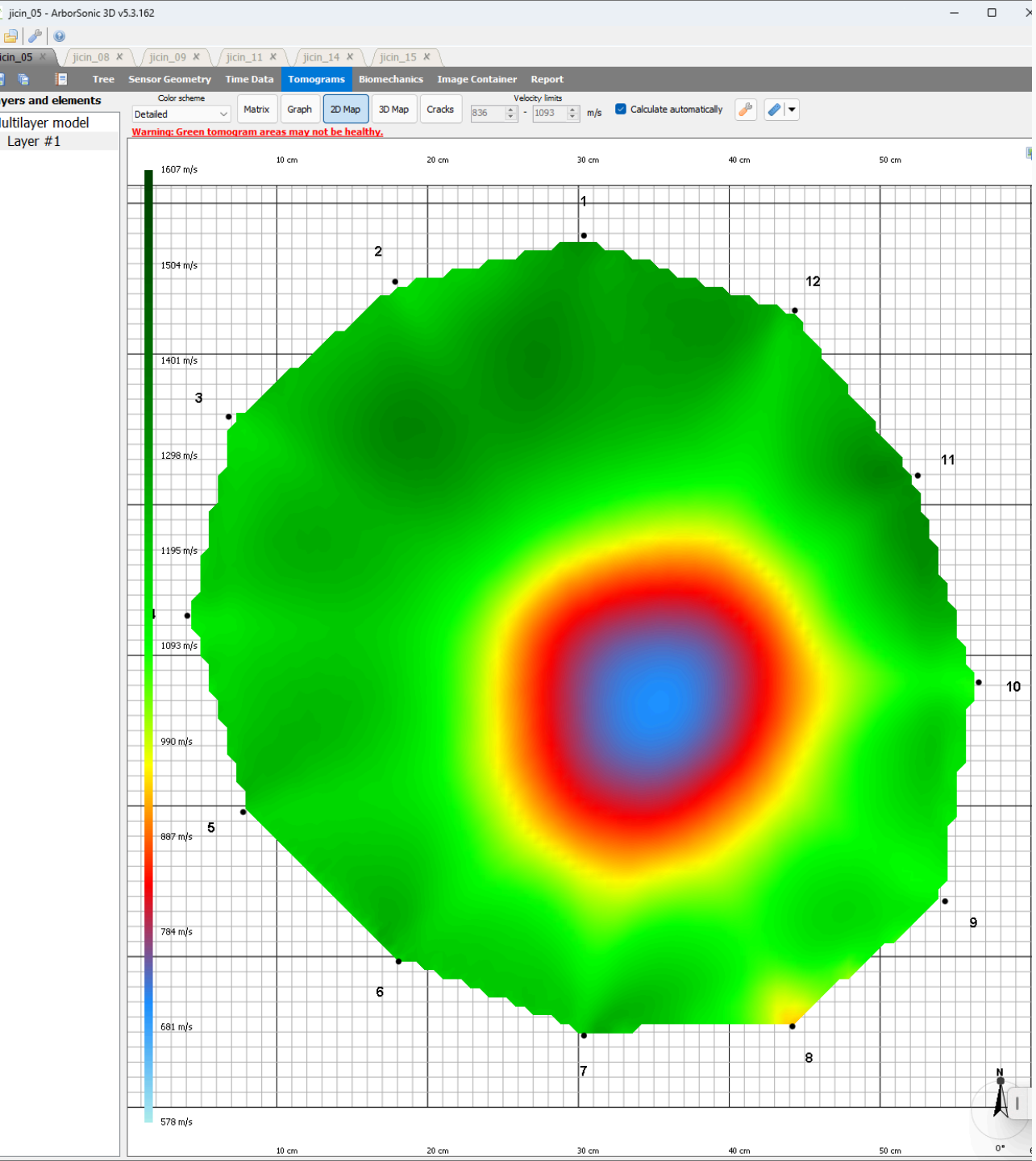
Resistograph meets tomograph

Robert Mařík & Valentino Cristini
Mendel University in Brno

MENDELU
Faculty of Forestry
and Wood
Technology

Content of the talk

- Resistograph and tomograph: strengths and limitations
- Combined approach: a Python library to merge data from both devices
- Vibe coding in 2025 (ChatGPT)
- Code sharing in 2025 (Docker)



