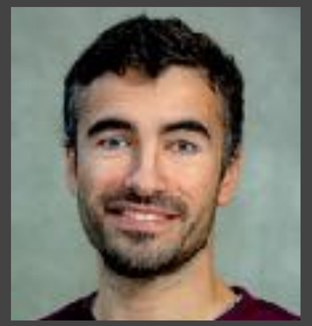
5. Scientific Guide for Reliable Energy Experiments

Sustainable Software Engineering
CS4295



Luís Cruz
L.Cruz@tudelft.nl

1. Scientific guide for energy measurements
2. Energy consumption data analysis

Energy tests are **flaky** ?

- Multiple runs might yield different results
- There are many **confounding factors** that need to be ~~controlled~~/**minimized**.

Zen mode 🧘🏾‍♀️

- **Close all applications.**
- **Turn off notifications.**
- **Only the required hardware** should be connected (avoid USB drives, external disks, external displays, etc.).
- **Kill unnecessary services** running in the background (e.g., web server, file sharing, etc.).
- If you do not need an internet or intranet connection, **switch off your network.**
- Prefer **cable over wireless** – the energy consumption from a cable connection is more stable than from a wireless connection.

Freeze and report your settings 🥶

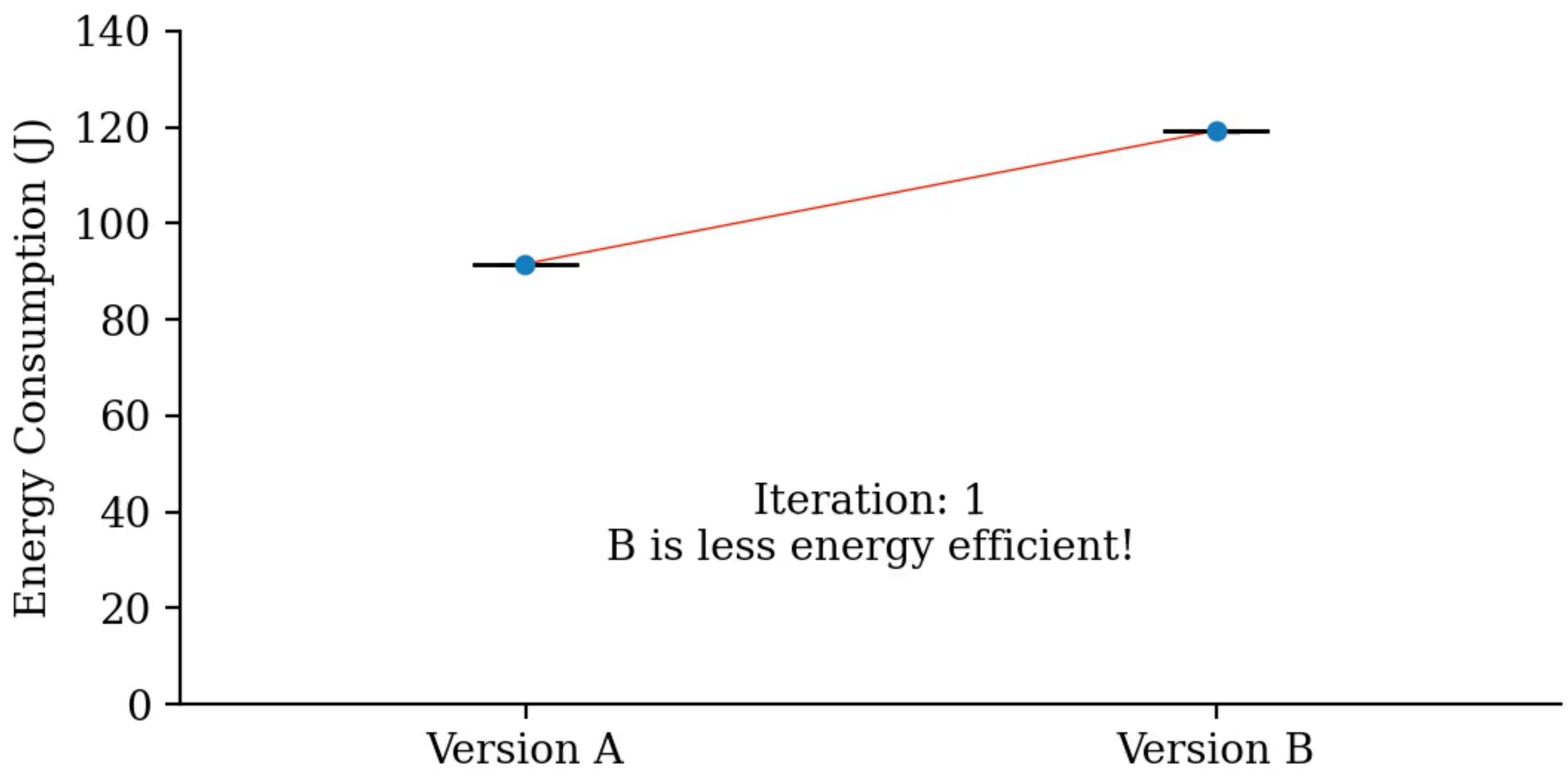
- Fix display brightness; **switch off auto brightness**
- If Wifi is on, it should always be on, connected to the same network/ endpoint….

