Use for GMM

The folowing import statement is required to avoid a warning when we use the GMM. There is a memory leakge when we use GMM. Do not use it for other entropy approzimations because it slows down the campaign in general.

Note: it causes a problem with the plots. They are created twice.

In [1]:
```
#import os
#os.environ["OMP_NUM_THREADS"] = '1'
```

In [2]:
```
import  MyPlotting as MyPlots
import matplotlib.pyplot as plt
from datastruct import Settings, Experiment, ExperimentStep, DataPoint
from entropy import calc_entropy, default_entropy_options
import numpy as np

from functions_gain_factor import recalculating_entropy
```

9 Reloading Experiments

In [3]:
```
# #Reloading Data
exp_control = Experiment.load('15/test1_control id15c.gz',recalc_yprofs=True)
exp_entropy1 = Experiment.load('15/test1_entropy id15e1.gz', recalc_yprofs=True)
exp_entropy2 = Experiment.load('15/test1_entropy id15e2.gz', recalc_yprofs=True)
exp_on_the_fly1 = Experiment.load('15/test1_on_the_fly1 id15o1.gz', recalc_yprofs=True
exp_on_the_fly2 = Experiment.load('15/test1_on_the_fly2 id15o2.gz', recalc_yprofs=True

################################################Use only when you have a GMM version of

###GMM case
# exp_control_gmm = Experiment.load('1/test1_control id1c_gmm.gz',recalc_yprofs=True)
# exp_entropy1_gmm = Experiment.load('1/test1_entropy id1e1_gmm.gz', recalc_yprofs=Tru
# exp_entropy2_gmm = Experiment.load('1/test1_entropy id1e2_gmm.gz', recalc_yprofs=Tru
# exp_on_the_fly1_gmm = Experiment.load('1/test1_on_the_fly1 id1o1_gmm.gz', recalc_ypr
# exp_on_the_fly2_gmm = Experiment.load('1/test1_on_the_fly2 id1o2_gmm.gz', recalc_ypr

#reloaded_exp.creation_time.isoformat()
```

In [ ]:
```

```

In [4]:
```
#Notice now all the outcomes have the same size except for the datax, datay and datady
#last unfitted point
len(exp_entropy1.entropy()) #34
len(exp_entropy1.load_yprofs()) #34
len(exp_entropy1.load_pts()) #34
len(exp_entropy1.getdata()[0]) #35
len(exp_entropy1.meastimes()) #34
```

```
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:346: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_yprof)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
```

Out[4]:
```
13
```

Warning

Remember that the maximum time spend in the control data is way greater than for the autonomous cases. Thus, when we are plotting the y-profiles and the histogram for the parameters we should may be truncated the values of control until a time that is closer to max amount of time spend in the autonomous cases.

10 Compare to Benchmark

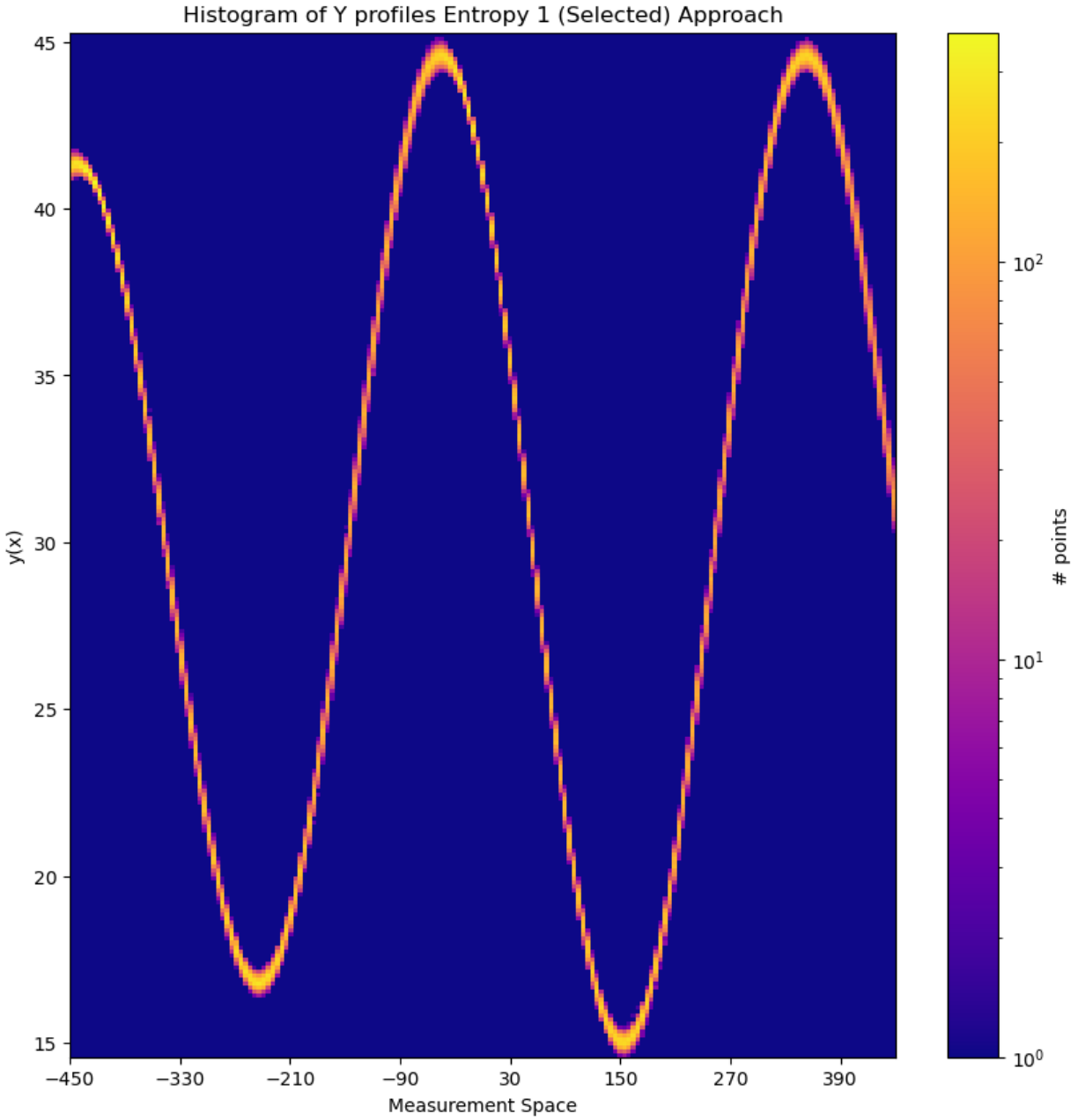
In [5]:
```
# from matplotlib.colors import LogNorm
# from numpy.matlib import repmat

print(exp_entropy1.settings.ground_truth_pars) #{'I0': 30.0, 'A': 15.0, 'phi0': 155.0,
print(exp_on_the_fly1.settings.ground_truth_pars)
```

```
{'I0': 30.0, 'A': 15.0, 'phi0': 155.0, 'T': 401.0, 'sigma': 797.0}
{'I0': 30.0, 'A': 15.0, 'phi0': 155.0, 'T': 401.0, 'sigma': 797.0}
```

10.1 Y-Profiles Histograms

In [6]:
```
##Plotting the last set of Y-profiles for the different approraches
##If you do not want to plot the figure of merits do not provide them. It still works

#Entropy 1
MyPlots.plot_y_profiles(exp_entropy1.load_yprofs()[-1],exp_entropy1.settings.x,
                        FOM = exp_entropy1.load_FOM()[-1],
                        this_title = "Histogram of Y profiles Entropy 1 (Selected) App

#Entropy 2
MyPlots.plot_y_profiles(exp_entropy2.load_yprofs()[-1],exp_entropy2.settings.x,
                        FOM = exp_entropy2.load_FOM()[-1],
                        this_title = "Histogram of Y profiles Entropy 2 (All) Approach
#On-the Fly 1
MyPlots.plot_y_profiles(exp_on_the_fly1.load_yprofs()[-1],exp_on_the_fly1.settings.x,
                        FOM = exp_on_the_fly1.load_FOM()[-1],
                        this_title = "Histogram of Y profiles On-The_Fly 1 Approach" )
#On-the Fly 1
MyPlots.plot_y_profiles(exp_on_the_fly2.load_yprofs()[-1],exp_on_the_fly2.settings.x,
                        FOM = exp_on_the_fly2.load_FOM()[-1],
                        this_title = "Histogram of Y profiles On-The_Fly 2 Approach" )
#Control Data
```

```
MyPlots.plot_y_profiles(exp_control.load_yprofs()[-1],exp_control.settings.x,
                        this_title = "Histogram of Y profiles My Control Evenly Approa
```



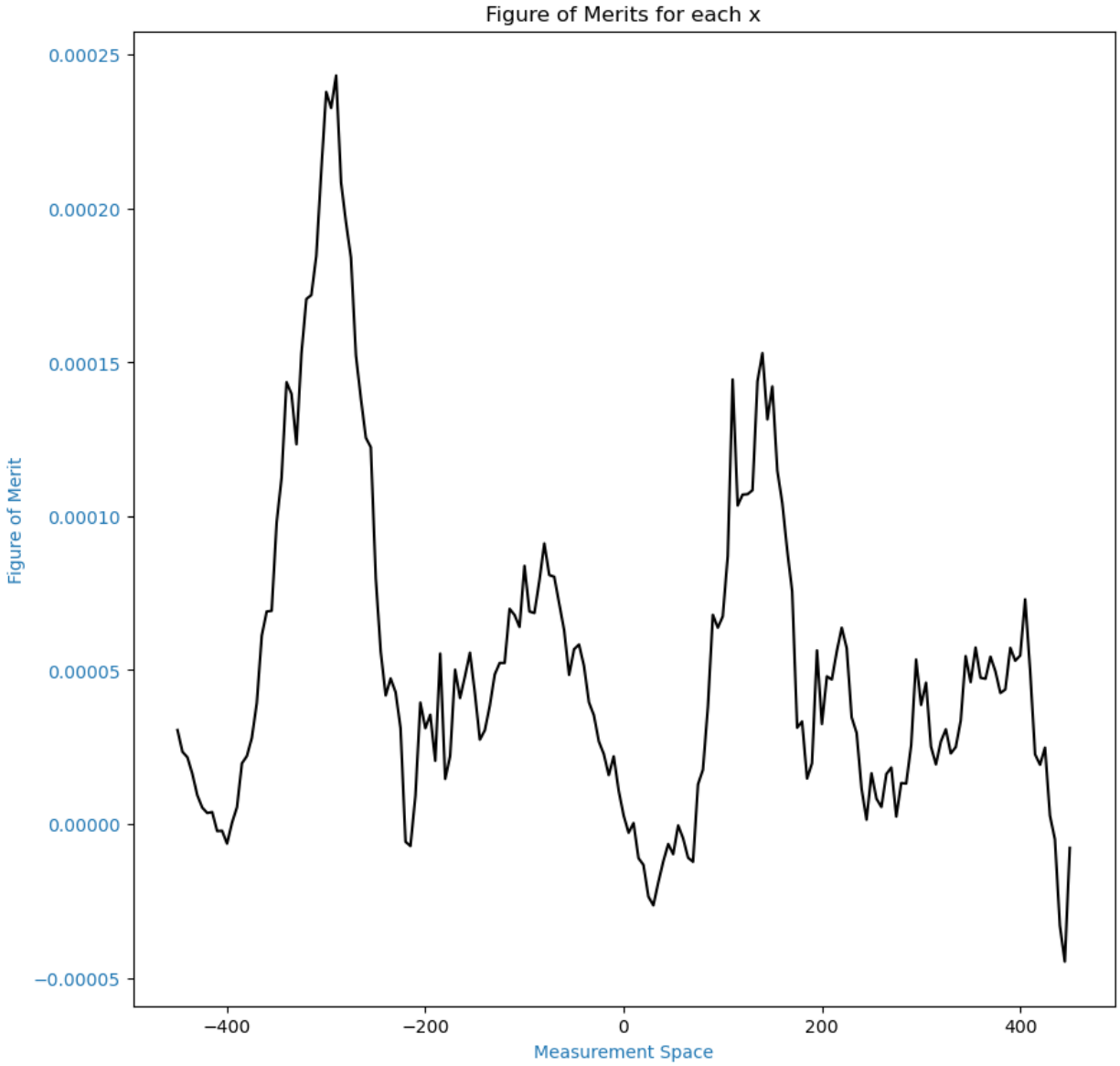
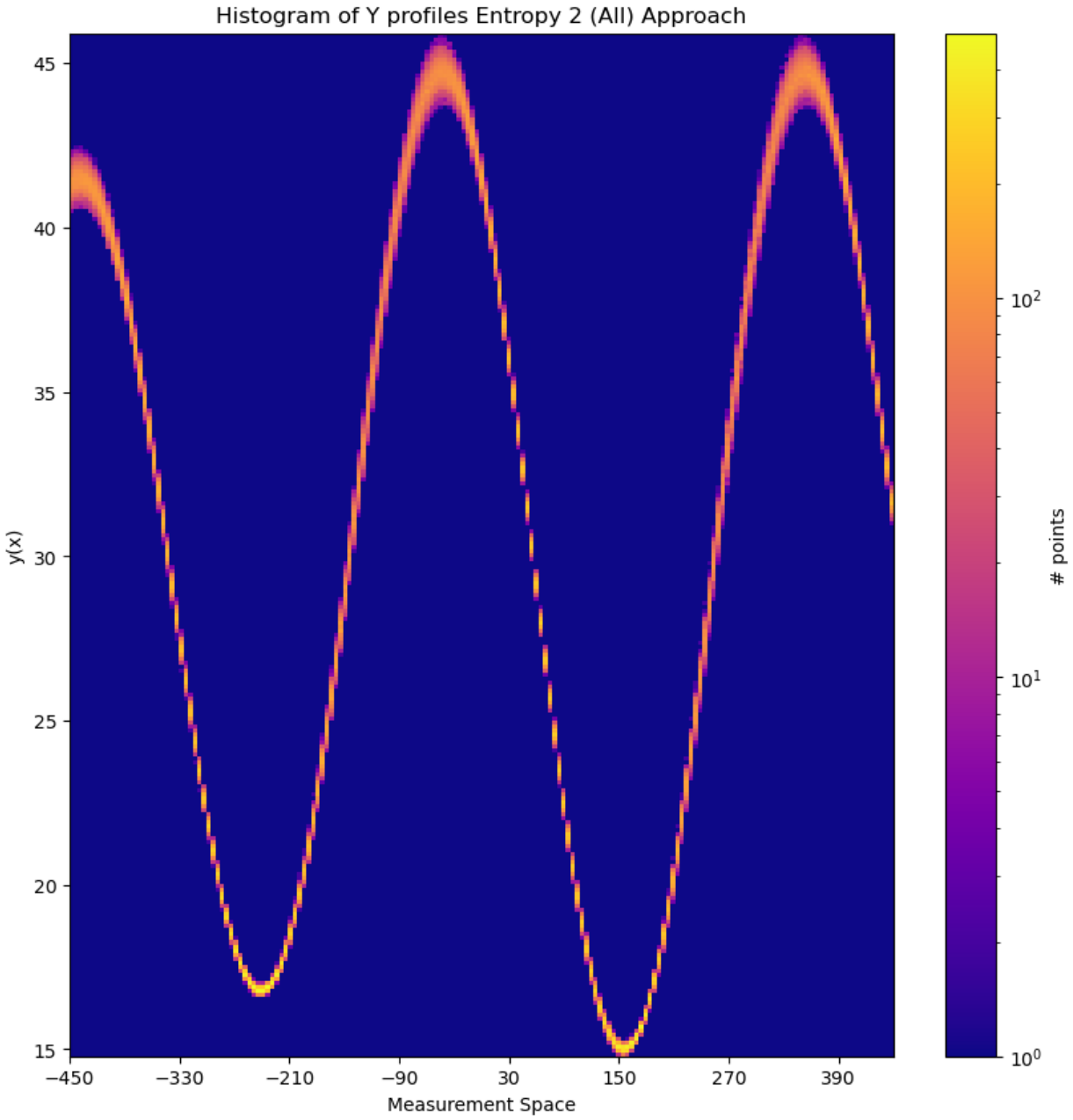


Figure of Merits for each x
Figure of Merit
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
−0.00005
−400
−200
0
200
400
Measurement Space


Histogram of Y profiles Entropy 2 (All) Approach
y(x)
Measurement Space
points
45
40
35
30
25
20
15
−450
−330
−210
−90
30
150
270
390
10^2
10^1
10^0

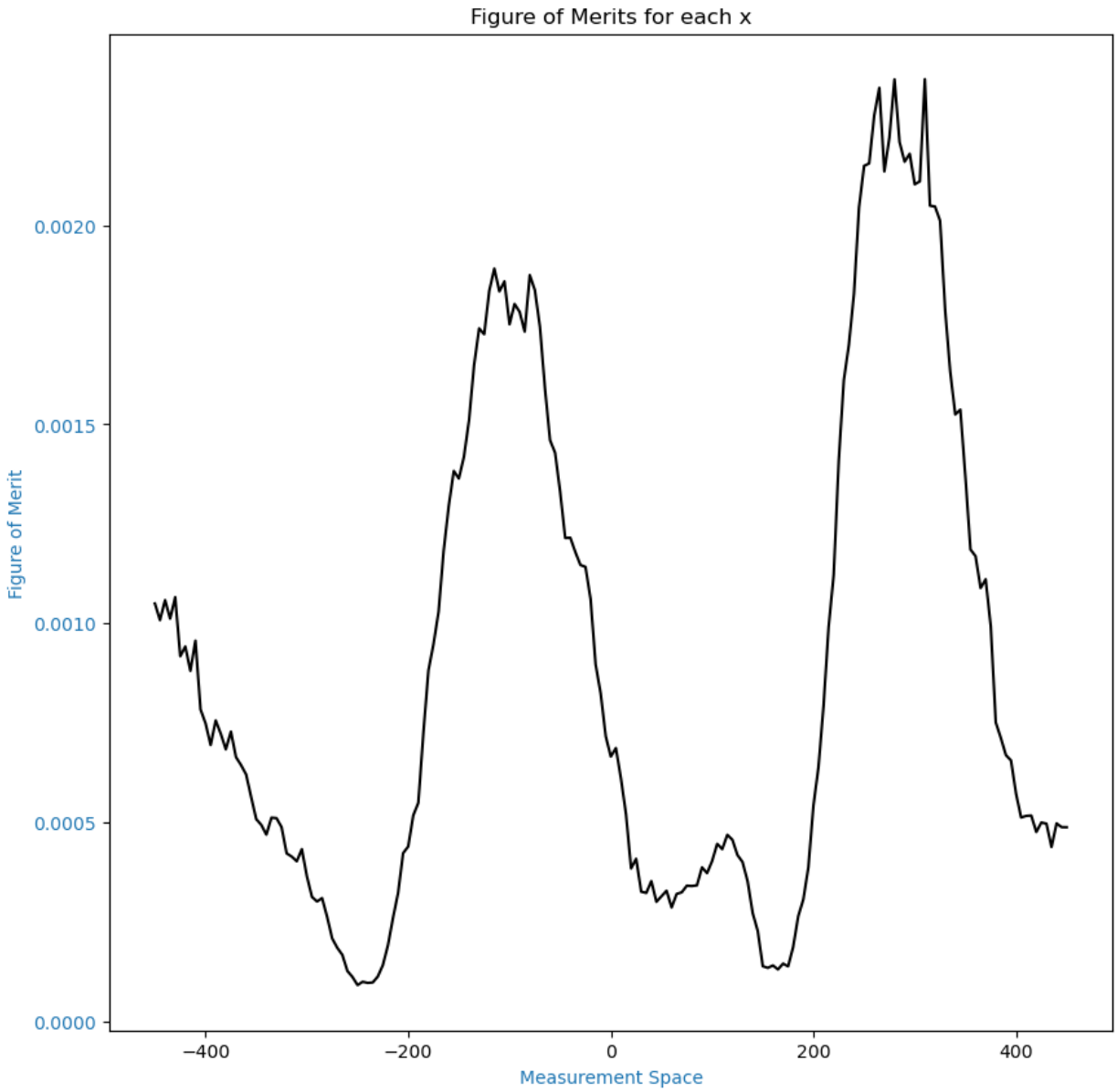
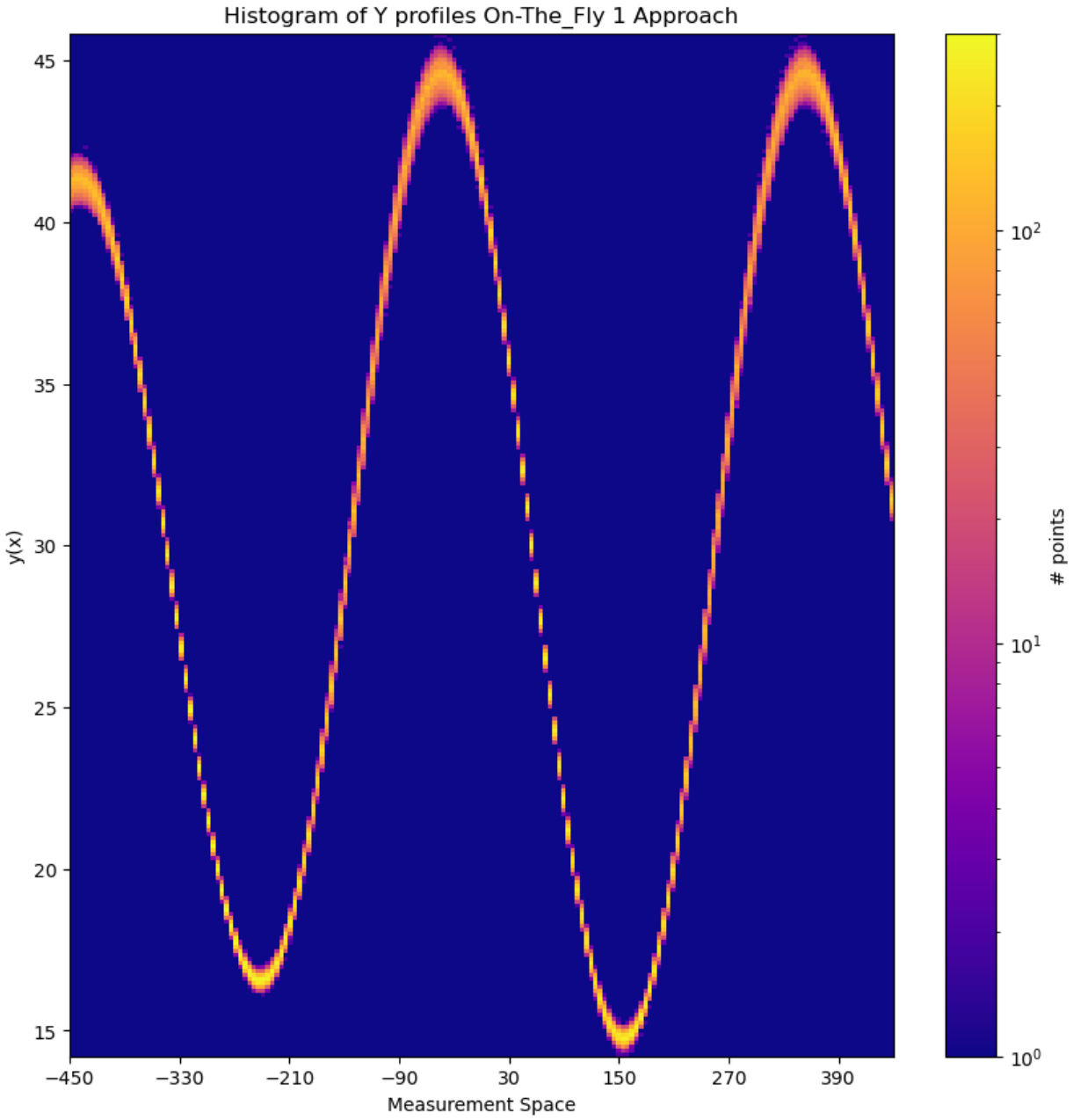