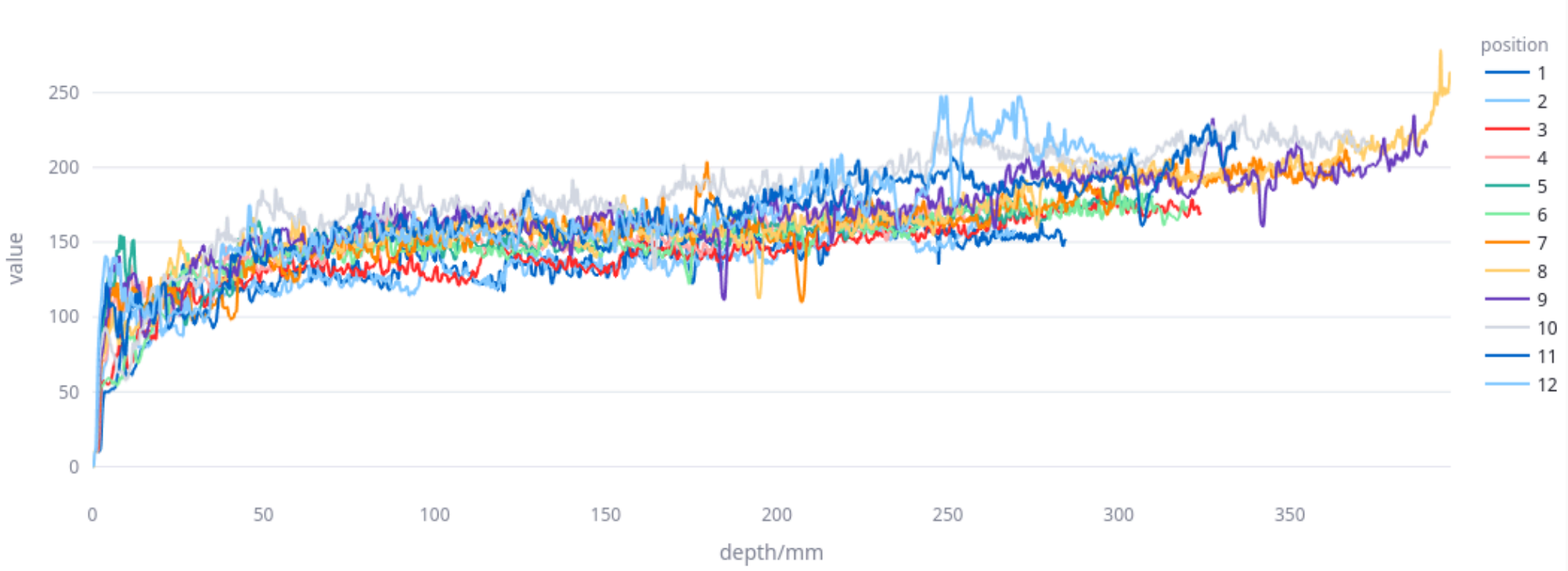
Tomograph

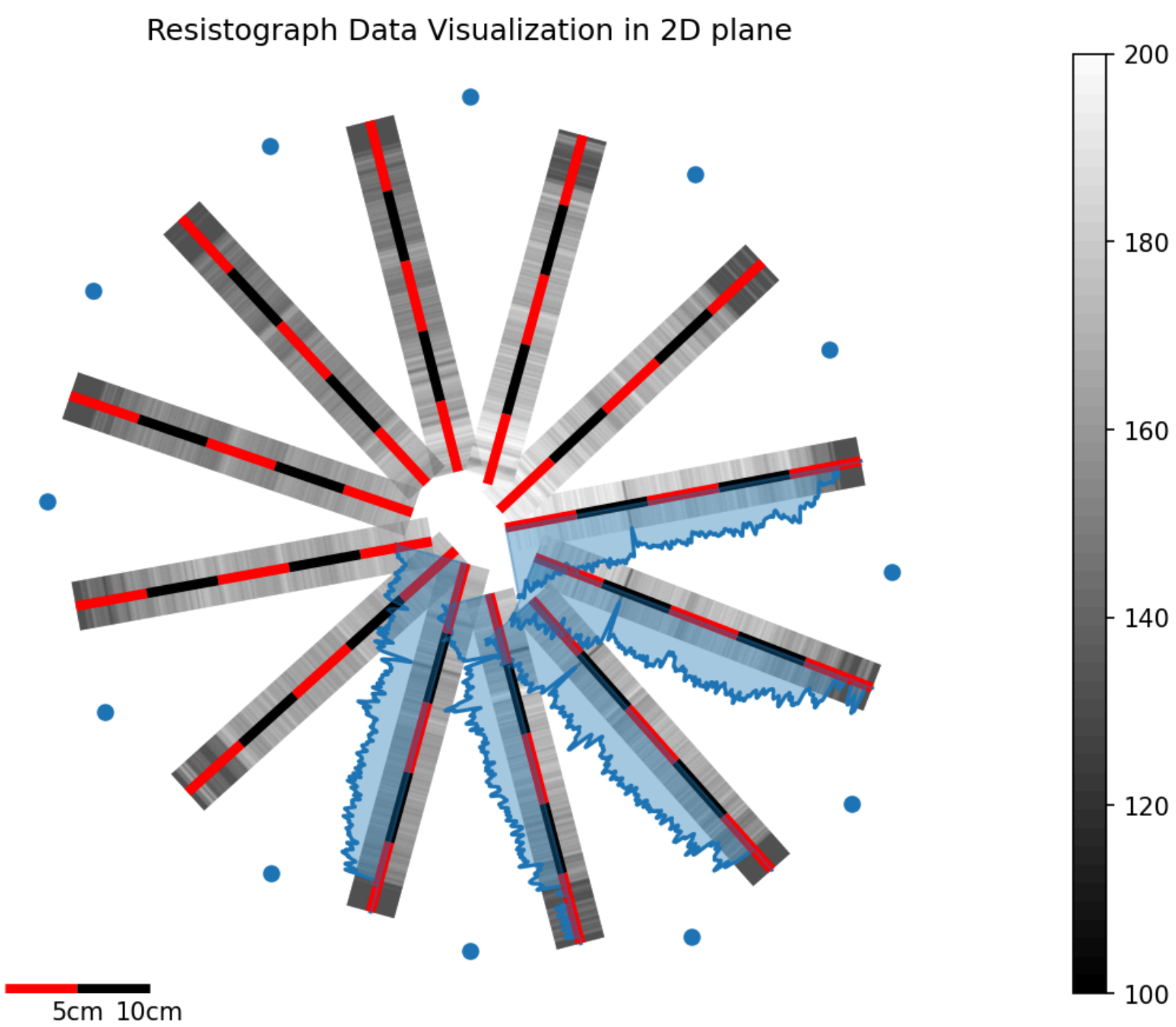
- fast and reliable tool for stem inspection
- global information from the whole cross section
- shows the size and shape of the internal defects
- cracks are reported as cavities