Warm-up 📶

- Energy consumption is highly affected by the **temperature of your hardware**.
- **Higher the temperature** -> higher the resistance of electrical conductors -> -> higher dissipation -> **higher energy consumption**
- The first execution will appear more efficient because the hardware is still cold.
- Run a **CPU-intensive task** before measuring energy consumption. E.g., Fibonacci sequence. At least 1min; 5min recommended.
  -

Repeat 🔁

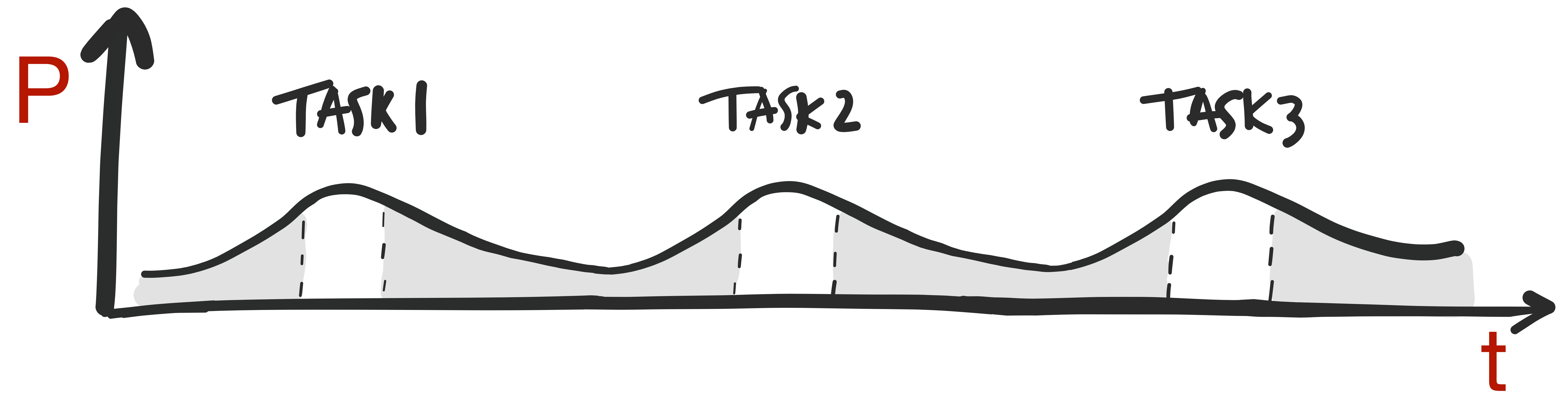
- The best way to make sure a measurement is **valid** is by **repeating** it.
- In a scientific project, the **magic number is 30.**


Energy Consumption (J)
140
120
100
80
60
40
20
0
Iteration: 1
B is less energy efficient!
Version A
Version B

Rest ⏸

- It is **common practice** to do a pause/sleep between executions/measurements.
- Prevent **tail energy consumption** from previous measurements. ?
- Prevent collateral tasks of previous measurement from affecting the next measurement.
- There is no golden rule but **one minute** should be enough. It can be more or less depending on your **hardware** or the **duration** of your energy test.

Tail Energy Consumption



Shuffle 🔀

- It is not a mystery that energy consumption depends on so many factors that it is impossible to control all of them.
- If you run 30 executions for version A and another batch for version B:
  - **External conditions that change over time** will have a **different bias** in the 2 versions (e.g., room temperature changes).
  - If you shuffle, you reduce this risk.

