```
import numpy as np
from numba import njit
#from sklearn.mixture import GaussianMixture
import matplotlib.pyplot as plt
import matplotlib as mpl
from matplotlib.colors import LogNorm, Normalize
import pandas as pd
import pickle
from datetime import datetime

#from tqdm.notebook import tqdm, trange

import copy
from matplotlib import cm


from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df_bkg_linked = pd.read_hdf('/content/drive/MyDrive/Dark
```

ALL COLUMND

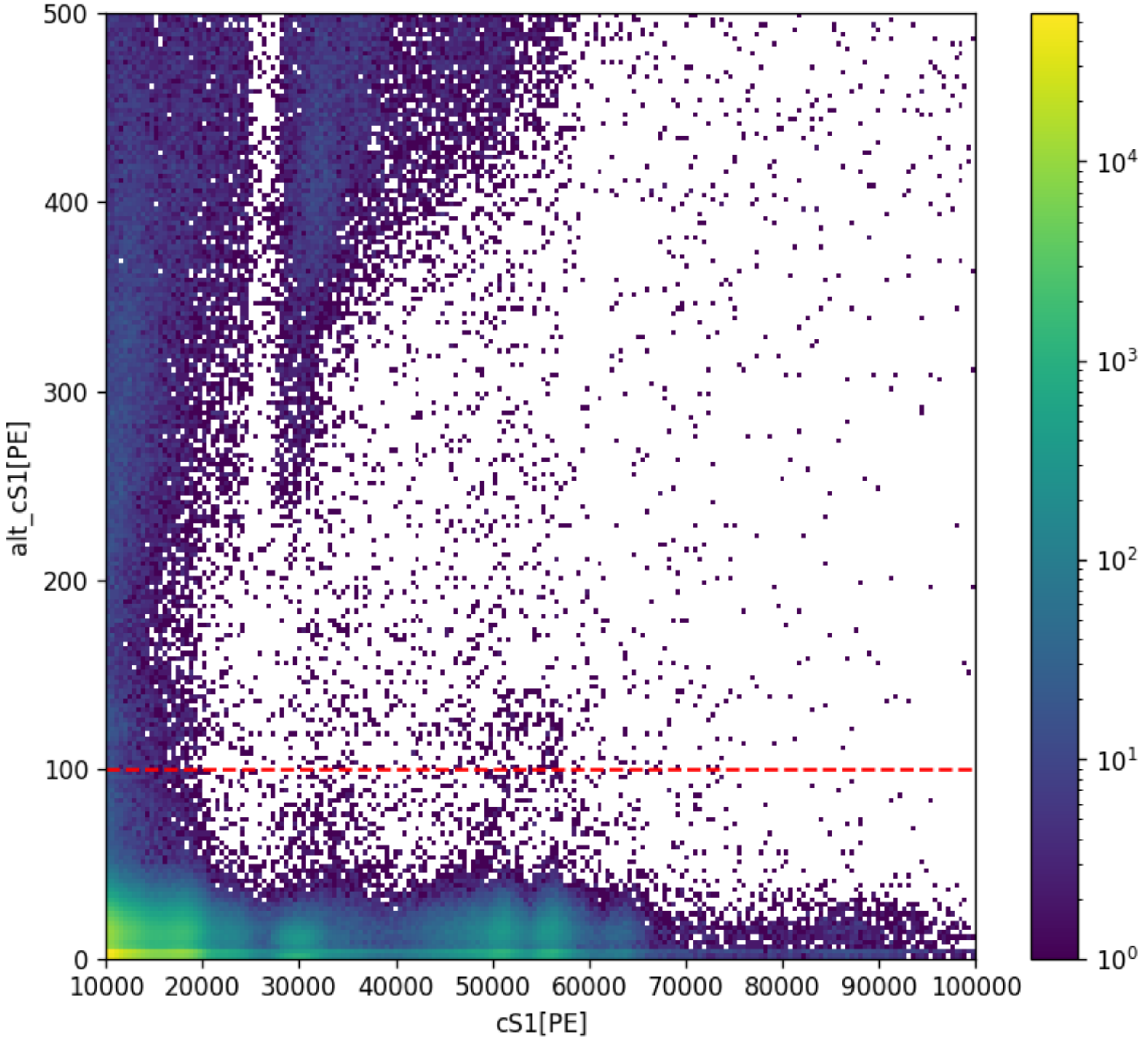
Show code

```
'\n['run_id',\n 'time',\n 'endtime',\n 'cs1',\n 'cs1_wo_timecorr',\n 'cs2_w
o_elifecorr',\n 'cs2_wo_timecorr',\n 'cs2_area_fraction_top',\n 'cs2_bottom
',\n 'cs2',\n 'alt_cs1',\n 'alt_cs1_wo_timecorr',\n 'alt_cs2_wo_elifecorr',
\n 'alt_cs2_wo_timecorr',\n 'alt_cs2_area_fraction_top',\n 'alt_cs2_bottom'
,\n 'alt_cs2',\n 'e_light',\n 'e_charge',\n 'e_ces',\n 'n_peaks',\n 'drift_
time',\n 'event_number',\n 's1_index',\n 'alt_s1_index',\n 's1_time',\n 'al
t_s1_time',\n 's1_center_time',\n 'alt_s1_center_time',\n 's1_endtime',\n '
alt_s1_endtime',\n 's1_area',\n 'alt_s1_area',\n 's1_n_channels',\n 'alt_s1
```

