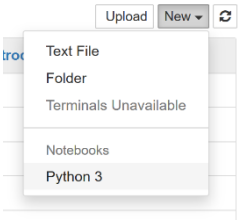
Importing a CMS open data file and creating overlaid histograms to analyze data

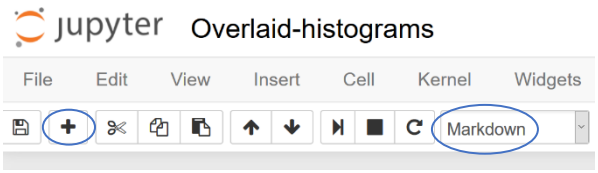
This exercise uses Jupyter Notebook which can be downloaded for free by installing Anaconda, available at https://www.continuum.io/downloads to analyze CMS data sets from the CERN Open Data Portal, http://opendata.cern.ch/.

Select a data file to analyze from http://opendata.cern.ch/record/545. For the example, download Dimuon_DoubleMu.cvs. Note, there are other files available, but the file must be comma separated values (.csv). Download the selected file, noting its exact location and file name. If the file name has spaces between the words replace them with underscores.



Open the Anaconda - Navigator. Launch Jupyter notebook. Select “New” from menu at right and then click “Python 3”. A blank notebook should appear to your browser.

Boxes that give instruction or explanation but not code, can be entered by pressing “+” button from the toolbar and selecting “Markdown” from the dropdown menu. To enter code, select "Code" from the dropdown menu. Cell content may be copied from other sources, pasted, and edited.



First, you will need to import the packages pandas and matplotlib.pyplot to be able to read files and plot histogram.

In [1]:
```
import pandas
import matplotlib.pyplot as plt
%matplotlib inline
```

After entering a code box, press *Ctrl+Enter* at the same time to run the code. An asterisk will appear in the *In[ ]* while the command is being processed. Wait until a number appears in that location before proceeding. Possible error messages will appear in pink and will indicate information regarding the error.

To use the file you have downloaded, it must be saved into a variable. Type the file location and name exactly as it appears on your computer. In this example, the file is located one directory up in a folder named *Data*. The path is therefore *../Data/Dimuon_DoubleMu.csv*.

Save the data into the variable *dataset* and check the contents of the first 5 rows.

In [2]:
```
dataset = pandas.read_csv('../Data/Dimuon_DoubleMu.csv')

dataset.head()
```

If the command is written properly, a data table will be generated.

For the given example, the invariant mass is of particular interest. Save the invariant mass column from the *dataset* into a variable *invariant_mass* by using the column heading as it appears in the table, e.g. *M*. If your data set does not contain invariant masses by default you must first calculate them. Plot a histogram, stating which variable to plot, setting the number of bins and range. In the example below, we use 50 bins and plot from 0 to 200 GeV.

```
In [3]: invariant_mass = dataset['M']

        plt.hist(invariant_mass, bins=50, range=(0,200))
        plt.show()
```

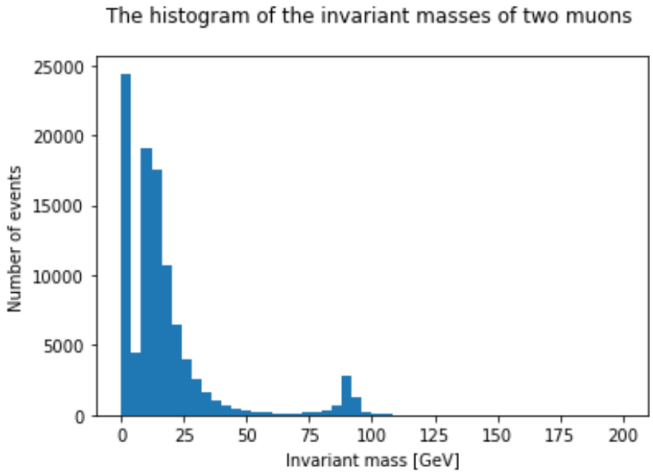
This plot was primarily to check our data and plotting commands. The number of bins and range can be varied in order to analyze the data more clearly.

Here are commands to replot the graph with a title and axis labels.

```
In [4]: plt.xlabel('Invariant mass [GeV]')
        plt.ylabel('Number of events')
        plt.title('The histogram of the invariant masses of two muons \n')

        plt.hist(invariant_mass, bins=50, range=(0,200))
        plt.show()
```

A sample histogram is shown below. The graph demonstrates background events below approximately 50 GeV and a peak at approximately 90 GeV.



Mathematical operations, such as addition or subtraction, may be performed on the data by defining new variables which allows the data to be further sorted.

In the example below, the original data is divided into two new data sets based on the energy of the collision. Each data set is given a name and organized in this case by high energy (>150 GeV), and low energy (<150 GeV).

```
In [5]: newsethighE = dataset[dataset.E1+dataset.E2>150]
        newsetlowE = dataset[dataset.E1+dataset.E2<150]
```

The new data sets can be plotted separately as was done previously or on one plot. The two histograms can be overlaid by adjusting the transparency using the alpha command. Labels for each data set are included in the legend located in the upper right corner. We can also set the range to focus on the event of interest.

```
In [6]: plt.xlabel('Invariant mass [GeV]')
        plt.ylabel('Number of events')
        plt.title('The invariant masses of two muons comparing high and low energy\n')
        plt.hist(newsetlowE ['M'], bins=50, range=(80,100),alpha=0.5, label='Low E')
        plt.hist(newsethighE ['M'], bins=50, range=(80,100),alpha=0.5, label='High E')
        plt.legend (loc='upper right')
        plt.show()
```

The final output is shown below.