In [1]:

```
import pandas as  pd
tel_data = pd.read_csv("telescope_data.csv")
```

In [2]:

```
tel_data.head()
```

Out[2]:

	Unnamed: 0	fLength	fWidth	fSize	fConc	fConc1	fAsym	fM3Long	fM3Trans
0	0	28.7967	16.0021	2.6449	0.3918	0.1982	27.7004	22.0110	-8.2027
1	1	31.6036	11.7235	2.5185	0.5303	0.3773	26.2722	23.8238	-9.9574
2	2	162.0520	136.0310	4.0612	0.0374	0.0187	116.7410	-64.8580	-45.2160
3	3	23.8172	9.5728	2.3385	0.6147	0.3922	27.2107	-6.4633	-7.1513
4	4	75.1362	30.9205	3.1611	0.3168	0.1832	-5.5277	28.5525	21.8393

change columns names lower letters

In [3]:

```
tel_data.columns = [cols.lower() for cols in tel_data.columns]
```

In [4]:

```
tel_data
```

Out[4]:

	unnamed: 0	flength	fwidth	fsize	fconc	fconc1	fasym	fm3long	fm3
0	0	28.7967	16.0021	2.6449	0.3918	0.1982	27.7004	22.0110	-8
1	1	31.6036	11.7235	2.5185	0.5303	0.3773	26.2722	23.8238	-9
2	2	162.0520	136.0310	4.0612	0.0374	0.0187	116.7410	-64.8580	-45
3	3	23.8172	9.5728	2.3385	0.6147	0.3922	27.2107	-6.4633	-
4	4	75.1362	30.9205	3.1611	0.3168	0.1832	-5.5277	28.5525	21
...	...	...	...	...	...	...	...	...	
19015	19015	21.3846	10.9170	2.6161	0.5857	0.3934	15.2618	11.5245	2
19016	19016	28.9452	6.7020	2.2672	0.5351	0.2784	37.0816	13.1853	-2
19017	19017	75.4455	47.5305	3.4483	0.1417	0.0549	-9.3561	41.0562	-9
19018	19018	120.5135	76.9018	3.9939	0.0944	0.0683	5.8043	-93.5224	-63
19019	19019	187.1814	53.0014	3.2093	0.2876	0.1539	-167.3125	-168.4558	31

19020 rows × 12 columns

In [5]:

```
tel_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19020 entries, 0 to 19019
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   unnamed: 0  19020 non-null  int64  
 1   flength      19020 non-null  float64
 2   fwidth       19020 non-null  float64
 3   fsize        19020 non-null  float64
 4   fconc        19020 non-null  float64
 5   fconc1       19020 non-null  float64
 6   fasym        19020 non-null  float64
 7   fm3long      19020 non-null  float64
 8   fm3trans     19020 non-null  float64
 9   falpha       19020 non-null  float64
 10  fdist        19020 non-null  float64
 11  class        19020 non-null  object 
dtypes: float64(10), int64(1), object(1)
memory usage: 1.7+ MB
```

In [6]:

