Scientific Computing in Python Lecture 1

Python and sampled measurements

Lecture notes

We begin with a temperature record. The calculation is elementary, so that we can examine exactly what the program represents: a value, its physical unit, the time at which it was observed, and the operations applied to it. A program is a sequence of state changes. An equation is a relation between mathematical quantities. Learning to distinguish them is the first step in translating a scientific calculation into Python.

1 Values and assignment

Slides for this section

A Python name refers to a value. Assignment evaluates the expression on the right and associates the resulting object with the name on the left. It does not create an equation that Python will solve, nor a permanent dependence between names. Integers represent whole numbers exactly within available memory. Floating-point values represent a finite set of real numbers. Strings are text; a quoted numerical label is not a number.

```
a = 4
b = 2*a
a = 7
```

The final values are $a = 7$ and $b = 8$. The second line was evaluated before the third.

Expression evaluation

```
x = 3
print(x**2 + 1)  # 10
print((x + 1)**2)  # 16
```

Exponentiation uses `**`. Parentheses determine which quantity is squared.

2 Paired observations

Slides for this section

Write the observations as ordered pairs (t_i, T_i), where the index identifies an observation. The two lists must have equal length and corresponding entries must refer to the same measurement. An increasing time axis is part of the data definition. Python indexing starts at zero, while a negative index counts backward from the end. A list preserves order; it does not attach a physical unit to its entries.

$$(t_i, T_i) = (0, 80), (60, 62), (120, 48), (180, 38), (240, 31).$$

Time is in seconds and temperature in degrees Celsius.

```
time_seconds = [0, 60, 120, 180, 240]
temperature_c = [80, 62, 48, 38, 31]
```

Indexing the endpoints

```
t0 = time_seconds[0]
t1 = time_seconds[-1]
T0 = temperature_c[0]
T1 = temperature_c[-1]
```

The endpoint calculation uses $(0, 80)$ and $(240, 31)$. Interior observations remain relevant to the plotted record.

3 Units and affine changes of scale

Slides for this section

A number acquires physical meaning through its unit. Converting seconds to minutes divides the numerical value by 60. Converting Celsius to kelvin adds 273.15. The latter is an affine transformation: it changes the origin but preserves temperature differences. More generally, if $y = ax + b$, then $y_j - y_i = a(x_j - x_i)$. The offset cancels; the scale factor remains.

$$t_{\mathrm{min}} = \frac{t_{\mathrm{s}}}{60}, \qquad T_K = T_C + 273.15.$$

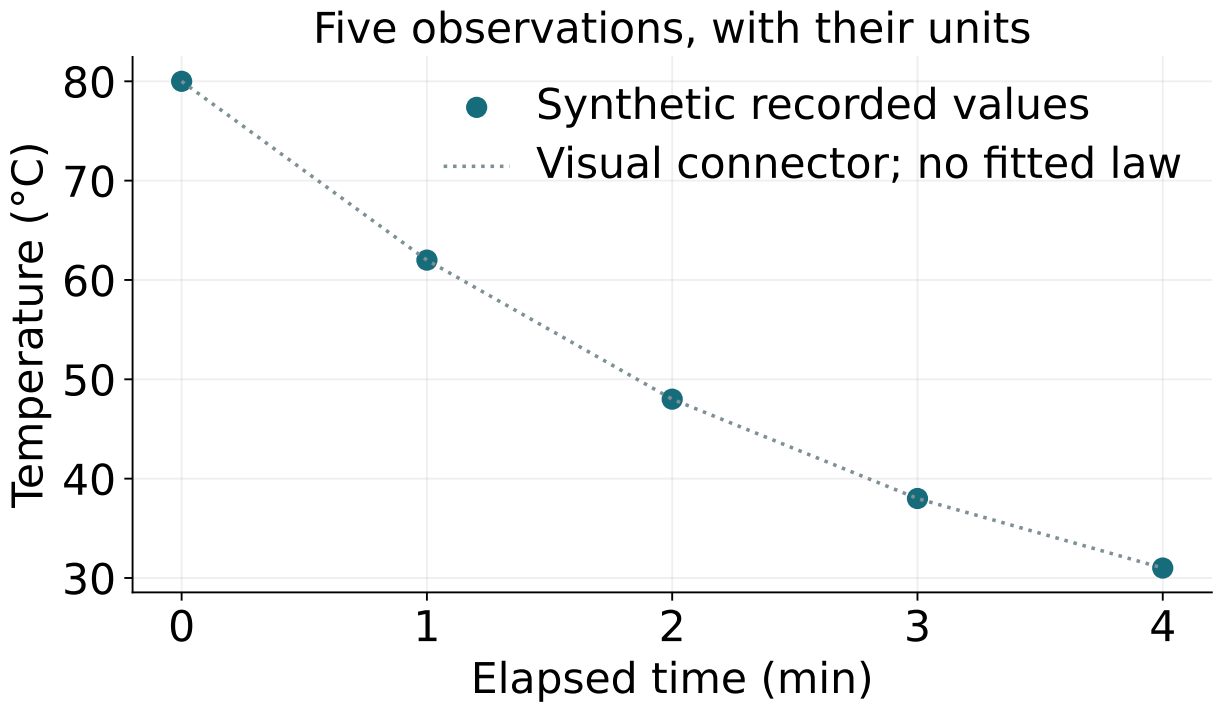
$$\Delta T_K = (T_{C,1} + 273.15) - (T_{C,0} + 273.15) = \Delta T_C.$$