Figure of Merits for each x
Figure of Merit
0.0020
0.0015
0.0010
0.0005
0.0000
−400
−200
0
200
400
Measurement Space


Histogram of Y profiles On-The_Fly 1 Approach
y(x)
Measurement Space
points
45
40
35
30
25
20
15
−450
−330
−210
−90
30
150
270
390
10^2
10^1
10^0

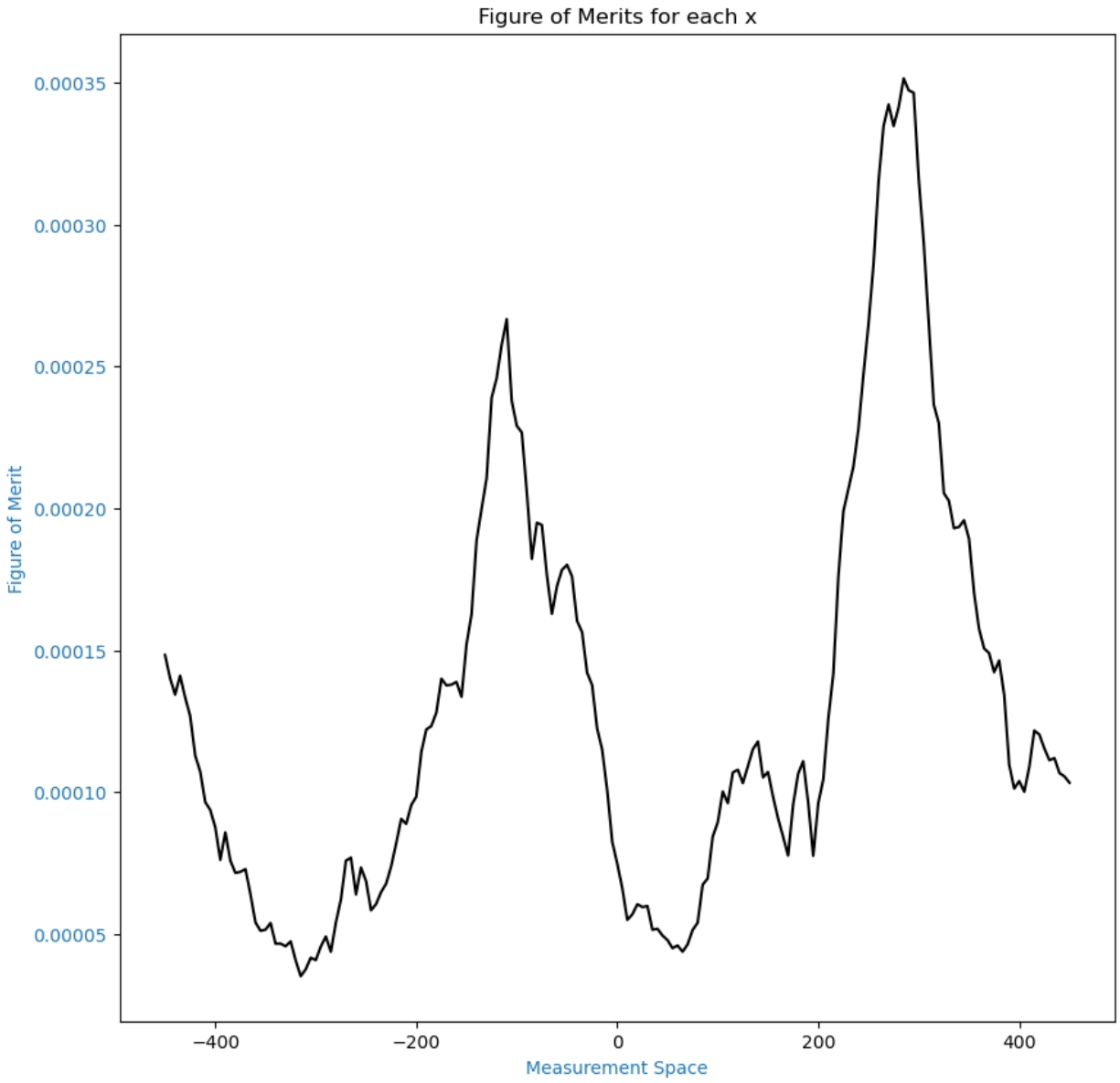
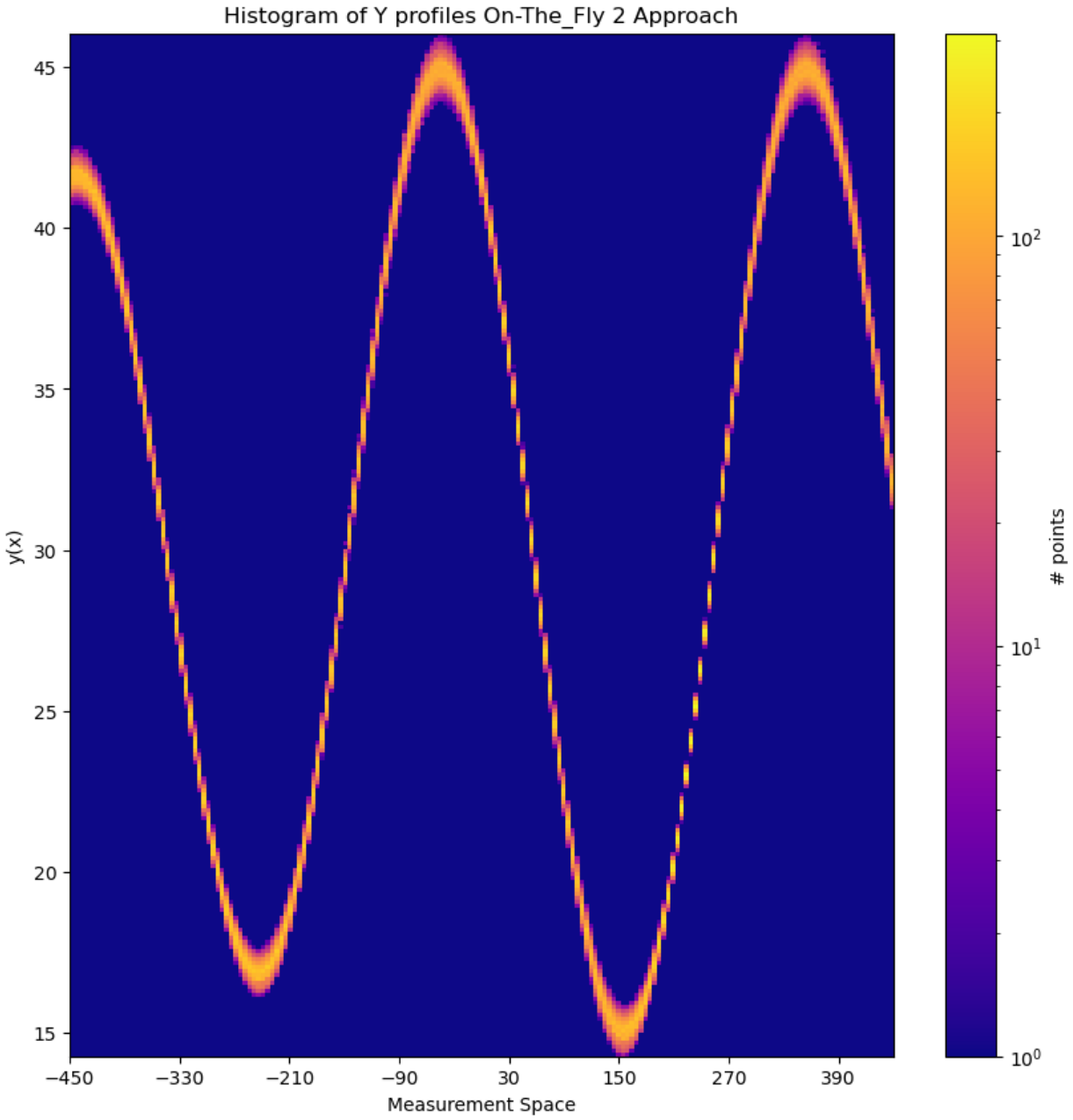


Figure of Merits for each x
Figure of Merit
0.00035
0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
−400
−200
0
200
400
Measurement Space


Histogram of Y profiles On-The_Fly 2 Approach
y(x)
Measurement Space
points
45
40
35
30
25
20
15
−450
−330
−210
−90
30
150
270
390
10^2
10^1
10^0

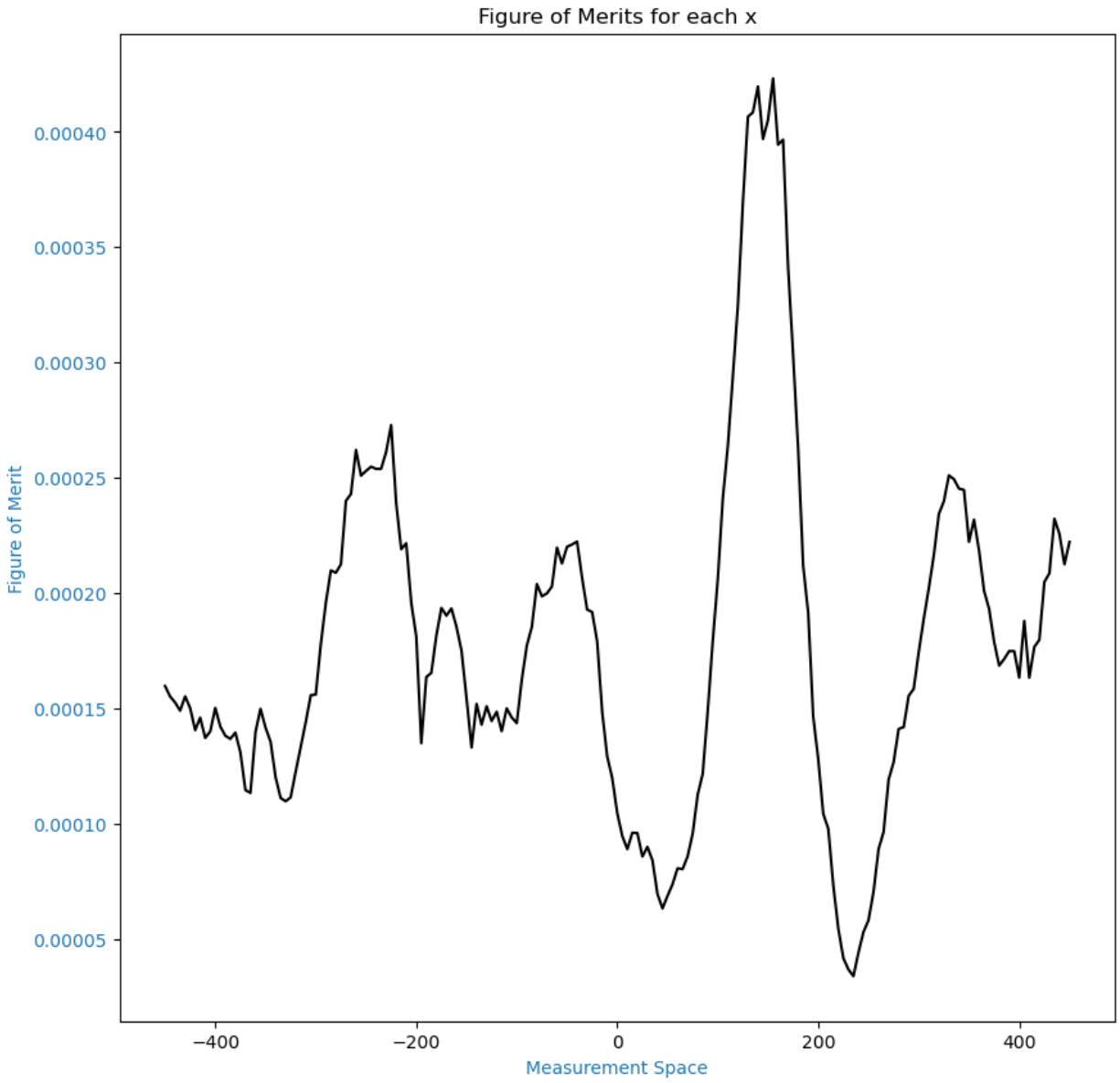
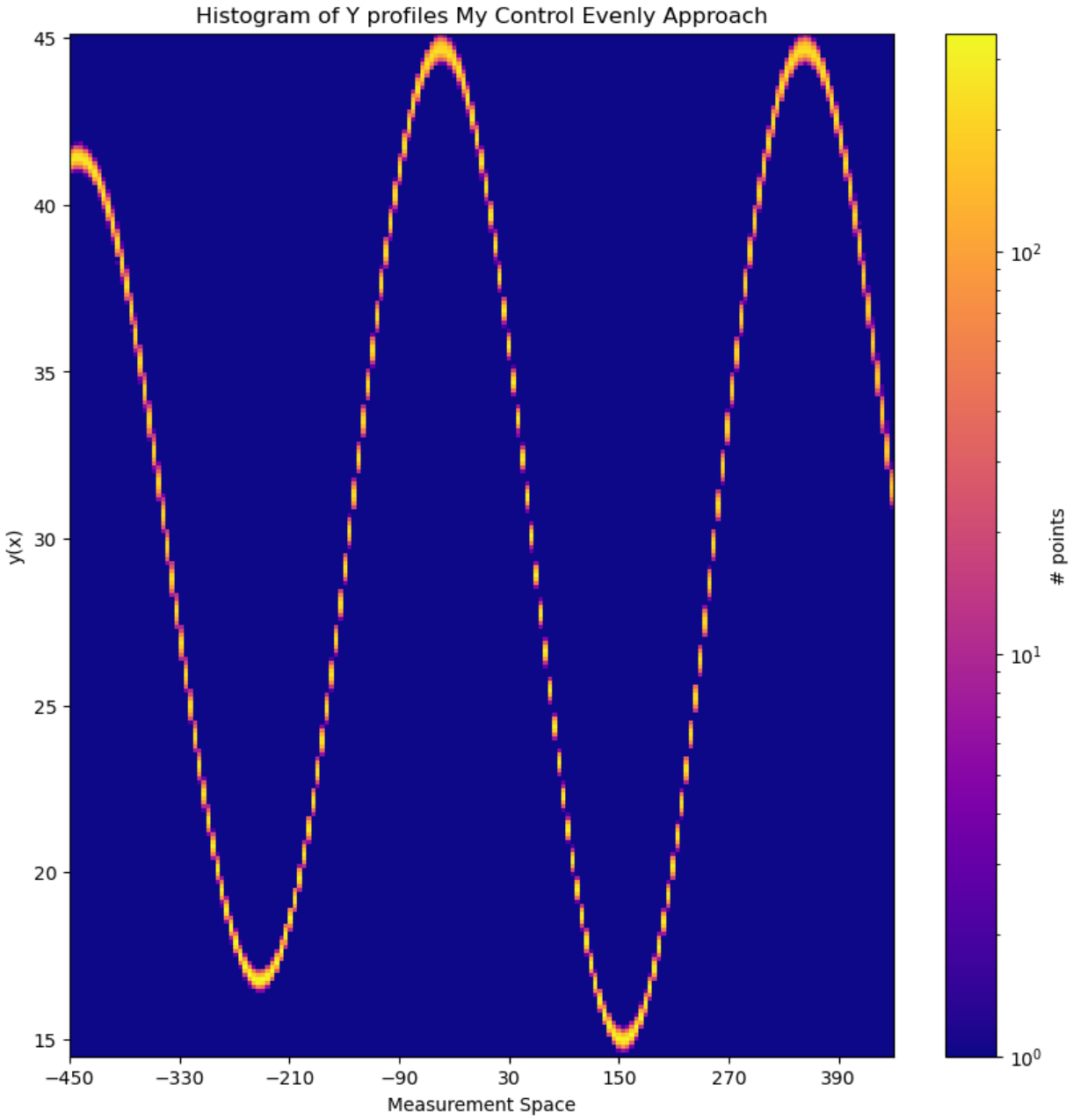


Figure of Merits for each x
Figure of Merit
0.00040
0.00035
0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
−400
−200
0
200
400
Measurement Space



In [7]:
```
############################## GMM TO COMPARE Commnet out everything below unless you ha


# #Entropy 1
# MyPlots.plot_y_profiles(exp_entropy1_gmm.load_yprofs()[-1],exp_entropy1_gmm.settings
#                         FOM = exp_entropy1_gmm.load_FOM()[-1],
#                         this_title = "Histogram of Y profiles Entropy 1 (Selected) G

# #Entropy 2
# MyPlots.plot_y_profiles(exp_entropy2_gmm.load_yprofs()[-1],exp_entropy2_gmm.settings
#                         FOM = exp_entropy2_gmm.load_FOM()[-1],
#                         this_title = "Histogram of Y profiles Entropy 2 (All) GMM Ap
# #On-the Fly 1
# MyPlots.plot_y_profiles(exp_on_the_fly1_gmm.load_yprofs()[-1],exp_on_the_fly1_gmm.se
#                         FOM = exp_on_the_fly1_gmm.load_FOM()[-1],
#                         this_title = "Histogram of Y profiles On-The_Fly 1 GMM Appro
# #On-the Fly 1
# MyPlots.plot_y_profiles(exp_on_the_fly2_gmm.load_yprofs()[-1],exp_on_the_fly2_gmm.se
#                         FOM = exp_on_the_fly2_gmm.load_FOM()[-1],
```

```
#                         this_title = "Histogram of Y profiles On-The_Fly 2 GMM Appro
# #Control Data
# MyPlots.plot_y_profiles(exp_control_gmm.load_yprofs()[-1],exp_control_gmm.settings.x
#                         this_title = "Histogram of Y profiles My Control Evenly GMM
```

10.2 Histogram for the Parameters Samples

Getting the Lists for the Parameters The containers total_pts (and similars) stores 2-D arrays on them; each 2-d array represents an iteration. The 2-d array contain the samples for all the parameters n_samples*n_parameters (where the number of samples can change per iteration, but the number of columns is constant). First,based on the columns of each array (each column represents a parameter) we will create lists that represent each parameter where each sub-list inside the list for a given paremeter represents the samples at an iterations, n_iterations*n_samples

Note: 1) Any of the 2-d arays in this_total_pts and this_total_pts2 have as columns representing each parameter. The columns represent respectvily A, I0, T,and phi0. They are in the same order than in the console output from the loop.

