SPONSORED BY THE
Federal Ministry of Education and Research

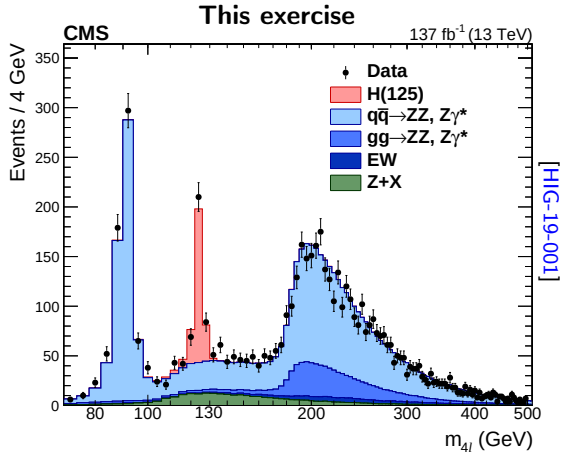
# PO&DAS HIG PAG Exercise

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

**Closer look at columnflow modules**

Matteo Bonanomi, **Philip Keicher**, Daniel Savoiu
October 14, 2023

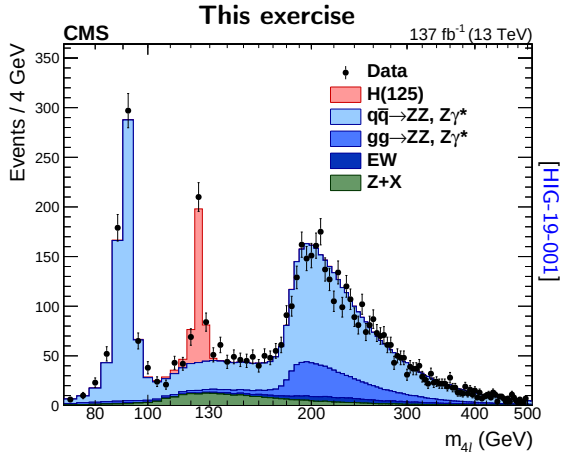
# Goal of ...


This exercise
CMS
137 fb-1 (13 TeV)
Data
H(125)
qq̄→ZZ, Zγ*
gg→ZZ, Zγ*
EW
Z+X
Events / 4 GeV
m4l (GeV)
[HIG-19-001]

# Goal of ...

## ... todays session

- Define $m_{4l}$
- Implement lepton selections
- *Bonus:* implement FSR correction



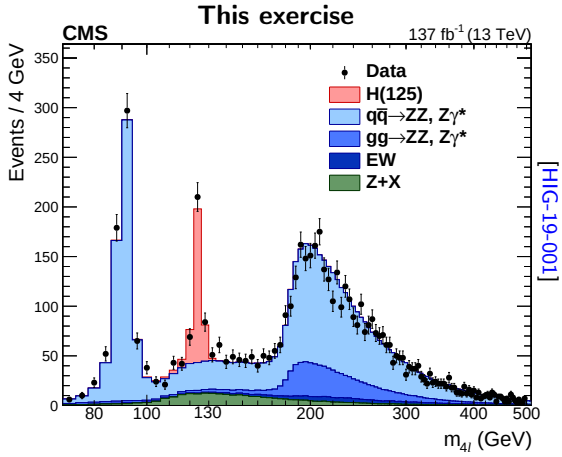
# Goal of ...

## ... todays session

- Define $m_{4l}$
- Implement lepton selections
- *Bonus:* implement FSR correction

## ... Monday's session

- Constuct Z-boson candidates
- Define analysis categories
- Answer physics questions from Matteo's talk!