The same change in two units

$$\Delta T = 31 - 80 = -49 \ ^\circ\mathrm{C} = -49 \ \mathrm{K}.$$

$$\Delta t = 240 \ \mathrm{s} = 4 \ \mathrm{min}.$$

Temperature differences have equal numerical values in Celsius and kelvin. Absolute temperatures do not.



Five supplied observations, with dotted visual connectors only; no continuous cooling law is fitted.

4 Average rate of change

Slides for this section

The signed average rate is a ratio of differences, not the final temperature divided by elapsed time. It is the slope of the secant through the endpoint observations. A negative value means that the measured quantity decreased over the interval. The unit in the denominator is essential: the numerical rate changes when the time unit changes.

$$\bar{r} = \frac{T_1 - T_0}{t_1 - t_0}, \qquad t_1 > t_0.$$

$$\bar{r} = \frac{-49}{4} = -12.25\ ^\circ\mathrm{C/min}.$$

This is an interval average, not an instantaneous derivative.

A direct Python calculation

```
elapsed_min = (time_seconds[-1]-time_seconds[0])/60
change_c = temperature_c[-1]-temperature_c[0]
rate_c_per_min = change_c/elapsed_min
```

A zero elapsed time would make the quotient undefined. The denominator must be checked before division.

5 Floating-point arithmetic

Slides for this section

Most decimal fractions do not have a finite binary expansion. Consequently a floating-point calculation usually carries a rounding error, even when the mathematical operation is exact. Equality of integer counters is appropriate; equality of independently computed floating-point expressions often requires a scale-dependent tolerance. A comparison tolerance should be small compared with the physical effect being measured.

```
0.1 + 0.2 == 0.3  # False in binary floating point
```