MENDELU
Faculty of Forestry and Wood Technology

Resistograph

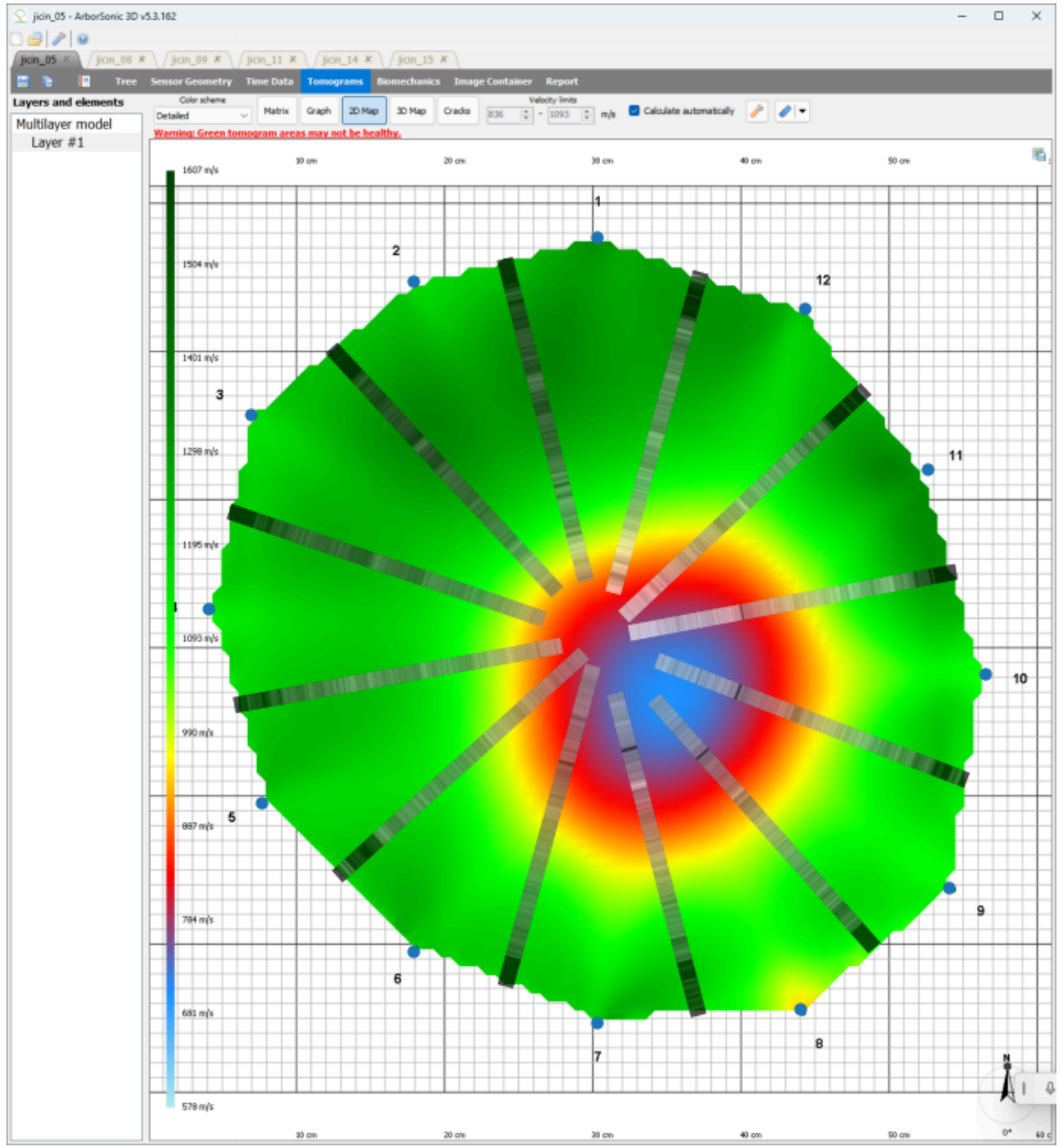
- scans the power required to microdrilling at given speed
- measures mechanical properties of the material
- local information