Following code shows how to make plots

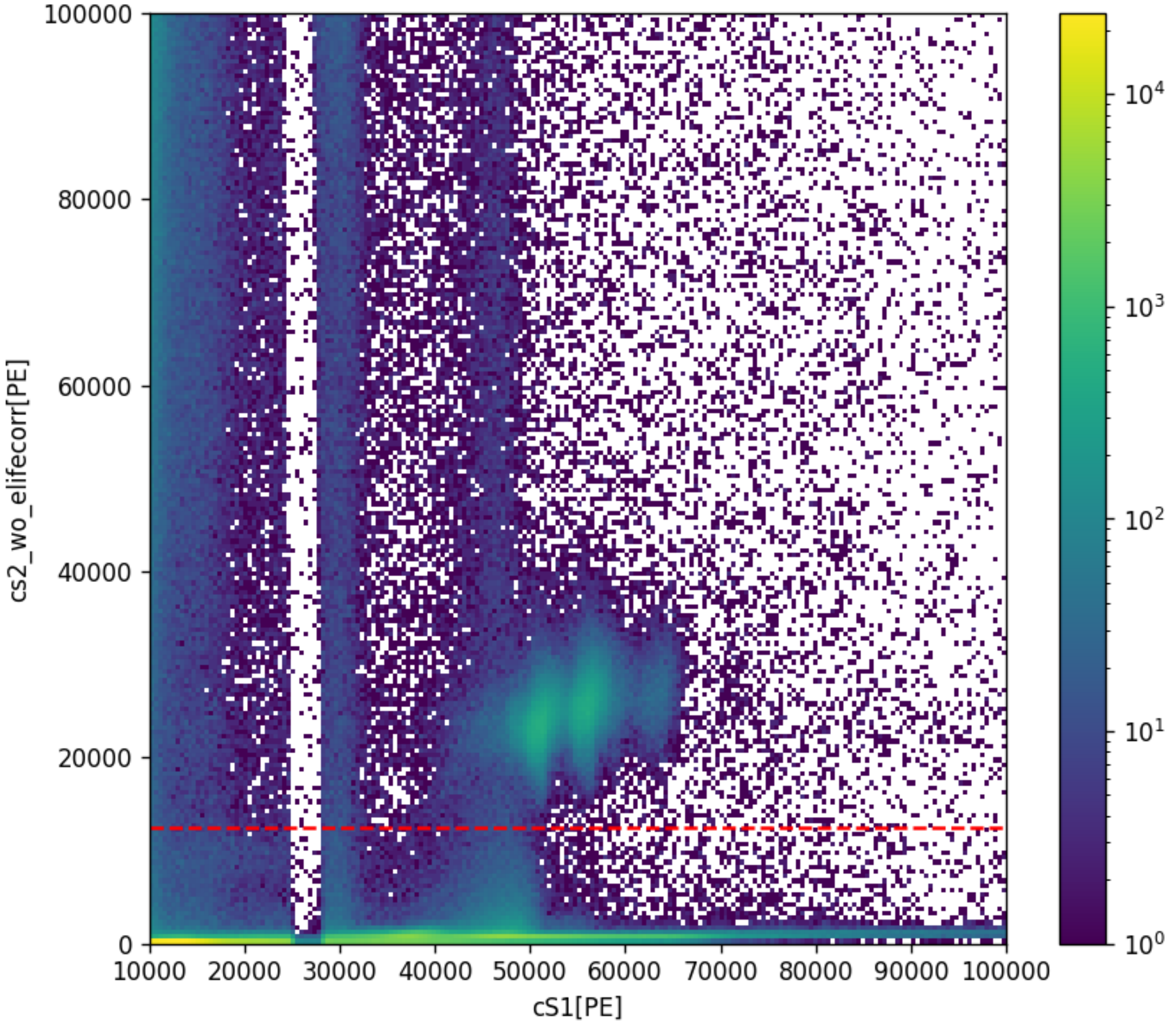
```
xlim = [1e4,1e5]
ylim = [0,500]
fig = plt.figure(figsize=(8, 7), dpi=120)
plt.hist2d(df_bkg_linked.cs1, df_bkg_linked.alt_cs1, ran
plt.axhline( 100, color = 'red', linestyle = '--', label

plt.xlim(xlim)
plt.ylim(ylim)
plt.xlabel('cS1[PE]')
plt.ylabel('alt_cS1[PE]')
plt.colorbar()
plt.show()
```