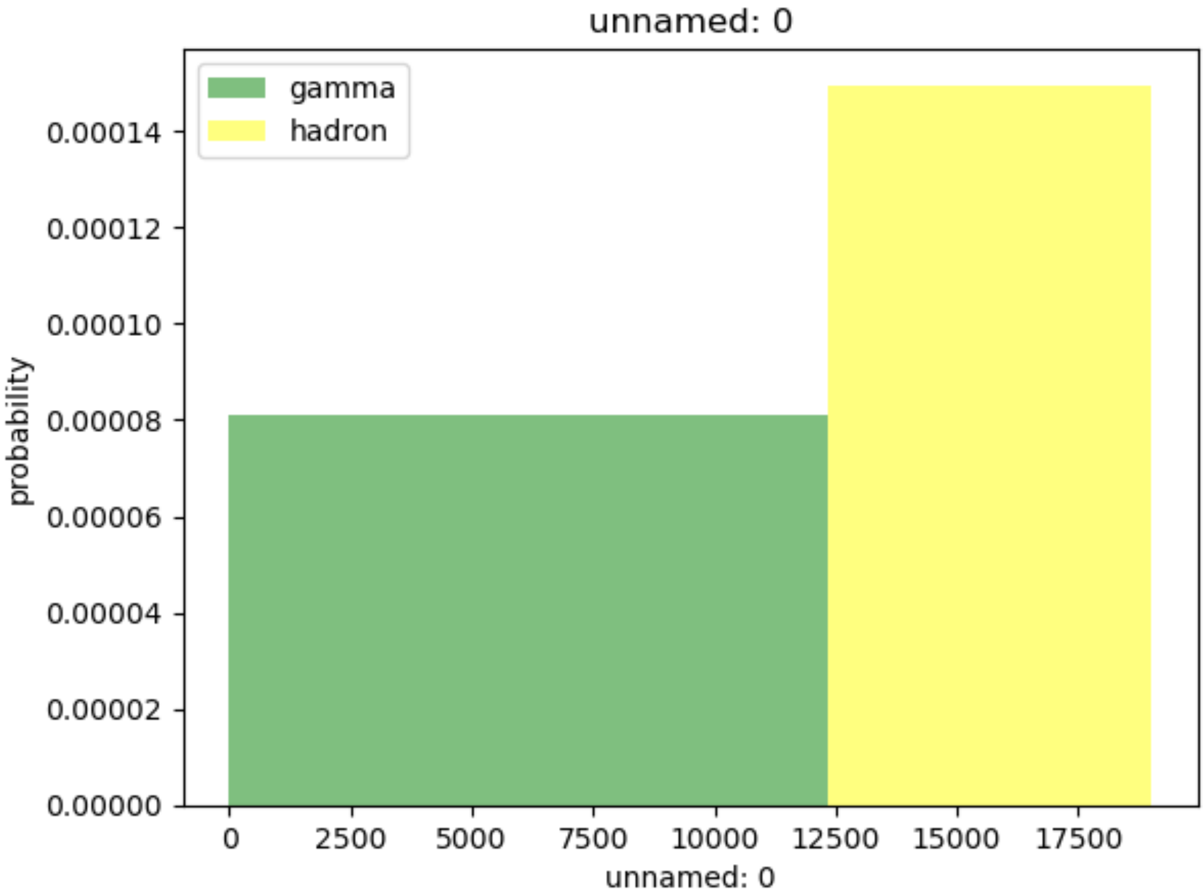
```
tel_data.describe()
```

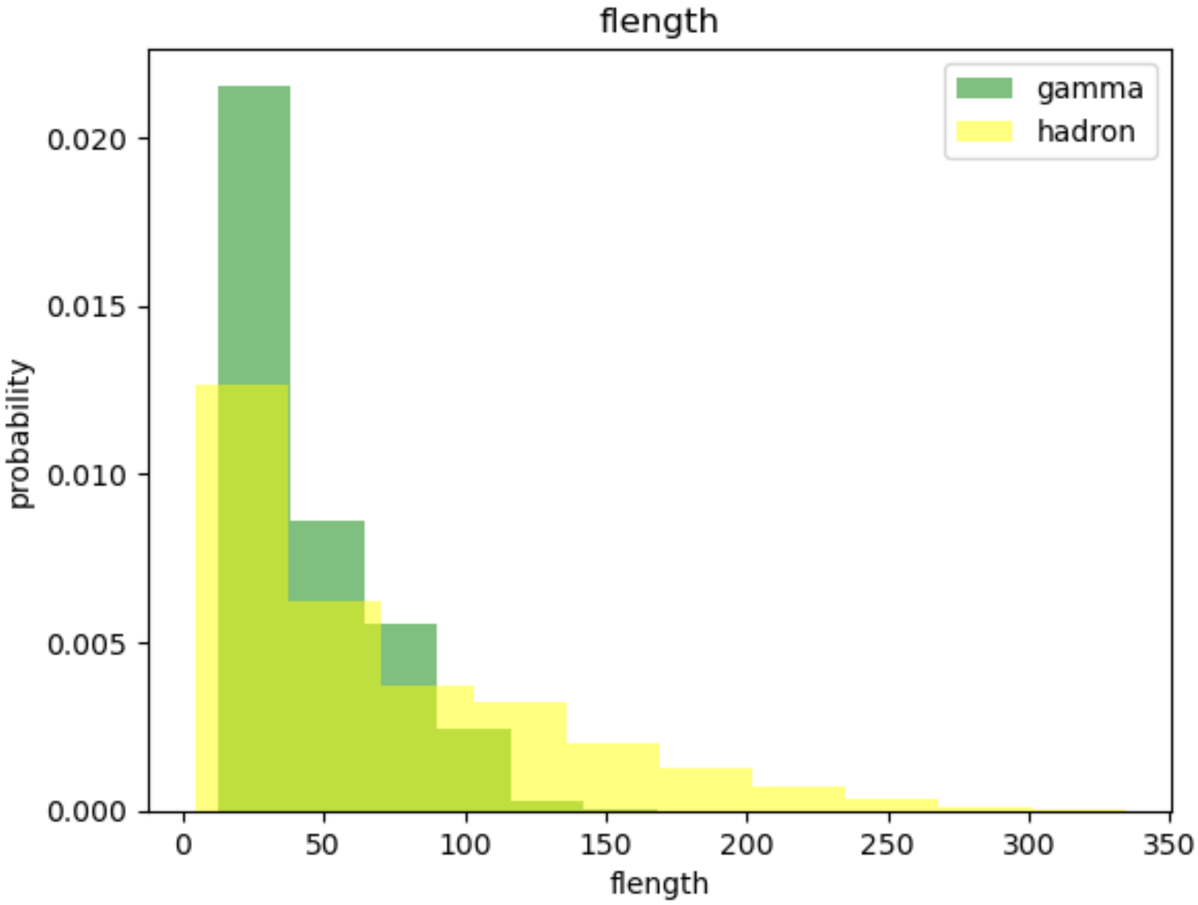
Out[6]:

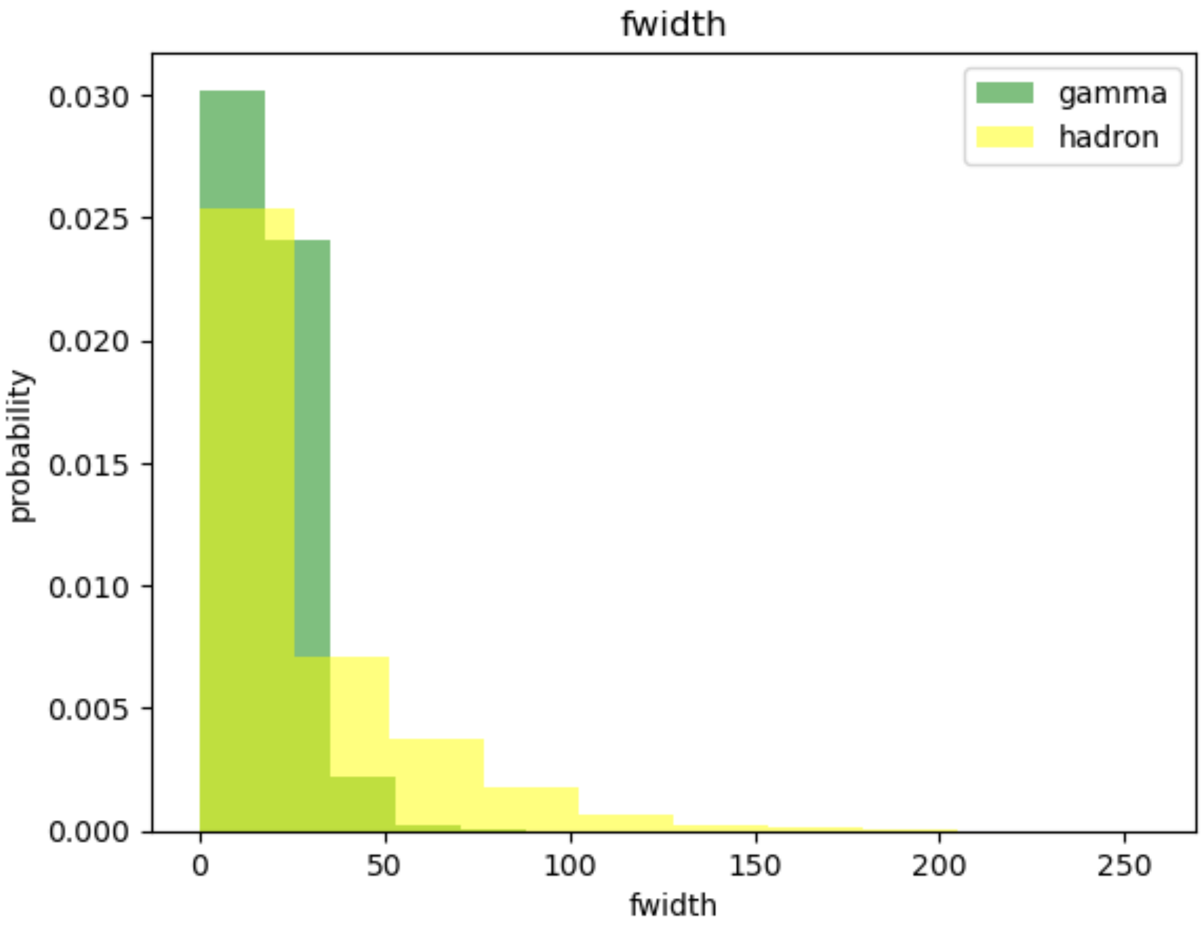
	unnamed: 0	flength	fwidth	fsize	fconc	f
count	19020.000000	19020.000000	19020.000000	19020.000000	19020.000000	19020.0
mean	9509.500000	53.250154	22.180966	2.825017	0.380327	0.2
std	5490.745396	42.364855	18.346056	0.472599	0.182813	0.
min	0.000000	4.283500	0.000000	1.941300	0.013100	0.0
25%	4754.750000	24.336000	11.863800	2.477100	0.235800	0.1
50%	9509.500000	37.147700	17.139900	2.739600	0.354150	0.1
75%	14264.250000	70.122175	24.739475	3.101600	0.503700	0.2
max	19019.000000	334.177000	256.382000	5.323300	0.893000	0.6