Merge data I

- Transform resistograph data to 2D geometry of the cross section
- Visualize the data in the new geometry



Merge data II

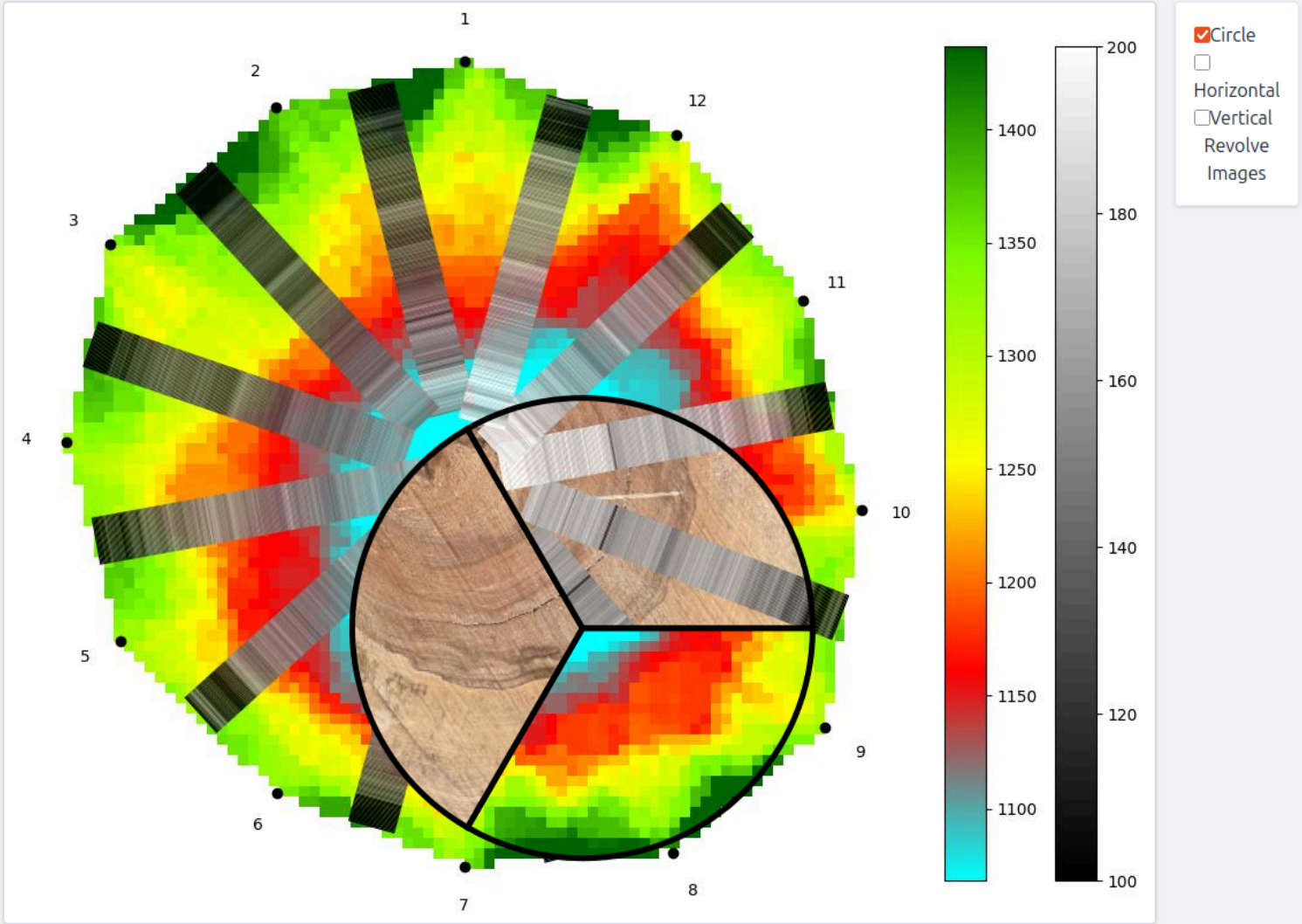
- merge resistograph data with tomograph data
- visualize the merged data
- look for short or long decreases in resistograph data. This indicates cracks and cavities, respectively

When resistograph meets tomograph

The demo of overlays of four images. See the repository for the code.