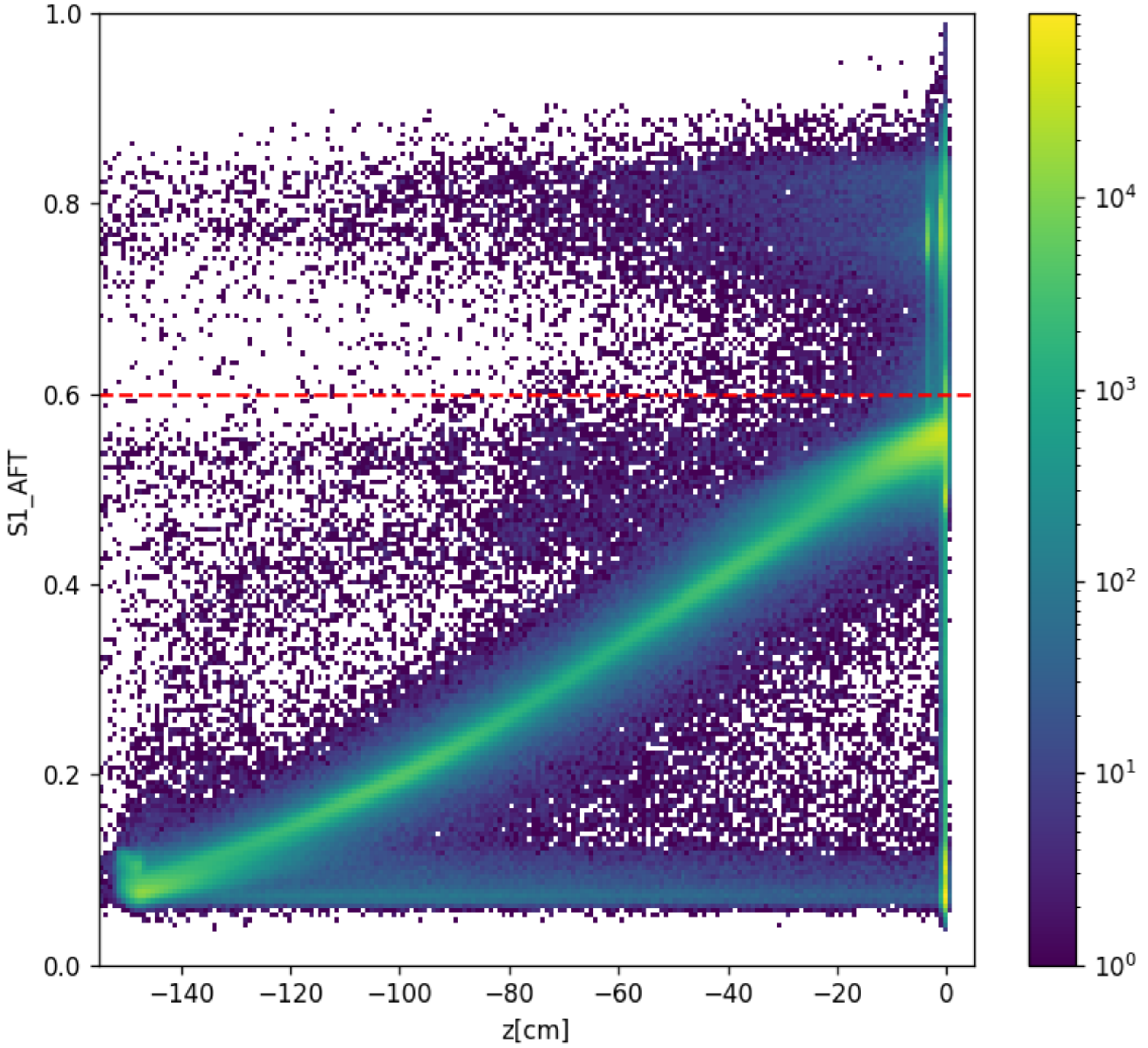

```
xlim = [1e4,1e5]
ylim = [0,1e5]
fig = plt.figure(figsize=(8, 7), dpi=120)
plt.hist2d(df_bkg_linked.cs1, df_bkg_linked.cs2_wo_elife
plt.axhline( 12500, color = 'red', linestyle = '--', lab
```

```
plt.xlim(xlim)
plt.ylim(ylim)
plt.xlabel('cS1[PE]')
plt.ylabel('cs2_wo_elifecorr[PE]')
plt.colorbar()
plt.show()
```



