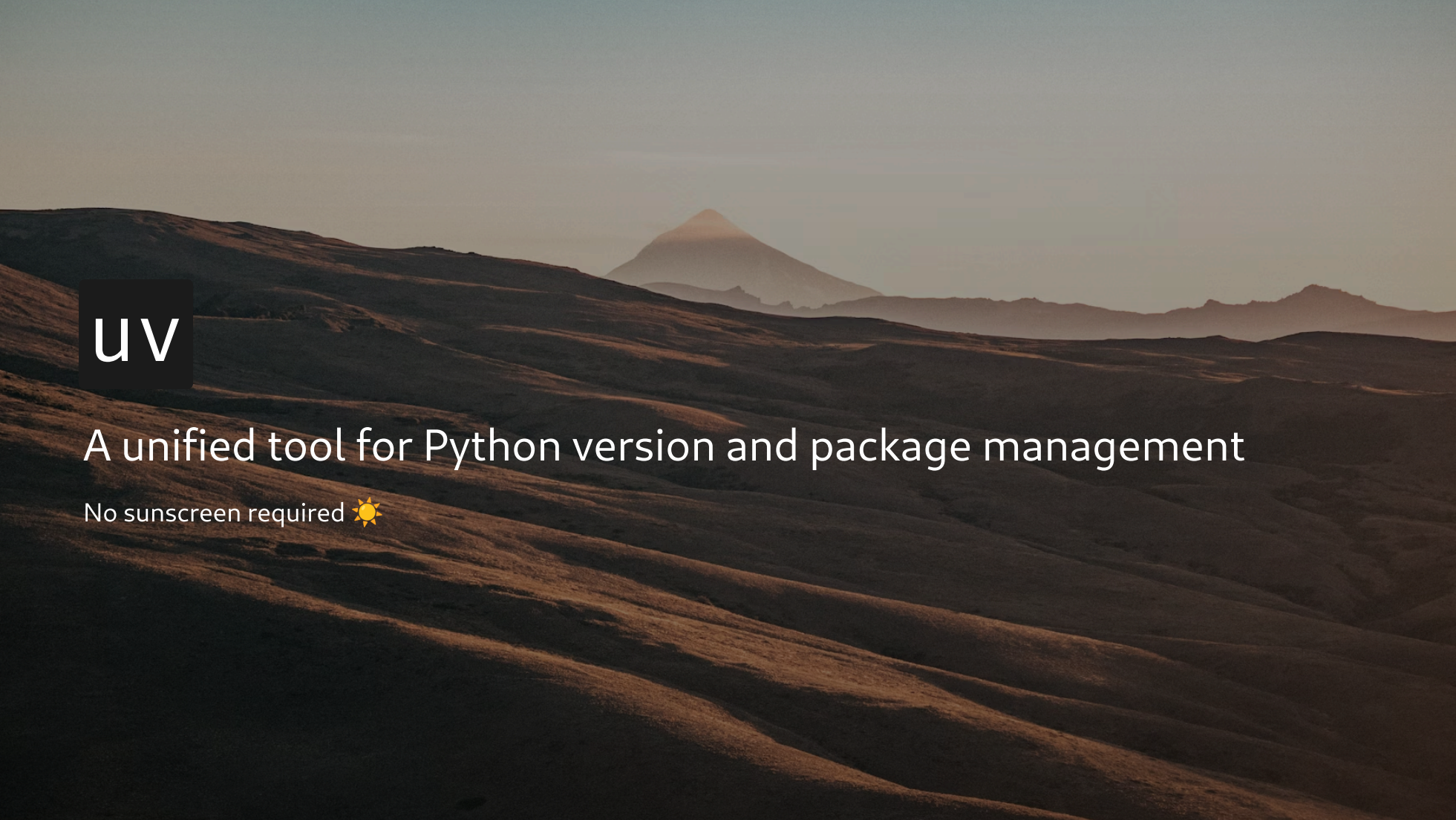
uv
A unified tool for Python version and package management
No sunscreen required ☀️

About

Florian Obersteiner (f.obersteiner@kit.edu), DM group, IMKASF

- data management & IT
- background: experimental atmospheric research
- tech stack: Python, go, LabVIEW, Zig, …
  - mostly on Linux
  - current projects: web applications and containerization

Outline

- challenges in Python version and package management
- what is `uv` and how can it help?
- usage examples & Q&A

Motivation

- multiple Python versions and virtual environments required for package development and collaboration
- many existing tools like pyenv, pip, pipx, poetry, venv, conda …
- especially for beginners, this can become a waste of time
  - …or even scare them away from Python, back to Matlab 😳