Keep it cool 🌡️

- Always make sure there is a **stable room temperature**.
- Tricky because, some times, experiments may have to run over a few days.
- If you cannot control room temperature: **collect temperature data** and **filter out** measurements where the room temperature is clearly deviating.

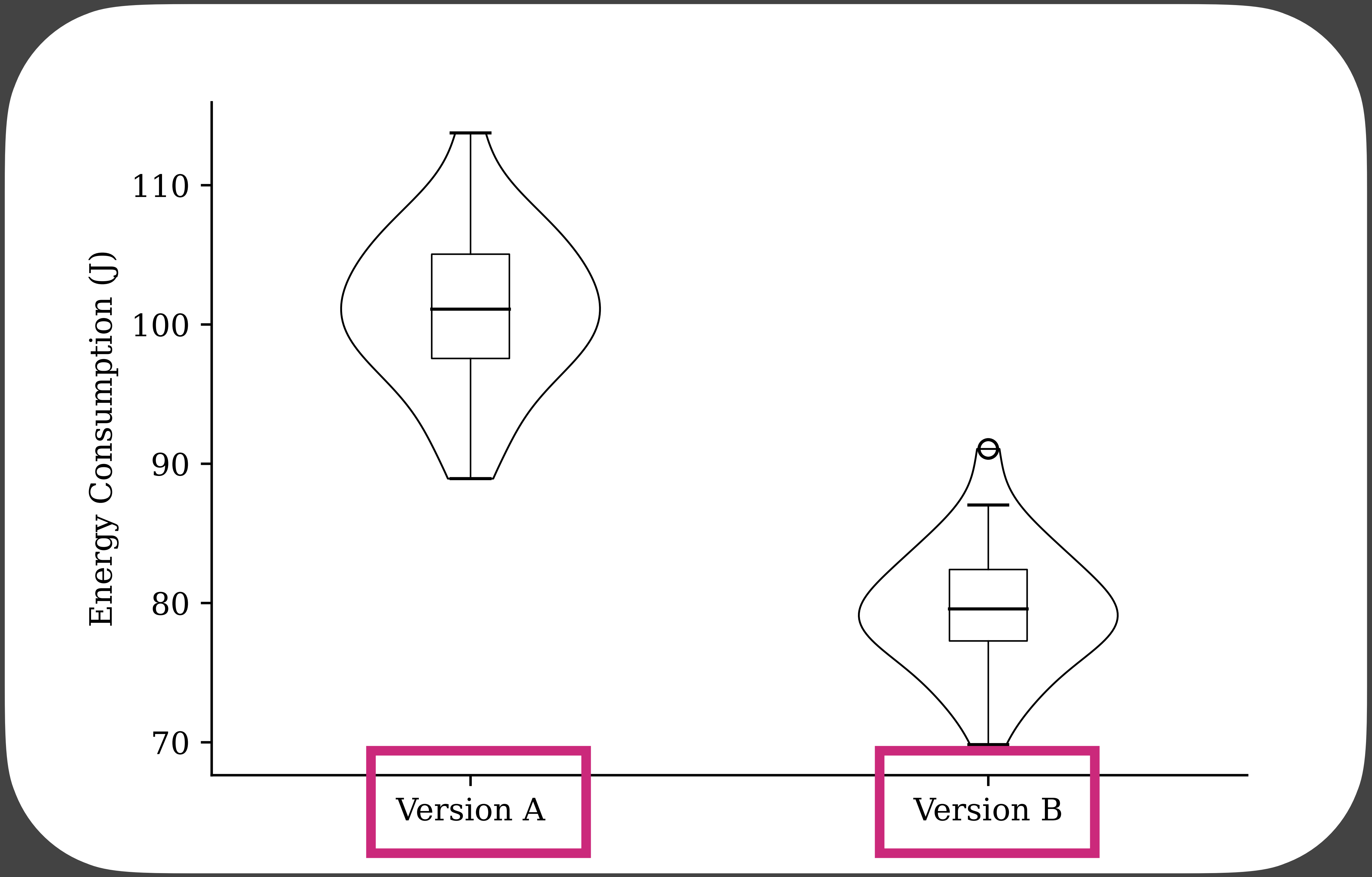
Automate Executions 🤖

- (Already mentioned in the previous classes)
- One cannot run 30 shuffled experiments per version without automation…