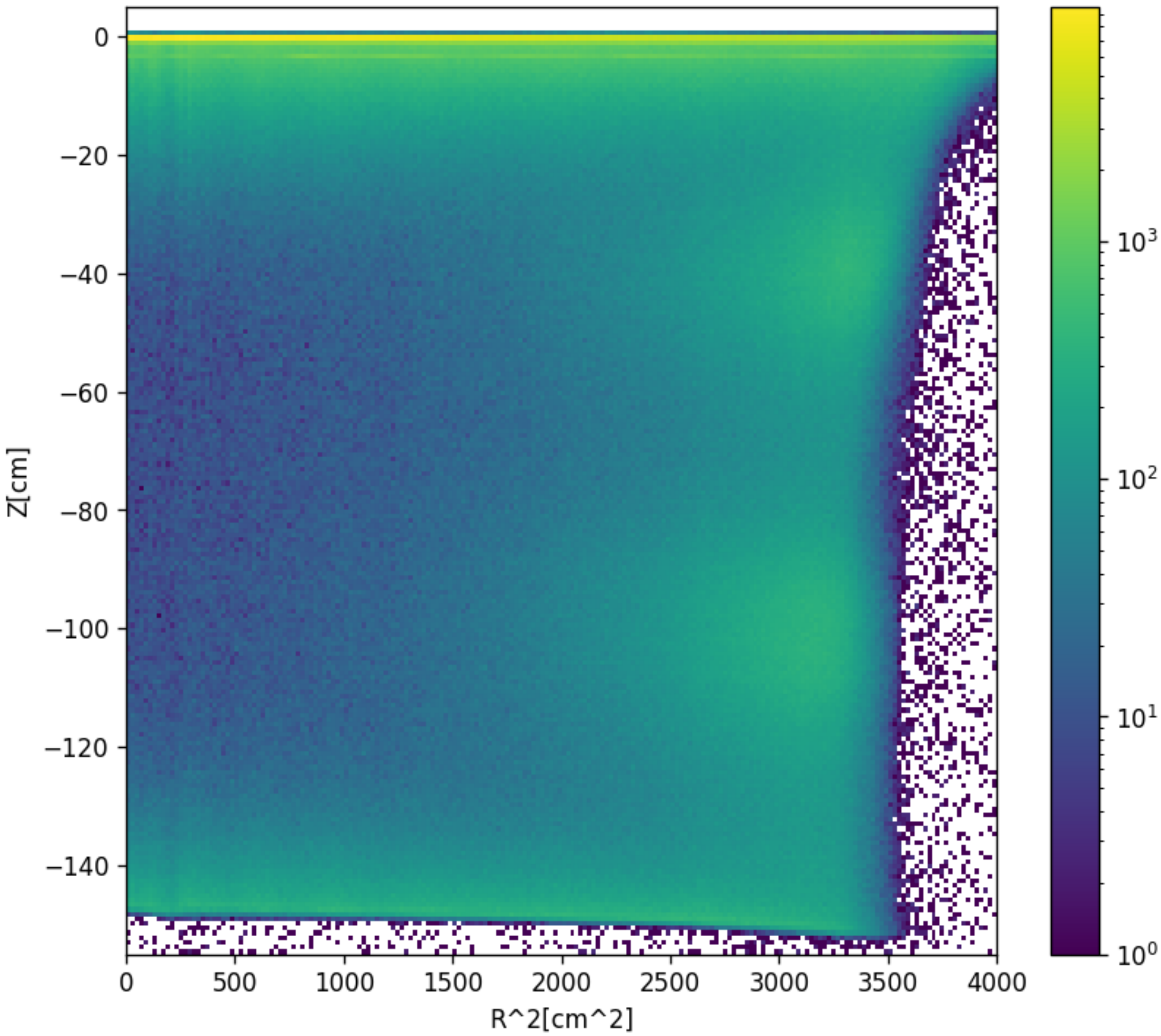

```
xlim = [-155,5]
ylim = [0,1]
fig = plt.figure(figsize=(8, 7), dpi=120)
plt.hist2d(df_bkg_linked.z, df_bkg_linked.s1_area_fracti
plt.axhline( 0.6, color = 'red', linestyle = '--', label
```

```
plt.xlim(xlim)
plt.ylim(ylim)
plt.xlabel('z[cm]')
plt.ylabel('S1_AFT')
plt.colorbar()
plt.show()
```




```
ylim = [-155,5]
xlim = [0,4000]
fig = plt.figure(figsize=(8, 7), dpi=120)
plt.hist2d(df_bkg_linked.r_naive**2, df_bkg_linked.z, ra
#plt.axhline( 0.6, color = 'red', linestyle = '--', labe
```

```
plt.xlim(xlim)
plt.ylim(ylim)
plt.xlabel('R^2[cm^2]')
plt.ylabel('Z[cm]')
plt.colorbar()
plt.show()
```



In above plots, red line indicate the region (in parameter space) which we are interested in. The following code-block shows how to implement a cut.

```
def cuts(df):
    mask = (df['alt_cs1'] < 100)|(np.isnan(df['alt_cs1']
    mask &= df['cs2_wo_elifecorr'] > 12.5e3
    mask &= df['s1_area_fraction_top'] < 0.6
    mask &= df['r_naive'] <= 50
    df_afc = df.loc[mask]

    return df_afc
```

```
df_bkg_linked_selected = cuts(df_bkg_linked)
```

```
len(df_bkg_linked_selected)/len(df_bkg_linked)
```

```
0.39918699261933094
```

Homework

Make cS1 versus z 2d-histogram.

