Self-introduction: Alexander Held

Alexander Held[1]

[1] University of Wisconsin–Madison

pyhep.dev 2024
https://indico.cern.ch/event/1375573/
Aug 26–30, 2024

This work was supported by the U.S. National Science Foundation (NSF) under Cooperative Agreements OAC-1836650 and PHY-2323298.

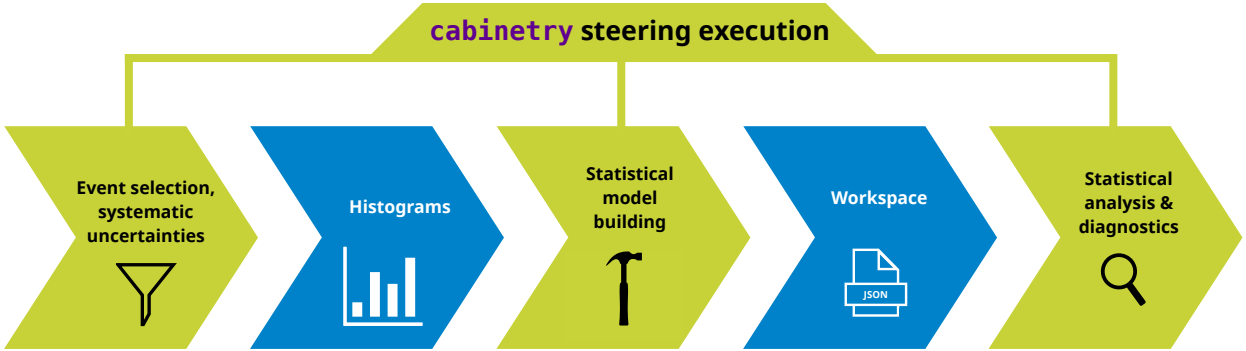
About myself

https://github.com/alexander-held

- **Research scientist** at the University of Wisconsin–Madison
- **ATLAS Collaboration**
  - focus on Higgs physics & statistical analysis
- Part of **IRIS-HEP**
  - software institute, R&D for HL-LHC
  - co-coordinator of
    - Analysis Grand Challenge project (https://agc.readthedocs.io/), see talk yesterday
    - 200 Gbps analysis challenge, see talk yesterday
- Author of **cabinetry**
  - statistical model building & inference with template fits (https://github.com/scikit-hep/cabinetry/)
- **Talk to me about**: scikit-hep stack, columnar analysis, statistics, distributed computing

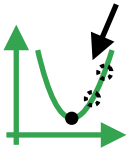
The `cabinetry` library

- **`cabinetry`** is a **Python library** for creating and operating HistFactory models
  - **design** and **construct statistical models** (workspaces) from instructions in **declarative configuration**
    - analyzers specify selections for signal/control regions, (Monte Carlo) samples, systematic uncertainties
    - `cabinetry` steers creation or collects provided template histograms (region ⊗ sample ⊗ systematic)
    - `cabinetry` produces `HistFactory` workspaces (serialized fit model)
  - perform **statistical inference**
    - including diagnostics and visualization tools to study and disseminate results



Full workflow example

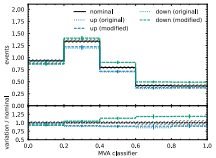
$$L(\vec{k},\vec{\theta}) = p(\vec{n},\vec{a} \mid \vec{k},\vec{\theta})$$



generate histograms for all samples: nominal & variations

can read histograms or ntuples *with* `cabinetry`

cabinetry



workspace: serialized statistical model

built by `cabinetry` or from *external source*

cabinetry

turn into likelihood function

a function, given (observed + auxiliary) data $\vec{n},\vec{a}$ *done via* `pyhf`

pyhf differentiable likelihoods

minimize NLL

using function minimizer, e.g. MINUIT (MIGRAD algorithm) *via* `iminuit`

iminuit

can wrap these steps with `cabinetry`

inference results

plots & tables *produced by* `cabinetry`

cabinetry