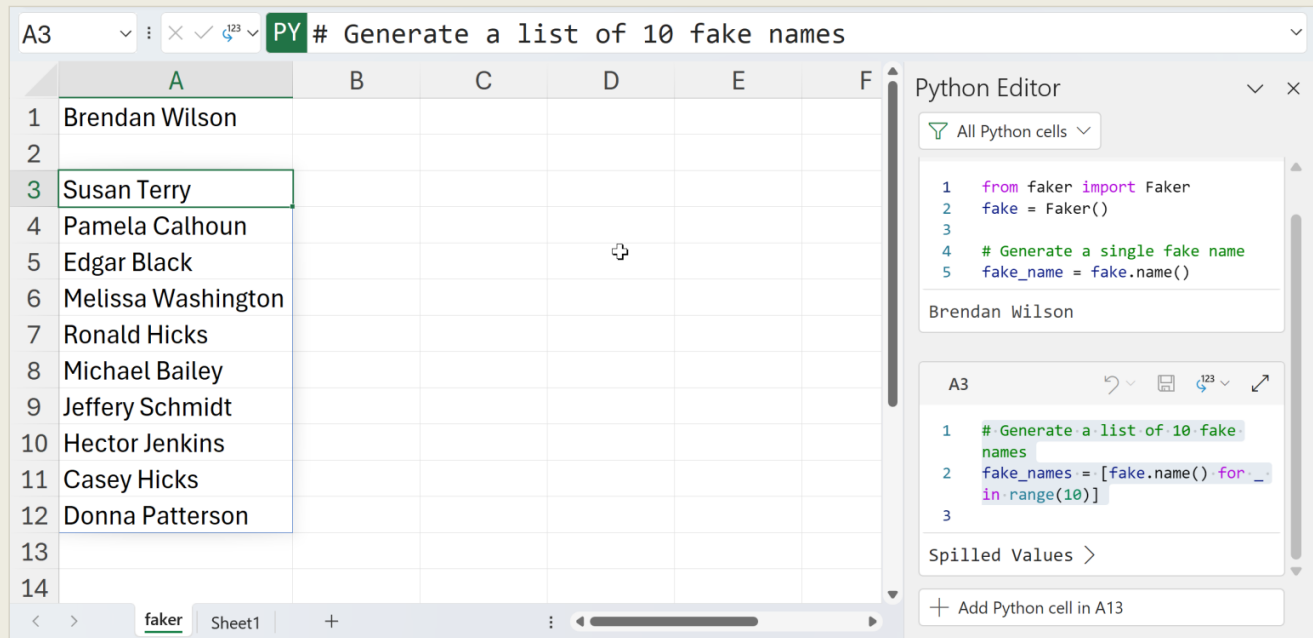
	A	B	C	D	E
1	Karen Johnson				
2					
3					

■ Two lines of setup, then a method call. That's it.

STEP 2 OF 5

Step 2. Generate many names with list comprehension.

One name is rarely enough. A list comprehension calls `fake.name()` repeatedly inside a single expression, and `range(10)` controls how many you want.



The screenshot shows a Google Sheets interface with a Python Editor on the right. The spreadsheet has a single cell (A3) containing a list of 10 names: Brendan Wilson, Susan Terry, Pamela Calhoun, Edgar Black, Melissa Washington, Ronald Hicks, Michael Bailey, Jeffery Schmidt, Hector Jenkins, Casey Hicks, and Donna Patterson. The Python Editor shows the following code:

```
1 from faker import Faker
2 fake = Faker()
3
4 # Generate a single fake name
5 fake_name = fake.name()
```

The output of the first cell is "Brendan Wilson". Below the code editor, the second cell (A3) contains the following code:

```
1 # Generate a list of 10 fake
  names
2 fake_names = [fake.name() for _
  in range(10)]
3
```