# Introduction to objects in columnflow: TaskArrayFunction

## Key features

- Define columns to **load** with **uses**
- Define columns to **write** with **produces**
- **init**: Things to do before evaluating graph
- **requires**: Set custom requirements for the TaskArrayFunction
- **setup**: Things to do before calling TaskArrayFunction instance

```python
class my_func(TaskArrayFunction):
    uses = {"Jet.pt"}
    produces = {"Jet.pt_weighted"}

    # put other keyword arguments here

    def call_func(self, events, **kwargs):
        # definition of callable of class `my_func`
        pass

    def init_func(self, **kwargs):
        # Things to do before evaluating graph
        pass

    def setup_func(self, **kwargs):
        # Things to do before calling `call_func`
        pass

    def require_func(self, **kwargs):
        # Set custom requirements for this TaskArrayFunction
        pass
```

# Introduction to objects in columnflow: TaskArrayFunction

## Key features

- Define columns to **load** with **uses**
- Define columns to **write** with **produces**
- **init**: Things to do before evaluating graph
- **requires**: Set custom requirements for the TaskArrayFunction
- **setup**: Things to do before calling TaskArrayFunction instance
- **Real application: decorators!**

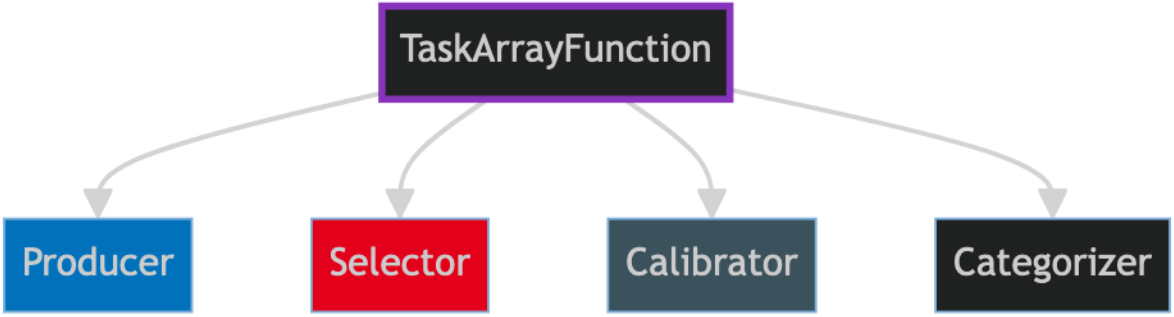
```python
@TaskArrayFunction(
    uses = {"Jet.pt"},
    produces = {"Jet.pt_weighted"},
    # put other keyword arguments here
)
def my_func(self, events, **kwargs):
    # this is automatically the callable of new class `my_func`
    pass

@my_func.init
def fancy_init_func(self, **kwargs):
    # Things to do before evaluating graph
    pass

@my_func.setup
def fancy_setup_func(self, **kwargs):
    # Things to do before calling `call_func`
    pass

@my_func.require
def fancy_require_func(self, **kwargs):
    # Set custom requirements for this TaskArrayFunction
        pass
```

