

trigger-studies report

Generated by Doxygen 1.8.11

Contents

1	Namespace Index	1
1.1	Packages	1
2	Namespace Documentation	3
2.1	dataModule Namespace Reference	3
2.1.1	Detailed Description	3
2.1.2	Function Documentation	3
2.1.2.1	getData(dataset, limits=None)	3
2.1.2.2	getLumi(data, path='data/lumi.csv')	4
2.1.2.3	getRunlist(data)	4
2.1.2.4	loadData(dataset, limits=None)	4
2.1.2.5	merge(dataset)	5
2.1.2.6	printRunlist(data)	5
2.1.2.7	save(dtset, event, header, mask)	5
2.2	triggerplotModule Namespace Reference	5
2.2.1	Detailed Description	6
2.2.2	Function Documentation	6
2.2.2.1	clearfile(path)	6
2.2.2.2	do2DPlot(dataset, trigger, quant1, quant2, texpath, cuts=None, mask="")	6
2.2.2.3	doEffPlot(dataset, trigger, quant, texpath, fit=False, x0=[0.9, mask="")	7
2.2.2.4	doFit(xdata, ydata, sigma, x0, cteff=0.99)	7
2.2.2.5	getEfficiency(data, trigger, quant, denominator)	8
2.2.2.6	getError(k, n, gamma=0.682)	8
2.2.2.7	makeSlide(name, texpath, caption="")	9
2.2.2.8	write2tex(txt, texpath)	9

Chapter 1

Namespace Index

1.1 Packages

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

dataModule	
Module to manage the data used for trigger efficiency plots	3
triggerplotModule	
Module to generate trigger efficiency plots	5

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

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

Return values

<i>pandas_DataFrame</i>	loaded data
-------------------------	-------------

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

calculates luminosity of data

Parameters

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

Return values

<i>float</i>	luminosity in 1/fb
--------------	--------------------

2.1.2.3 `def dataModule.getRunlist (data)`

generates a runlist containing all runs present in the dataset

Parameters

<i>data</i>	pandas_DataFrame like loaded with this module
-------------	---

Return values

<i>numpy_array</i>	sorted runlist
--------------------	----------------

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

loads data; runs `merge()` before loading

Parameters

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

Return values

<code>pandas_DataFrame</code>	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

<code>dataset</code>	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

<code>data</code>	<code>pandas_DataFrame</code> like loaded with this module
-------------------	--

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

write single event to file

Parameters

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

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 formatted 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, cteff=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-threshold for datapoints to be considered in fitting
- string `worklabel` = 'CMS private work'
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

<code>path</code>	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

<code>dataset</code>	same as dataset in <code>doEffPlot()</code> , but subsets are combined and not shown separately
----------------------	---

Parameters

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

<i>dataset</i>	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. <code>data={'data': data, 'label': 'SingleMuon', 'key': 'dataset', 'sets': ['SingleMuon-postfix'], 'denom': 'Mu50_OR_IsoMu27'}</code>
<i>trigger</i>	list of trigger to generate efficiency plots of
<i>quant</i>	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. <code>quant={'key': 'Mjj', 'label': 'invariant dijetmass M_{jj} in GeV', 'limits': [500, 2000, 30]}</code>
<i>texpath</i>	path to textfile
<i>fit</i>	(optional) enables fit of modified errorfunction
<i>x0</i>	(optional) startparameter for fit
<i>mask</i>	(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

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

Return values

<i>numpy_array</i>	x value of fit
<i>numpy_array</i>	y value of fit
<i>str</i>	fitlabel
<i>bool</i>	fit succeeded
<i>list</i>	best parameter estimation
<i>list</i>	uncertainty of parameter estimation
<i>float</i>	x value where fit reaches cteff

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

calculates efficiency

Parameters

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

Return values

<i>numpy_array</i>	center of bin
<i>numpy_array</i>	efficiency
<i>numpy_array</i>	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

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

Return values

<i>list</i>	containing the lower and upper error
-------------	--------------------------------------

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

writes LaTeX formatted slide with graphic to textfile

Parameters

<i>name</i>	filename of the graphic
<i>texpath</i>	filepath of textfile
<i>caption</i>	(optional) caption of the slide

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

writes text to textfile

Parameters

<i>txt</i>	text to write in file
<i>texpath</i>	path of textfile