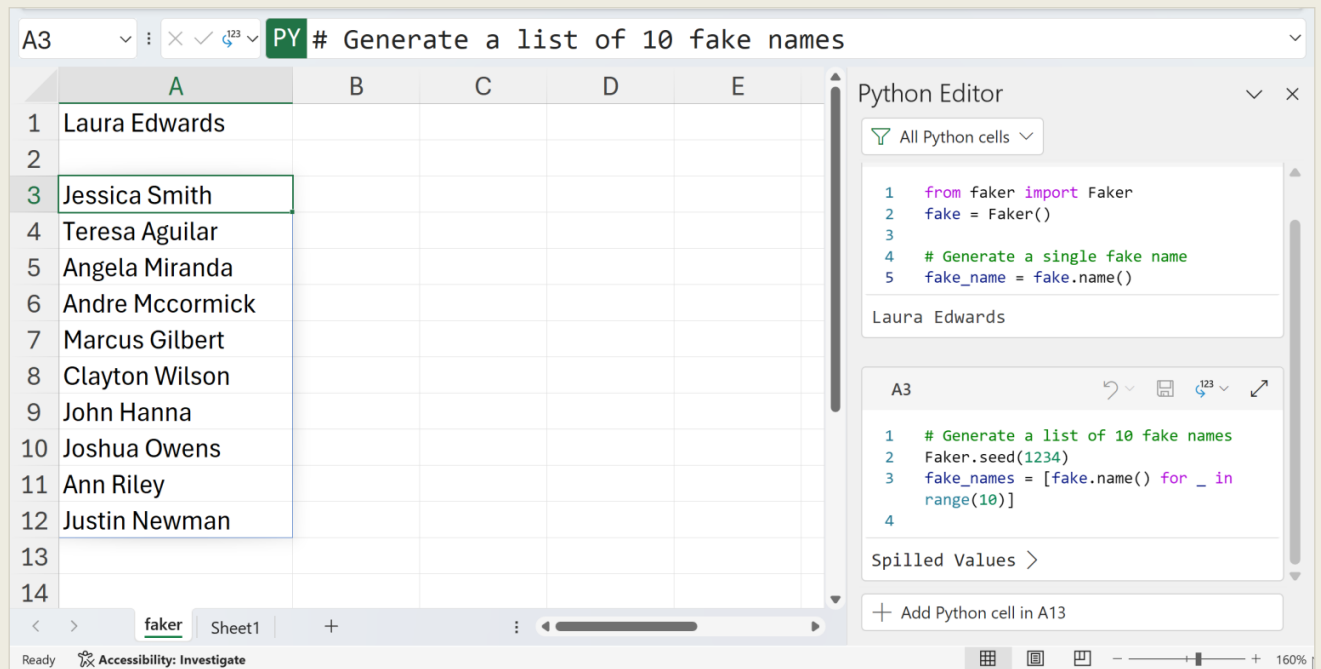
The output of the second cell is "Spilled Values >".

■ Re-run the cell and you'll get ten different names each time.

STEP 3 OF 5

Step 3. Lock the output with a random seed.

`Faker.seed(1234)` tells Faker to use the same starting point every time. Now the same ten names come back on every run, which matters when you want demos or training files to stay stable across sessions. Pick any integer and stick with it.



The screenshot shows a Jupyter Notebook interface. The active cell is a Python cell with the following code:

```
# Generate a list of 10 fake names
```

```
1 from faker import Faker
2 fake = Faker()
3
4 # Generate a single fake name
5 fake_name = fake.name()
```

The output of the cell is a list of 10 fake names, displayed in a table with columns A through E and rows 1 through 14:

	A	B	C	D	E
1	Laura Edwards				
2					
3	Jessica Smith				
4	Teresa Aguilar				
5	Angela Miranda				
6	Andre McCormick				
7	Marcus Gilbert				
8	Clayton Wilson				
9	John Hanna				
10	Joshua Owens				
11	Ann Riley				
12	Justin Newman				
13					
14					

The Python Editor on the right shows the code and the output: Laura Edwards. Below the editor, the cell content is repeated, and the output is shown as a list of 10 fake names.

STEP 4 OF 5

Step 4. Combine multiple columns into a DataFrame.

When you need a table instead of a list, build a Python dictionary where each key is a column name and each value is a list of fake data. Pass it to `pd.DataFrame()` and you get a structured table that spills into Excel like any other Python output.

The screenshot shows an Excel spreadsheet with a Python cell (A14) that has spilled its output into columns A and B. The output is a table with 10 rows of fake names and emails. The Python Editor on the right shows the code used to generate this data.

Name	Email
Mr. David Mahoney	fduran@example.com
Joseph Smith	yalvarez@example.com
Abigail Thomas	scottnorman@example.com
James Brown	joshua26@example.com
Carly Grant	crosbydakota@example.com
Timothy Wise	qroth@example.com
Brenda Torres	michaelpark@example.org
Donald Drake	teresa57@example.net
Christopher Blevins	millerjason@example.com
Roy Williams	robinsonmaria@example.net