> Indented block

> Indented block

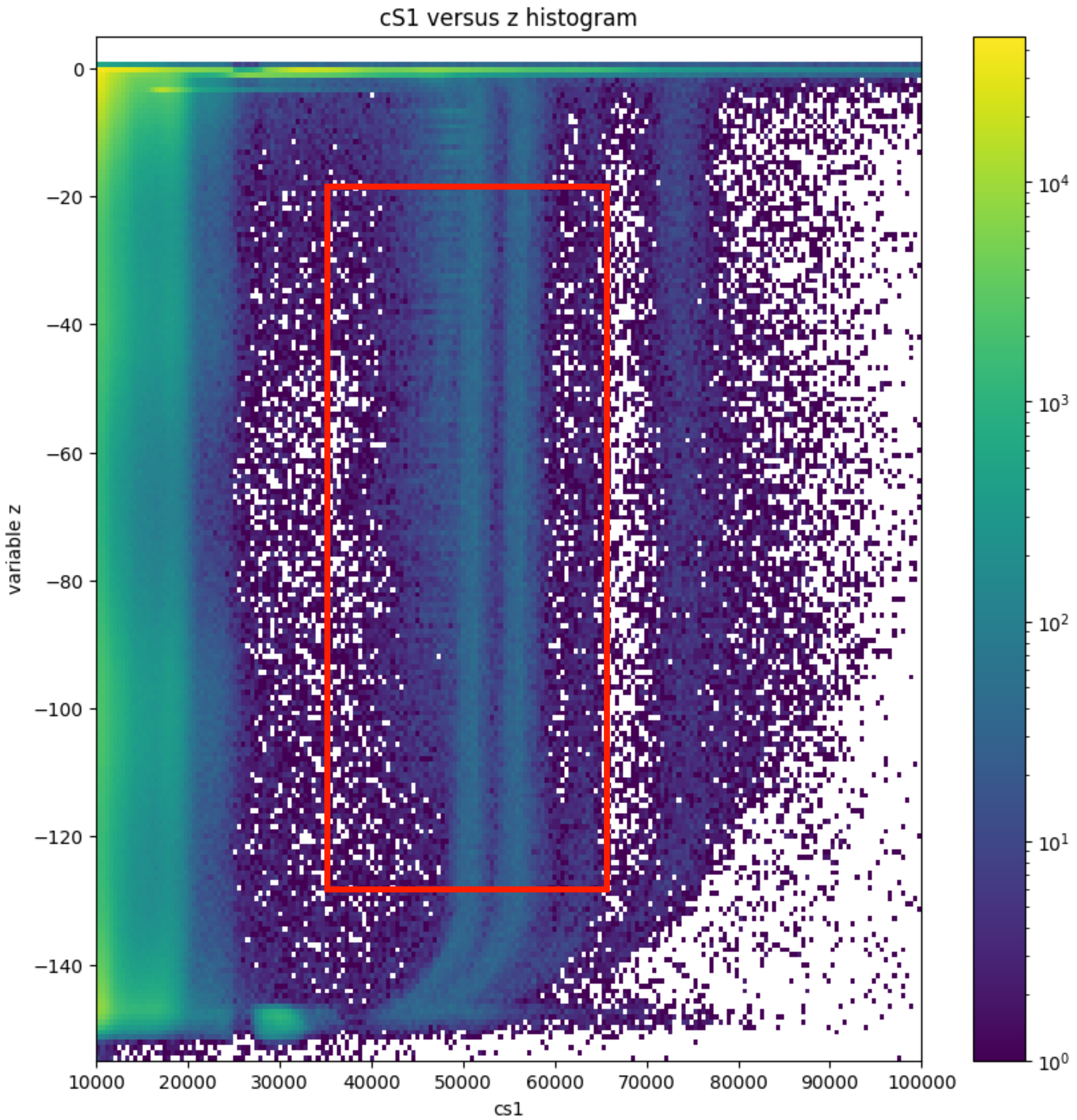
You will see 2-bands. Make a cut to select data in these bands.

Plot the selected data in cS1-cS2 space. Rn222 has smaller energy than Po218. Make 2 dataframes containing Rn222 data and Po218 data.

If you want more challenging exercise, use GaussianMixture from sklearn to select Rn222 and Po218 data by fitting 2 2-d Gaussian in cS1-cS2 space.