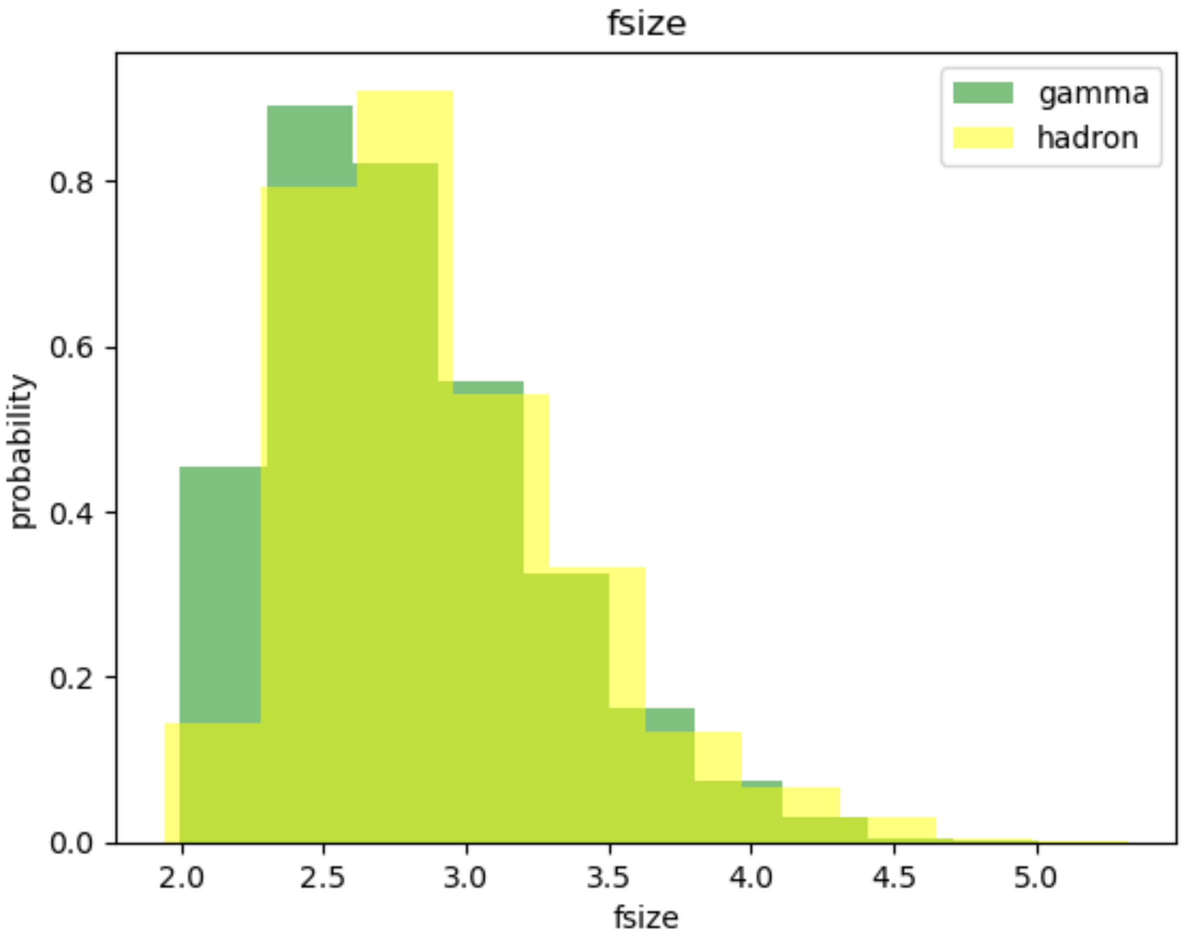
In [7]:

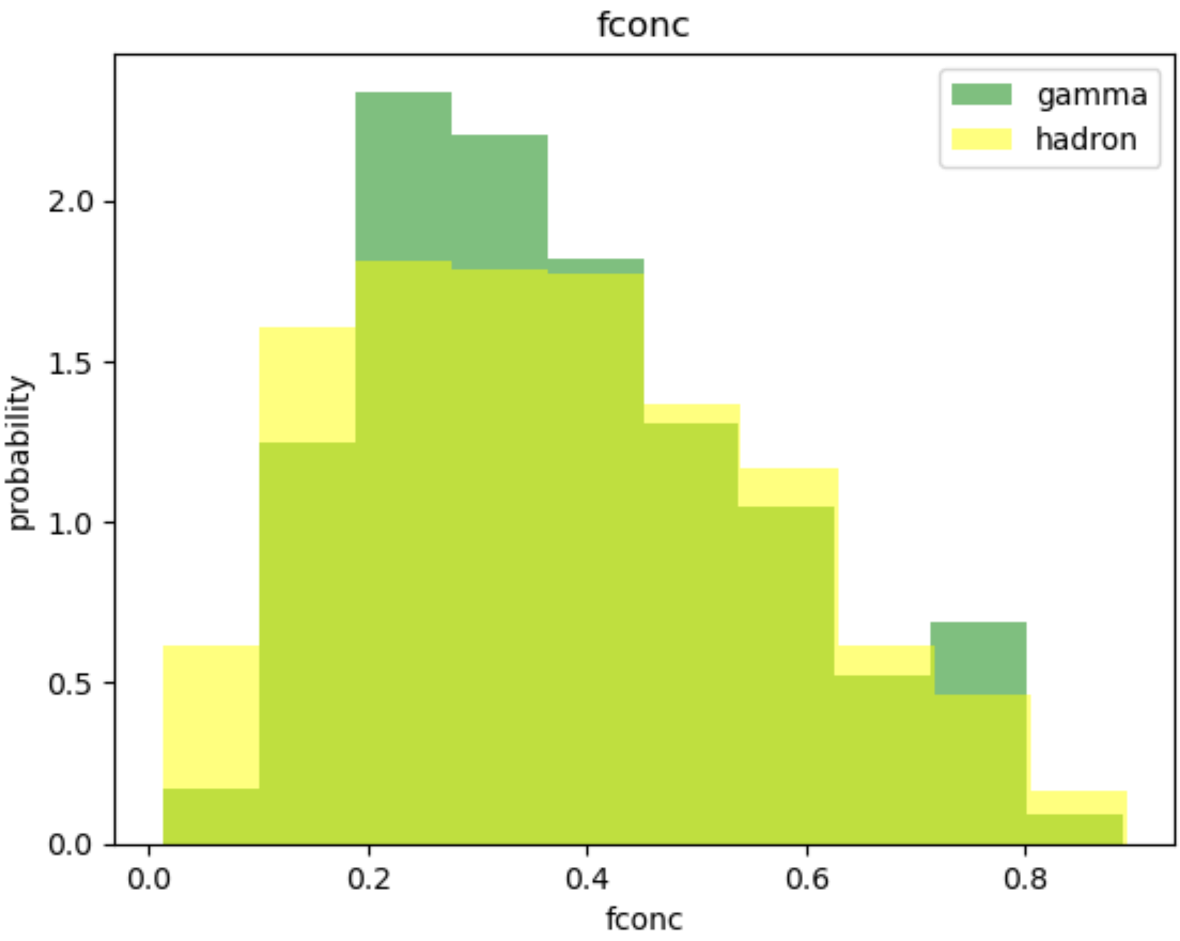
```
import matplotlib.pyplot as plt
for label in (tel_data.columns):
    plt.hist(tel_data[tel_data["class"]=="g"][label],color="green",label=
    plt.hist(tel_data[tel_data["class"]=="h"][label],color="yellow",label
    plt.title(label)
    plt.ylabel("probability")
    plt.xlabel(label)
    plt.legend()
    plt.show()
```