Data analysis

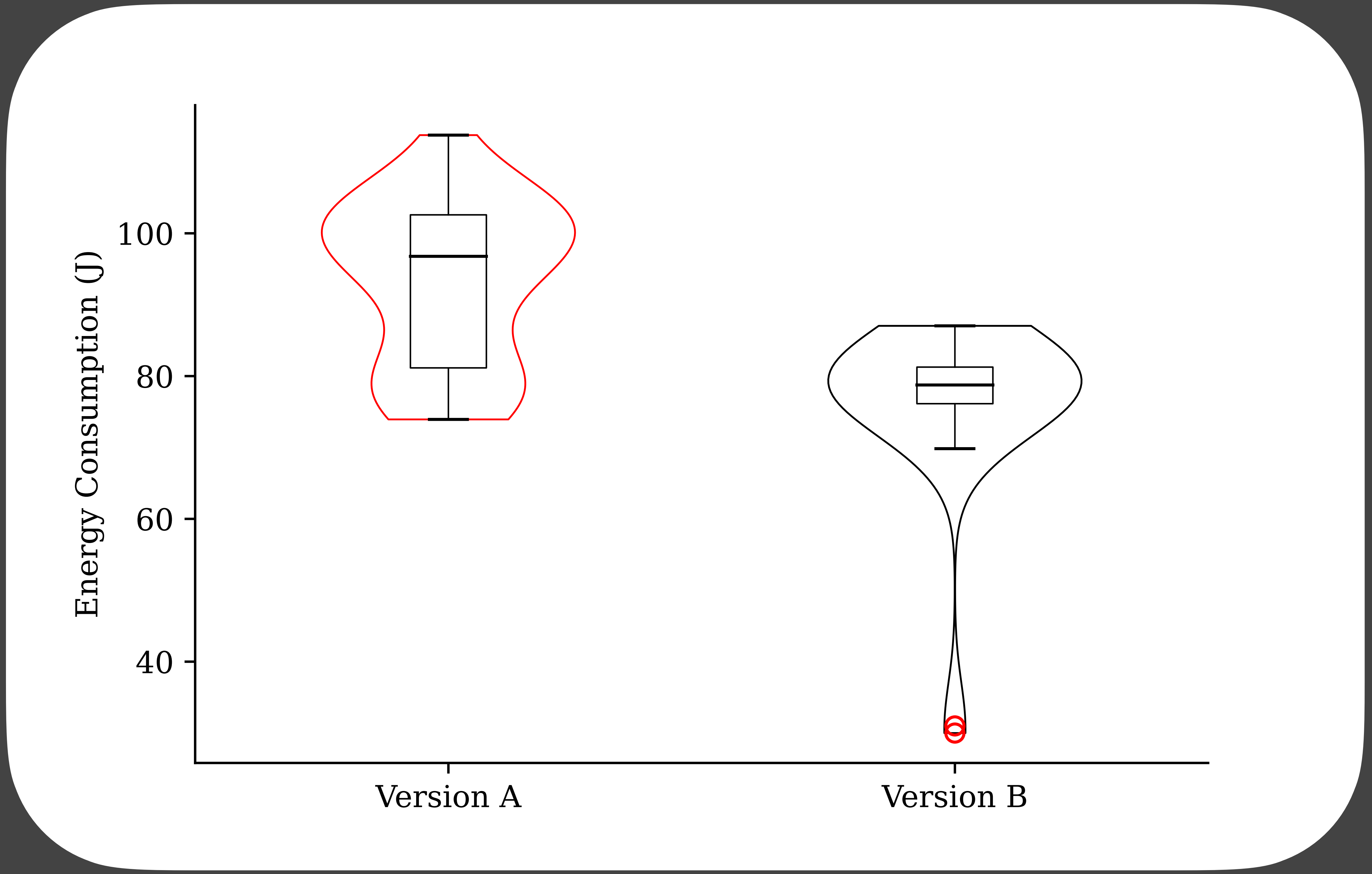
1. Exploratory Analysis

- Plot the data and inspect outliers or unexpected biases.
- **Violin+box plots** are usually handy. (?)
  - It's a nice way of combining the 30 experiments, and of showing descriptive statistics. (?)
  - Shows the **shape of the distribution** of the data.



1. Exploratory Analysis (II)

- Data should be **Normal**. Unless there's a good reason.
- E.g., somewhere amongst the 30 executions, there might be 1 or 2 that failed to finish due to some unexpected error.
  - (It happens and that's ok!)– consequently, the execution is shorter and spends less energy – **falsely appearing as more energy efficient**.
- If data is not Normal there might be some issues affecting the measurements that might be ruining results. It is important to investigate this.