2) We Cannot create 2D arrays for each parameter because the number of samples per iteration can change. Thus, we create lists for each parameter.

plottinh_hist() Here we create an auxiliary funtion to plot each of the sublists (that represent the samples at all the iterations for a given parameter) of the output of using the function above.

WARNING: Remember that the number of samples per iteration can change. Thus, a histogram may not be the best way to visualize converge. we have two options.

1. Normalize the number of samples for al iterations.
2. commnet out the line of code mark_outliers() in the main loop. So, we do not rule outliers and then we will have the same number of samples for all iterations.

plottinh_hist()

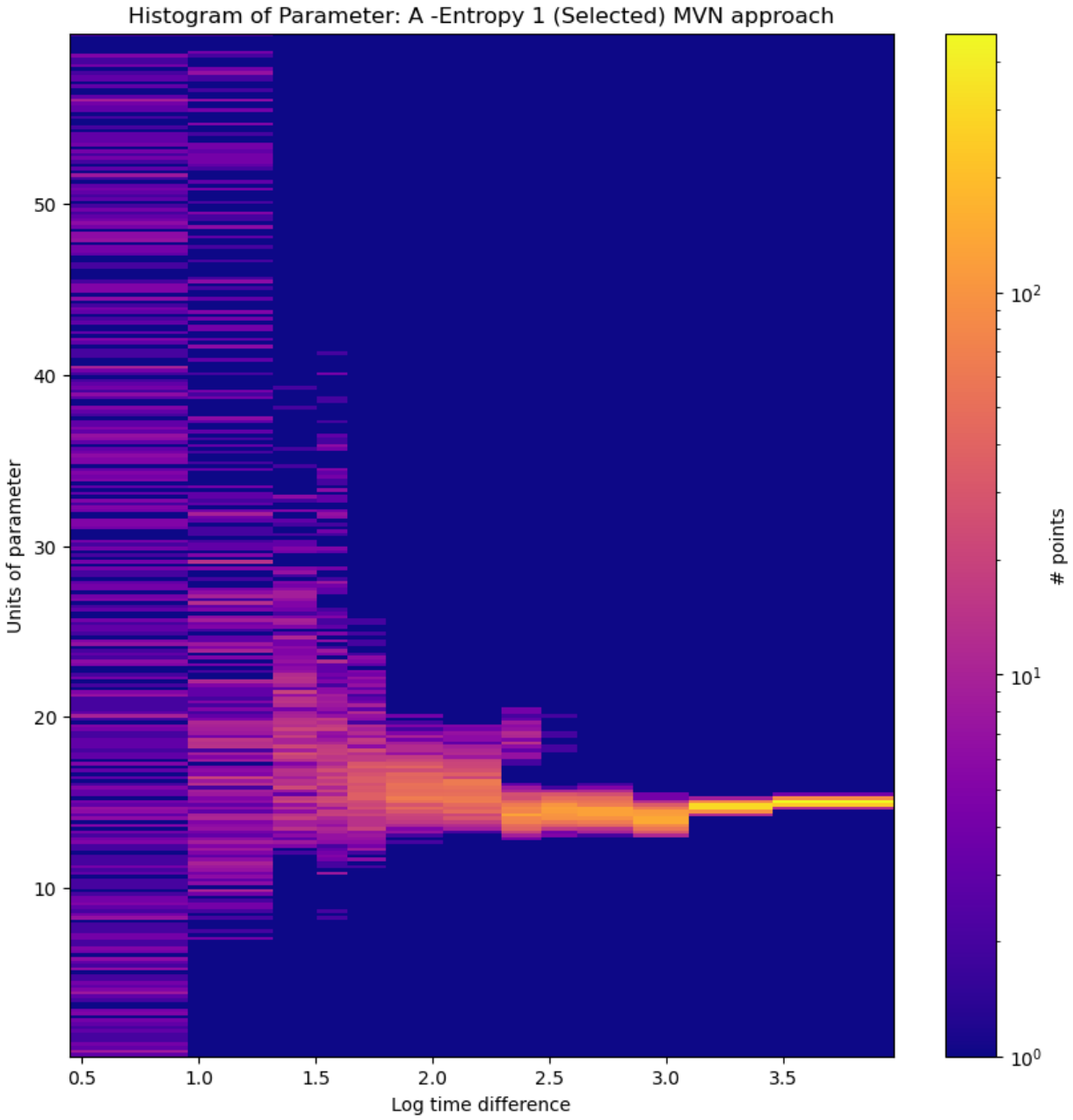
In [8]:
```
#getting the lists for both approaches. The lists contains lists where each sublists i
#sublists that represent a parameter there are sublists that represent the samples for
#at an iteration.

#Entropy Method
list_par_separated_e1 = MyPlots.getting_list_each_par(exp_entropy1.load_pts()) #This l
list_par_separated_e2 = MyPlots.getting_list_each_par(exp_entropy2.load_pts())
#On-the-fly 1 Method
list_par_separated_o1 = MyPlots.getting_list_each_par(exp_on_the_fly1.load_pts())
#On-the-fly 2 Method
list_par_separated_o2 = MyPlots.getting_list_each_par(exp_on_the_fly2.load_pts())
#Control method
list_par_separated_c = MyPlots.getting_list_each_par(exp_control.load_pts())

#######################################GMM VERSION


# #Entropy Method
```

```
# list_par_separated_e1_gmm = MyPlots.getting_list_each_par(exp_entropy1_gmm.load_pts(
# list_par_separated_e2_gmm = MyPlots.getting_list_each_par(exp_entropy2_gmm.load_pts(
# #On-the-fly 1 Method
# list_par_separated_o1_gmm = MyPlots.getting_list_each_par(exp_on_the_fly1_gmm.load_p
# #On-the-fly 2 Method
# list_par_separated_o2_gmm = MyPlots.getting_list_each_par(exp_on_the_fly2_gmm.load_p
# #Control method
# list_par_separated_c_gmm = MyPlots.getting_list_each_par(exp_control_gmm.load_pts())
```

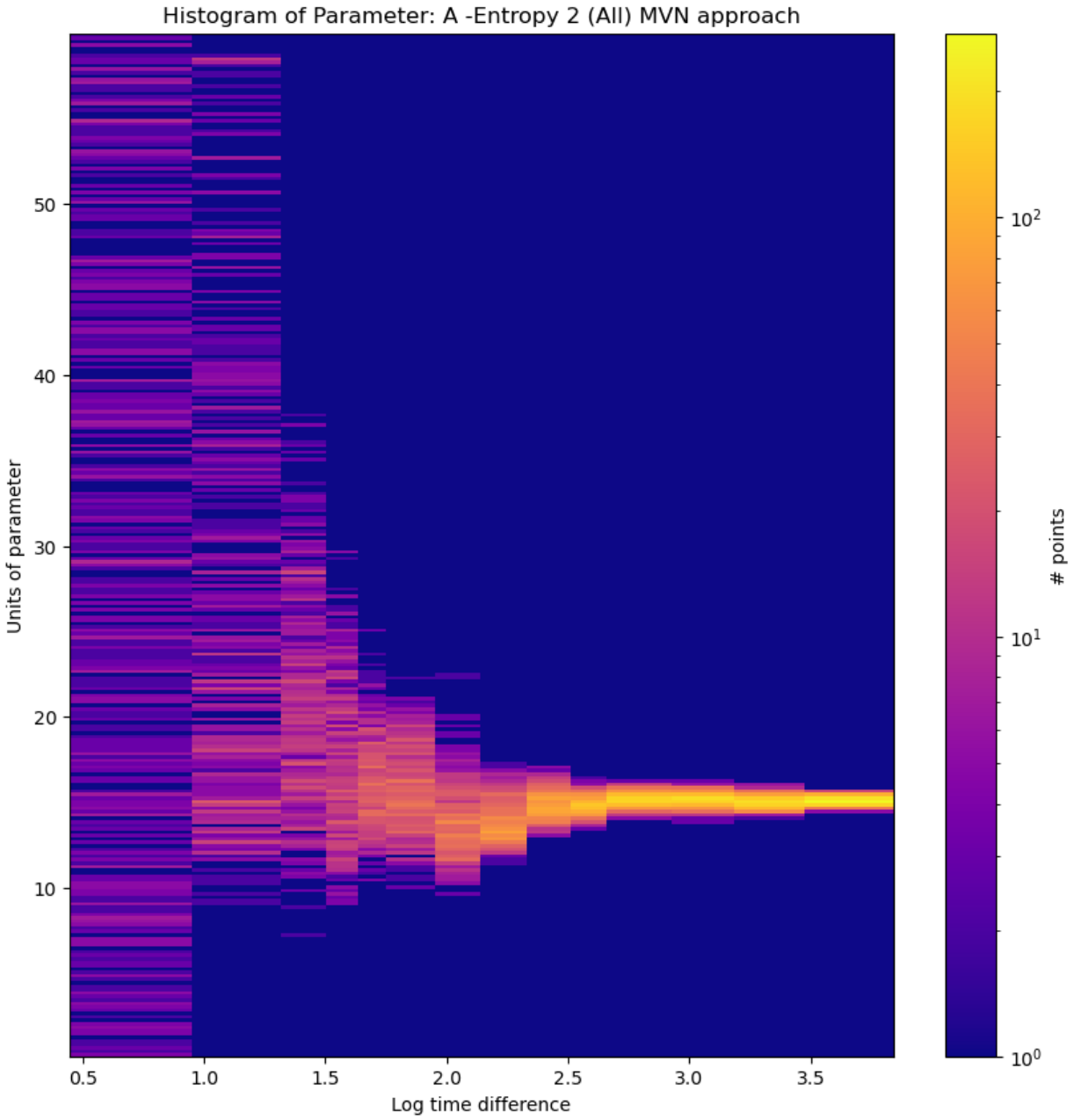
In [9]:

```
aux_list = [list_par_separated_e1[0],list_par_separated_e2[0], list_par_separated_o1[0
            list_par_separated_o2[0], list_par_separated_c[0]]

list_times = [exp_entropy1.totaltimes(), exp_entropy2.totaltimes(), exp_on_the_fly1.to
             exp_on_the_fly2.totaltimes(), exp_control.totaltimes()]

names_for_approaches = ["Entropy 1 (Selected) MVN", "Entropy 2 (All) MVN","On-the fly
                        "Control-45 even MVN"]
par_name = "A"



# specify y-edges for all three histograms
y_min, y_max = MyPlots.finding_max_min(aux_list)

for i in range(len(aux_list)):
    MyPlots.plotting_hist_logtime(aux_list[i], names_for_approaches[i], par_name, y_mi



#################GMM VERSION COMMNENT OUT IF YOU DO NOT HAVE A GMM VERSION

# aux_list = [list_par_separated_e1_gmm[0],list_par_separated_e2_gmm[0], list_par_sepa
#             list_par_separated_o2_gmm[0], list_par_separated_c_gmm[0]]

# names_for_approaches = ["Entropy 1 (Selected) GMM", "Entropy 2 (All) GMM","On-the fl
#                         "Control-45 even GMM"]
# par_name = "A"



# # specify y-edges for all three histograms
# y_min, y_max = MyPlots.finding_max_min(aux_list)

# for i in range(len(aux_list)):
#     MyPlots.plotting_hist(aux_list[i], names_for_approaches[i], par_name, y_min, y_m
```

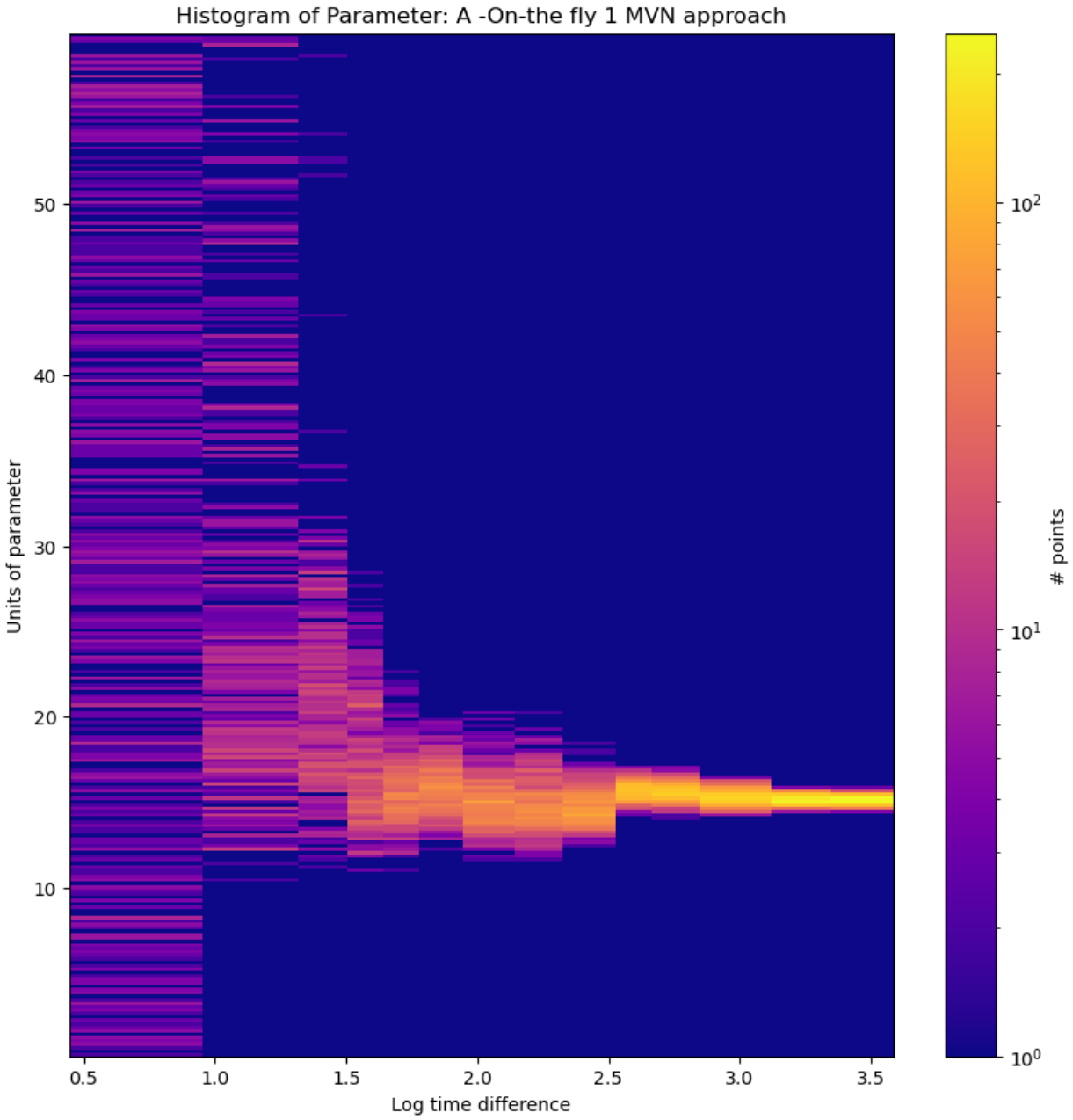
```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.79924284
 2.04446783 2.29341009 2.46357343 2.61964425 2.85901898 3.09807981
 3.45534982 3.97317463]
```



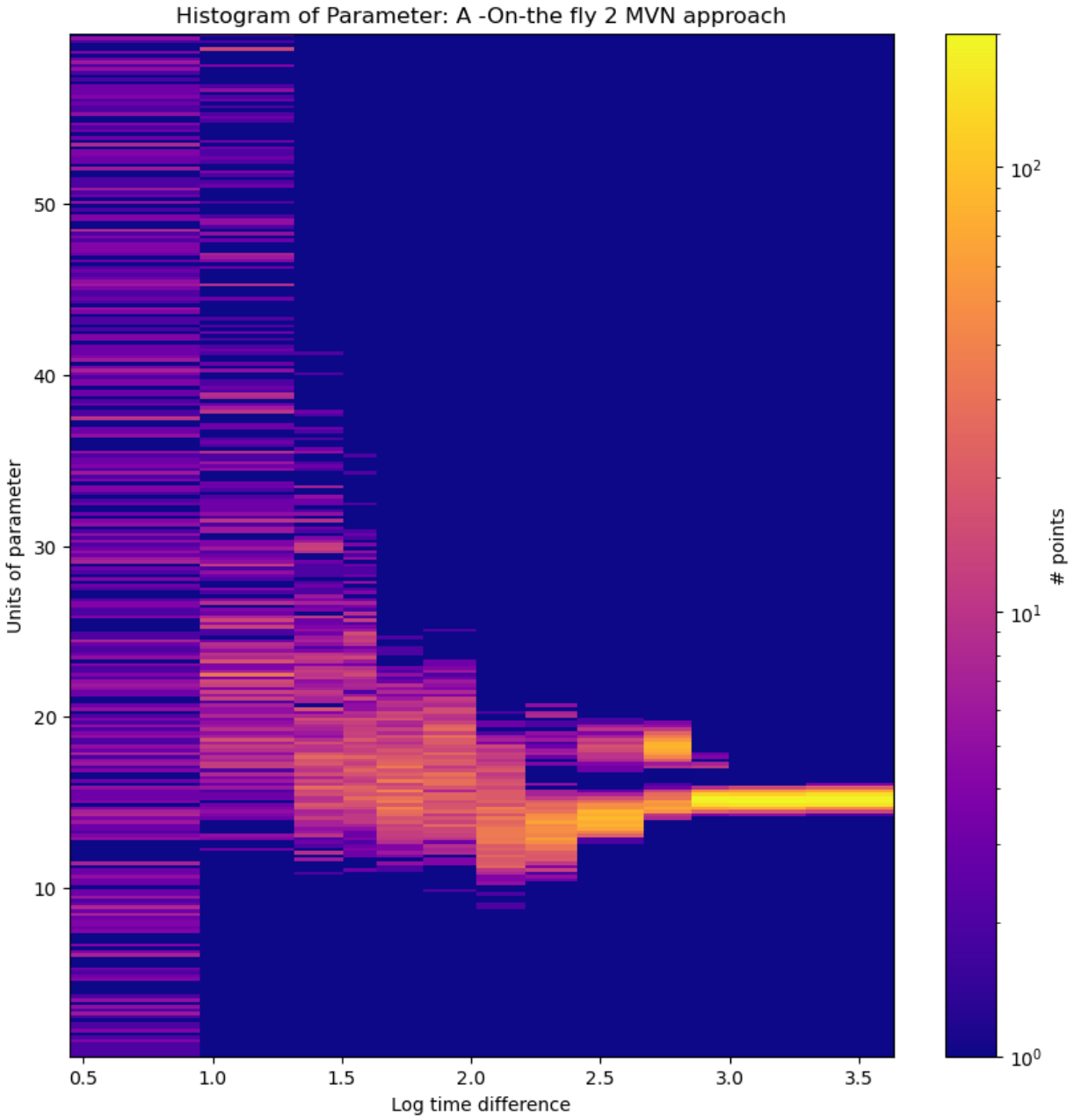

```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.75088287
 1.9497747  2.14092984 2.33077169 2.50838469 2.6594732  2.92690925
 3.18585244 3.4724728  3.84196472]
```