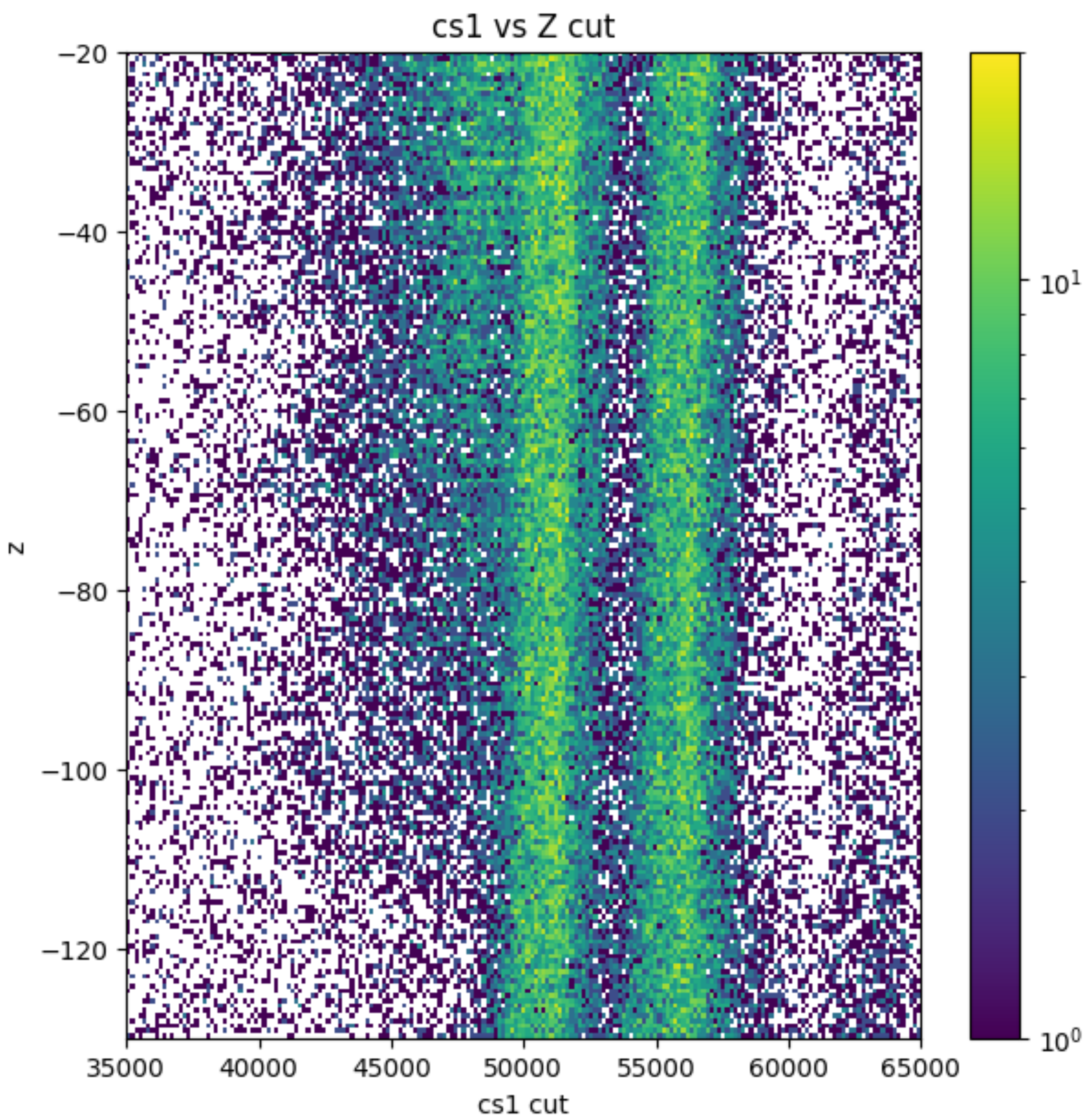
```
plt.figure(figsize = (10, 10))
plt.hist2d(df_bkg_linked.cs1, df_bkg_linked.z, bins = [n
plt.xlabel("cs1")
plt.ylabel("variable z")
```

```
plt.title("cS1 versus z histogram")
plt.colorbar()
#plt.xlim(df_bkg_linked.cs1.min(), df_bkg_linked.cs1.max
#plt.ylim(df_bkg_linked.z.min(), df_bkg_linked.z.max())
plt.show(block = True)
```


cS1 versus z histogram
variable z
cs1
0
−20
−40
−60
−80
−100
−120
−140
10000
20000
30000
40000
50000
60000
70000
80000
90000
100000
10^4
10^3
10^2
10^1
10^0

```
lowerLimit = 35e+3
upperLimit = 65e+3
plt.figure(figsize = (7, 7))
mainCut = (df_bkg_linked.cs1 > lowerLimit) & (df_bkg_lin
cs1Cut = df_bkg_linked.cs1[mainCut]
zCut = df_bkg_linked.z[mainCut]
plt.hist2d(cs1Cut, zCut, bins = [np.linspace(lowerLimit,
plt.colorbar()
plt.title("cs1 vs Z cut")
plt.xlabel("cs1 cut")
plt.ylabel("z")

plt.show(block = True)
```



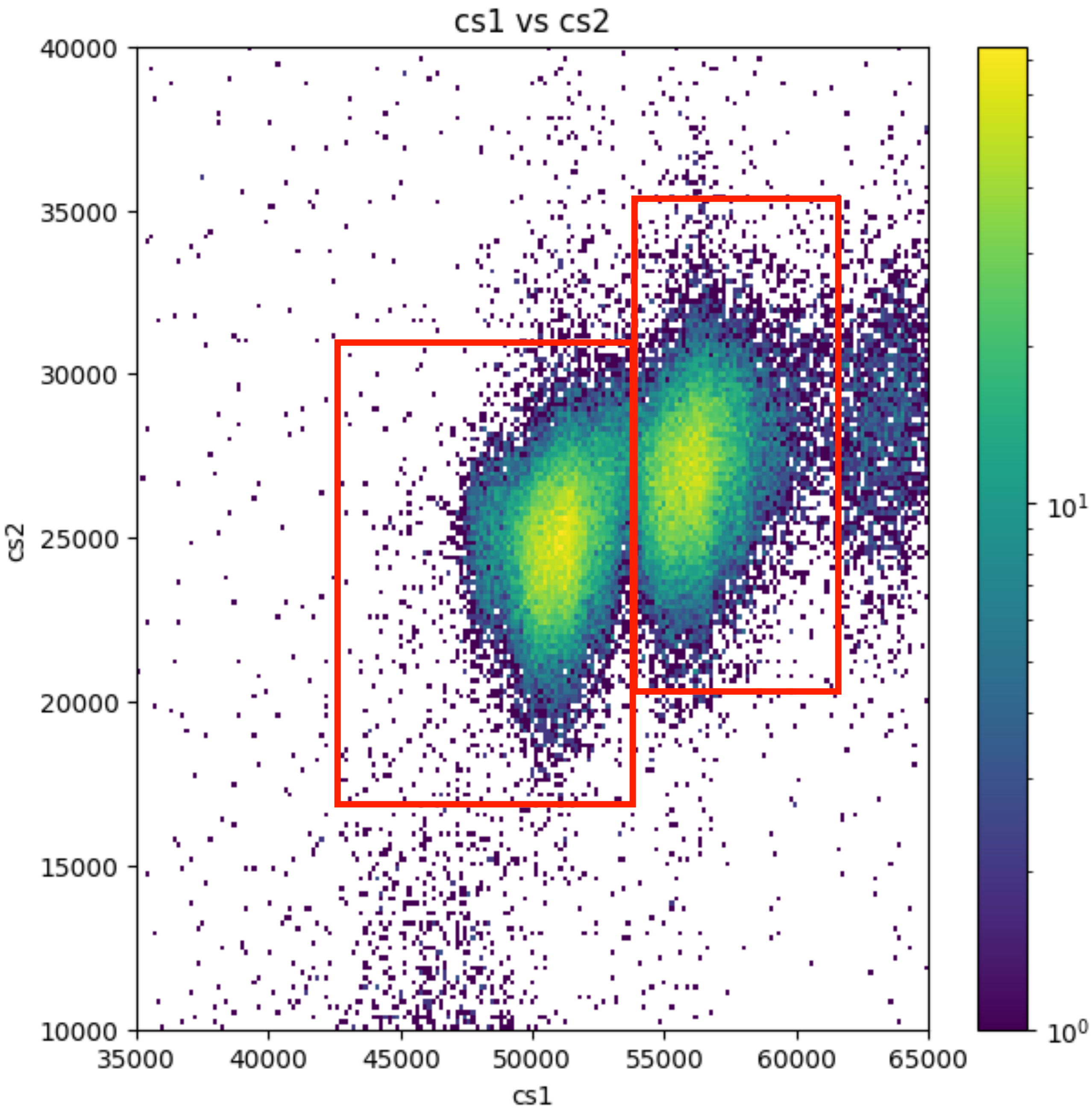

```
plt.figure(figsize = (7, 7))

df_cs1cs2_cut = df_bkg_linked[mainCut]
plt.hist2d(df_cs1cs2_cut.cs1, df_cs1cs2_cut.cs2, bins =

plt.colorbar()

plt.title("cs1 vs cs2")
plt.xlabel("cs1")
```

```
plt.ylabel("cs2")
plt.show(block = True)
```



```
df_bkg_linked.cs1.describe()
```

```
count    5.424305e+06
mean     2.323289e+04
std      9.023302e+04
min      1.000000e+04
25%      1.092634e+04
50%      1.288159e+04
75%      1.820678e+04
max      7.986324e+06
Name: cs1, dtype: float64
```

```
df_bkg_linked.z.describe()
```