Energy data is not normal. Why?

- It might be caused by one of the following reasons:
  - **There was an error** in some of the executions. If not detected and fixed it might ruin results.
  - Your tests are **not fully replicable** or are **not deterministic**. Quite frequent when you have **internet requests** or **random-based algorithms**.
  - There is an **unusual task** being run by the system during some experiments.
  - The computer entered a **different power mode**.
  - External physical conditions have changed. E.g., someone opened a window.

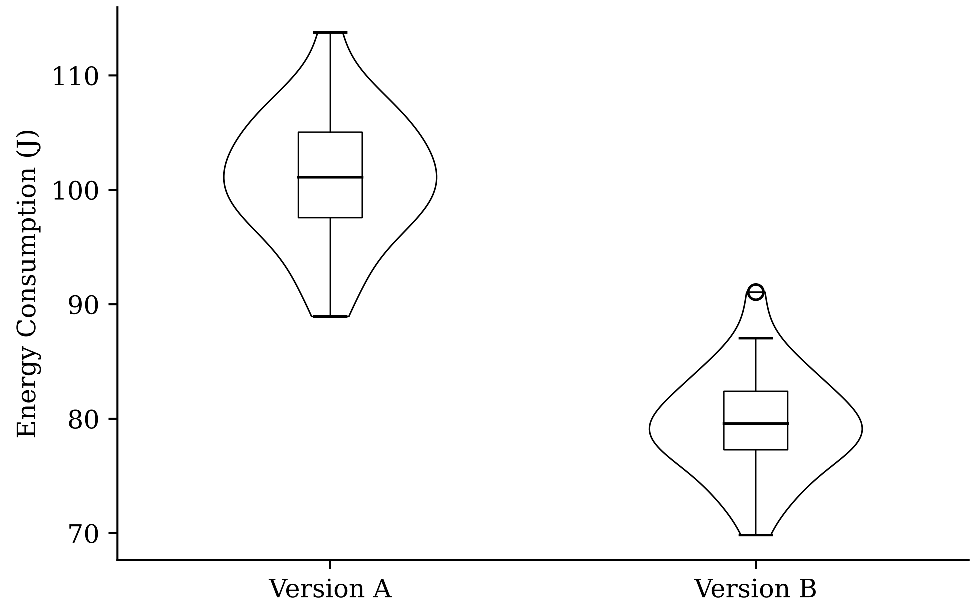
Energy data is not normal. How to fix?

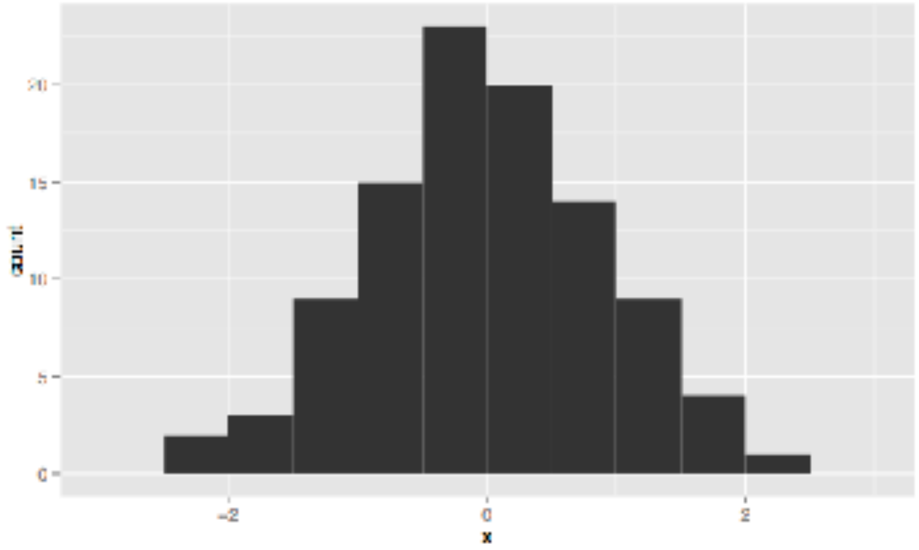
- We have **2**+1 options:

