

2nd Belle II Academy, 1-5 Aug 2022  
University of Bonn



Improving your workflow  
by organizing your codes

Zuzana Gruberová

# The goal

## Write a code for a data analysis

### » The analysis

- MC simulation and data (input)
- prepare the sample (signal definition, background suppression, cuts, MVA)
- do some checks (variable distributions, data/MC agreement)
- perform the measurement (fitting, signal yield, efficiency, purity)
- get the results (numbers, plots, limits)

```
bands_list = [
[data_central_list, 'data'],
[data_band_list, 'data'],
[mc_central_list, 'mc'],
[mc_band_list, 'mc_upper'],
]

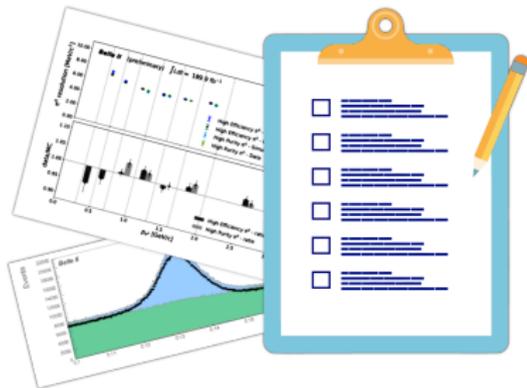
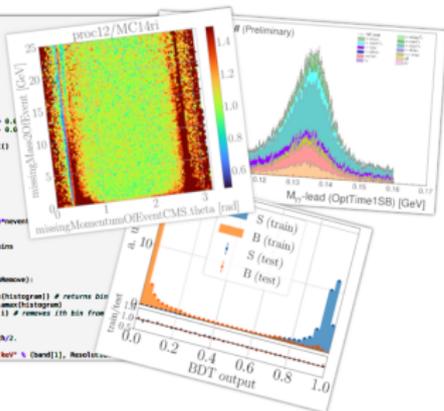
for i in range(len(bands_list)):
    data_central_name, data = bands_list[i]
    for band in bands_list:
        skins = len(bands_list)

        n = np.linspace(0.135 - 0.06, 0.135 + 0.06, 5)
        binsWidth = (0.135 + 0.06) - (0.135 - 0.06)
        list = array([band[i], volume.tolist()])
        histogram = np.ma.array(list)
        plt.plot(histogram)
        plt.show()

widthFraction = 0.83 #0.75
events = np.sum(histogram)
eventsToRemove = widthFraction*events
eventsToRemove = 0
binsWidthOfDifference = widthFraction*skins
binsToRemove = 0

while (eventsToRemove > eventsWidthOfDifference):
    i = np.where(histogram == np.ma.max(histogram)) # returns data
    eventsToRemove = eventsToRemove - binsWidthOfDifference
    histogram = np.delete(histogram, i) # removes ith bin from
    binsToRemove = binsToRemove + 1

Resolution = 100*(binsWidthOfDifference/2.
band[i], upper(bands_list))
print("Resolution for %s: %f keV" % (band[i], Resolution))
```



### » The code

- load the input (switch between samples)
- change the signal selection (adjust cuts, MVA)
- view various checks (distributions, ROC curve, FOM)
- do the computations (fitting, calculations)
- produce the results (numbers, plots, limits)

# Tips and tricks

## Simple but very very handy

### » Bad habits

- ✗ one huge code for everything
- ✗ absence of comments and headlines
- ✗ ambiguous variable names
- ✗ one output folder for all plots and other files
- ✗ no intermediate steps saved

### » Organized approach

- ✓ separate code files for different steps  
e.g. function definitions, applying corrections, applying cuts, fitting, final plots
- ✓ inline comments, headlines separating different steps/checks  
in this case more is better than less
- ✓ clear variable names  
e.g. `BDT_variables_list`,  
`pi_sample_MC_hist_LIDcorrection_no_cuts`
- ✓ organized output folders  
e.g. `data/MC distributions, histograms, CSV files, final plots`
- ✓ save intermediate steps  
e.g. save histograms after applying cuts to use in the fitting code later

```
band_list = [
    'pd_list', 'data'],
    'list', 'data_upper'],
    'list', 'mc'],
    'list', 'mc_upper'],
    'range(0, len(bands)) :
    'error_array_all[]]
    'nd in bands_list:
    'bin = len(error_array)

    = np.linspace(0.135 - 0.06, 0.135 + 0.06, nbins)
    'width = ((0.135 + 0.06) - (0.135 - 0.06))/nbins

    'at = error[band[1]].values.tolist()
    'histogram = np.asarray(list_

    plt.plot(histogram)
    'plt.show()

    widthFraction = 0.03 #66.7%

    nevents = np.sum(histogram)
    'neventsToRemove = widthFraction*nevents
    'neventsToRemove = 0

    binsToRemove = widthFraction*nbins
    'binsToRemove = 0

    while (neventsToRemove - neventsToRemove):
        i = np.where(histogram == np.max(histogram)) # returns bin in y with max value
        'neventsToRemove = neventsToRemove - np.max(histogram)
        'histogram = np.delete(histogram, i) # removes ith bin from y array
        'binsToRemove = binsToRemove - 1

    Resolution = 1000*binsToRemove*binwidth/2.
    'band[0].append(Resolution)
    'print("Resolution for %s: %f %f %e" % (band[1], Resolution*1000))

ImportError: cannot o
<ipython-input-12-5a0b>
51148
51149
51150
51151
51152
ImportError: cannot o
```

# Conclusion

» **Keeping your code and folders organized pays off!**

- saves time
- prevents unnecessary moments of despair
- makes it easier to share the code with others

» **There is no "one and only correct way"**

- try what works for you...
- ...but do not be afraid to adopt others' habits and tricks

No need to learn all thing the hard way



huge messy code  
with no comments  
and one output dump  
folder



couple of codes with  
clear subsections  
and organized  
structure of output  
folders

Good luck with your codes!