trigger-studies report

Generated by Doxygen 1.8.11

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Namespace Documentation

2.1 dataModule Namespace Reference

module to manage the data used for trigger efficiency plots

Functions

- def loadData (dataset, limits=None)

  loads data; runs merge() before loading
- def getData (dataset, limits=None)

  loads data; use loadData() to process new files from the IN folder
- def merge (dataset)

  splits ∗.csv files from the IN folder into files with there runnumber in the data/dataset/ folder; after succesfull run the files are moved from the IN folder to data/raw
- def save (dtset, event, header, mask)

  write single event to file
- def getRunlist (data)

  generates a runlist containing all runs present in the dataset
- def printRunlist (data)

  prints runlist; the list is generated using getRunlist()
- def getLumi (data, path='data/lumi.csv')

  calculates luminosity of data

2.1.1 Detailed Description

module to manage the data used for trigger efficiency plots

Note

requires numpy and pandas

2.1.2 Function Documentation

2.1.2.1 def dataModule.getData (*dataset, limits =* `None`)

loads data; use loadData() to process new files from the IN folder

Parameters

dataset	name of the dataset to load; there should be a similar names folder in data
limits	(optional) list containing the lower and upper limit for the runnumber to load

Return values

pandas_DataFrame	loaded data

2.1.2.2 def dataModule.getLumi (*data, path* = `'data/lumi.csv'`)

calculates luminosity of data

Parameters

data	pandas_DataFrame to calculate luminosity of
path	(optional) path to lumifile e.g. generated with BRIL

Return values

float	luminosity in 1/fb

2.1.2.3 def dataModule.getRunlist (*data*)

generates a runlist containing all runs present in the dataset

Parameters

data	pandas_DataFrame like loaded with this module

Return values

numpy_array	sorted runlist

2.1.2.4 def dataModule.loadData (*dataset, limits* = `None`)

loads data; runs merge() before loading

Parameters

dataset	name of the dataset to load; there should be a similar names folder in data
limits	(optional) list containing the lower and upper limit for the runnumber to load

Return values

pandas_DataFrame	loaded data

2.1.2.5 def dataModule.merge (*dataset*)

splits ∗.csv files from the IN folder into files with there runnumber in the data/dataset/ folder; after succesfull run the files are moved from the IN folder to data/raw

Parameters

dataset	name of the dataset to merge (the file in the folder IN should contain this in their filename)

2.1.2.6 def dataModule.printRunlist (*data*)

prints runlist; the list is generated using getRunlist()

Parameters

data	pandas_DataFrame like loaded with this module

2.1.2.7 def dataModule.save (*dtset, event, header, mask*)

write single event to file

Parameters

dtset	name of the dataset
event	array containing data of one event
header	header for the file (only used if file does not exist)
mask	mask to write event in file, e.g. mask='{:.3f}, {:.3f}, {:.3f}, {:.0f}, {:.0f}'

2.2 triggerplotModule Namespace Reference

module to generate trigger efficiency plots

Functions

- def clearfile (path)

  clears a file; usually called before generating plots.
- def write2tex (txt, texpath)

  writes text to textfile

- def makeSlide (name, texpath, caption='')

  writes LaTeX formated slide with graphic to textfile
- def getError (k, n, gamma=0.682)

  function to calculate the asymmetric error for the trigger efficiency using Clopper-Pearson interval like defined in https://de.wikipedia.org/wiki/Konfidenzintervall_f%C3%BCr_die_Erfolgswahrscheinlichkeit_der_Binomialverteilung
- def getEfficiency (data, trigger, quant, denominator)

  calculates efficiency
- def doEffPlot (dataset, trigger, quant, texpath, fit=False, x0=[0.9, mask='')

  generates a trigger efficiency plot for list of triggers on different subsets and writes a LaTeX formatted slide to textfile
- def doFit (xdata, ydata, sigma, x0, cuteff=0.99)

  fits a modified error function to data
- def do2DPlot (dataset, trigger, quant1, quant2, texpath, cuts=None, mask='')

  generates 2D trigger efficiency plots

Variables

- float fitthresh = 0.8

  y-threshhold for datapoints to be considered in fitting
- string worklabel = ''

  label shown in the top left corner of generated plots

2.2.1 Detailed Description

module to generate trigger efficiency plots

Note

requires numpy, scipy and matplotlib

2.2.2 Function Documentation

2.2.2.1 def triggerplotModule.clearfile (*path*)

clears a file; usually called before generating plots.