# Day to day objects in columnflow



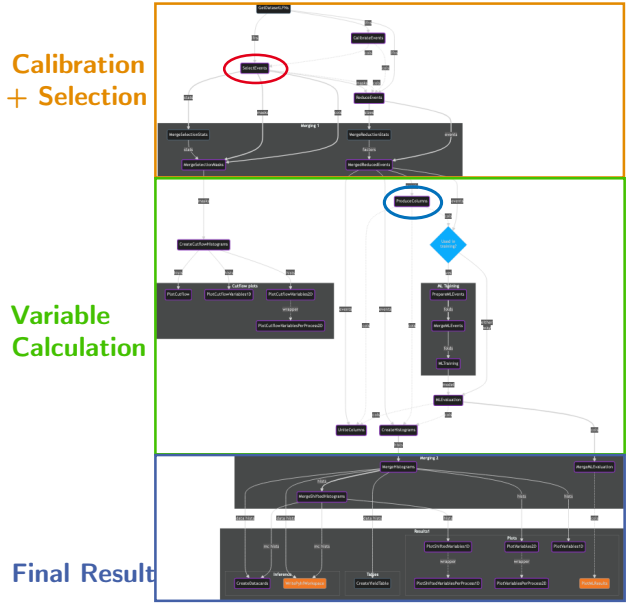
# Todos

## $m_{4l}$

- Implement Producer for $m_{4l}$

→ Profit from coffea behavior

- Define variable in analysis config `h4l/config/variables.py`



# HANDS-ON

# Todos

## $m_{4l}$ ✓

- Implement Producer for $m_{4l}$

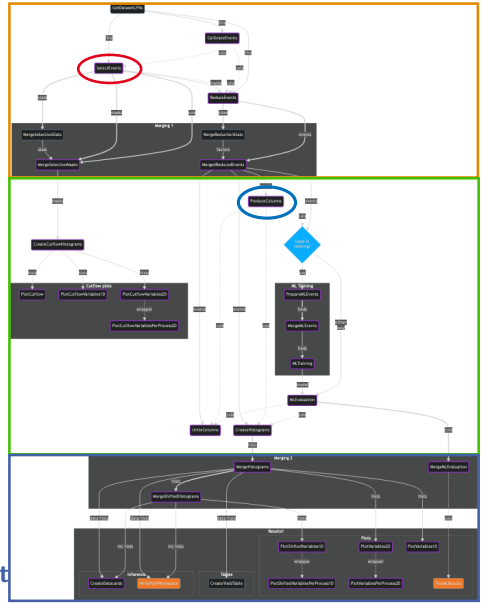
→ Profit from coffea behavior

- Define variable in analysis config `h4l/config/variables.py`

Calibration + Selection

Variable Calculation

Final Result



# Todos

## $m_{4l}$ ✓

- Implement Producer for $m_{4l}$

→ Profit from coffea behavior

- Define variable in analysis config `h4l/config/variables.py`

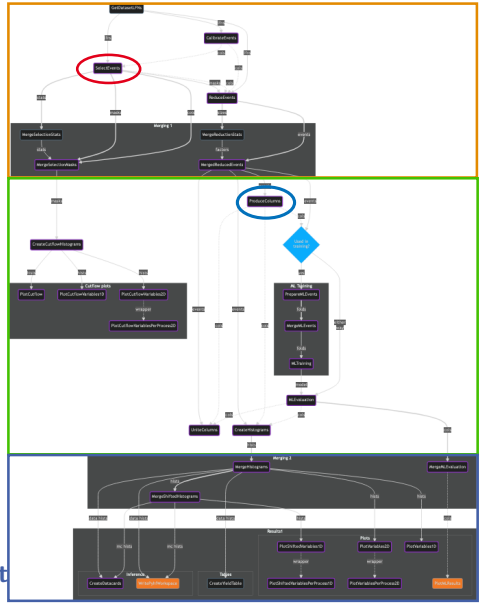
## Lepton selections

- Implement Selectors for electrons and muons [electron ID criteria]
- Include them in the workflow, e.g. by calling them in `h4l/selection/default.py`

Calibration + Selection

Variable Calculation

Final Result



# HANDS-ON

# *Bonus*: FSR photon correction

| FSR photons |
| --- |
| $p_T^{fl} > 2\,\text{GeV} \qquad \lvert\eta^\gamma\rvert < 2.4$ |
| $\mathcal{I}^\gamma_{PF} < 1.8$ |
| $\Delta R(\ell,\gamma) < 0.5 \qquad \frac{\Delta R(\ell,\gamma)}{(p_T^\gamma)^2} < 0.012\,\text{GeV}^{-2}$ |

- Consider electrons (muons) passing $p_T > 7\,\text{GeV}(5\,\text{GeV})$ w/o ID criteria
- Write **Calibrator** that finds all photons fulfilling above selections with a given lepton and add the four momenta
- Each photon can only be associated with one lepton $\rightarrow$ check for double matching!
- In case criteria are fulfilled with multiple leptons: assign to lepton with smallest $\frac{\Delta R(l,\gamma)}{(p_T^\gamma)^2}$

# Additional material

- columnflow repository
- Documentation (work in progress)
- Overview of columnflow
- coffea behavior