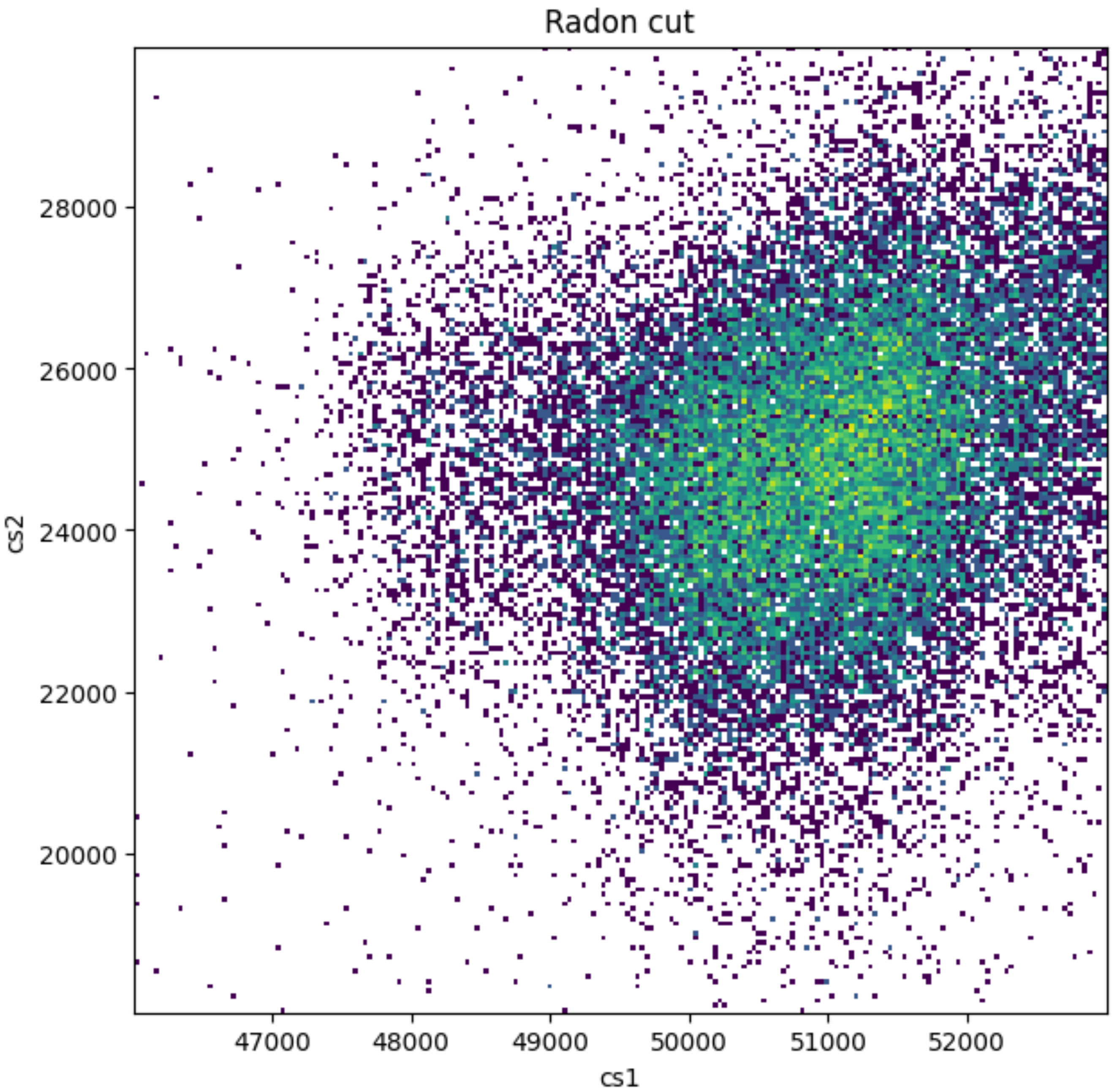
```
count    5.424305e+06
mean    -4.015494e+01
std      5.124940e+01
min     -4.926520e+02
25%     -8.241682e+01
50%     -9.232852e+00
75%     -9.528300e-01
max      2.766825e-01
Name: z, dtype: float64
```

```
#Radon df
plt.figure(figsize = (7, 7))

Rn_cut = (df_bkg_linked.cs1 > 46e+3) & (df_bkg_linked.cs
df_Rn = df_bkg_linked[mainCut & Rn_cut]
plt.hist2d(df_Rn.cs1, df_Rn.cs2, bins = [200, 200], norm

plt.title("Radon cut")
plt.xlabel("cs1")
plt.ylabel("cs2")

plt.show(block = True)
```




```
#Polonium df

plt.figure(figsize = (7, 7))

Po_cut = (df_bkg_linked.cs1 > 53e+3) & (df_bkg_linked.cs
df_Po = df_bkg_linked[mainCut & Po_cut]
plt.hist2d(df_Po.cs1, df_Po.cs2, bins = [200, 200], norm

plt.title("Polonium cut")
```

```
plt.xlabel("cs1")
plt.ylabel("cs2")

plt.show(block = True)
```