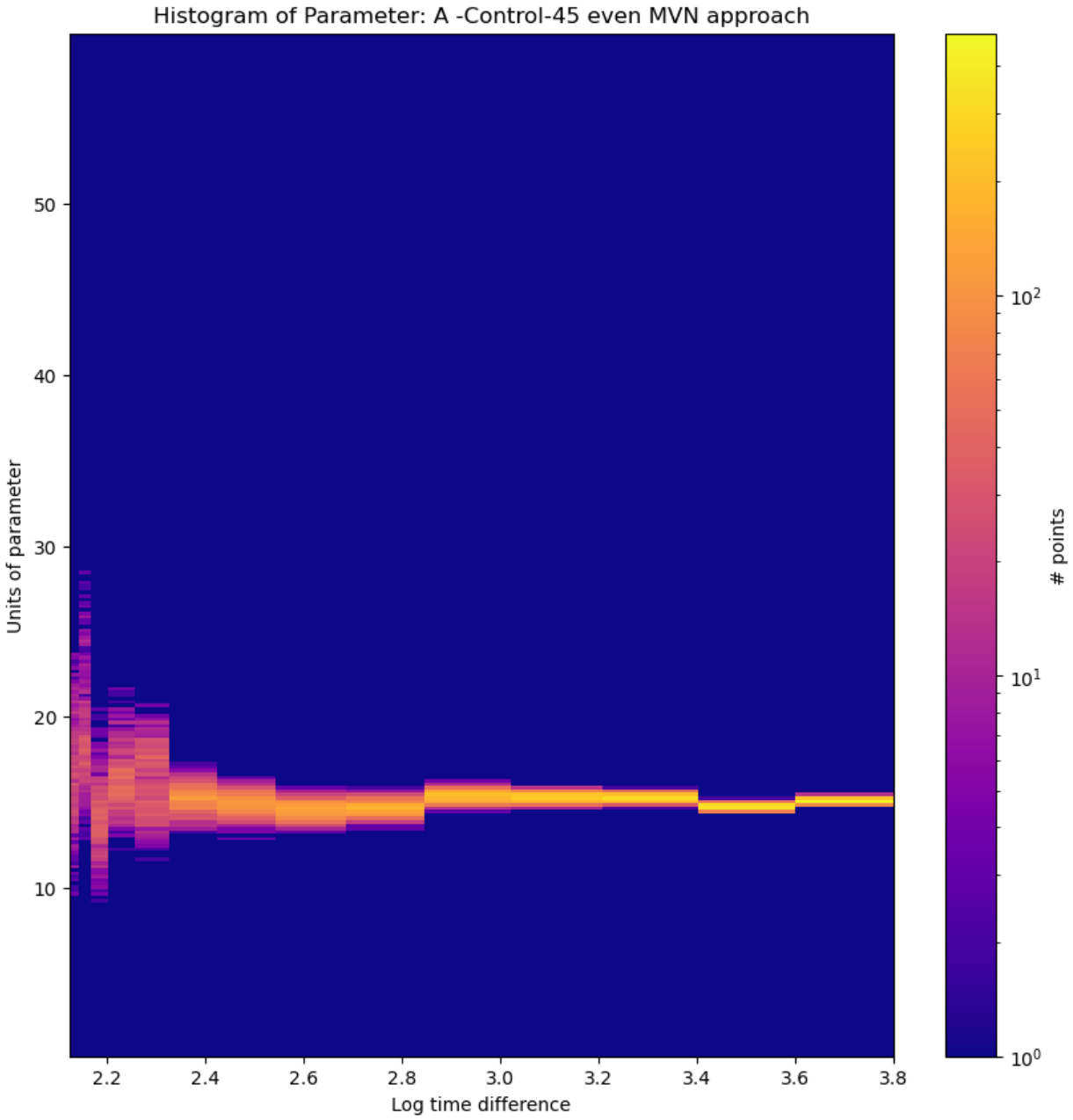

```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63925132 1.78039166
 1.94541665 2.14036554 2.32329382 2.52597764 2.6659401  2.84474297
 3.12212717 3.34835091 3.58641497]
```




```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.81627773
 2.02180203 2.21024151 2.41133864 2.6672224  2.8538082  2.9992758
 3.29278834 3.63448291]
```



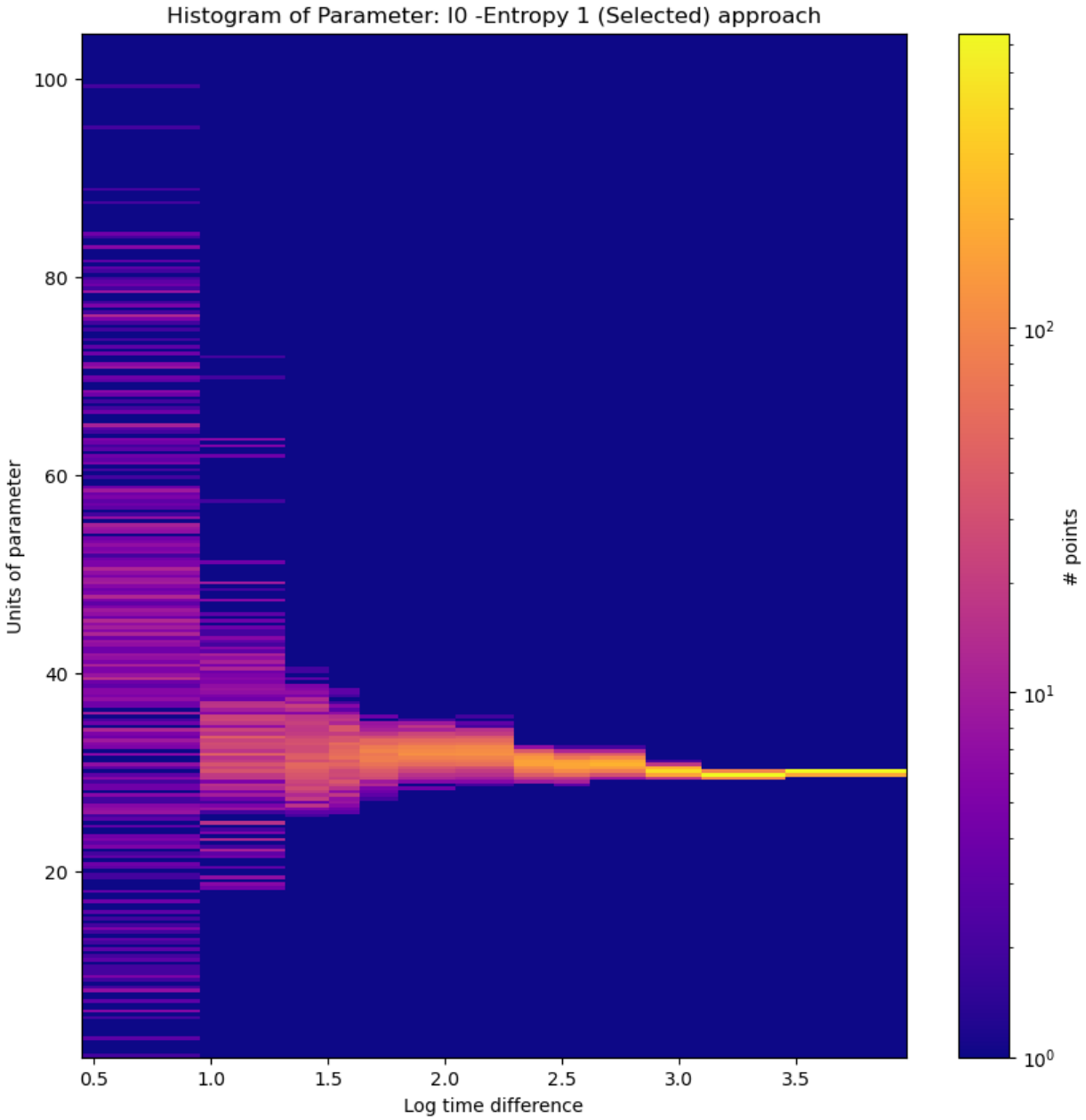

```
my x edges
[2.12400496 2.14307285 2.16713147 2.203179   2.25542118 2.32797081
 2.42377446 2.54362306 2.68585652 2.84696932 3.02270427 3.209021
 3.40261866 3.60105122 3.80139375]
```



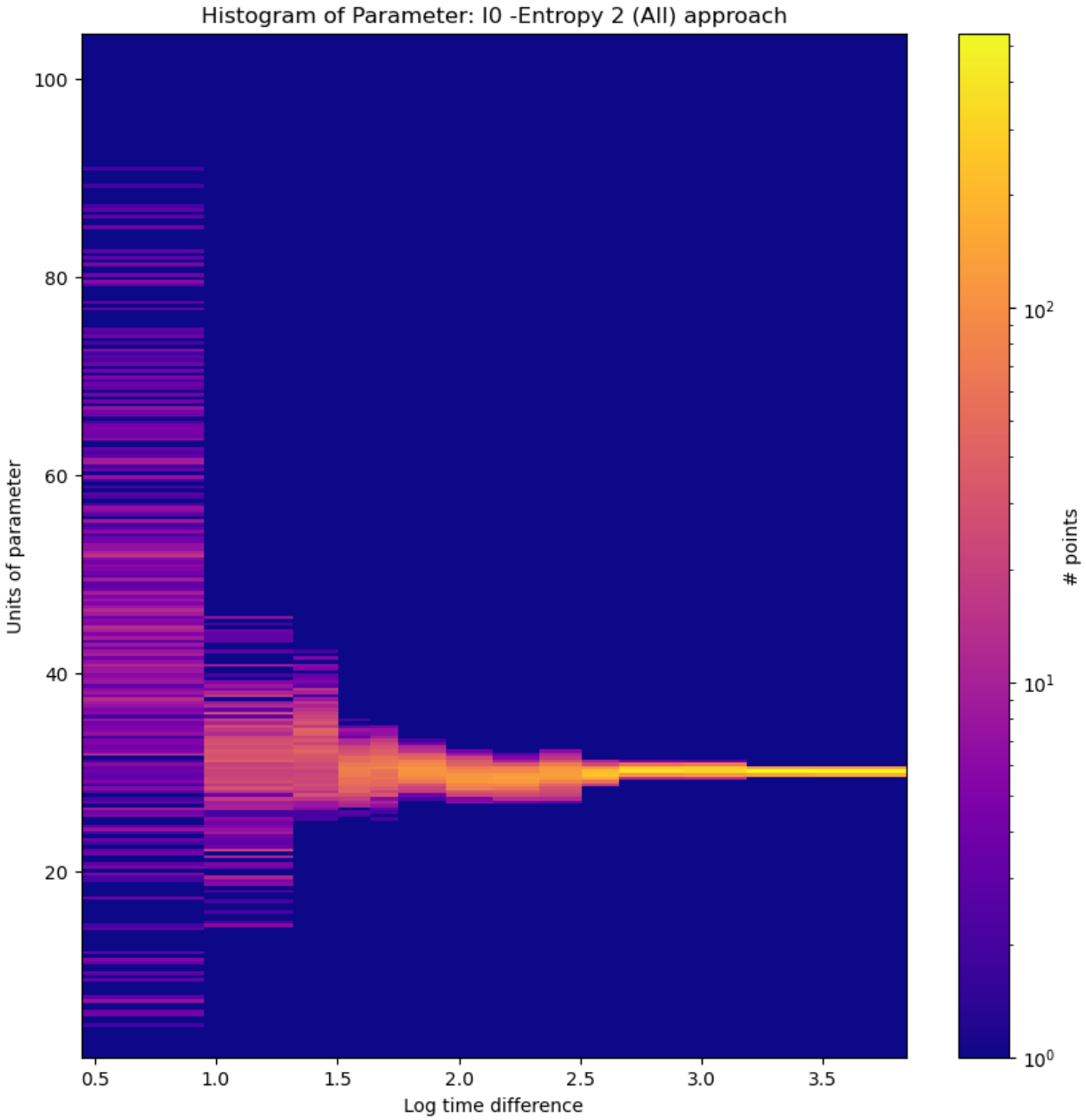
In [10]:
```
aux_list = [list_par_separated_e1[1],list_par_separated_e2[1], list_par_separated_o1[1
            list_par_separated_o2[1], list_par_separated_c[1]]

list_times = [exp_entropy1.totaltimes(), exp_entropy2.totaltimes(), exp_on_the_fly1.to
             exp_on_the_fly2.totaltimes(), exp_control.totaltimes()]


names_for_approaches = ["Entropy 1 (Selected)", "Entropy 2 (All)","On-the fly 1","On-t
                        "Control-45 even"]
par_name = "I0"



# specify y-edges for all three histograms

#WARNING: If you are plotting the GMM versions, you will need to include the values us
#values below
y_min, y_max = MyPlots.finding_max_min(aux_list)
```

```
for i in range(len(aux_list)):
    MyPlots.plotting_hist_logtime(aux_list[i], names_for_approaches[i], par_name, y_mi



################GMM VERSION COMMNENT OUT IF YOU DO NOT HAVE A GMM VERSION

# aux_list = [list_par_separated_e1_gmm[1],list_par_separated_e2_gmm[1], list_par_sepa
#             list_par_separated_o2_gmm[1], list_par_separated_c_gmm[1]]

# names_for_approaches = ["Entropy 1 (Selected) GMM", "Entropy 2 (All) GMM","On-the fl
#                         "Control-45 even GMM"]
# par_name = "I0"



# # specify y-edges for all three histograms
# y_min, y_max = MyPlots.finding_max_min(aux_list)

# for i in range(len(aux_list)):
#     MyPlots.plotting_hist(aux_list[i], names_for_approaches[i], par_name, y_min, y_m
```

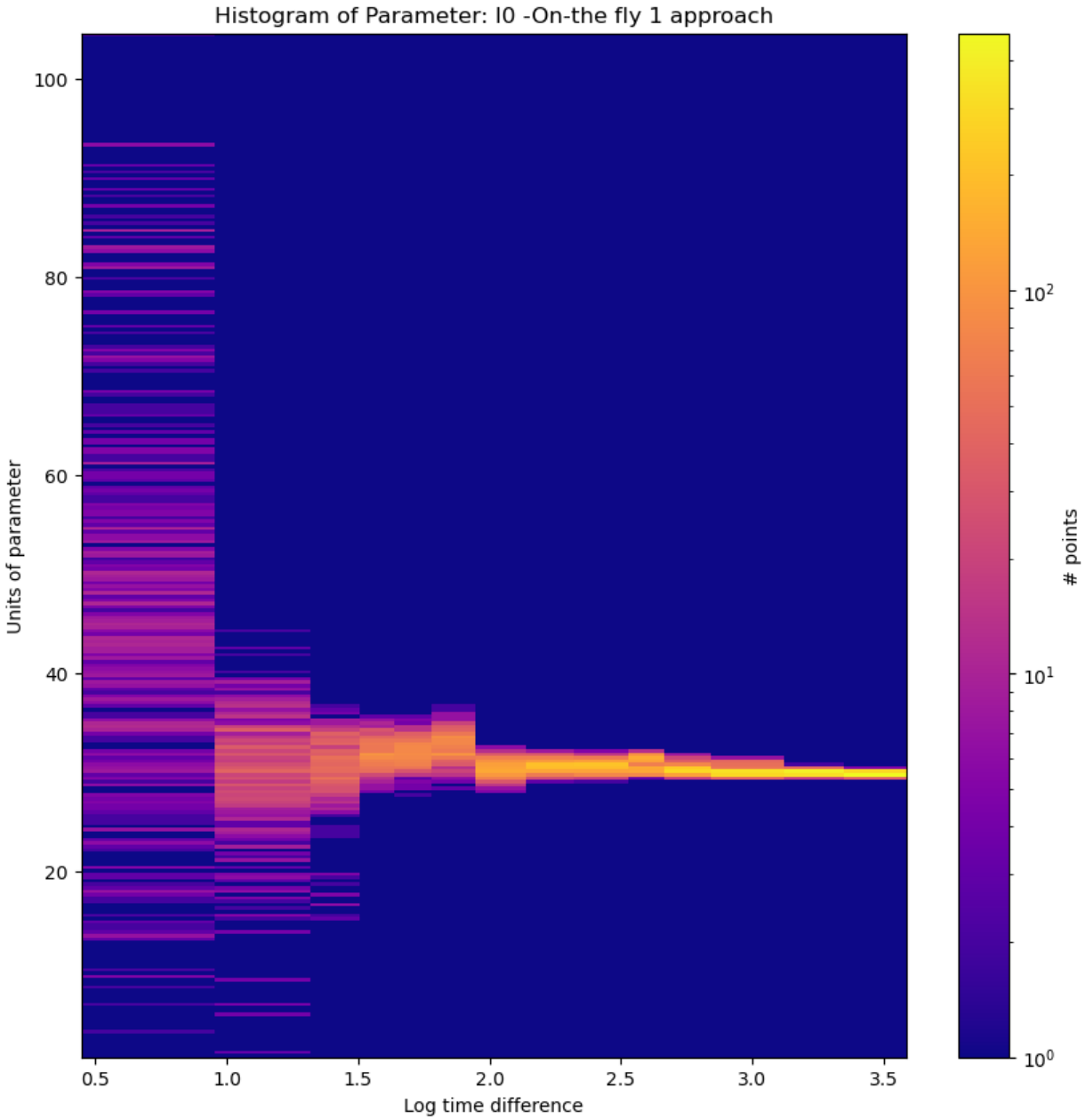
```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.79924284
 2.04446783 2.29341009 2.46357343 2.61964425 2.85901898 3.09807981
 3.45534982 3.97317463]
```



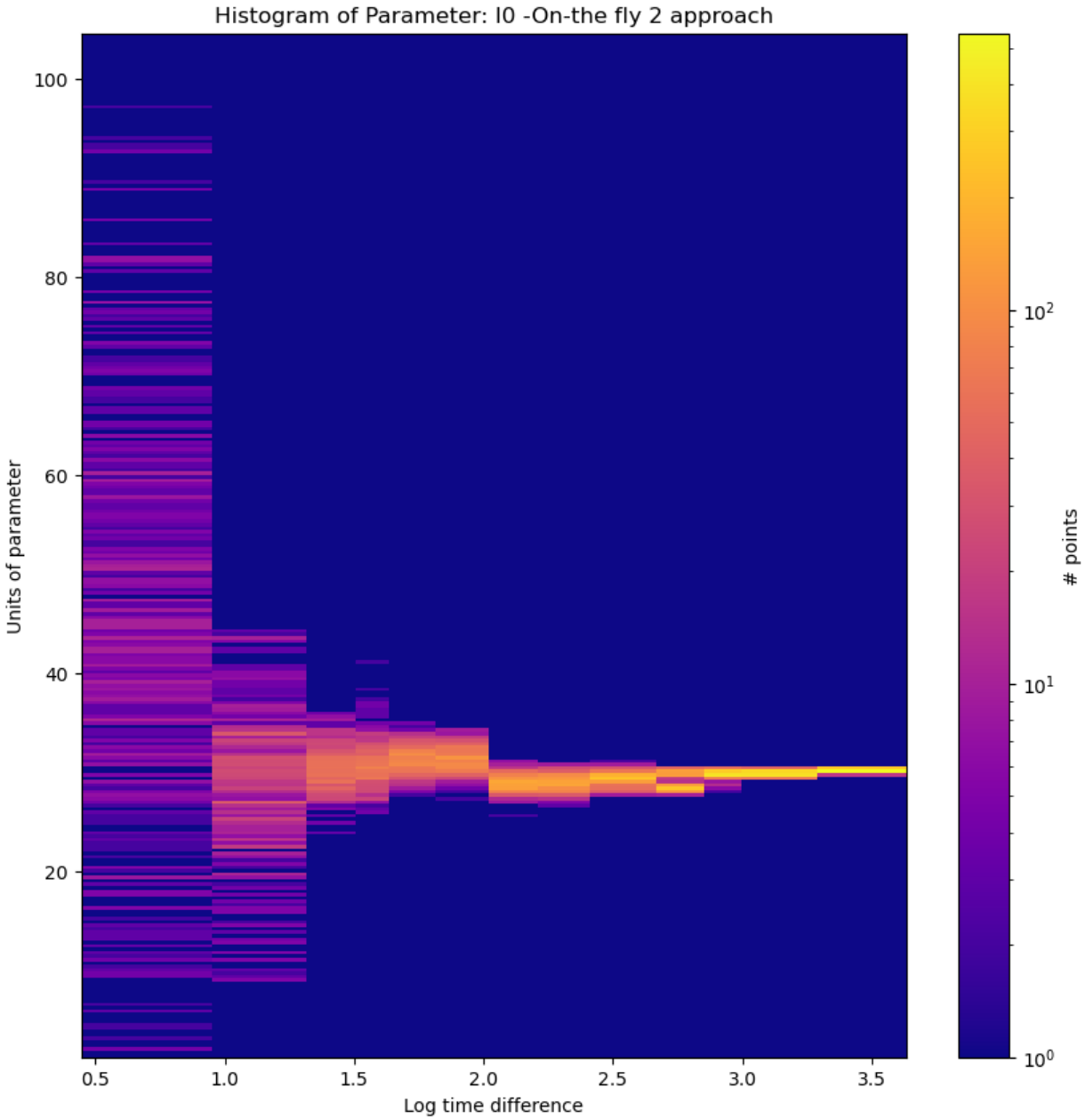

```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.75088287
 1.9497747  2.14092984 2.33077169 2.50838469 2.6594732  2.92690925
 3.18585244 3.4724728  3.84196472]
```