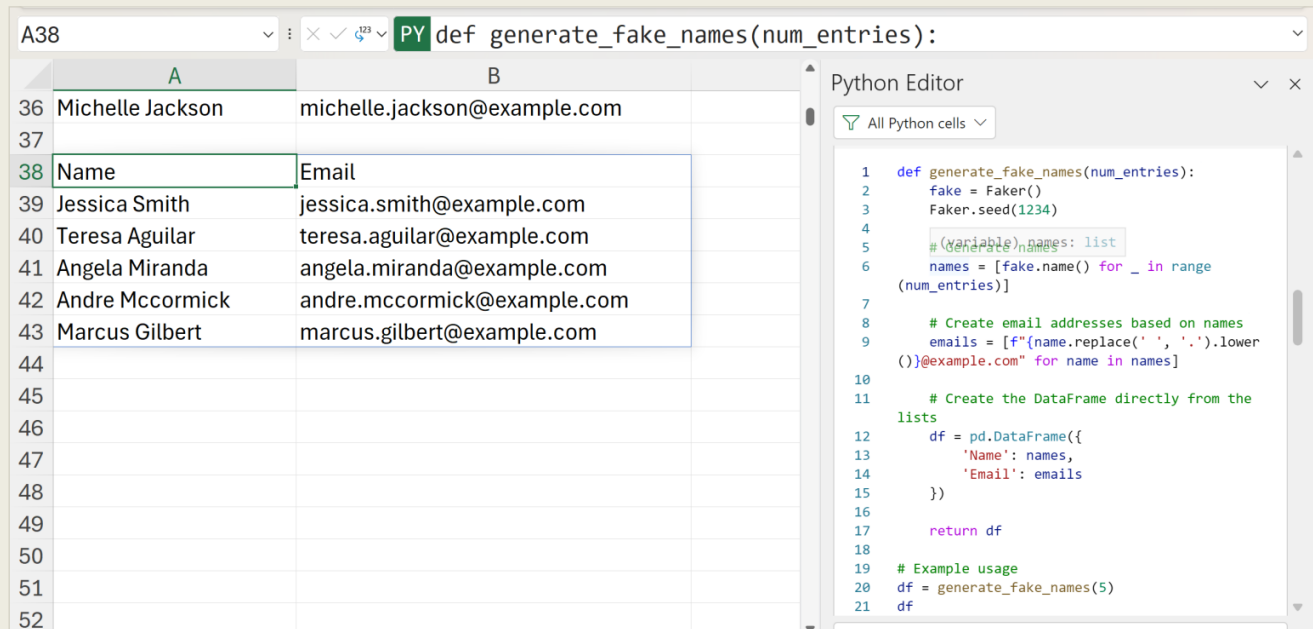
```
1 # Generate a list of 10 fake names
2 Faker.seed(1234)
3 fake_names = [fake.name() for _ in range(10)]
4
```

```
1 data = {
2     'Name': [fake.name() for _ in range(10)],
3     'Email': [fake.email() for _ in range(10)]
4 }
5 df = pd.DataFrame(data)
```

STEP 5 OF 5

Step 5. Wrap it in a function for reuse.

If you'll regenerate this dataset at different sizes, package the logic into a function. The Faker instance and the seed live inside, so `generate_fake_names(5)` and `generate_fake_names(500)` both return reproducible output.



The screenshot shows a Jupyter Notebook interface. On the left, a DataFrame is displayed with two columns: 'Name' and 'Email'. The data is as follows:

Name	Email
Michelle Jackson	michelle.jackson@example.com
Jessica Smith	jessica.smith@example.com
Teresa Aguilar	teresa.aguilar@example.com
Angela Miranda	angela.miranda@example.com
Andre McCormick	andre.mccormick@example.com
Marcus Gilbert	marcus.gilbert@example.com

On the right, the Python Editor shows the function definition for `generate_fake_names`:

```
1 def generate_fake_names(num_entries):
2     fake = Faker()
3     Faker.seed(1234)
4
5     # Generate names: list
6     names = [fake.name() for _ in range
7               (num_entries)]
8
9     # Create email addresses based on names
10    emails = [f"{name.replace(' ', '.').lower
11              ()}@example.com" for name in names]
12
13    # Create the DataFrame directly from the
14    lists
15    df = pd.DataFrame({
16        'Name': names,
17        'Email': emails
18    })
19
20    return df
21
22 # Example usage
23 df = generate_fake_names(5)
24 df
```

■ Call it anywhere in the workbook with any row count.

ALTERNATIVE

Or: ask Copilot to write the script for you.

If you'd rather skip writing the Python yourself, Copilot in Excel can generate the script. Describe the columns you want, the row count, any distributions, and a random seed. Copilot returns code you can paste straight into a PY() cell.

EXAMPLE PROMPT

Generate a Python script for a 500-row employee performance dataset. Columns: Employee ID, Department, Performance Score (normal, mean 70, SD 10), Salary (lognormal). Departments: Sales, HR, IT, Marketing, Finance. Set `Faker.seed(1234)`.

WHERE TO GO NEXT

That's the full Faker workflow.

There's more on the blog: setting the locale for international data, when to swap Faker for NumPy on quantitative variables, and a longer Copilot prompt for bigger datasets.

READ THE FULL POST

stringfestanalytics.com/blog