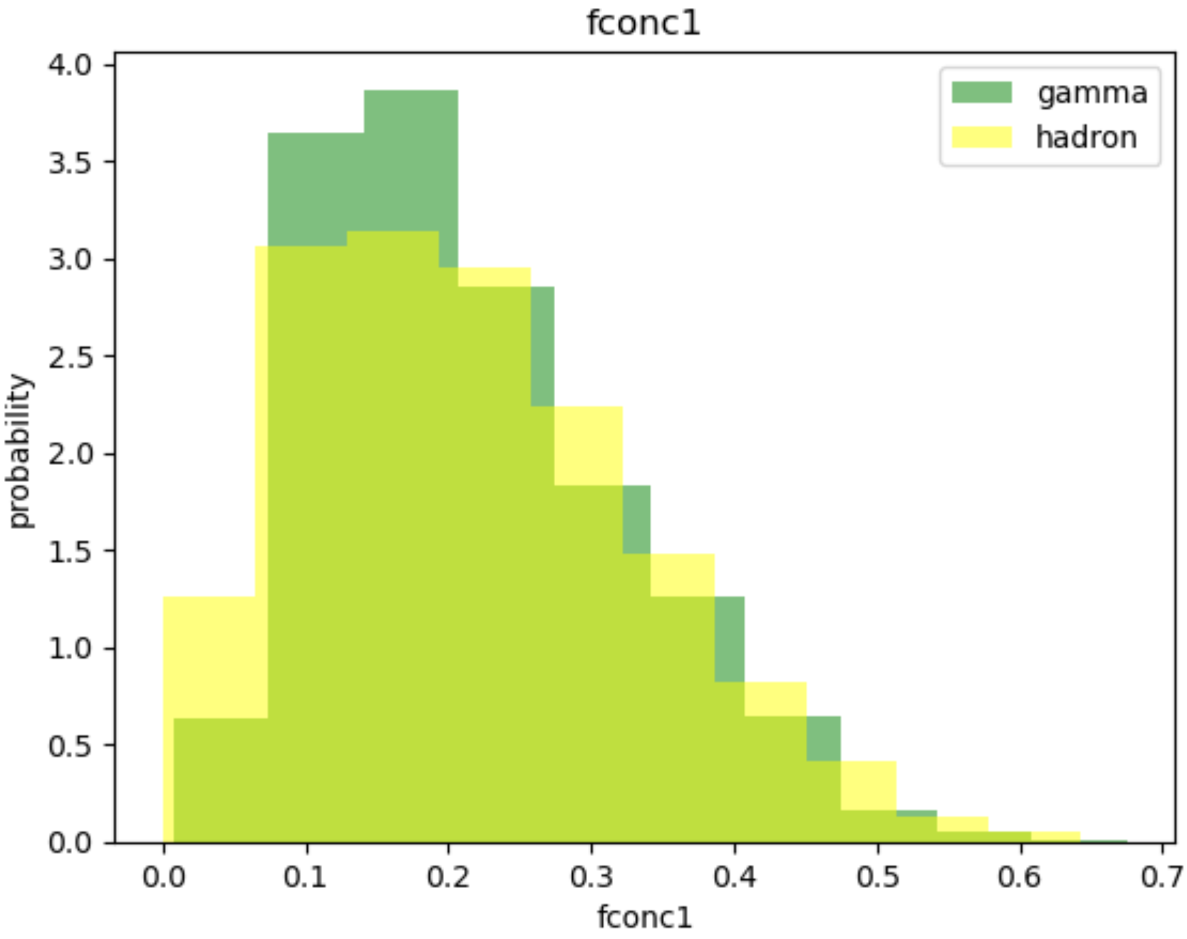

unnamed: 0
gamma
hadron
probability
0.00000
0.00002
0.00004
0.00006
0.00008
0.00010
0.00012
0.00014
0
2500
5000
7500
10000
12500
15000
17500
unnamed: 0

