

Tax Simulation & Policy Sandbox Results

Name: Paul Andy **Date:** 27 December 2025

1. Overview

In preparation for the Family Resources Survey work, I expanded my initial microsimulation script into a functional Desktop Application.

My objectives were:

1. **Validation:** Ensure the Python logic matches the Stata baseline 100%.
2. **Tooling:** Build a user-friendly interface ([wxPython](#)) to allow non-coders to test tax policy changes dynamically.

2. Tax Parameters (2024/25 Rules)

The model currently uses the standard FY 24/25 parameters:

- **Personal Allowance:** £12,570 (tapers by £1 for every £2 earned over £100k).
- **Basic Rate:** 20% (up to £50,270).
- **Higher Rate:** 40% (up to £125,140).
- **Additional Rate:** 45% (over £125,140).
- **National Insurance:** 8% main rate / 2% upper rate.

3. Baseline Validation (Stata vs. Python)

I generated a control dataset in Stata to serve as the "correct" answer. I then ran the same scenario in my Python App to verify accuracy.

A. Stata Output (Control) For a high earner (£150,000 gross), the Stata script calculates a **Net Income of £91,286.40**.

StataNow/MP 19.5

Log View

Results

(2 real changes made)

```
.  
. gen net_income = income - tax_due - ni_due  
. format income tax_due ni_due net_income %9.2f  
. list name income tax_due ni_due net_income, separator(5)
```

	name	income	tax_due	ni_due	net_inve
1.	Low Earner	12000.00	0.00	0.00	12000.00
2.	Basic Rate	25000.00	2486.00	994.40	21519.60
3.	Threshold	50270.00	7540.00	3016.00	39714.00
4.	Higher Rate	80000.00	19432.00	3610.60	56957.40
5.	Add. Rate	150000.00	53703.00	5010.60	91286.40

```
.  
end of do-file
```

(Caption: Stata terminal showing the control figures for the 5 mock individuals)

B. Python App Output (Test) I ran the exact same £150,000 scenario through the new Python GUI. The figures match the Stata baseline exactly.

The image shows a Python GUI titled "IPPR Policy Sandbox". It has two main sections: "Policy Inputs" and "Simulation Results".

Policy Inputs:

- Gross Income (£):** A text input field containing "150000".
- Adjust Basic Tax Rate (%):** A slider control ranging from 0 to 50, with a white knob positioned at 20.

Simulation Results:

Basic Rate Used: 20%

Income Tax: £53,703.00

National Insurance: £5,010.60

Net Income: £91,286.40

At the bottom, there is a "Run Simulation" button.

(Caption: Python GUI confirming £91,286.40 Net Income, matching the Stata model)

4. Policy Sandbox (New Functionality)

To demonstrate the "Desirable" requirement for **wxPython**, I built a graphical interface that allows users to modify tax parameters in real-time.

Scenario: "What if the Basic Tax Rate increased from 20% to 28%?"

Simulation Result:

- **Gross Income:** £150,000
- **New Basic Rate:** 28%
- **Impact:** Net Income drops from £91,286.40 to **£88,270.40**.

The image shows a software window titled "IPPR Policy Sandbox". It has a dark theme. Under the "Policy Inputs" section, there is a text box for "Gross Income (£)" containing the value "150000". Below it is a slider for "Adjust Basic Tax Rate (%)" with a range from 0 to 50, and a white knob positioned at 28. The "Simulation Results" section displays the following information: "Basic Rate Used: 28%", "Income Tax: £56,719.00", "National Insurance: £5,010.60", a dashed line separator, and "Net Income: £88,270.40". At the bottom center is a button labeled "Run Simulation".

Category	Value
Gross Income (£)	150000
Basic Rate Used (%)	28%
Income Tax (£)	£56,719.00
National Insurance (£)	£5,010.60
Net Income (£)	£88,270.40

(Caption: Dynamic policy testing showing the immediate financial impact of an 8% tax rise)

5. Conclusion

- **Accuracy:** The Python logic is validated against the Stata script.
- **Usability:** The GUI allows for rapid policy testing without touching the code.
- **Readiness:** The environment (Pandas, wxPython, Stata) is fully operational for the January start.