- Tomogram
- Tomogram with resistograph data
- Section photo
- Section photo with resistograph data

You can move the mouse over the image to reveal the other layers or click the image to switch layers.



Python library

- language widely used in scientific data processing
- many libraries for data processing and visualization
- easy to automate, scale, modify, share and reuse
- easy to integrate with other tools

MENDELU
Faculty of Forestry and Wood Technology

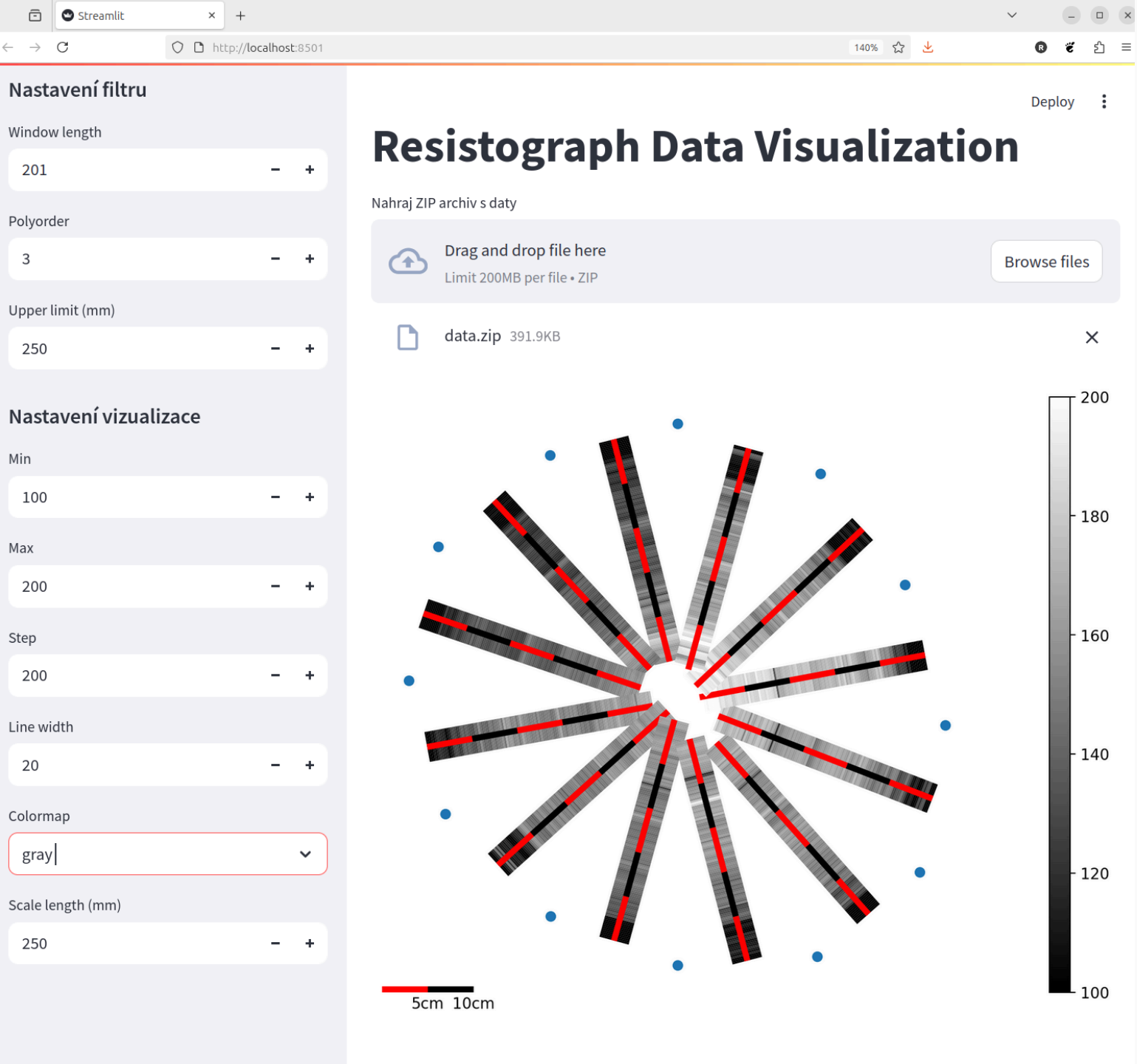
```
# See LICENSE file or https://creativecommons.org/licenses/by/4.0/
Run Cell | Run Below | Debug Cell
#%%
"""
This script visualizes resistograph data on a tomogram.
It processes resistograph data files and node coordinates to generate a plot
with resistograph data overlaid on a tomographic representation.

Configuration and validation are handled via Pydantic models.
"""
Run Cell | Run Above | Debug Cell
#%%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import glob
from scipy.signal import savgol_filter
import logging
from matplotlib.collections import LineCollection
from matplotlib.transforms import Affine2D

# --- NEW: importy pro konfiguraci ---
from pydantic import BaseModel, Field, PositiveInt, DirectoryPath, model_validator
from typing import List, Optional

# Logging configuration
logging.basicConfig(level=logging.WARNING, format='%(levelname)s - %(message)s')

# --- NEW: Pydantic models for configuration ---
class FilterSettings(BaseModel):
    window_length: PositiveInt = Field(201, description="Window length for Savitzky-Golay filter")
    polyorder: int = Field(3, description="Polynomial order for filter")
    upper_limit: int = Field(250, description="Maximum depth in mm")

    @model_validator(mode="after")
    def check_polyorder_vs_window(self):
        if self.polyorder >= self.window_length:
            raise ValueError("polyorder must be smaller than window_length")
        return self


class PlotSettings(BaseModel):
    min: int = Field(100, description="Minimum value for color normalization")
    max: int = Field(200, description="Maximum value for color normalization")
    step: int = Field(200, description="Step for downsampling")
    linewidth: int = Field(20, description="Line width")
    cmap: str = Field("gray", description="Matplotlib colormap")
```