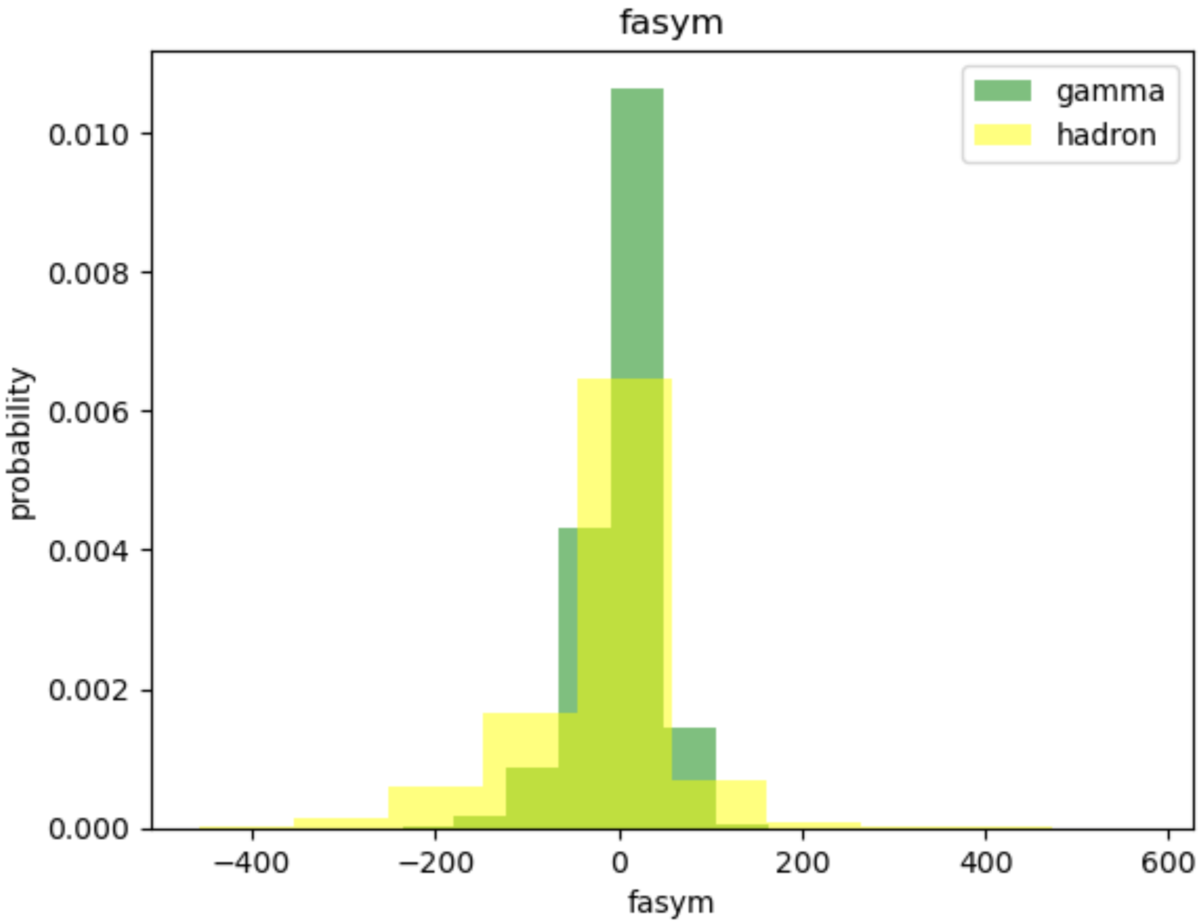

flength
gamma
hadron
probability
0.000
0.005
0.010
0.015
0.020
0
50
100
150
200
250
300
350
flength

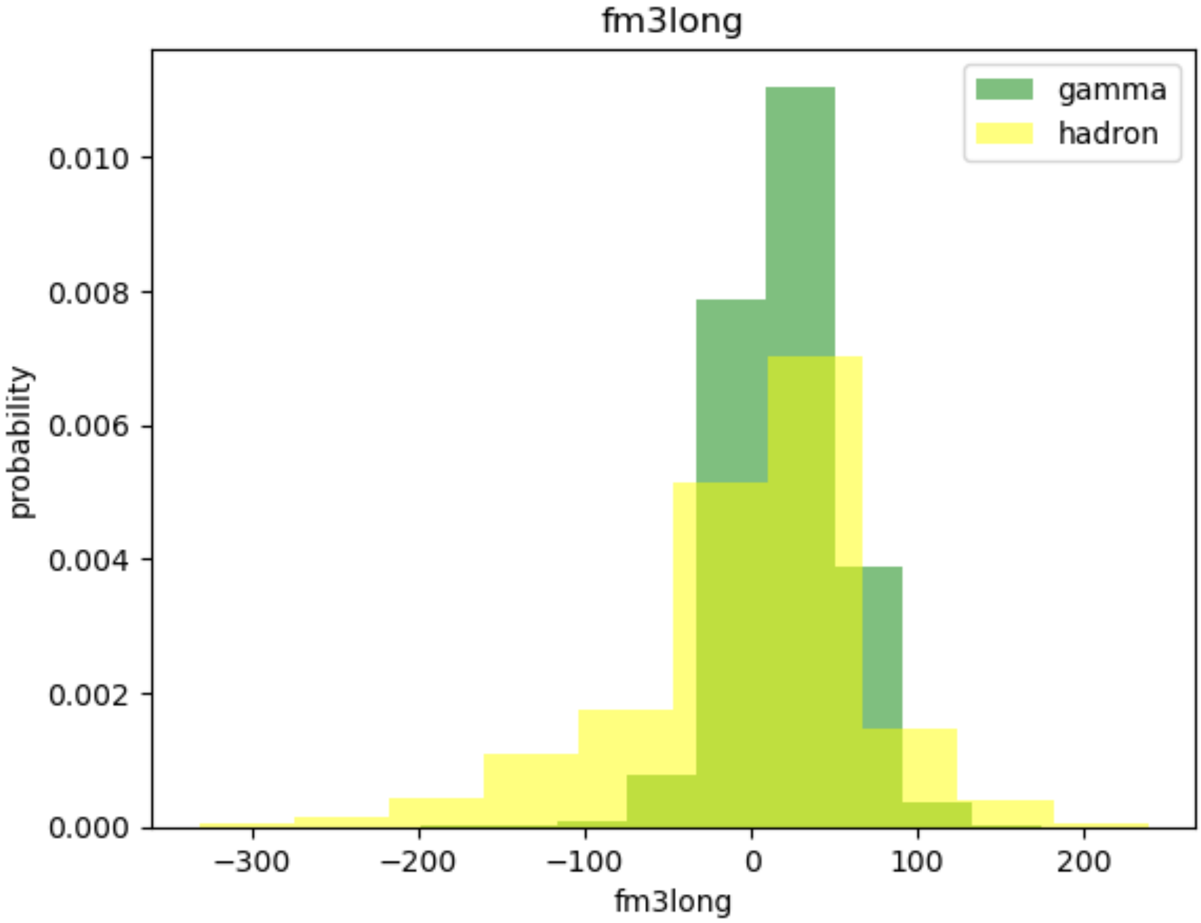

fwidth
gamma
hadron
probability
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
50
100
150
200
250
fwidth