A useful comparison has the form

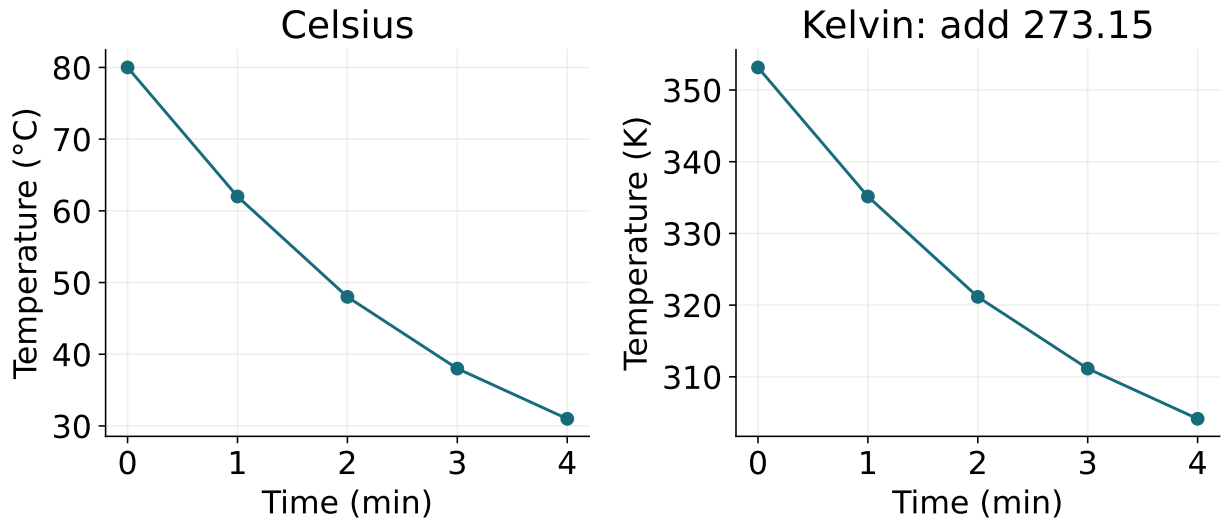
$$|a - b| \leq \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}}|b|.$$

Scale in an error tolerance

The absolute term controls comparisons near zero. The relative term scales with the reference magnitude.

$$|60r_{\text{s}} - r_{\text{min}}| \leq 10^{-12}$$

is a numerical unit-conversion check for this small example. It is not a statement about thermometer accuracy.



Celsius and kelvin panels represent the same values; adding 273.15 changes absolute numbers while preserving differences.

6 Plots and reproducible execution

Slides for this section

Plot the pairs (t_i, T_i) with labeled axes. A line segment between observations is a graphical interpolation unless a continuous model has been specified. Restarting a notebook and executing cells in order removes dependence on hidden earlier assignments. The supplied figure helper recreates the documented baseline;

editing a separate list does not alter that helper's internal data. Use the editable plotting cell when changing the record.

Five observations determine five plotted points. They do not determine the complete function $T(t)$.

If $g(t_i) = 0$ at every sampled time, both $T(t)$ and $T(t) + g(t)$ fit those observations.

A family of indistinguishable interpolants

For any constant c,

$$g(t) = c \prod_{i=0}^{4} (t - t_i)$$

vanishes at all five sample times.

Additional assumptions are required to infer behavior between observations.

Exercises

H1-1. Values, text and execution

New practice; 10 feedback points.
a An experiment records 150 seconds. Convert it to minutes and state the unit cancellation.

b x is assigned 150; y is assigned x/60; x is then assigned 240. State y before and after rerunning its assignment.

c Explain why the text value "150" should not be silently treated as a numerical duration. Describe a deliberate correction.

H1-2. A different temperature record

Synthetic readings are 50 and 32 degrees Celsius at times 0 and 90 seconds. Total: 10 points.
a Give both temperatures in kelvin.

b Compute the endpoint change and average rate in degrees Celsius per minute.

c Must the temperature at 45 seconds be 41 degrees Celsius? Explain.

H1-3. A plot that someone else can read

A file contains time_seconds = [0,30,60] and readings_C = [70,55,46]. Total: 10 points.
a State the horizontal values if the plot uses minutes, and write both axis labels.

b A colleague changes only the time-axis label from seconds to minutes. Explain the error and a check.

c What goes wrong if the temperature list is reversed while time order stays fixed?

H1-4. Prepare a reproducible handoff

A notebook runs only when its author executes cells in a special order. Total: 10 points.
a Give a sequence that makes its dependencies visible.

b Describe one execution check and two data checks.

c Write one supported claim and one unresolved question for a two-endpoint temperature record.

References

Original Python Lecture 0. PDF pages 15–21 and 38–43.
https://zanahmad.com/files/scientific-python/Lecture_0__Intro_to_Python.pdf

Python source materials: Phillip Kerger (2023). Course adaptation and additional exposition: Zan Ahmad.