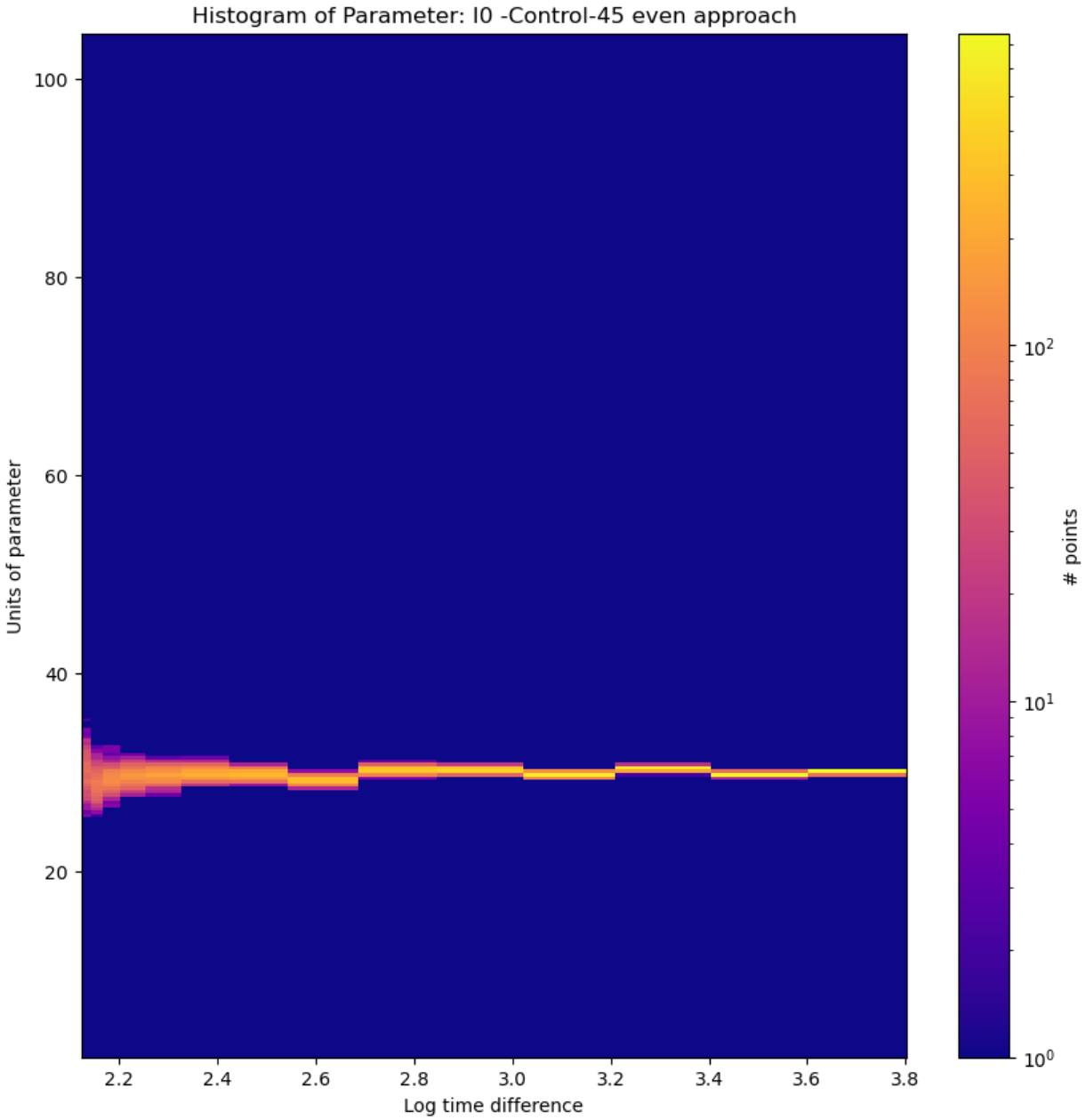

```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63925132 1.78039166
 1.94541665 2.14036554 2.32329382 2.52597764 2.6659401  2.84474297
 3.12212717 3.34835091 3.58641497]
```




```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.81627773
 2.02180203 2.21024151 2.41133864 2.6672224  2.8538082  2.9992758
 3.29278834 3.63448291]
```



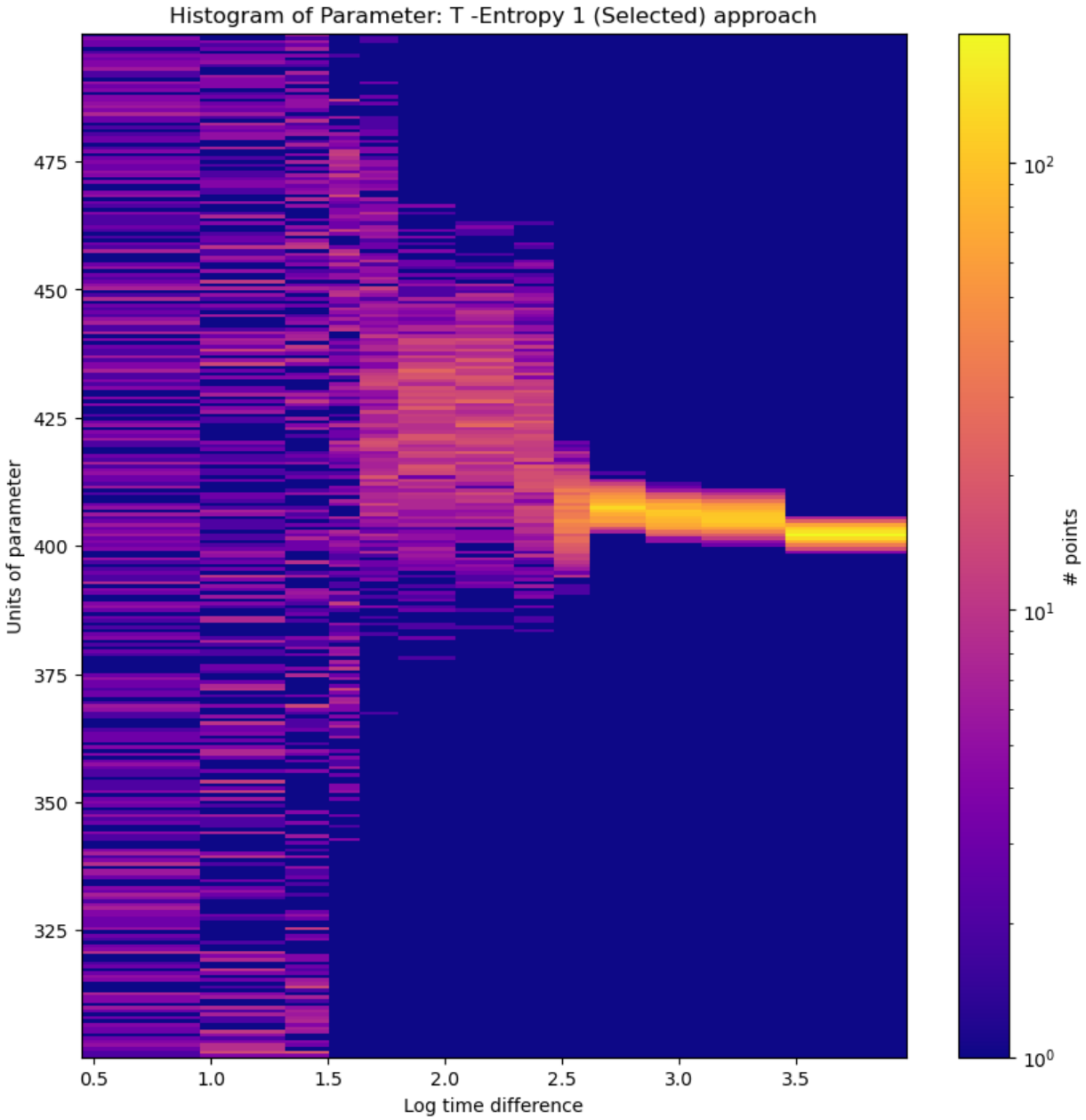

```
my x edges
[2.12400496 2.14307285 2.16713147 2.203179   2.25542118 2.32797081
 2.42377446 2.54362306 2.68585652 2.84696932 3.02270427 3.209021
 3.40261866 3.60105122 3.80139375]
```



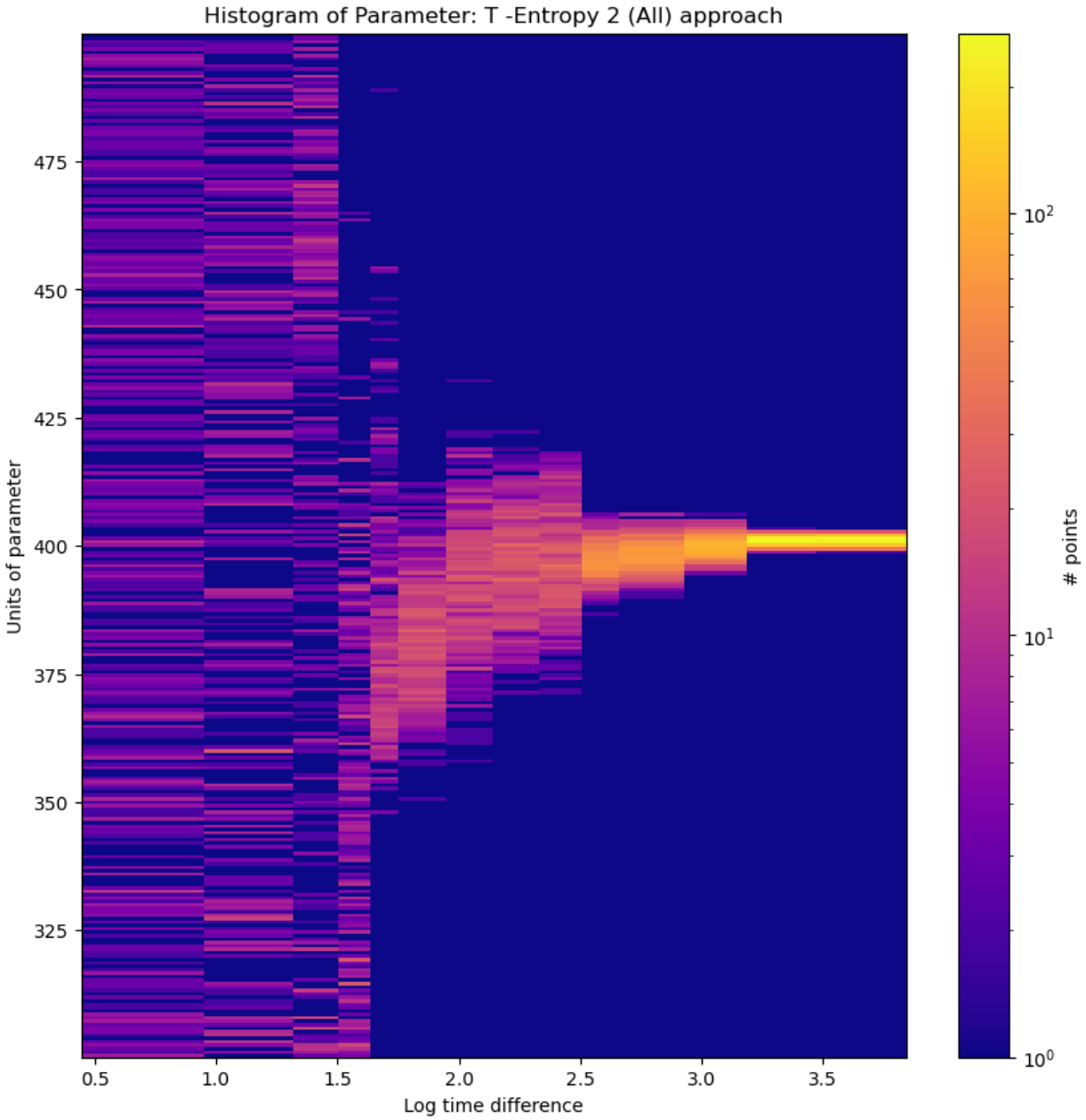
In [11]:
```
#Plotting histograms for T
aux_list = [list_par_separated_e1[2],list_par_separated_e2[2], list_par_separated_o1[2
            list_par_separated_o2[2], list_par_separated_c[2]]

list_times = [exp_entropy1.totaltimes(), exp_entropy2.totaltimes(), exp_on_the_fly1.to
              exp_on_the_fly2.totaltimes(), exp_control.totaltimes()]


names_for_approaches = ["Entropy 1 (Selected)", "Entropy 2 (All)","On-the fly 1","On-t
                        "Control-45 even"]
par_name = "T"



# specify y-edges for all three histograms
y_min, y_max = MyPlots.finding_max_min(aux_list)

for i in range(len(aux_list)):
    MyPlots.plotting_hist_logtime(aux_list[i], names_for_approaches[i], par_name, y_mi
```

```
################GMM VERSION COMMNENT OUT IF YOU DO NOT HAVE A GMM VERSION

# aux_list = [list_par_separated_e1_gmm[2],list_par_separated_e2_gmm[2], list_par_sepa
#             list_par_separated_o2_gmm[2], list_par_separated_c_gmm[2]]

# names_for_approaches = ["Entropy 1 (Selected) GMM", "Entropy 2 (All) GMM","On-the fl
#                         "Control-45 even GMM"]
# par_name = "T"



# # specify y-edges for all three histograms
# y_min, y_max = MyPlots.finding_max_min(aux_list)

# for i in range(len(aux_list)):
#     MyPlots.plotting_hist(aux_list[i], names_for_approaches[i], par_name, y_min, y_m
```

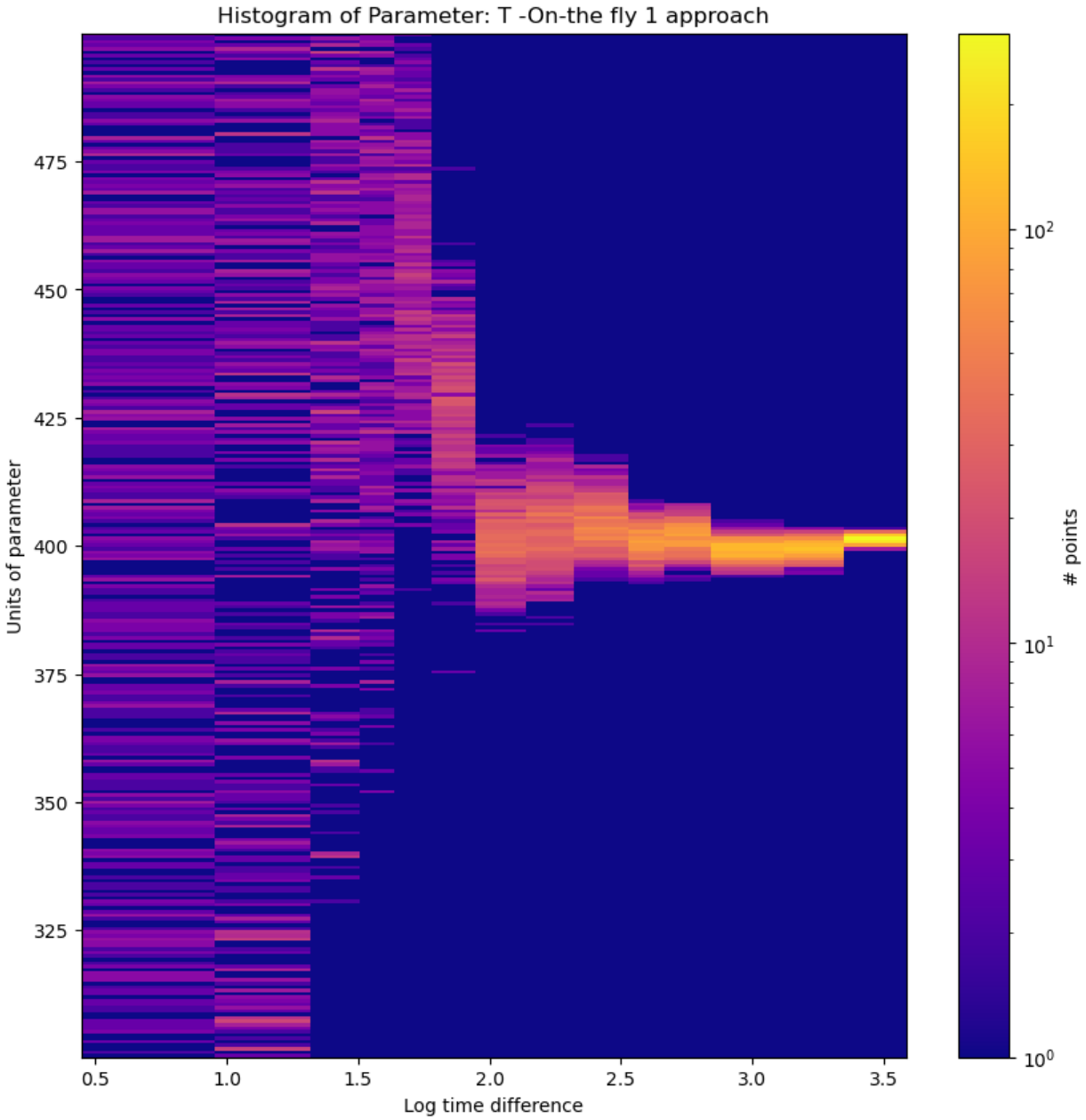
```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.79924284
 2.04446783 2.29341009 2.46357343 2.61964425 2.85901898 3.09807981
 3.45534982 3.97317463]
```



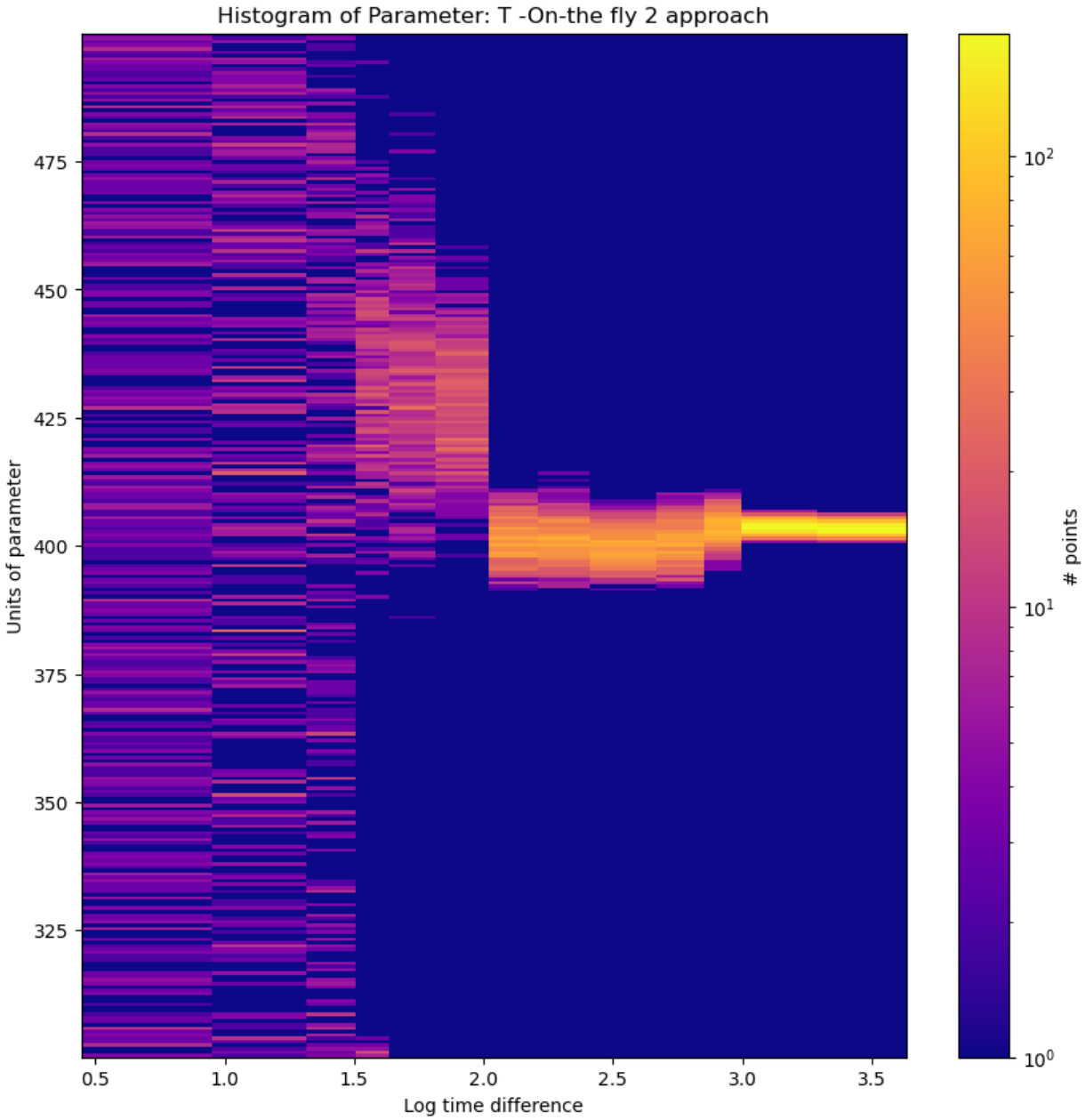

```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.75088287
 1.9497747  2.14092984 2.33077169 2.50838469 2.6594732  2.92690925
 3.18585244 3.4724728  3.84196472]
```