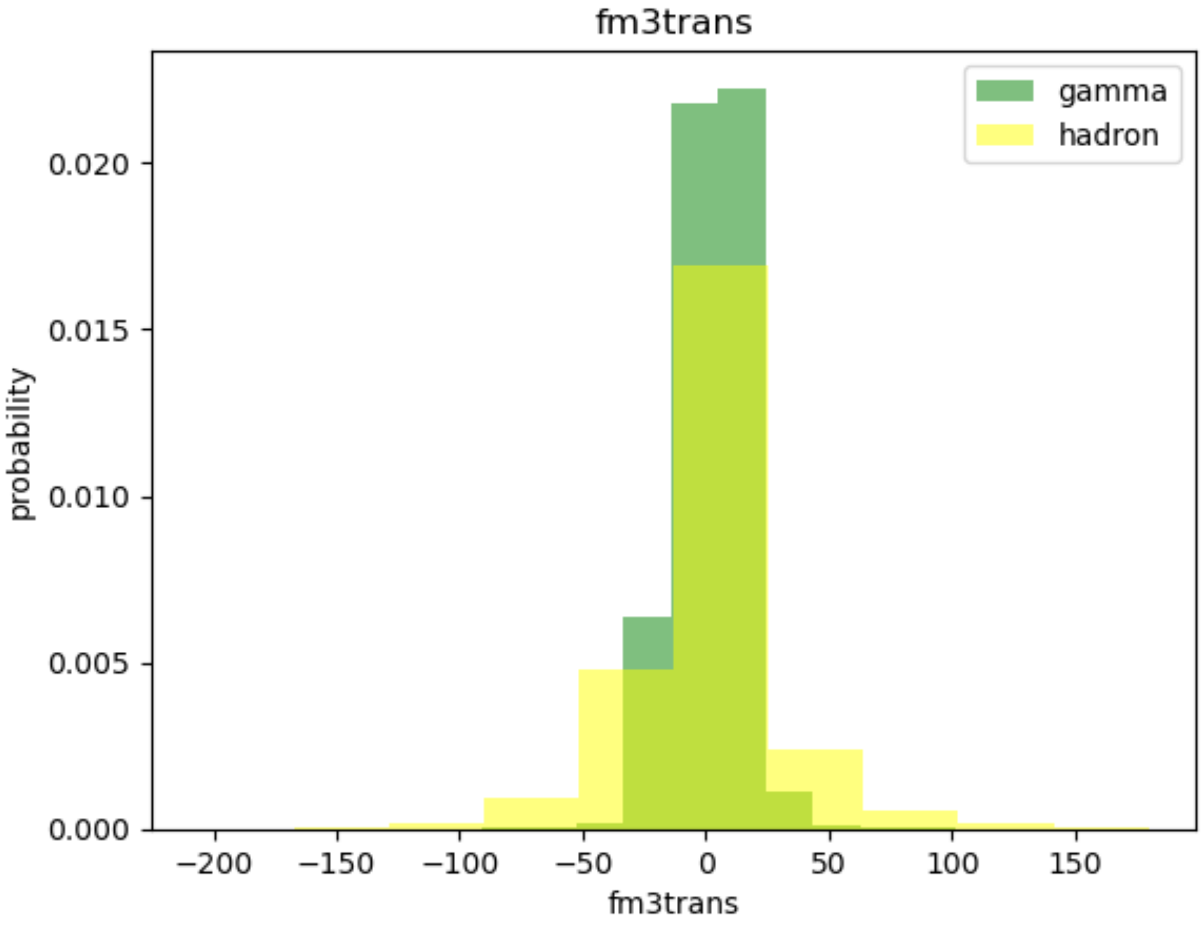

fsize
gamma
hadron
probability
0.8
0.6
0.4
0.2
0.0
2.0
2.5
3.0
3.5
4.0
4.5
5.0
fsize

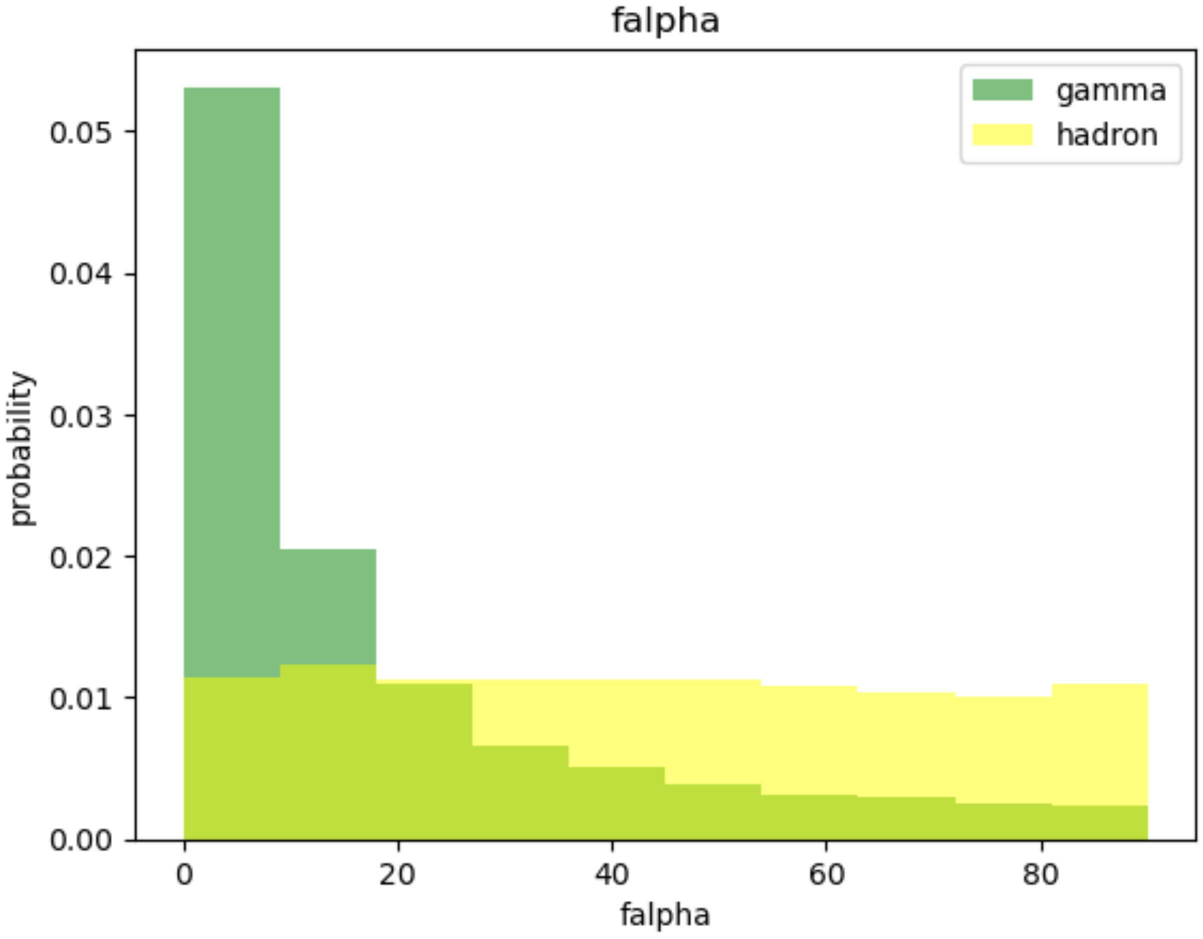

fconc
gamma
hadron
probability
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
fconc