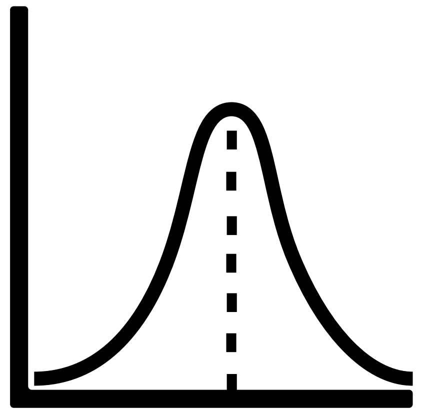
1. **Remove outliers**. If there are only a few points that deviate from the normal distribution, it is okay to simply remove them.
    - Use the **z-score outlier removal**. (?)
        - **Remove** all data points that **deviate** from the **mean** more than **3 standard deviations**: $|\bar{x} - x| > 3s$
2. **Fix the issue** and **rerun** experiments.
3. Conclude that **nothing can be done about it** and data will never be normal. (e.g., in AI, executions are rarely deterministic). ⚠️ Only after ruling out the previous points.

How do we know whether data is Normal?

- Visualise distribution shape: **violin plots**, **histograms**, **density plot**.







- **Shapiro-Wilk test**.
  p-value $< 0.05 \Rightarrow$ **data is not normal**;
  p-value $\geq 0.05 \Rightarrow$ we are not sure but it is okay to assume that **data is normal**.

⚠️ assuming normal distribution

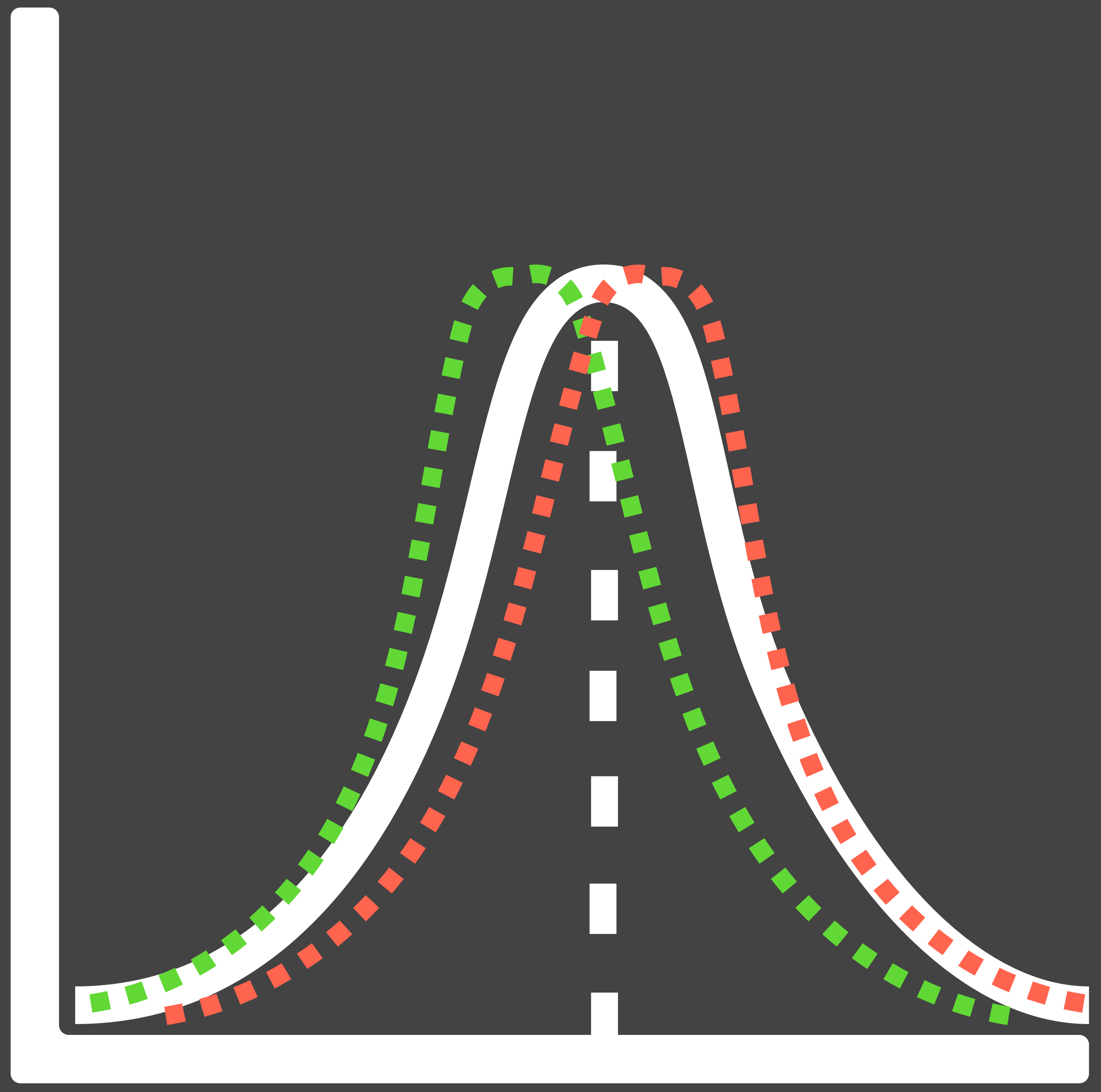
After having all data ready, which artefact is more energy efficient(?)