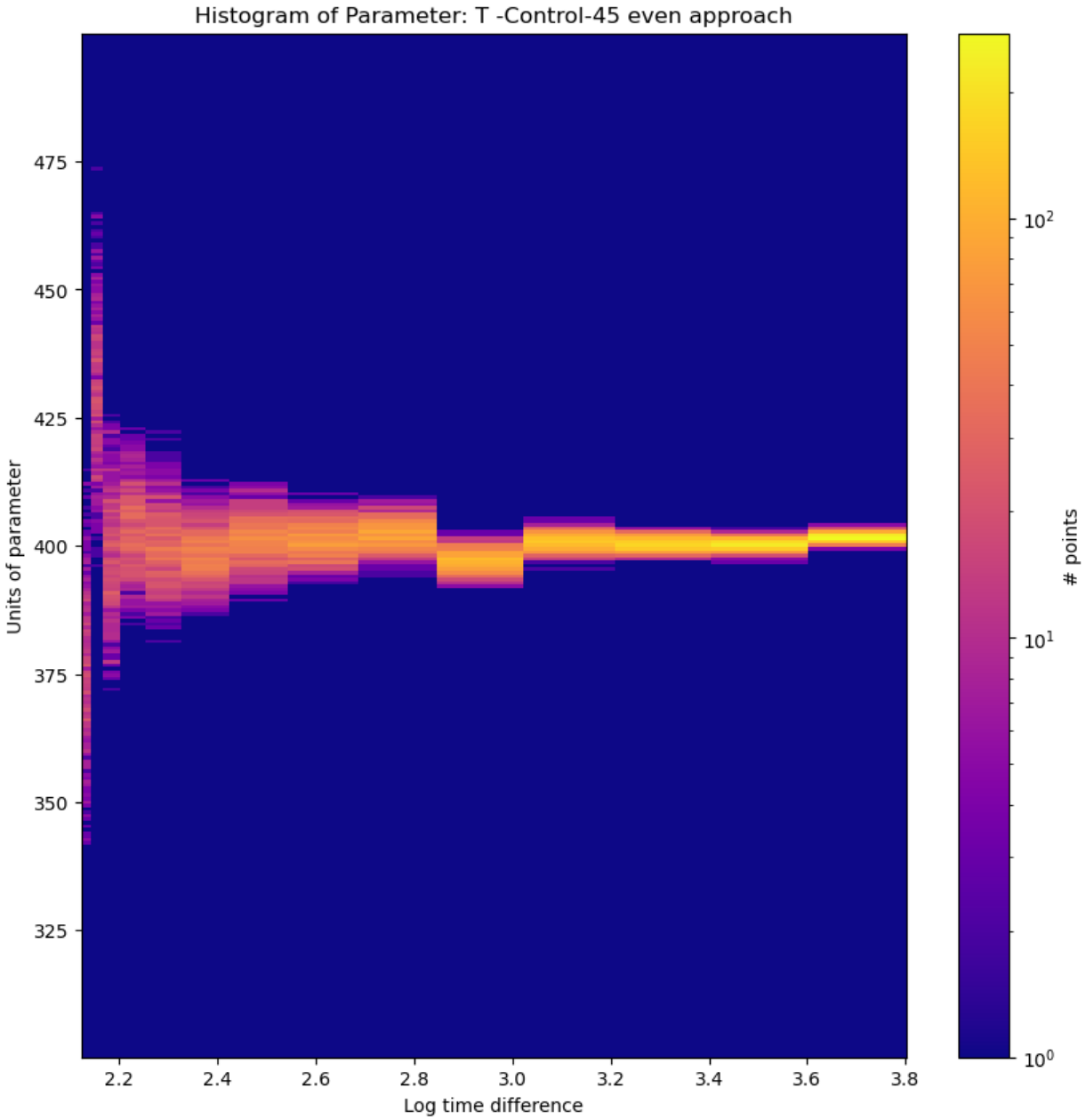

```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63925132 1.78039166
 1.94541665 2.14036554 2.32329382 2.52597764 2.6659401  2.84474297
 3.12212717 3.34835091 3.58641497]
```




```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.81627773
 2.02180203 2.21024151 2.41133864 2.6672224  2.8538082  2.9992758
 3.29278834 3.63448291]
```




```
my x edges
[2.12400496 2.14307285 2.16713147 2.203179   2.25542118 2.32797081
 2.42377446 2.54362306 2.68585652 2.84696932 3.02270427 3.209021
 3.40261866 3.60105122 3.80139375]
```

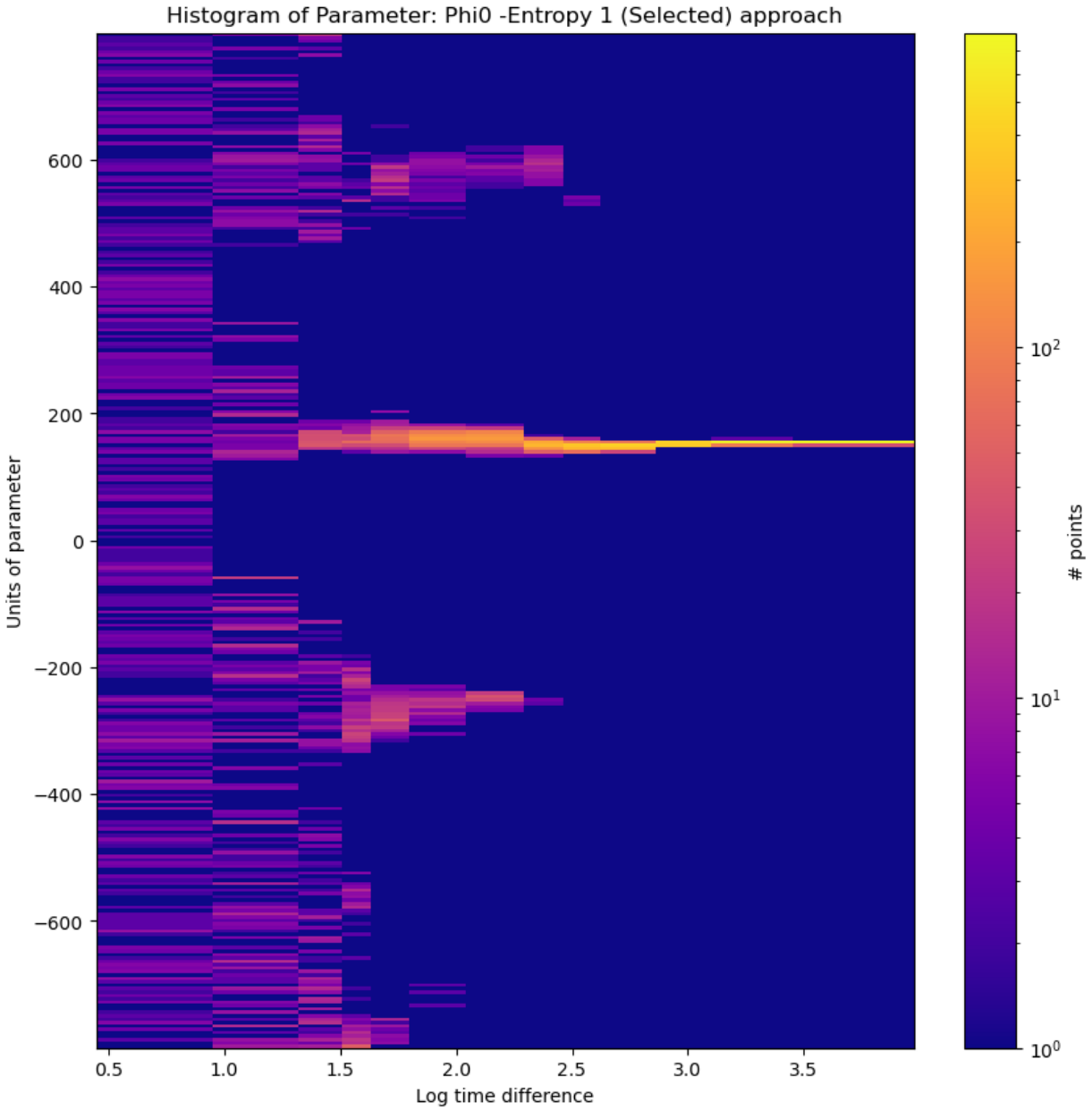


In [ ]:

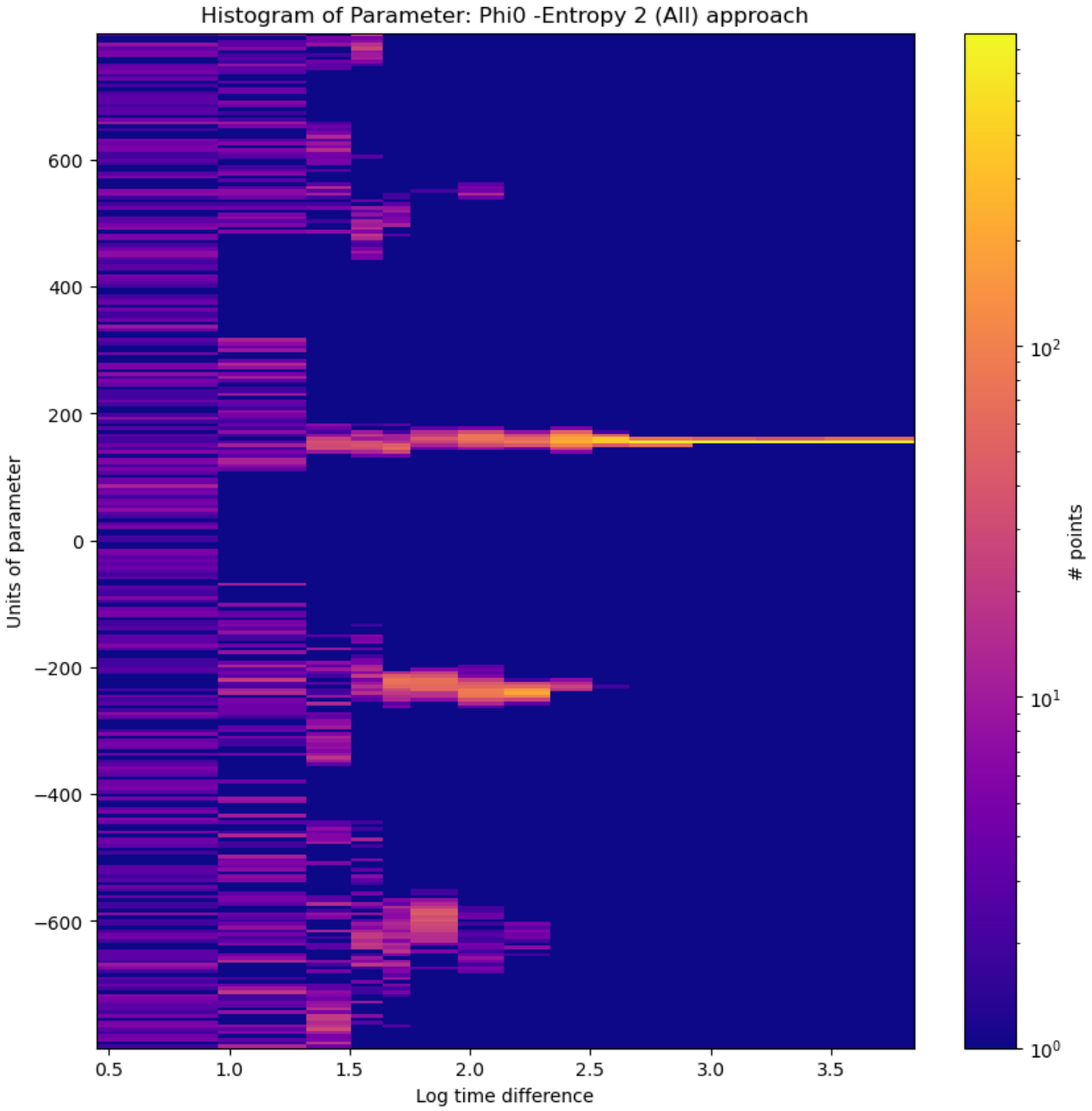
In [12]:
```
#Plotting histograms for Phi0
aux_list = [list_par_separated_e1[3],list_par_separated_e2[3], list_par_separated_o1[3
            list_par_separated_o2[3], list_par_separated_c[3]]

list_times = [exp_entropy1.totaltimes(), exp_entropy2.totaltimes(), exp_on_the_fly1.to
             exp_on_the_fly2.totaltimes(), exp_control.totaltimes()]


names_for_approaches = ["Entropy 1 (Selected)", "Entropy 2 (All)","On-the fly 1","On-t
                        "Control-45 even"]
par_name = "Phi0"



# specify y-edges for all three histograms
y_min, y_max = MyPlots.finding_max_min(aux_list)
```

```
for i in range(len(aux_list)):
    MyPlots.plotting_hist_logtime(aux_list[i], names_for_approaches[i], par_name, y_mi



################GMM VERSION COMMNENT OUT IF YOU DO NOT HAVE A GMM VERSION

# aux_list = [list_par_separated_e1_gmm[3],list_par_separated_e2_gmm[3], list_par_sepa
#             list_par_separated_o2_gmm[3], list_par_separated_c_gmm[3]]

# names_for_approaches = ["Entropy 1 (Selected) GMM", "Entropy 2 (All) GMM","On-the fl
#                         "Control-45 even GMM"]
# par_name = "Phi0"



# # specify y-edges for all three histograms
# y_min, y_max = MyPlots.finding_max_min(aux_list)

# for i in range(len(aux_list)):
#     MyPlots.plotting_hist(aux_list[i], names_for_approaches[i], par_name, y_min, y_m
```

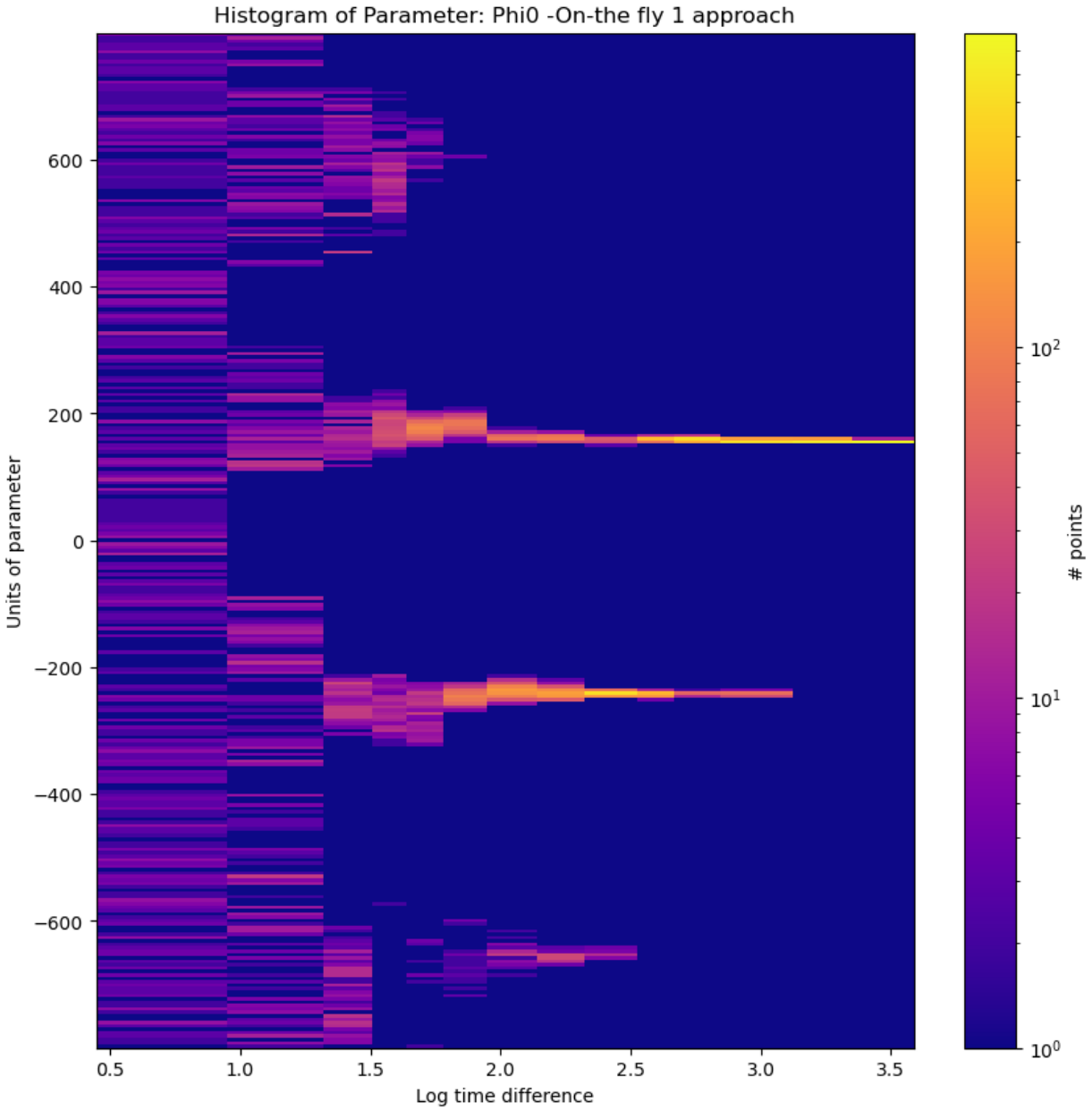
```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.79924284
 2.04446783 2.29341009 2.46357343 2.61964425 2.85901898 3.09807981
 3.45534982 3.97317463]
```



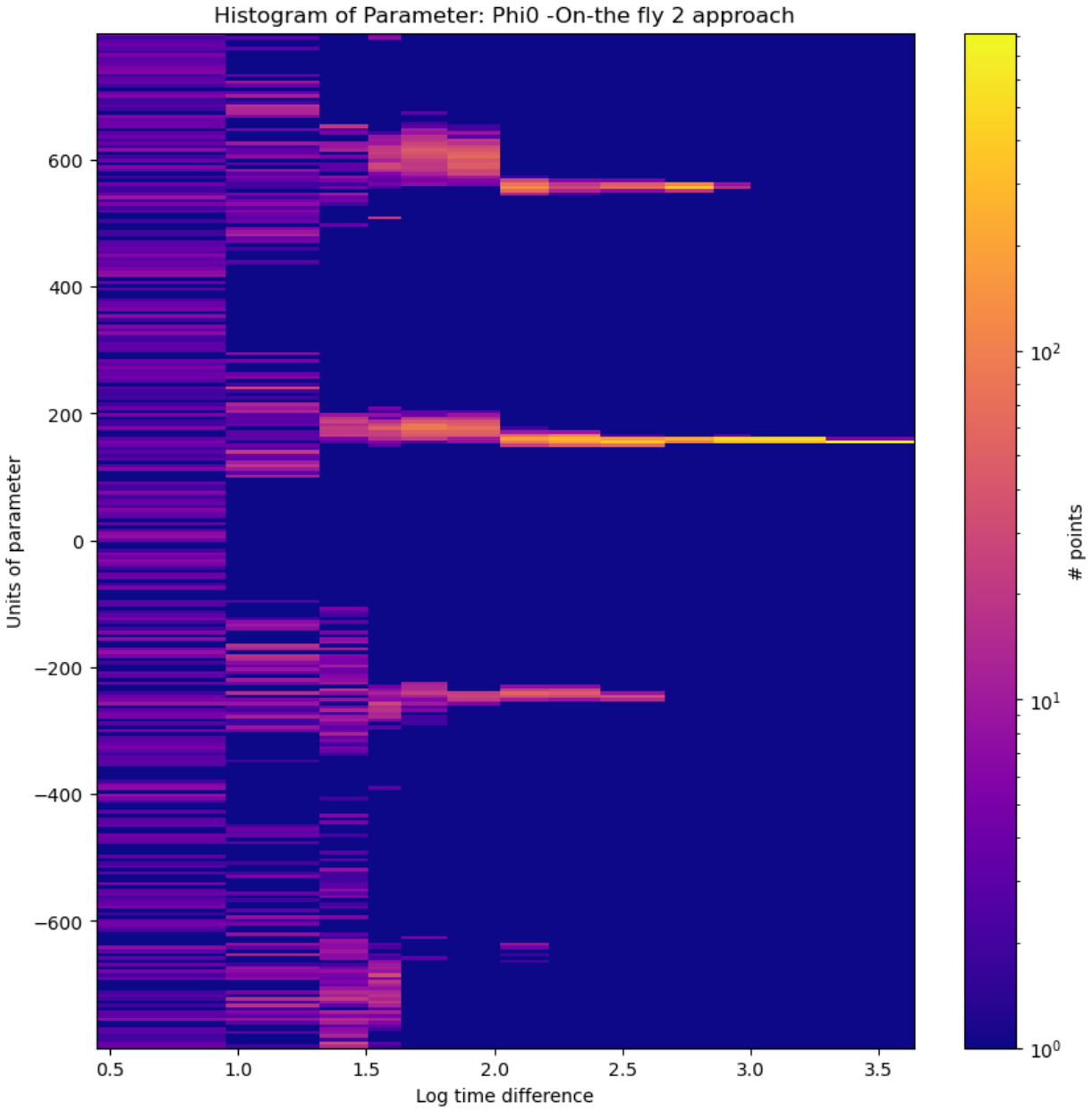

```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.75088287
 1.9497747  2.14092984 2.33077169 2.50838469 2.6594732  2.92690925
 3.18585244 3.4724728  3.84196472]
```