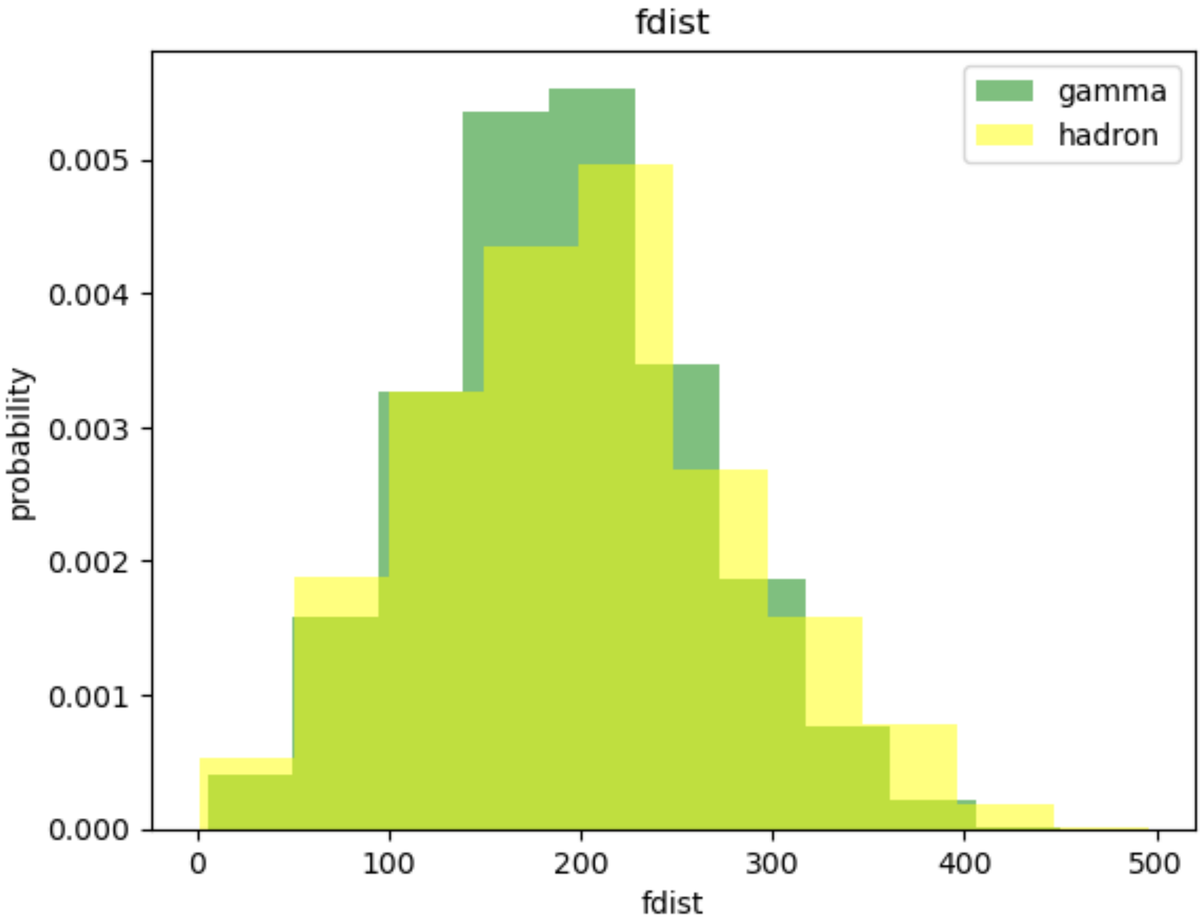

fconc1
gamma
hadron
probability
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
fconc1

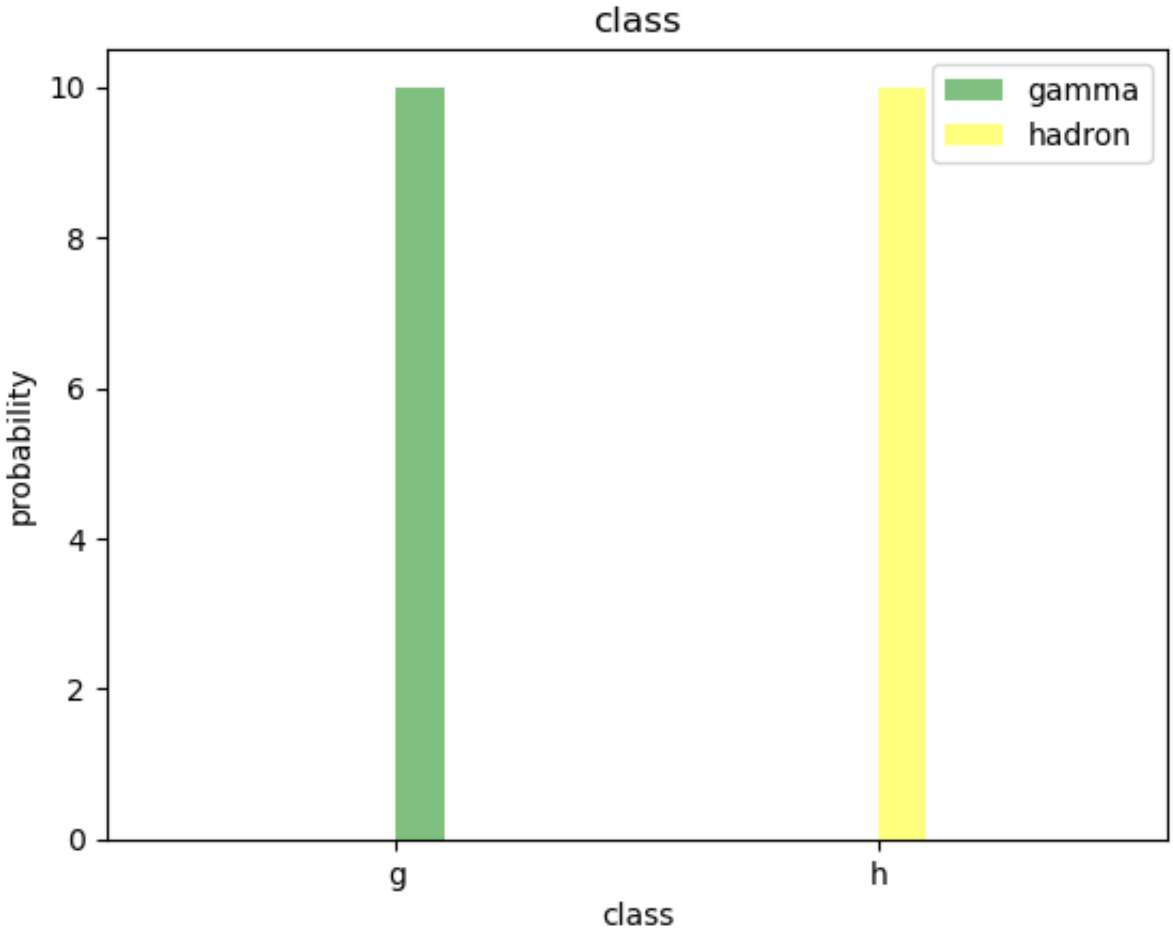

fasym
gamma
hadron
probability
0.000
0.002
0.004
0.006
0.008
0.010
−400
−200
0
200
400
600
fasym


fm3long
gamma
hadron
probability
0.000
0.002
0.004
0.006
0.008
0.010
−300
−200
−100
0
100
200
fm3long


fm3trans
gamma
hadron
probability
0.000
0.005
0.010
0.015
0.020
−200
−150
−100
−50
0
50
100
150
fm3trans


falpha
gamma
hadron
probability
0.00
0.01
0.02
0.03
0.04
0.05
0
20
40
60
80
falpha


fdist
probability
0.000
0.001
0.002
0.003
0.004
0.005
0
100
200
300
400
500
fdist
🟩 gamma
🟨 hadron


class
probability
0
2
4
6
8
10
g
h
class
🟩 gamma
🟨 hadron

In [8]:
```
from sklearn.model_selection import train_test_split
ds_train , ds_test = train_test_split(tel_data , test_size = 0.2)
```