Python library

Limitations

- requires programming skills
- requires installation of Python, Python IDE and libraries
- no GUI

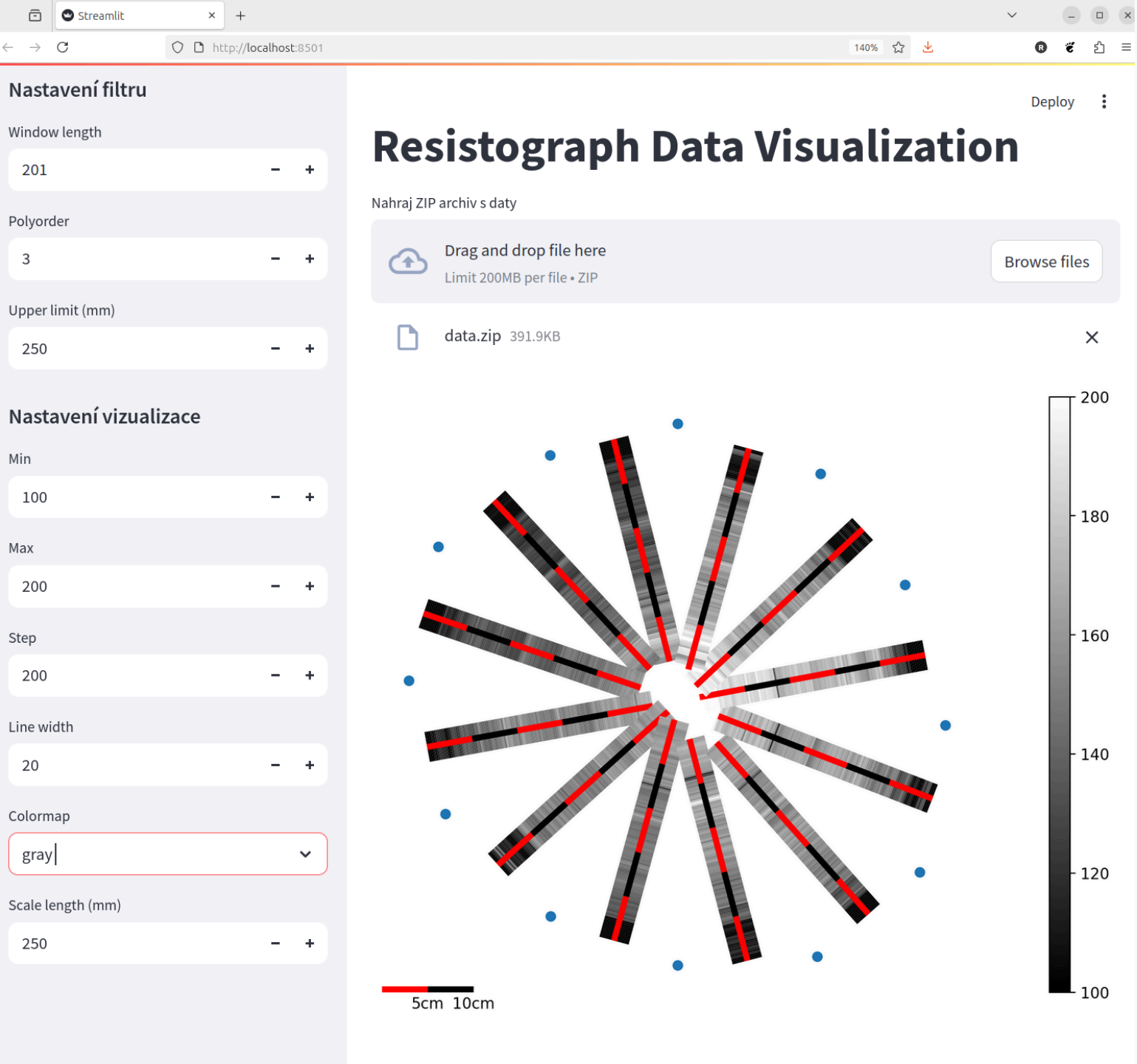
MENDELU
Faculty of Forestry and Wood Technology