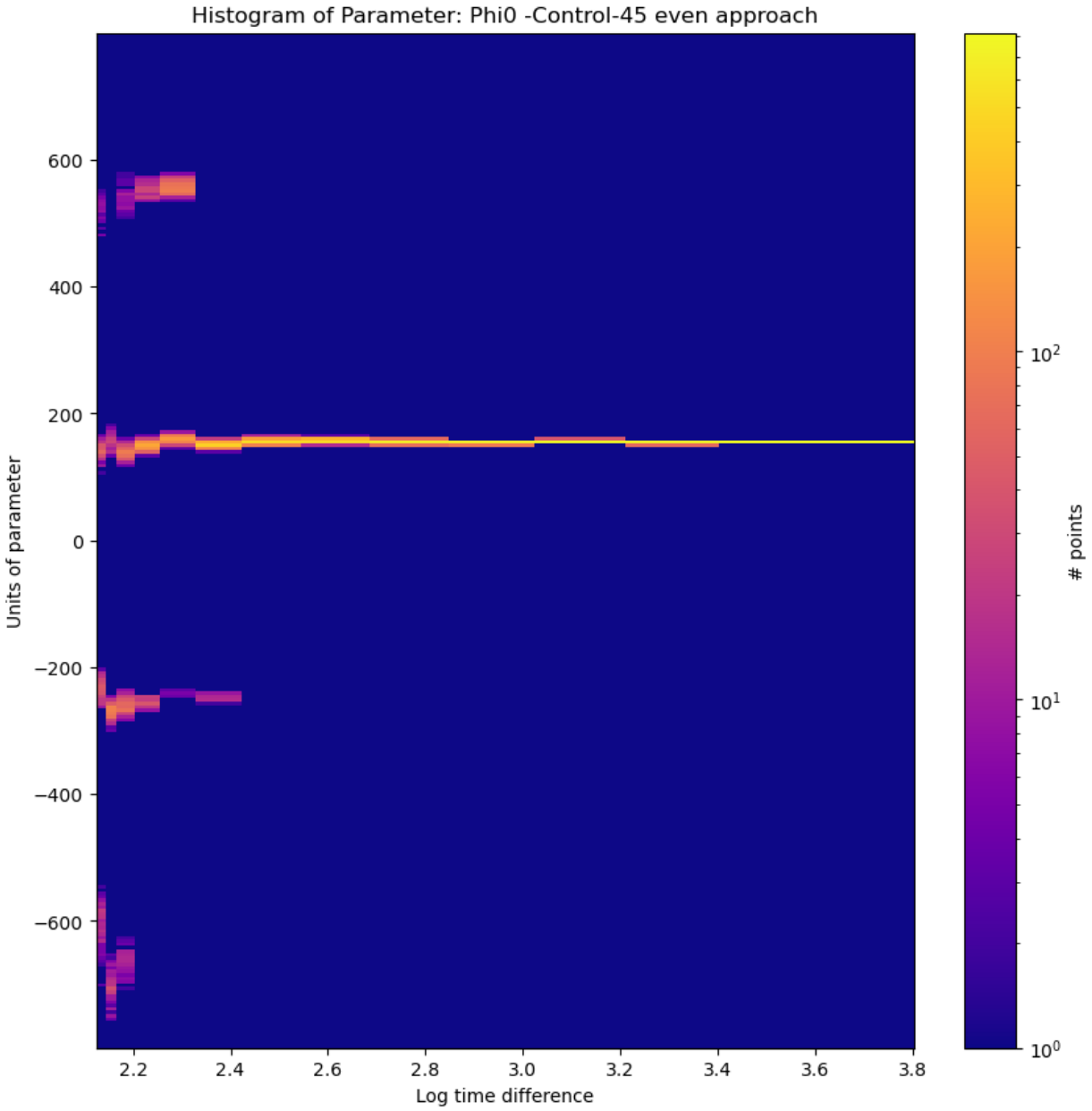

```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63925132 1.78039166
 1.94541665 2.14036554 2.32329382 2.52597764 2.6659401  2.84474297
 3.12212717 3.34835091 3.58641497]
```




```
my x edges
[0.44639502 0.95154499 1.31774187 1.50557368 1.63498984 1.81627773
 2.02180203 2.21024151 2.41133864 2.6672224  2.8538082  2.9992758
 3.29278834 3.63448291]
```



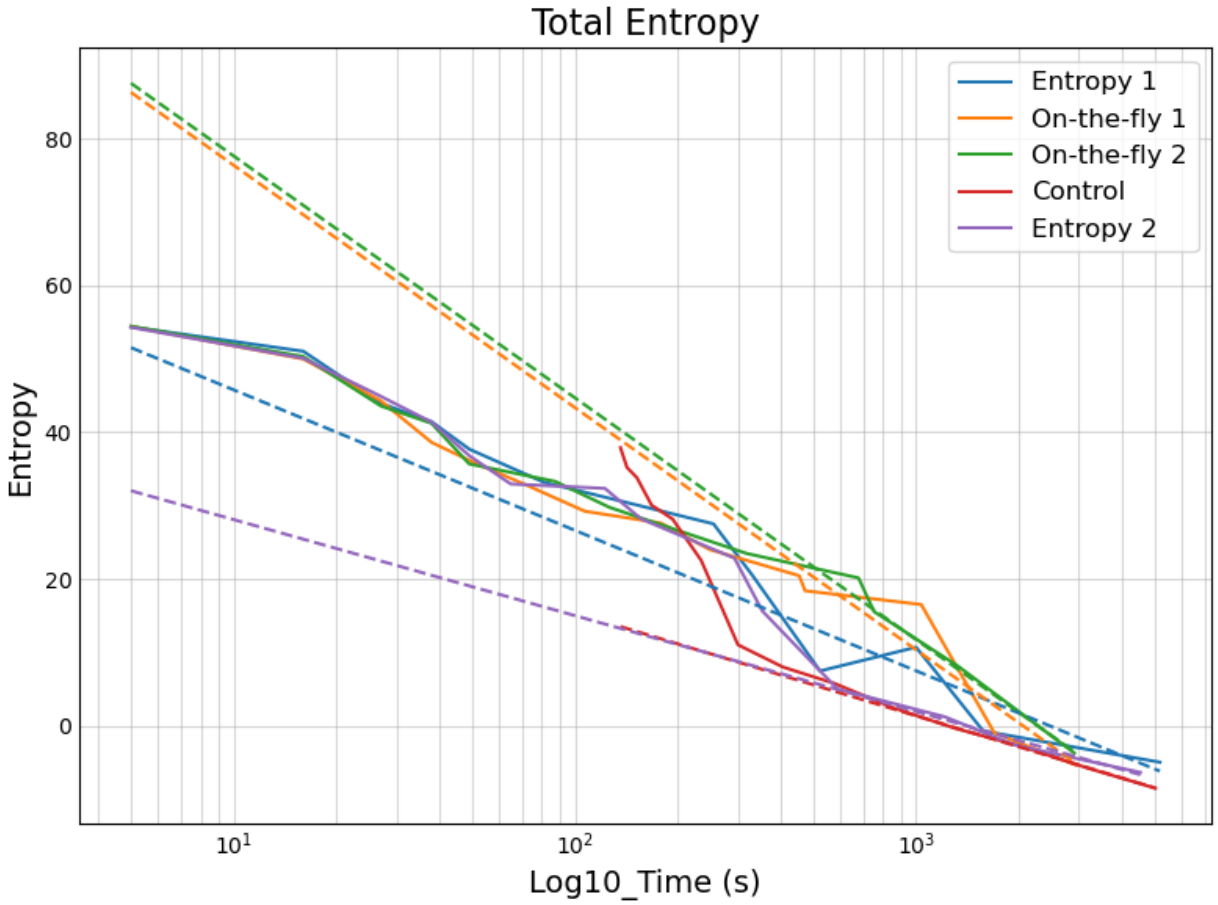

```
my x edges
[2.12400496 2.14307285 2.16713147 2.203179   2.25542118 2.32797081
 2.42377446 2.54362306 2.68585652 2.84696932 3.02270427 3.209021
 3.40261866 3.60105122 3.80139375]
```



10.3 Entropy V.S Time

In [13]:
```
#Notice we could also print the total entropy, instead, of the marginalized entropy.
#On-thefly always deals with the total entropy. You cannot choose a parameter of inter
times = [exp_entropy1.totaltimes(), exp_on_the_fly1.totaltimes(),
         exp_on_the_fly2.totaltimes(),exp_control.totaltimes(), exp_entropy2.totaltime

entropies = [exp_entropy1.entropy(), exp_on_the_fly1.entropy(),
             exp_on_the_fly2.entropy(), exp_control.entropy(), exp_entropy2.entropy()]

list_names = ["Entropy 1 ", "On-the-fly 1", "On-the-fly 2", "Control", "Entropy 2"]

MyPlots.plot_entropy_times(times, entropies,list_names, "Total Entropy")


#C1 blue entrop
#C2 yellow on the
#C3 green on the 2
```

```
#C4 red control
#C5 Purple entropy2

#Total Entropy
```



In [14]:
```
#DELETE LATER THIS WORK ONLY IF YOU HAVE A GMM

#Entropy 1 Comparisson
#Blue fast mvn
#Orange gmm


# MyPlots.plot_entropy_times([exp_entropy1.totaltimes(), exp_entropy1_gmm.totaltimes()
#                             [exp_entropy1.entropy(), exp_entropy1_gmm.entropy()])
# plt.savefig("entropy1 total entropy mvn_blue gmm_orange.png")
```

In [15]:
```
#DELETE LATER THIS WORK ONLY IF YOU HAVE A GMM

#Entropy 2 Comparisson
#Blue fast mvn
#Orange gmm

#COMMENT OUT IF YOU DO NOT HAVE A GMM VERSION

# MyPlots.plot_entropy_times([exp_entropy2.totaltimes(), exp_entropy2_gmm.totaltimes()
#                             [exp_entropy2.entropy(), exp_entropy2_gmm.entropy()])
```

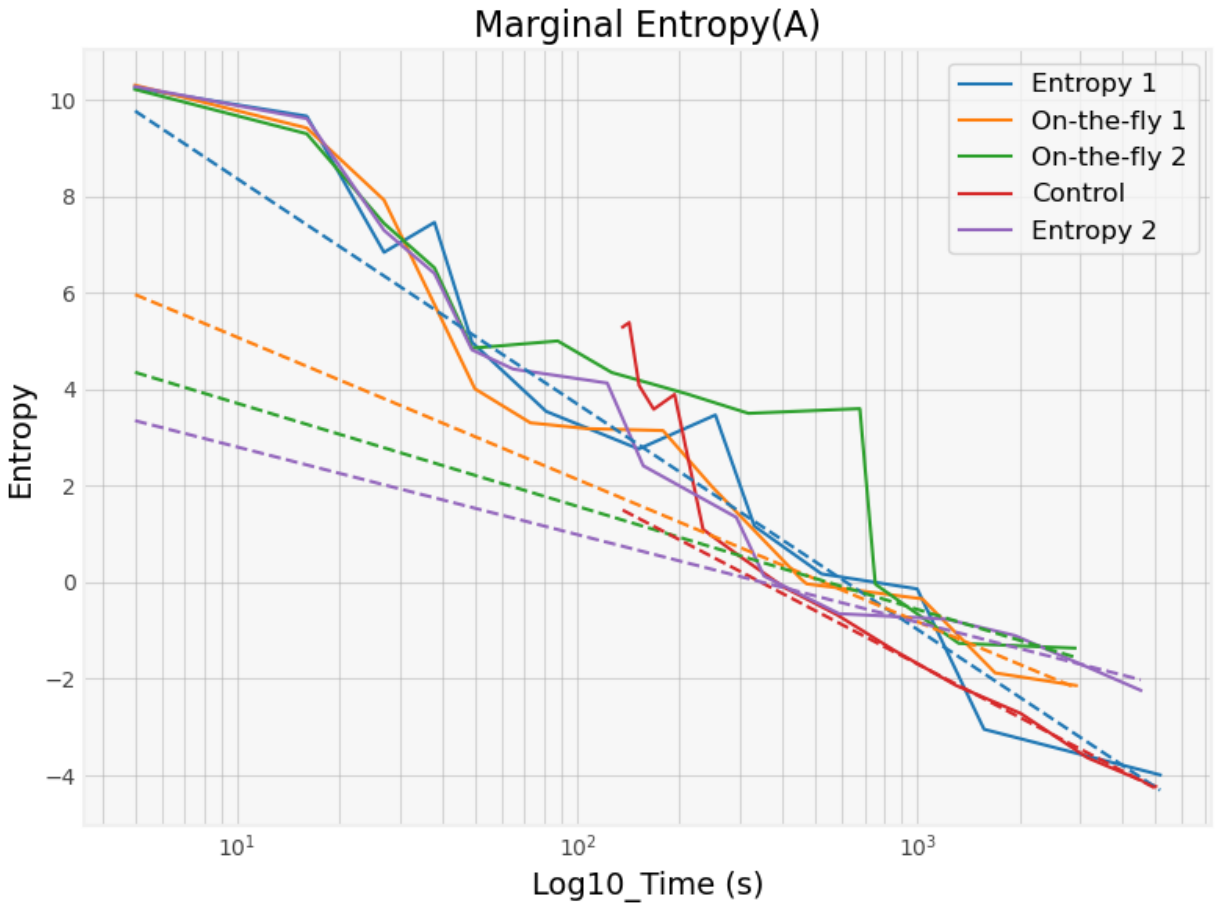
In [16]:
```
#On the fly Comparisson
#Blue fast mvn
#Orange gmm
```

```
# MyPlots.plot_entropy_times([exp_on_the_fly1.totaltimes(), exp_on_the_fly1_gmm.totalt
#                             [exp_on_the_fly1.entropy(), exp_on_the_fly1_gmm.entropy()]
```

In [17]:
```
#On the fly 2- Total Entropy Comparisson
#Blue fast mvn
#Orange gmm


# MyPlots.plot_entropy_times([exp_on_the_fly2.totaltimes(), exp_on_the_fly2_gmm.totalt
#                             [exp_on_the_fly2.entropy(), exp_on_the_fly2_gmm.entropy()]
```

In [ ]:
```
```

In [18]:
```
#Ask REMEMBER THE RETURN OF ENOTROPY_MARG() BC entropy2 does not have a marginal
times = [exp_entropy1.totaltimes(), exp_on_the_fly1.totaltimes(),
         exp_on_the_fly2.totaltimes(),exp_control.totaltimes(), exp_entropy2.totaltime


list_names = ["Entropy 1 ", "On-the-fly 1", "On-the-fly 2", "Control", "Entropy 2"]

#Rememeber that the marginal entropy for entropy2 apporach is the same as total entrop
#is not selecting any parameters. Thus, we calculate here using the select of any of t
#approaches that it must be same.
entro = [calc_entropy(this_pts, select_pars=exp_on_the_fly2.settings.sel,
                options=exp_entropy2.settings.entropy_options)[0] for this_pts in exp_

entropies = [exp_entropy1.entropy_marg(), exp_on_the_fly1.entropy_marg(),
             exp_on_the_fly2.entropy_marg(), exp_control.entropy_marg(), entro]
MyPlots.plot_entropy_times(times, entropies, list_names,"Marginal Entropy(A)")
```



In [19]:
```
#DELETE LATER THIS WORK ONLY IF YOU HAVE A GMM

#Entropy 1 Marginal  Comparisson
#Blue fast mvn
#Orange gmm


# MyPlots.plot_entropy_times([exp_entropy1.totaltimes(), exp_entropy1_gmm.totaltimes()
#                             [exp_entropy1.entropy_marg(), exp_entropy1_gmm.entropy_mar
```

In [20]:
```
#Entropy 2- Marginal  Comparisson

# ###################Comment out unless you have a GMM version
# gmm_setting= {'method': 'gmm', 'n_components': None}
# entro1 = recalculating_entropy(exp_entropy2, exp_on_the_fly2.settings.sel,gmm_settir
# entro2 = recalculating_entropy(exp_entropy2_gmm, exp_on_the_fly2.settings.sel,gmm_se
# #Blue fast mvn
# #Orange gmm
# MyPlots.plot_entropy_times([exp_entropy2.totaltimes(), exp_entropy2_gmm.totaltimes()
#                             [entro1, entro2])
```

In [21]:
```
#DELETE LATER THIS WORK ONLY IF YOU HAVE A GMM

#On the fly 1 Marginal  Comparisson
#Blue fast mvn
#Orange gmm

#Comment out if you do not have GMM version
```

```
# MyPlots.plot_entropy_times([exp_on_the_fly1.totaltimes(), exp_on_the_fly1_gmm.totalt
#                           [exp_on_the_fly1.entropy_marg(), exp_on_the_fly1_gmm.entro
```

In [22]:
```
#DELETE LATER THIS WORK ONLY IF YOU HAVE A GMM

#On the fly 1 Marginal  Comparisson
#Blue fast mvn
#Orange gmm


# MyPlots.plot_entropy_times([exp_on_the_fly2.totaltimes(), exp_on_the_fly2_gmm.totalt
#                           [exp_on_the_fly2.entropy_marg(), exp_on_the_fly2_gmm.entro
```

Helper Functions for Estimator of PDF and Log-likelihoods

These function wll be used in the remaining sections

"""

estimator_avg_at

Note: In my opinion, the estimator_avg_at causes an error in some cases bc at some points of the iteration the values are concentrate around the ground truth. Thus, there are no many points to take on the left or right of the interval. So the checking below alert us!

```
if (left_value_index < 0) or (right_value_index > (NUMBER_SAMPLES -
1)): Alert!
```

""" Note: Notice compute_likeloohs assumes independence of the parameters (columns). It is using David's estimator for pdf. Ask But can we assume that?

""" Notes likelihood_helper_kde:

1. Using KDE to estimate the pdf. Notice that KDE is a non-parametric estimator. However, it assumes independent samples; this could be a problem. Remember each sample is measurememnt point in the main loop is chosen based in the previous results.