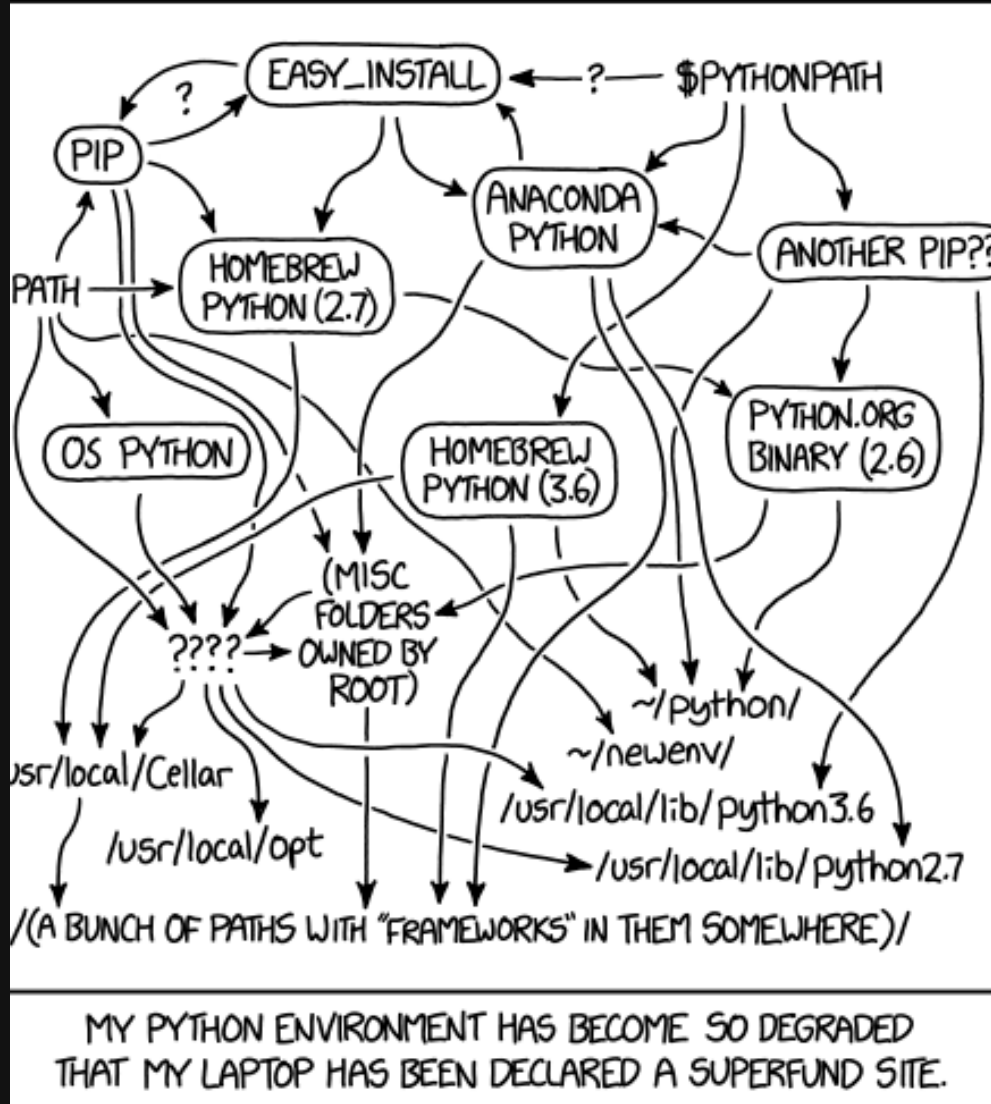

image source: xkcd

What I have worked with

- Python version management: `pyenv`
  - user installations and `win-pyenv` on Windows, back in the days
- package installation: `pip`
- virtual environments: Python / `venv`
- package development / management: `poetry` (comes with its own venv…)

∑ **5+ tools** - and no code written…

A simplified, all-in-one solution ?!

Introducing… What is `uv`?

- 👾 a CLI tool
- Python version, package, and project manager in one executable
- based on rye, now developed by Astral, the creators of the `ruff` Python linter
- available for Linux, Mac and Windows

Exemplary commands

```
uv python install 3.12.7 # install a Python version

uv init [project-name] # create a Python project, including a venv

uv add [name-of-dependency] # add a dependency to a project

uv lock -U && uv sync # make a lock file and upgrade everythin in the venv

uv run [script-name] # run a script in the project
```

CLI reference

Demo

Conclusion: why use uv?

- simplifies environment setup and management, from Python version to virtual environment
- reduces the learning curve associated with multiple tools
- easily reproduce environments across different systems
  - `uv.lock` and derived `requirements.txt` for use with other package managers
  - CI integration
  - Docker support
- overall pleasant user experience; fast dependency resolution and intelligent caching

Q&A