Streamlit

- library for building web apps
- requires minimal code
- interactive widgets for user input
- real-time updates
- widely used in data science and machine learning, in industry and academia

MENDELU
Faculty of Forestry and Wood Technology



Vibe coding

- ChatGPT 5 on August 2025
- web app in two prompts

```
Mam nasledujici knihovnu. Napis streamlit
program, ktery umozni nahrat zazipovany
adresar s daty a spusti na nem prikazy
odpovidajici main funkci. Vystup se zobrazi.
```

```
OK. V levem panelu chci mit moznost menit
prednastavene volby.
```

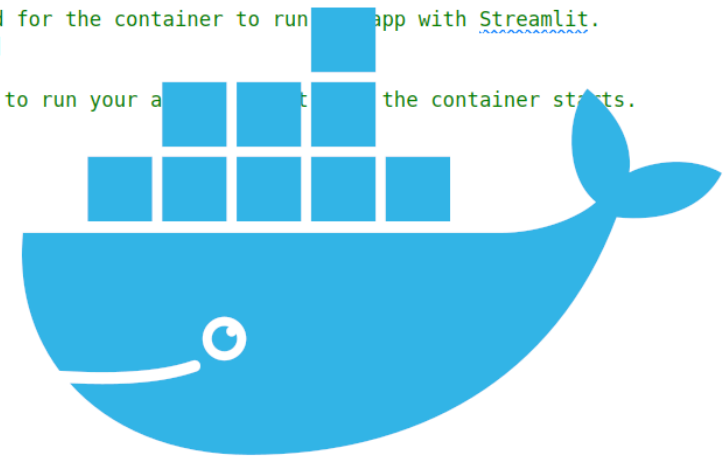
MENDELU
Faculty of Forestry and Wood Technology

compose.yml

```yaml
services:
  resisto:
    network_mode: bridge
    working_dir: /app/app
    ports:
      - 8501:8501
    image: resisto:latest
    build: .
```

Dockerfile 1

```dockerfile
# This sets up the container with Python 3.10 installed.
FROM python:3.10-slim (last pushed 3 weeks ago)

WORKDIR /app
COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

# This tells Docker to listen on port 80 at runtime. Port 80 is the standard port for HTTP.
EXPOSE 80

# This command creates a .streamlit directory in the home directory of the container.
RUN mkdir ~/.streamlit

# This copies your Streamlit configuration file into the .streamlit directory you just created.
RUN cp config.toml ~/.streamlit/config.toml

# This sets the default command for the container to run app with Streamlit.
ENTRYPOINT ["streamlit", "run"]

# This command tells Streamlit to run your a t the container st ts.
CMD ["app.py"]
```



Docker

A containerization platform

- packages application and its dependencies into a container
- ensures consistency across different environments
- easy to share and deploy
- widely used in industry, academia, research

MENDELU
Faculty of Forestry and Wood Technology

compose.yml > ...

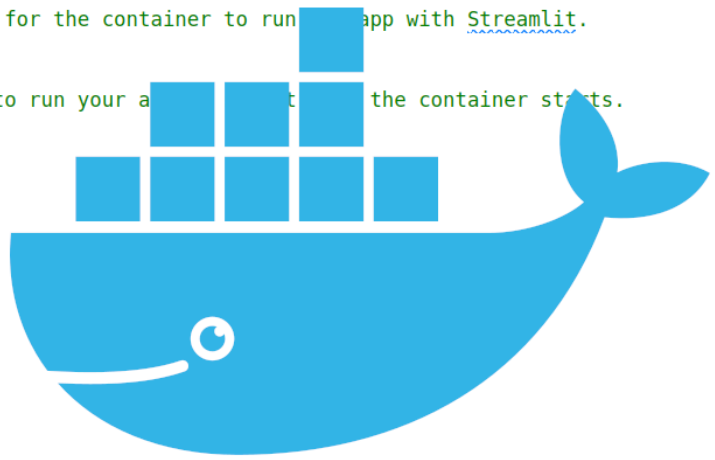
Run All Services

```yaml
services:
    Run Service
    resisto:
        network_mode: bridge
        working_dir: /app/app
        ports:
            - 8501:8501
        image: resisto:latest
        build: .
```

Dockerfile 1 ×

Dockerfile > ...

```dockerfile
# This sets up the container with Python 3.10 installed.
FROM python:3.10-slim (last pushed 3 weeks ago)

WORKDIR /app
COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

# This tells Docker to listen on port 80 at runtime. Port 80 is the standard port for HTTP.
EXPOSE 80

# This command creates a .streamlit directory in the home directory of the container.
RUN mkdir ~/.streamlit

# This copies your Streamlit configuration file into the .streamlit directory you just created.
RUN cp config.toml ~/.streamlit/config.toml

# This sets the default command for the container to run app with Streamlit.
ENTRYPOINT ["streamlit", "run"]

# This command tells Streamlit to run your a t the container st ts.
CMD ["app.py"]
```