First approach: compare sample **means**.

⚠️ assuming normal distribution

Statistical significance

- Even if, on average, one artefact has lower energy consumption than other, it might be just a random difference.
- When we extract a sample from a normal distribution it will never be the exact same
- **Statistical significance** tests help you understand the differences in the average are conclusive/significant or inconclusive/insignificant.



⚠ assuming normal distribution

Statistical significance test

- Two-sided (?) parametric test **Welch's t-test**.
  - Less known alternative to **student's t-test**.

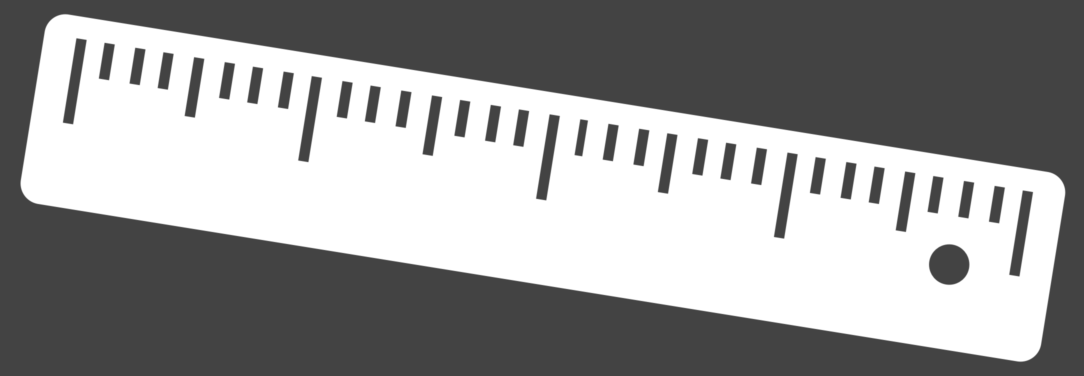
Welch's t-test in Python

```
from scipy.stats import ttest_ind

_,pvalue = ttest_ind(sample_a, sample_b,
                     equal_var=False,
                     alternative='two-sided')
```

⚠️ assuming normal distribution

Effect Size analysis



- Now that we know that results are statistically significant we need to **measure the difference** between the two samples.
    - **Mean difference**: $\Delta\bar{x}$
    - **Percent change**: $\dfrac{x_B - x_A}{x_A} \times 100\,\% = \dfrac{\Delta\bar{x}}{x_A} \times 100\,\%$
    - **Cohen's d** (informal definition: mean difference normalized by a **combined standard deviation**): $\dfrac{\Delta\bar{x}}{\frac{1}{2}\sqrt{s_1^2 + s_2^2}}$

(?)

Imagine that version **A** spends **70J** and version **B** spends **68J** with a p-**value** $= 0.04$.

On average, version **B** spent **less energy than** version **A** — ✅

There is statistical significance — ✅

Effect size, percent-difference is ≈3% — 🤔

⚠️ Do we care?

Practical Significance

- Depending on the case, a 2% improvement might be either **wonderful** or **completely useless**.
  - Effect size analyses help assess practical significance but **might not be enough**.
  - A critical discussion always needs to be performed. **Consider context** and explain in what sense the effect size might be **relevant**.
    - E.g.:
      - to improve 2% in energy efficiency the code will be less readable or the user experience is not so appealing.
      - A particular method improves 2% but it will only be used 1% of the time.
- **There is no particular metric or structure**, but this kind of critical analysis is very important.

What if data is **not Normal**?

Same approach but different tests/metrics!