Parameters

path	filepath of the file to clear

2.2.2.2 def triggerplotModule.do2DPlot (*dataset, trigger, quant1, quant2, texpath, cuts* = None, *mask* = '')

generates 2D trigger efficiency plots

Parameters

dataset	same as dataset in doEffPlot(), but subsets are combined and not shown separately

Parameters

trigger	label of trigger
quant1	dict used for x axis (same format as quant in doEffPlot())
quant2	dict used for y axis (same format as quant in doEffPlot())
texpath	path to textfile
cuts	(optional) shows cuts with red lines in plot; e.g. [[-1, 10000],[65, 105]]
mask	(optional) additional mask to apply to data

2.2.2.3 def triggerplotModule.doEffPlot (*dataset, trigger, quant, texpath, fit* = `False`, *x0* = `[0.9`, *mask* = `''`)

generates a trigger efficiency plot for list of triggers on different subsets and writes a LaTeX formatted slide to textfile

Parameters

dataset	dict with entries: data: Pandas dataframe loaded with the dataModule label: label of the dataset to use for filename key: key where the subsets are stored sets: list with labels of the subsets denom: label of the denominator or combination of denominator e.g. data={'data': data, 'label': 'SingleMuon', 'key': 'dataset', 'sets': ['SingleMuon-postfix'], 'denom': 'Mu50_OR_IsoMu27'}
trigger	list of triggers to generate efficiency plots of
quant	dict with entries: key: key of the quantity used as x-axis label: label used for x-axis label; can use LaTeX commands limits: list with lower, upper limits and stepsize for bins e.g. quant={'key': 'Mjj', 'label': 'invariant dijetmass M_{jj} in GeV', 'limits': [500, 2000, 30]}
texpath	path to textfile
fit	(optional) enables fit of modified errorfunction
x0	(optional) startparameter for fit
mask	(optional) additional mask to apply to data

2.2.2.4 def triggerplotModule.doFit (*xdata, ydata, sigma, x0, cuteff* = `0.99`)

fits a modified error function to data

Parameters

xdata	x value of datapoints
ydata	y value of datapoints
sigma	asymmetric uncertainity
x0	startparameter for fit
cuteff	(optional) value used to determine the plateau point (efficiency>cuteff)

Return values

numpy_array	x value of fit
numpy_array	y value of fit
str	fitlabel
bool	fit succeded
list	best parameter estimation
list	uncertainity of parameter estimation
float	x value where fit reaches cuteff

2.2.2.5 def triggerplotModule.getEfficiency (*data, trigger, quant, denominator*)

calculates efficiency

Parameters

data	data
trigger	label of trigger or trigger combination to calculate efficiency of
quant	quantity used for x-axis
denominator	denominator used for filtering

Return values

numpy_array	center of bin
numpy_array	efficiency
numpy_array	asymmetric error interval

2.2.2.6 def triggerplotModule.getError (*k, n, gamma* = `0.682`)

function to calculate the asymmetric error for the trigger efficiency using Clopper-Pearson interval like defined in https://de.wikipedia.org/wiki/Konfidenzintervall_f%C3%BCr_die_↩Erfolgswahrscheinlichkeit_der_Binomialverteilung

Parameters

k	number of hits
n	number of experiments
gamma	(optional) confidence level for interval; default is one sigma (68,2%)

Return values

list	containing the lower and upper error

2.2.2.7 def triggerplotModule.makeSlide (*name, texpath, caption = ' '*)

writes LaTeX formated slide with graphic to textfile

Parameters

name	filename of the graphic
texpath	filepath of textfile
caption	(optional) caption of the slide

2.2.2.8 def triggerplotModule.write2tex (*txt, texpath*)

writes text to textfile

Parameters

txt	text to write in file
textpath	path of textfile