In [9]:
```
print(len(ds_train))
print(len(ds_test))
```

```
15216
3804
```

In [10]:
```
print(len(ds_train[ds_train["class"]=="g"]))
print(len(ds_train[ds_train["class"]=="h"]))
```

```
9849
5367
```

resampling

In [11]:
```
pip install -U scikit-learn imbalanced-learn
```

```
Requirement already satisfied: scikit-learn in /opt/homebrew/anaconda3/li
b/python3.11/site-packages (1.4.2)
Requirement already satisfied: imbalanced-learn in /opt/homebrew/anaconda
3/lib/python3.11/site-packages (0.12.2)
Requirement already satisfied: numpy>=1.19.5 in /opt/homebrew/anaconda3/l
ib/python3.11/site-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: scipy>=1.6.0 in /opt/homebrew/anaconda3/li
b/python3.11/site-packages (from scikit-learn) (1.11.1)
Requirement already satisfied: joblib>=1.2.0 in /opt/homebrew/anaconda3/l
ib/python3.11/site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/homebrew/anac
onda3/lib/python3.11/site-packages (from scikit-learn) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

In [12]:
```
from imblearn.over_sampling import RandomOverSampler
r_sam = RandomOverSampler()
in_ds_train = ds_train.drop(columns = ["class"])
out_ds_train = ds_train["class"]
```

In [13]:
```
print(in_ds_train)
```

```
       unnamed: 0   flength   fwidth   fsize   fconc  fconc1    fasym  \
16964       16964   46.3735  22.7618  2.7980  0.3003  0.1489  12.7939   
13830       13830   34.9814  13.3332  2.5508  0.4529  0.3165 -22.4672   
5532         5532   29.4022  15.8372  2.7604  0.4236  0.2370  11.3791   
5058         5058   17.6309  10.8487  2.2227  0.5748  0.3323  -0.7018   
15927       15927  139.1430  89.1839  3.6977  0.1659  0.0938  -0.2938   
...           ...       ...      ...     ...     ...     ...      ...   
11835       11835   34.4505   9.7234  2.5145  0.5138  0.2859  36.3998   
13410       13410   21.3338  10.0117  2.2201  0.5904  0.3283   5.4090   
13314       13314   26.3861  15.1486  2.7889  0.4163  0.2447   8.0143   
15451       15451   32.7050  11.1606  2.3891  0.5847  0.3258 -15.0011   
8762         8762   79.2728  14.8430  2.6618  0.2723  0.1383  87.9695   

        fm3long  fm3trans   falpha     fdist  
16964   15.4585    6.6745  65.4746   90.0872  
13830  -30.2027   -9.7852  49.9020  278.4310  
5532   -17.9492    9.4850   3.0127  139.4030  
5058     5.3114    6.2969  12.6160  132.3030  
15927 -127.2560  -61.0442  74.7645  337.4880  
...         ...       ...      ...       ...  
11835   31.0299    5.9910  14.8240  141.0140  
13410  -16.8477   10.4250  23.4930   82.3087  
13314   21.5488   -9.4032  52.1800  211.4810  
15451    1.8022   11.7110  50.3919  184.7431  
8762    76.2804  -11.1655  12.4240  248.1300  

[15216 rows x 11 columns]
```

In [14]:

```python
print(out_ds_train)
```

```
16964    h
13830    h
5532     g
5058     g
15927    h
        ..
11835    g
13410    h
13314    h
15451    h
8762     g
Name: class, Length: 15216, dtype: object
```

(69.7459 [flength] - 53.250154 [flength] mean] / 42.364855 [staned devision]

standed scaller

In [15]:

```python
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler ()
in_ds_train = scaler.fit_transform(in_ds_train)
print(in_ds_train)
```

```
[[ 1.35478584 -0.16503933  0.02845156 ...  0.31062075  1.44558598
  -1.38106986]
 [ 0.78379494 -0.43160643 -0.48608658 ... -0.47385908  0.84974787
   1.14036147]
 [-0.72803739 -0.56215577 -0.34943813 ...  0.44457098 -0.94432841
  -0.72086016]
 ...
 [ 0.68978367 -0.63273038 -0.38701646 ... -0.45565271  0.9369086
   0.24407591]
 [ 1.07912877 -0.48487258 -0.60464984 ...  0.55066356  0.8684924
  -0.1138747 ]
 [-0.13955602  0.60478106 -0.40369368 ... -0.53964506 -0.58423368
   0.73471026]]
```

In [16]:

```
in_ds_train, out_ds_train = r_sam.fit_resample (in_ds_train,out_ds_train)
print (len(out_ds_train))
print (sum(out_ds_train =="g"))
print (sum(out_ds_train =="h"))
```

```
19698
9849
9849
```

In [17]:

```
in_ds_test = ds_test.drop(columns = ["class"])
out_ds_test = ds_test["class"]
in_ds_test = scaler.fit_transform(in_ds_test)
```

In [18]:

```
print (len(in_ds_test))
```

```
3804
```

In [19]:

```
from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(n_neighbors=1)
knn_model.fit(in_ds_train , out_ds_train)
```

Out[19]:

```
▾ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=1)
```

In [20]:

```
out_pred = knn_model.predict(in_ds_test)
```

In [21]:

```
from sklearn.metrics import accuracy_score
accuracy_score_info = accuracy_score(out_ds_test , out_pred)
accuracy_score_info
```

Out[21]:

```
0.9729232386961093
```

In [22]:

```
from sklearn.metrics import classification_report
print(classification_report(out_ds_test , out_pred))
```

```
              precision    recall  f1-score   support

           g       0.97      0.99      0.98      2483
           h       0.97      0.95      0.96      1321

    accuracy                           0.97      3804
   macro avg       0.97      0.97      0.97      3804
weighted avg       0.97      0.97      0.97      3804
```

In [ ]:

In [ ]:

In [ ]: