Dispersione di soglia e ILDAC (no outliers) RD53B

May 11, 2021

1 In questo script recupero informazioni relative alla dispersione di soglia dei FE RD53A e RD53B e ne elimino gli outliers

Devo prendere i valori al 50% dello scan di soglia e calcolare la varianza della loro distribuzione. Infine, plottare l'andamento delle dispersioni di soglia delle misurazioni fatte in funzione della corrente I_{LDAC}, cercando di eliminare l'effetto di eventuali outliers.

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
%run ./Functions/selectFiles.ipynb #gestisce button e schermata di dialogo con␣
 ↪il file system
%run ./Functions/parseCalibs.ipynb #parsing dei file txt. input: path[]; output:␣
 ↪dataframe[]
```

[13]:
```
def plotHist(dataframe, tipo): #plot istogramma distribuzione delle soglie␣
 ↪filtrate per tipo di FE. Ritorna lista di soglie
    C = dataframe[0][0]
    D = C.loc[C['Tipo'] == tipo]
    E = np.asarray(D['Thresholds'].values.tolist())

    fig, ax = plt.subplots(1, 1)

    kws = dict(histtype= "stepfilled",alpha= 0.5, linewidth = 2)

    ax.hist(E, color="lightblue", edgecolor = "k", **kws,  density = False)
    ax.legend(["Dev ~ " + str(round(np.std(E), 3))])
    ax.set_xlabel("Thresholds")
    ax.set_ylabel("Frequency")


    plt.show()
    return E
```

[18]:
```
def plotNoOutliers(array):
    fig, ax = plt.subplots(1, 1)
```

```
    kws = dict(histtype= "stepfilled",alpha= 0.5, linewidth = 2)

    ax.hist(array, color="green", edgecolor = "k", **kws,  density = False)
    ax.legend(["Dev ~ " + str(round(np.std(array), 3))])
    ax.set_xlabel("Thresholds")
    ax.set_ylabel("Frequency")
    plt.show()
```

[57]:
```
#inizializzo vettori che userò per plottare i grafici. Ad ogni passo aggiungo in␣
↪coda la dev std
varianzeB_pulite = []
varianzeB = []
```

[6]:
```
button = selectFiles() #è possibile selezionare più files
```

```
Selected files:
C:/Users/andre/Documents/CMSAFE/CMSAFE_calibs/calib_I_LDAC_8_0_soglia_1000_elett
roni.txt
```

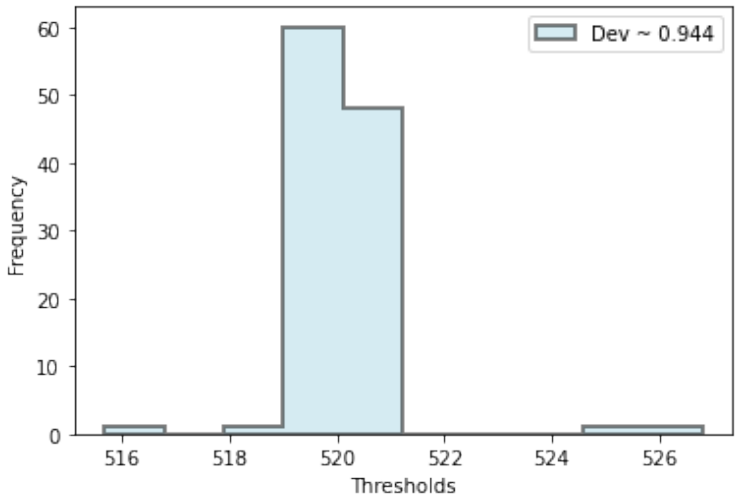
1.1 Dati per corrente $I_{LDAC} = 8\mu A$

[7]:
```
data8 = parseCalibs(button.files) #ad ogni file è associato un dataframe          ␣
↪
```

```
1 dataframe crated!
```

Dati sporchi

[14]:
```
array8 = plotHist(data8, 'B')
```



Rimuovo gli outliers lavorando sul coefficiente m

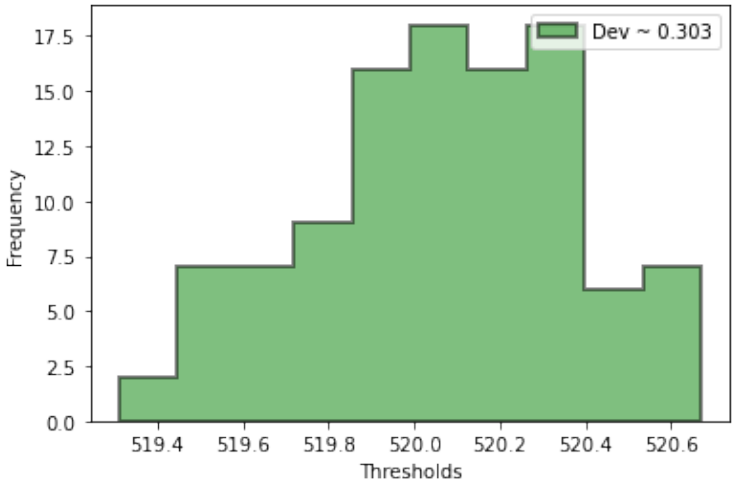
Il dato viene considerato se:

$$x - \overline{x} < m \cdot \lambda_x$$

con λ_x deviazione standard

[16]:
```
no_outliers_8 = array8[abs(array8 - np.mean(array8)) < 0.85*np.std(array8)]␣
↪#Rimuovo gli outliers
```

[19]:
```
plotNoOutliers(no_outliers_8)
```



[58]:
```
varianzeB.append(np.std(array8))
varianzeB_pulite.append(np.std(no_outliers_8))
```

1.2 Dati per corrente $I_{LDAC} = 10\mu A$

[20]:
```
button10 = selectFiles() #è possibile selezionare più files
```

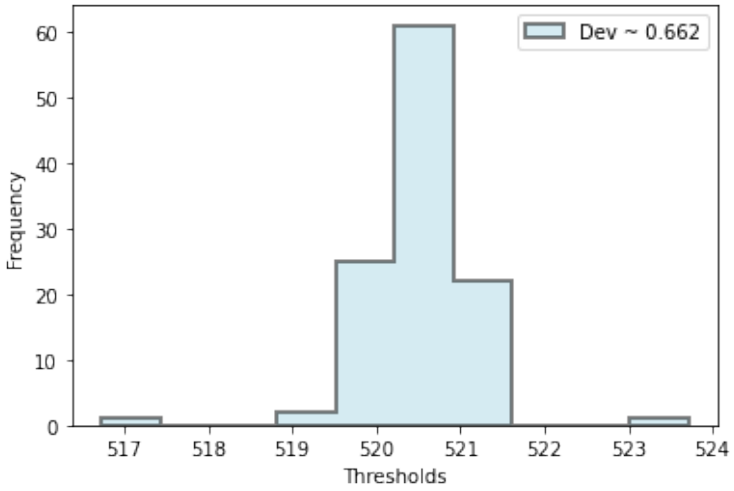
```
Selected files:
C:/Users/andre/Documents/CMSAFE/CMSAFE_calibs/calib_I_LDAC_10_0_soglia_1000_elet
troni.txt
```

[21]:
```
data10 = parseCalibs(button10.files) #ad ogni file è associato un dataframe ␣
↪
```

```
1 dataframe crated!
```

Dati sporchi

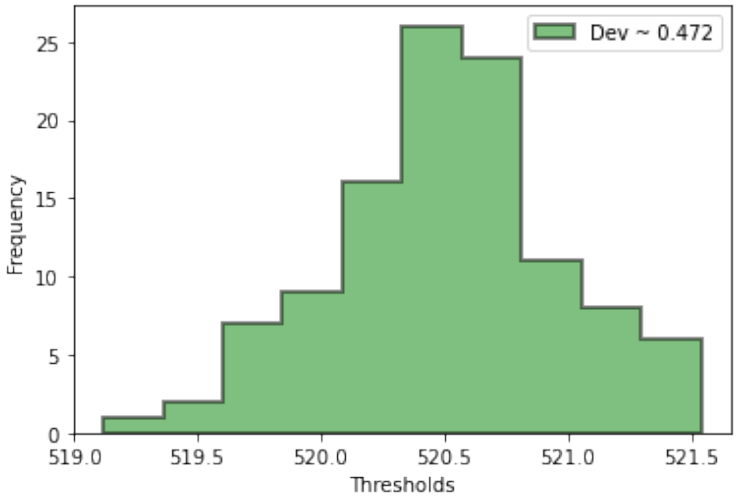
[22]:
```
array10 = plotHist(data10, 'B')
```



Rimuovo gli outliers lavorando sul coefficiente m

[26]:
```
no_outliers_10 = array10[abs(array10 - np.mean(array10)) < 4*np.std(array10)]
```

[27]:
```
plotNoOutliers(no_outliers_10)
```



[59]:
```
varianzeB.append(np.std(array10))
varianzeB_pulite.append(np.std(no_outliers_10))
```

1.3 Dati per corrente $I_{LDAC} = 12\mu A$

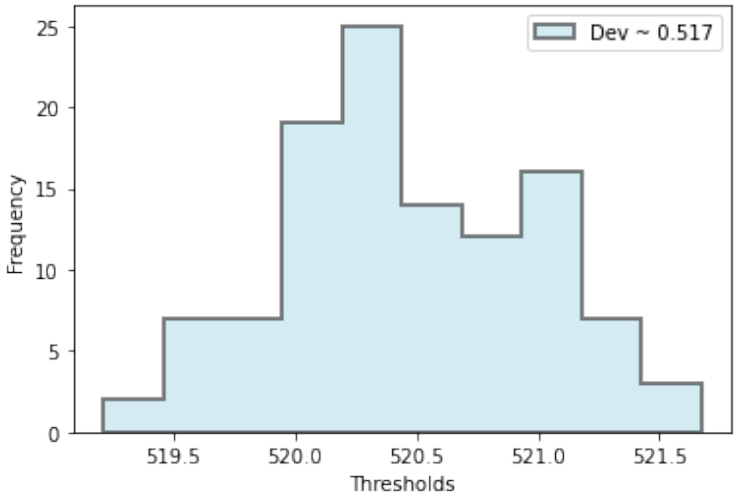
[29]:
```
button12 = selectFiles() #è possibile selezionare più files
```

```
Selected files:
C:/Users/andre/Documents/CMSAFE/CMSAFE_calibs/calib_I_LDAC_12_0_soglia_1000_elet
troni.txt
```

[30]:
```
data12 = parseCalibs(button12.files) #ad ogni file è associato un dataframe
↪
```

```
1 dataframe crated!
```

[31]:
```
array12 = plotHist(data12, 'B')
```



[60]:
```
varianzeB.append(np.std(array12))
varianzeB_pulite.append(np.std(array12))
```

1.4 Dati per corrente $I_{LDAC} = 13\mu A$

[34]:
```
button13 = selectFiles() #è possibile selezionare più files
```

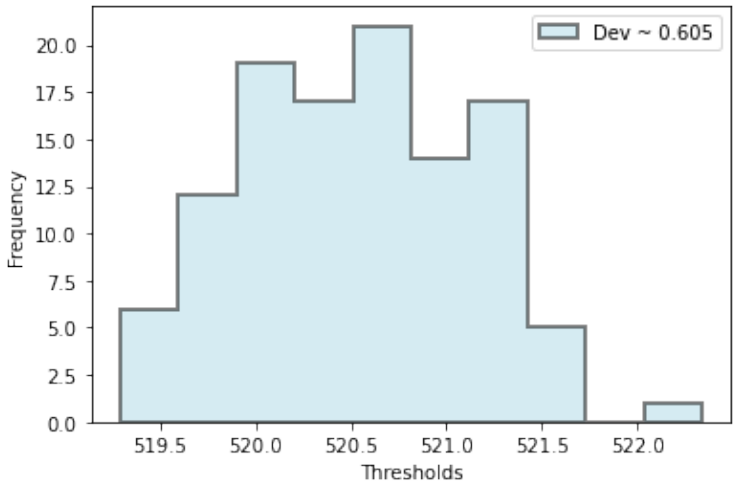
```
Selected files:
C:/Users/andre/Documents/CMSAFE/CMSAFE_calibs/calib_I_LDAC_13_0_soglia_1000_elet
troni.txt
```

[35]:
```
data13 = parseCalibs(button13.files) #ad ogni file è associato un dataframe
↪
```

```
1 dataframe crated!
```

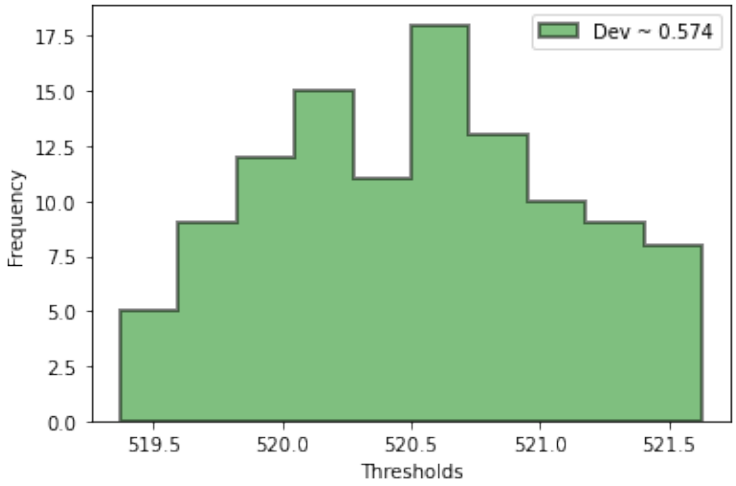
Dati sporchi

[36]:
```
array13 = plotHist(data13, 'B')
```



Rimuovo gli outliers lavorando sul coefficiente m

[37]:
```
no_outliers_13 = array13[abs(array13 - np.mean(array13)) < 2*np.std(array13)]
plotNoOutliers(no_outliers_13)
```



[61]:
```
varianzeB.append(np.std(array13))
varianzeB_pulite.append(np.std(no_outliers_13))
```

1.4.1 Dati per corrente $I_{LDAC} = 14\mu A$

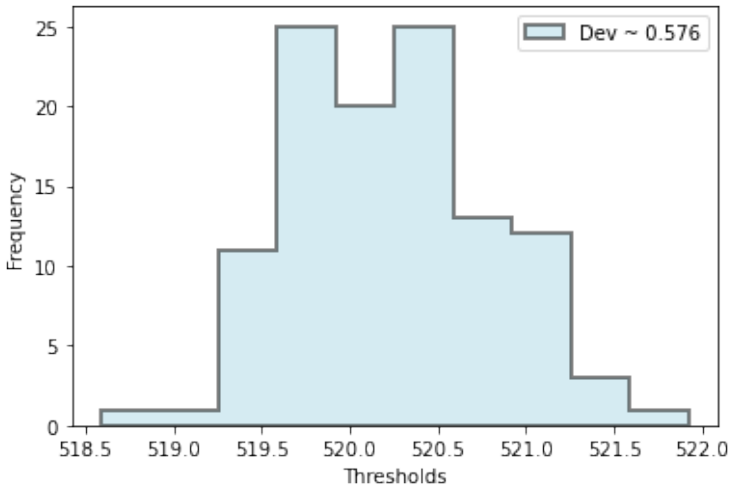
[39]:
```
button14 = selectFiles() #è possibile selezionare più files
```

```
Selected files:
C:/Users/andre/Documents/CMSAFE/CMSAFE_calibs/calib_I_LDAC_14_0_soglia_1000_elet
troni.txt
```

[40]:
```
data14 = parseCalibs(button14.files) #ad ogni file è associato un dataframe
```

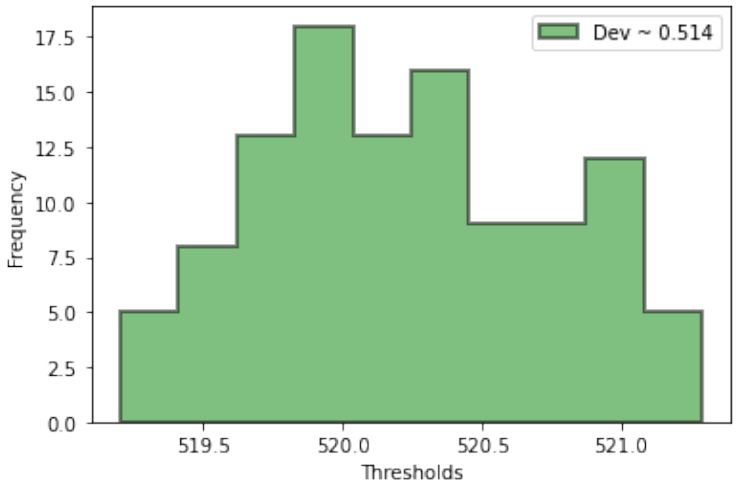
```
1 dataframe crated!
```

[41]:
```
array14 = plotHist(data14, 'B')
```



[42]:
```
array_x=array14
no_outliers_14 = array_x[abs(array_x - np.mean(array_x)) < 2*np.std(array_x)]

plotNoOutliers(no_outliers_14)
```



[62]:
```
varianzeB.append(np.std(array14))
varianzeB_pulite.append(np.std(no_outliers_14))
```

1.4.2 Dati per corrente $I_{LDAC} = 16\mu A$

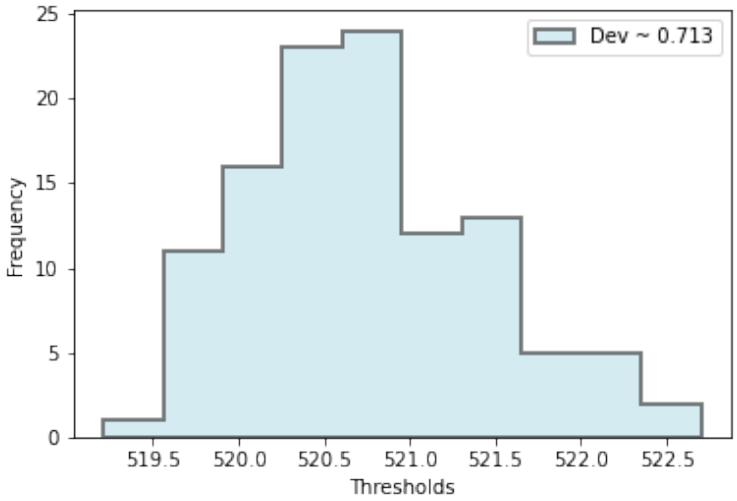
[43]:
```
button16 = selectFiles() #è possibile selezionare più files
```

```
Selected files:
C:/Users/andre/Documents/CMSAFE/CMSAFE_calibs/calib_I_LDAC_16_0_soglia_1000_elet
troni.txt
```

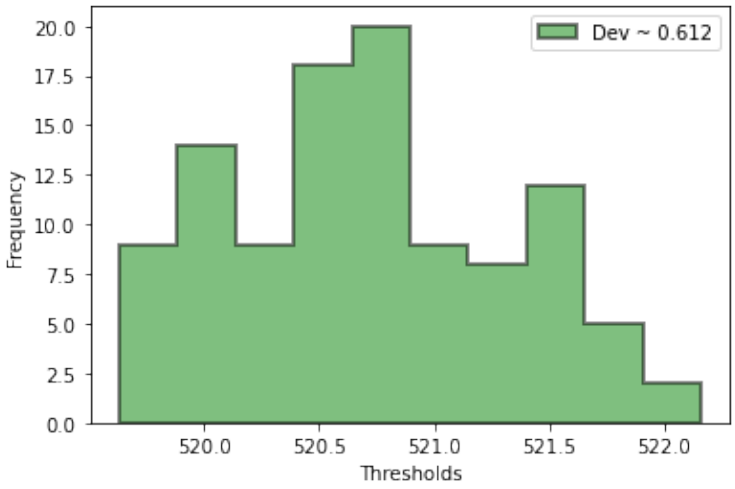
[44]:
```
data16 = parseCalibs(button16.files) #ad ogni file è associato un dataframe
```

```
1 dataframe crated!
```

[45]:
```
array16 = plotHist(data16, 'B')
```



[47]:
```
array_x=array16
no_outliers_16 = array_x[abs(array_x - np.mean(array_x)) < 2*np.std(array_x)]
plotNoOutliers(no_outliers_16)
```



[63]:
```
varianzeB.append(np.std(array16))
varianzeB_pulite.append(np.std(no_outliers_16))
```

1.4.3 Dati per corrente $I_{LDAC} = 18\mu A$

[48]:
```
button18 = selectFiles() #è possibile selezionare più files
```

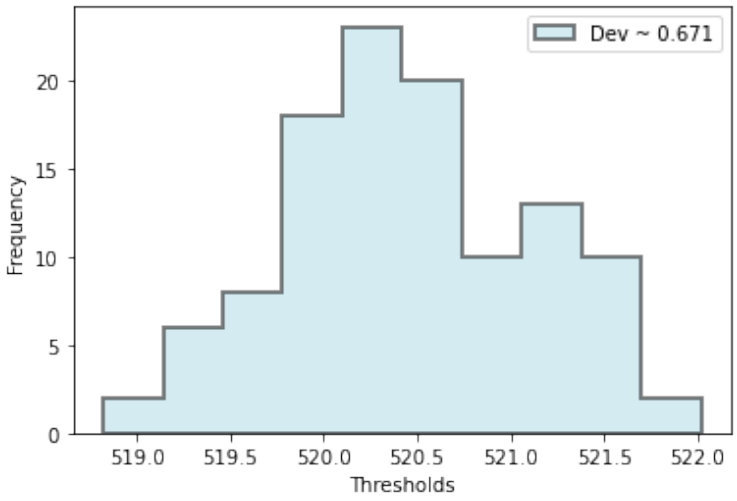
```
Selected files:
```

```
C:/Users/andre/Documents/CMSAFE/CMSAFE_calibs/calib_I_LDAC_18_0_soglia_1000_elet
troni.txt
```

[49]:
```
data18 = parseCalibs(button18.files) #ad ogni file è associato un dataframe
```

```
1 dataframe crated!
```

[50]:
```
array18 = plotHist(data18, 'B')
```



[64]:
```
varianzeB.append(np.std(array18))
varianzeB_pulite.append(np.std(array18))
```

1.4.4 Dati per corrente $I_{LDAC} = 20\mu A$

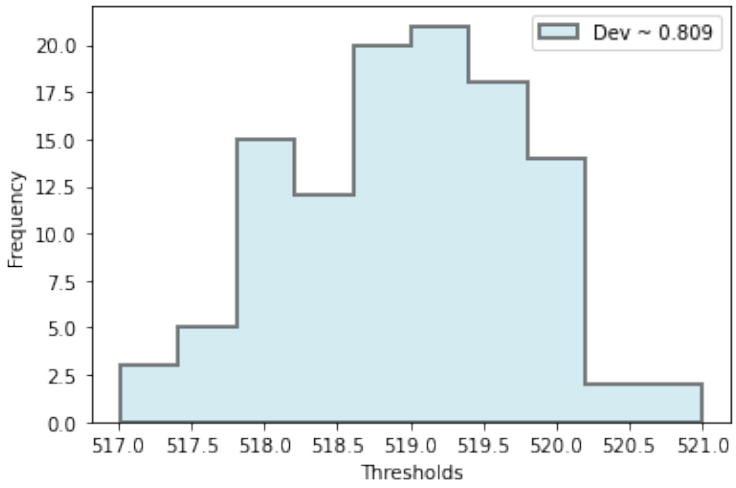
[51]:
```
button20 = selectFiles() #è possibile selezionare più files
```

```
Selected files:
C:/Users/andre/Documents/CMSAFE/CMSAFE_calibs/calib_I_LDAC_20_0_soglia_1000_elet
troni.txt
```

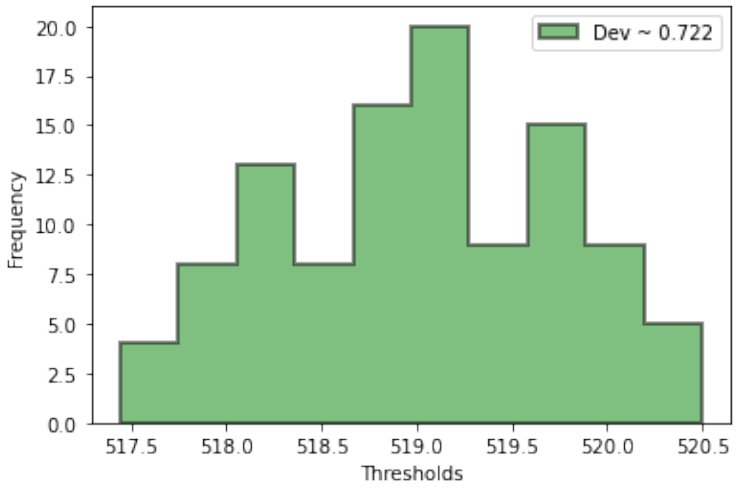
[52]:
```
data20 = parseCalibs(button20.files) #ad ogni file è associato un dataframe
```

```
1 dataframe crated!
```

[53]:
```
array20 = plotHist(data20, 'B')
```



[55]:
```
array_x=array20
no_outliers_20 = array_x[abs(array_x - np.mean(array_x)) < 2*np.std(array_x)]
plotNoOutliers(no_outliers_20)
```



[65]:
```
varianzeB.append(np.std(array20))
varianzeB_pulite.append(np.std(no_outliers_20))
```

1.5 Plot Dispersioni di soglia (senza outliers) in funzione di I_{LDAC}

[66]:
```
correnti = [8.0, 10.0, 12.0, 13.0, 14.0, 16.0, 18.0, 20.0]
```

[67]:
```
fig, ax = plt.subplots(figsize = (15,4))

fig.suptitle("Andamento dispersione di soglia in funzione di $I_{LDAC}$ -␣
 ↪RD53B", fontsize=16)
ax.set_ylabel("Dispersione di soglia (dev std)")
ax.set_xlabel("Corrente $I_{LDAC}  (\mu A)$")

ax.plot(correnti, varianzeB, '-o', color = 'coral', label="Dirty")
ax.plot(correnti, varianzeB_pulite, '-o', color = 'blue', label="No outliers")


ax.legend(loc="upper right")

#plt.xticks(np.arange(8, 20, 0.5))
#plt.savefig('disp.png', bbox_inches='tight')
plt.show()
```