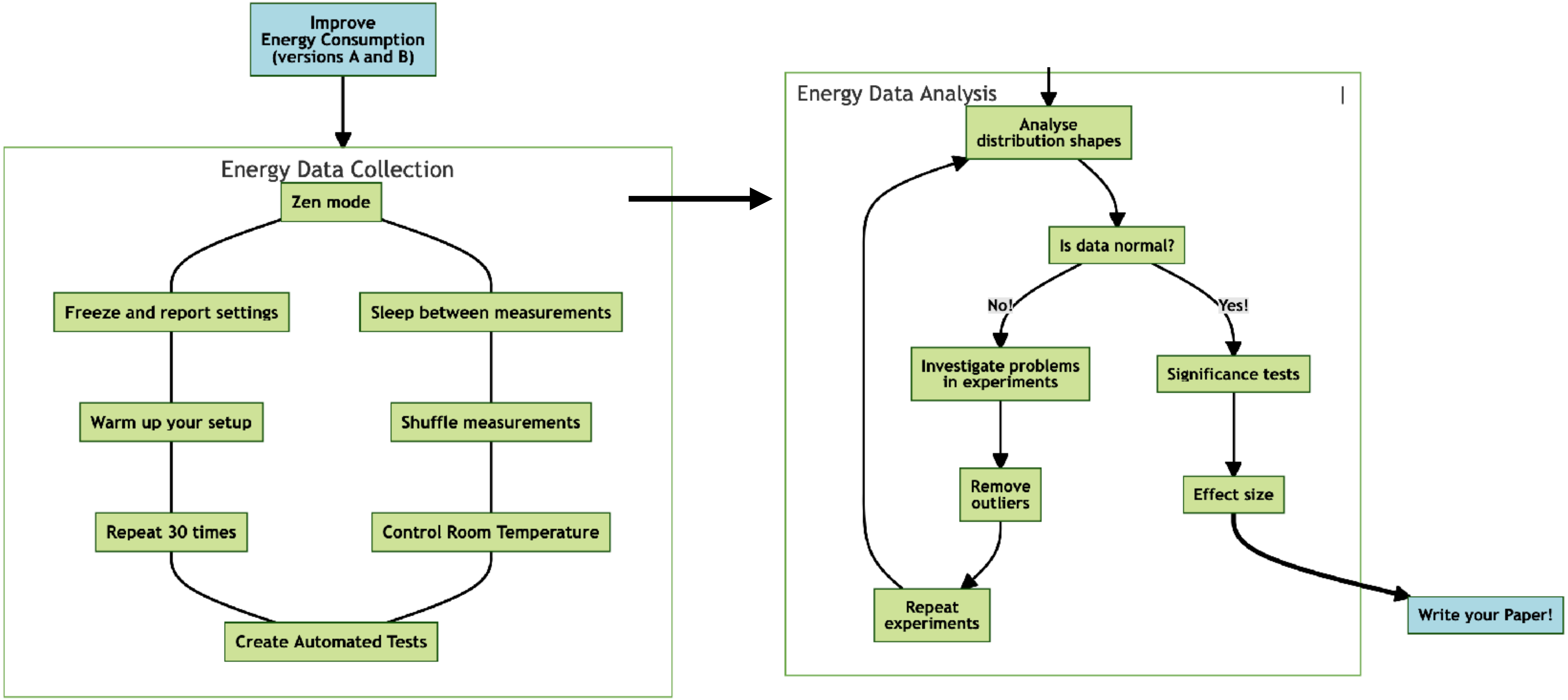
Non-normal data

- Statistical significance: **non-parametric** test (?)
  - **Mann-Whitney *U*** test. Instead of looking at standard deviation or mean, it orders samples and compares with each other.
  - **Less power** than parametric-tests (?)
- Effect size
  - Median difference: ΔM
  - **Percentage of pairs** supporting a conclusion (i.e., # pairs where version A > version B/ total pairs)
  - **Common language effect size**: $\dfrac{U_1}{N_1 N_2}$

Version A	Version B
100	101
90	89
88	89
88	89
87	86
86	70
86	60

A is more energy efficient
in 57% of the cases
(4 out of 7)

Recap



Improve
Energy Consumption
(versions A and B)
Energy Data Collection
Zen mode
Freeze and report settings
Sleep between measurements
Warm up your setup
Shuffle measurements
Repeat 30 times
Control Room Temperature
Create Automated Tests
Energy Data Analysis
Analyse
distribution shapes
Is data normal?
No!
Yes!
Investigate problems
in experiments
Significance tests
Remove
outliers
Effect size
Repeat
experiments
Write your Paper!