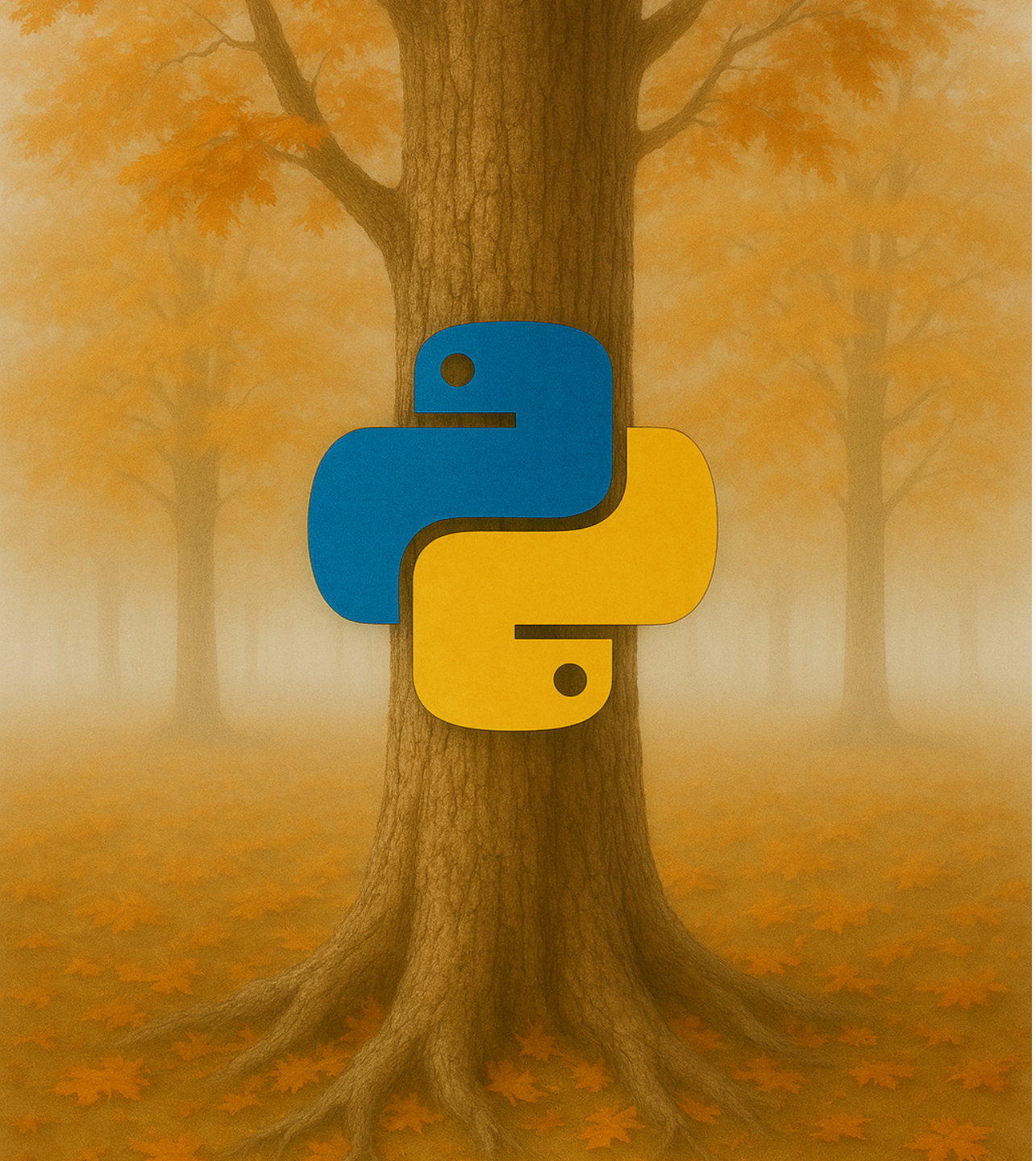


Run dockerized app

```
docker compose up
```

- 🚫 No Python install
- 🚫 No dependency issues
- 🖥️ Works on Win / Mac / Linux
- 📂 Just clone repo with `Dockerfile` and `docker-compose.yml`
- ⏳ First run = minutes, later = ms

MENDELU
Faculty of Forestry and Wood Technology



Summary

- Resistograph and tomograph are complementary tools for tree stem inspection
- A Python library was developed to simplify data merging and visualization
- GUI for Python is possible with Streamlit
- Installation can be made simple and repeatable with Docker

MENDELU
Faculty of Forestry and Wood Technology