2. Ask about the bandwidth for this function. In the next functions, we use a kde in a single feature (a parameter or a y at a given measurement place); so we can use the std as the bandwidth. We use the std bc all the other methods s.a. silverman or scott gave us terrible estimators. Thus, it may be the case that they are terrible estimators since we are using silverman. In this case, I am using "silverman". Since it is a multidimesonal pdf I cannot use just the std. Should I tried to use something smilar to the std?
   """ #likelihood_helper_kde

10.4 Estimator of Log Likelihoods for the Y's

We will use David's estimator and a multidimensional Kde

Using David's Estimator

In [23]:
```
#all_total_likelihoods
#At each iteration, we have a y-profiles. We compute the likelihhood at each x and ado
Sum_likelihoods_at_each_iter_entropy1_for_ys = MyPlots.compute_likelihoods(exp_entropy
                                                                           exp_entropy2
Sum_likelihoods_at_each_iter_entropy2_for_ys = MyPlots.compute_likelihoods(exp_entropy
                                                                           exp_entropy2
Sum_likelihoods_at_each_iter_on_the_fly1_for_ys = MyPlots.compute_likelihoods(exp_on_t
                                                                              exp_on_th

Sum_likelihoods_at_each_iter_on_the_fly2_for_ys = MyPlots.compute_likelihoods(exp_on_t
                                                                              exp_on_th

Sum_likelihoods_at_each_iter_control_for_ys = MyPlots.compute_likelihoods(exp_control.
                                                                          exp_contr
#Sum_likelihoods_at_each_iter_control_for_ys = compute_likelihoods(total_yprofs_contro
```

```
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
```

```
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
818
this is rigth index
819
```

```
C:\Users\rober\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3432: RuntimeWar
ning: Mean of empty slice.
  return _methods._mean(a, axis=axis, dtype=dtype,
C:\Users\rober\anaconda3\lib\site-packages\numpy\core\_methods.py:190: RuntimeWarnin
g: invalid value encountered in double_scalars
  ret = ret.dtype.type(ret / rcount)
```

```
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
```

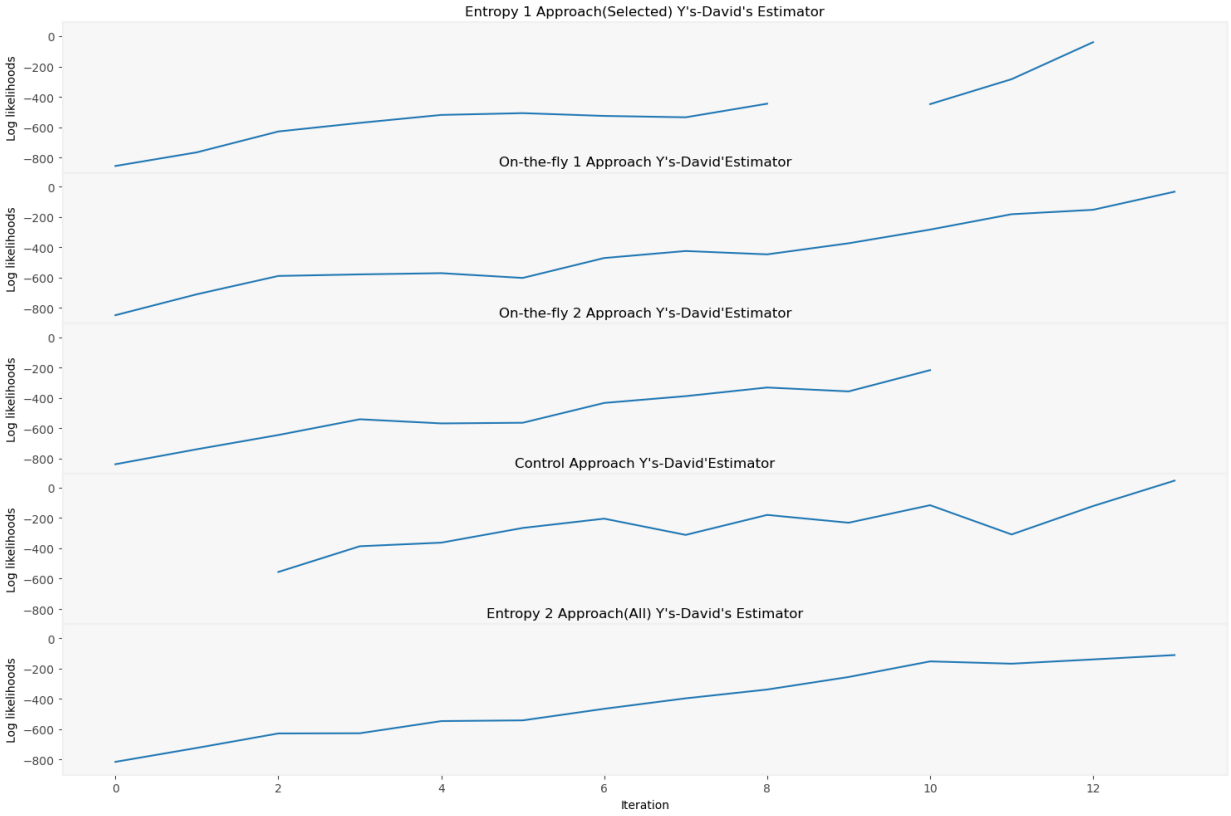
```
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
0
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
```

```
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
839
this is rigth index
840
```

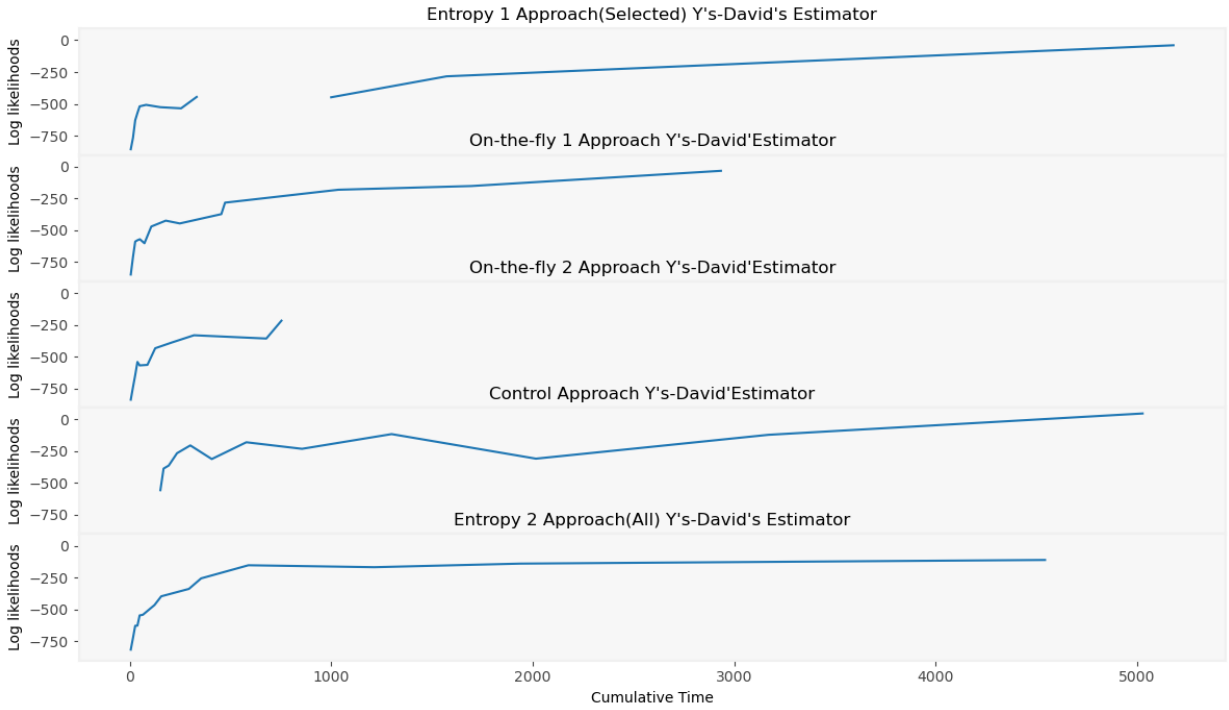
In [24]:

```
MyPlots.plot_likelihood_iterations(5,1, [Sum_likelihoods_at_each_iter_entropy1_for_ys,
                                         Sum_likelihoods_at_each_iter_on_the_fly1_for_
                                         Sum_likelihoods_at_each_iter_on_the_fly2_for_
                                         Sum_likelihoods_at_each_iter_control_for_ys,
                                         Sum_likelihoods_at_each_iter_entropy2_for_ys
```

```
                          ,["Entropy 1 Approach(Selected) Y's-David's Estimator",
                            "On-the-fly 1 Approach Y's-David'Estimator",
                            "On-the-fly 2 Approach Y's-David'Estimator",
                            "Control Approach Y's-David'Estimator",
                           "Entropy 2 Approach(All) Y's-David's Estimator"])

#REMEMBER: CONTROL TAKES A DIFFERENT NUMBER OF POINTS THAN THE OTHER APPORACHES.
```



In [25]:
```
#Add later the other apporaches to this plot
#[exp_entropy.totaltimes()[1:], exp_on_the_fly.totaltimes()[1:], exp_control.totaltime
MyPlots.plot_likelihood_time(5,1, [Sum_likelihoods_at_each_iter_entropy1_for_ys,
                                   Sum_likelihoods_at_each_iter_on_the_fly1_for_
                                   Sum_likelihoods_at_each_iter_on_the_fly2_for_
                                   Sum_likelihoods_at_each_iter_control_for_ys,
                                   Sum_likelihoods_at_each_iter_entropy2_for_ys]
                          ,[exp_entropy1.totaltimes(), exp_on_the_fly1.totaltimes(),
                            exp_on_the_fly2.totaltimes(), exp_control.totaltimes(),
                            exp_entropy2.totaltimes()],
                            ["Entropy 1 Approach(Selected) Y's-David's Estimator",
                            "On-the-fly 1 Approach Y's-David'Estimator",
                            "On-the-fly 2 Approach Y's-David'Estimator",
                            "Control Approach Y's-David'Estimator",
                           "Entropy 2 Approach(All) Y's-David's Estimator"])
```



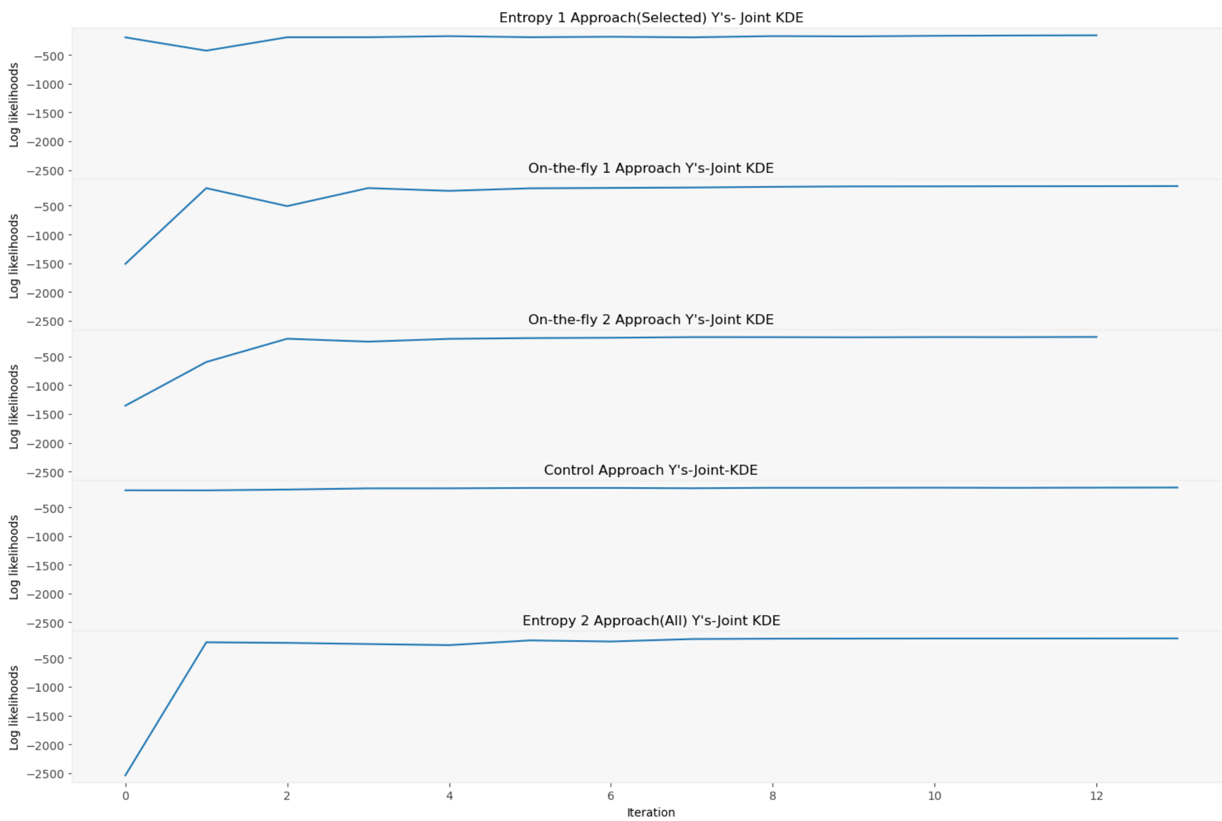
Using Multi-Dimensional KDE

In [26]:
```
#all_total_likelihoods
Sum_likelihoods_at_each_iter_entropy1_for_ys_kde = MyPlots.likelihood_helper_kde(exp_e
                                                                                 exp_er

Sum_likelihoods_at_each_iter_entropy2_for_ys_kde = MyPlots.likelihood_helper_kde(exp_e
                                                                                 exp_er
Sum_likelihoods_at_each_iter_on_the_fly1_for_ys_kde = MyPlots.likelihood_helper_kde(ex
                                                                                   exp
Sum_likelihoods_at_each_iter_on_the_fly2_for_ys_kde = MyPlots.likelihood_helper_kde(ex
                                                                                   exp
Sum_likelihoods_at_each_iter_control_for_ys_kde = MyPlots.likelihood_helper_kde(exp_co
                                                                                   exp
#Sum_likelihoods_at_each_iter_control_for_ys_kde = likelihood_helper_kde(total_yprofs_
```

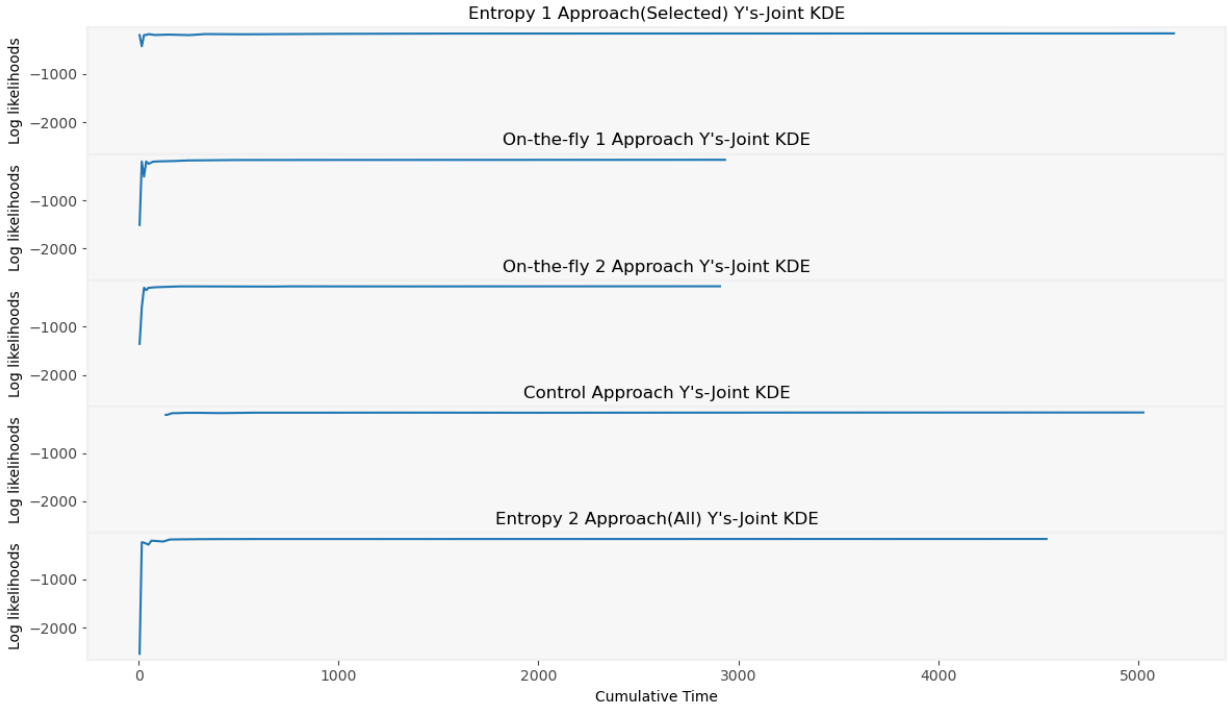
```
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:346: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_yprof)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:346: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_yprof)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:346: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_yprof)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:346: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_yprof)
```

In [27]:

```
#Add later the other apporaches to this plot
MyPlots.plot_likelihood_iterations(5,1
                         ,[Sum_likelihoods_at_each_iter_entropy1_for_ys_kde,
                           Sum_likelihoods_at_each_iter_on_the_fly1_for_ys_kde,
                          Sum_likelihoods_at_each_iter_on_the_fly2_for_ys_kde,
                          Sum_likelihoods_at_each_iter_control_for_ys_kde,
                          Sum_likelihoods_at_each_iter_entropy2_for_ys_kde]
                         , ["Entropy 1 Approach(Selected) Y's- Joint KDE",
                            "On-the-fly 1 Approach Y's-Joint KDE",
                            "On-the-fly 2 Approach Y's-Joint KDE",
                            "Control Approach Y's-Joint-KDE",
                           "Entropy 2 Approach(All) Y's-Joint KDE"])
```



In [28]:
```
MyPlots.plot_likelihood_time(5,1
                        ,[Sum_likelihoods_at_each_iter_entropy1_for_ys_kde,
                          Sum_likelihoods_at_each_iter_on_the_fly1_for_ys_kde,
                         Sum_likelihoods_at_each_iter_on_the_fly2_for_ys_kde,
                         Sum_likelihoods_at_each_iter_control_for_ys_kde,
                         Sum_likelihoods_at_each_iter_entropy2_for_ys_kde
                         ],
                         [ exp_entropy1.totaltimes(),
                          exp_on_the_fly1.totaltimes(),
                          exp_on_the_fly2.totaltimes(),
                          exp_control.totaltimes(),
                          exp_entropy2.totaltimes()
                         ],

                        ["Entropy 1 Approach(Selected) Y's-Joint KDE",
                            "On-the-fly 1 Approach Y's-Joint KDE",
                            "On-the-fly 2 Approach Y's-Joint KDE",
                            "Control Approach Y's-Joint KDE",
                           "Entropy 2 Approach(All) Y's-Joint KDE"])
```



8.5 Estimator of log-likelihood for the Parameters

The log likelihhods estimator will be computed with two fitting: The loglikelihhods are computed for the ground truth values.

We compute the loglikelihoods in two ways:

-David's estimator for pdf: by default this estimator calculate the pdf for each parameter separatly. It is equivalent to the second fitting in Kde

-Kde estimator for pdf: For the kde we use two apporaches The first fitting will use all the samples for all the parameters at a given iteration

The second fitting will use only the data for one parameter to estimate a pdf of each parameter at a given iteration

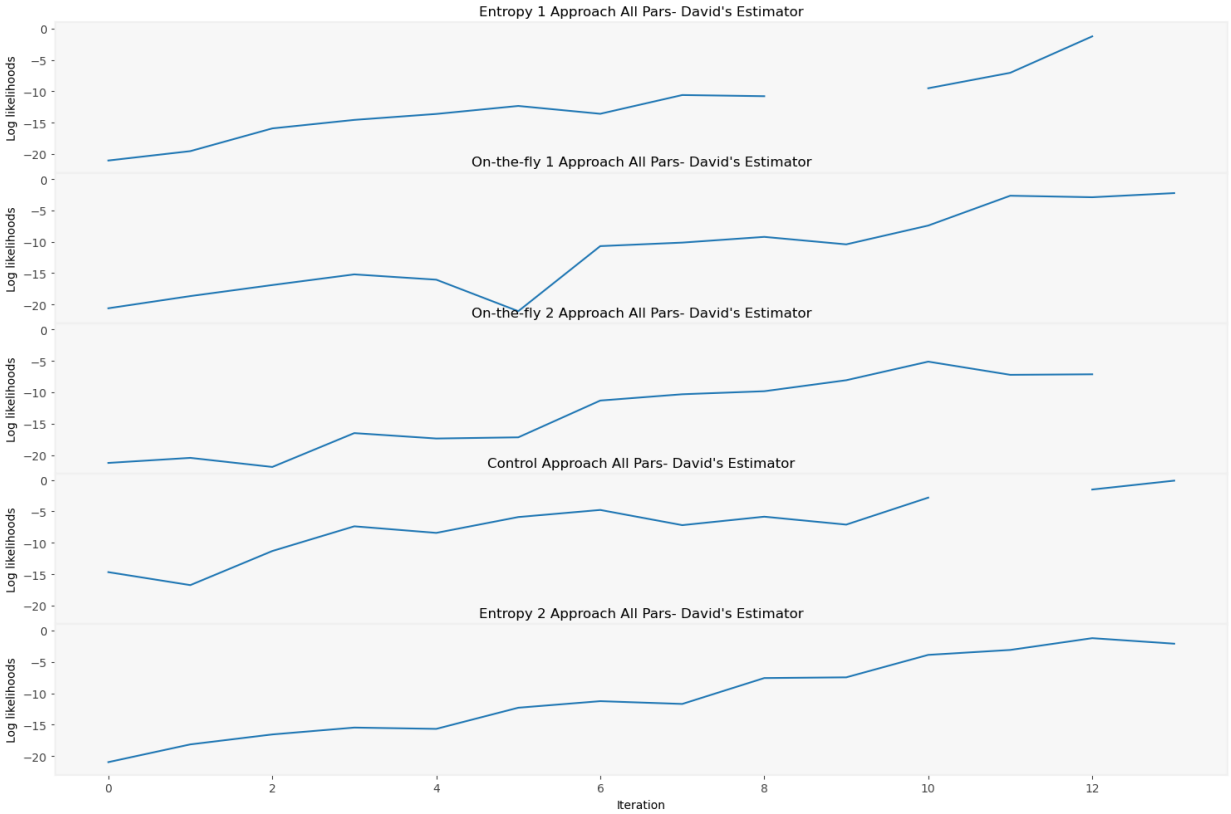
David's Estimator for Parameters

In [29]:
```
#all_total_likelihoods
Sum_likelihoods_at_each_iter_entropy1_for_pars = MyPlots.compute_likelihoods(exp_entro
                                                                             exp_entrop
                                                                             3)
Sum_likelihoods_at_each_iter_entropy2_for_pars = MyPlots.compute_likelihoods(exp_entro
                                                                             exp_entrop
                                                                             3)

Sum_likelihoods_at_each_iter_on_the_fly1_for_pars = MyPlots.compute_likelihoods(exp_or
                                                                                exp_on_
                                                                                 3)

Sum_likelihoods_at_each_iter_on_the_fly2_for_pars = MyPlots.compute_likelihoods(exp_or
                                                                                exp_on_
                                                                                 3)

Sum_likelihoods_at_each_iter_control_for_pars = MyPlots.compute_likelihoods(exp_contro
                                                                            exp_cor
                                                                             3)
#Sum_likelihoods_at_each_iter_control_for_pars = compute_likelihoods(total_pts_control
```

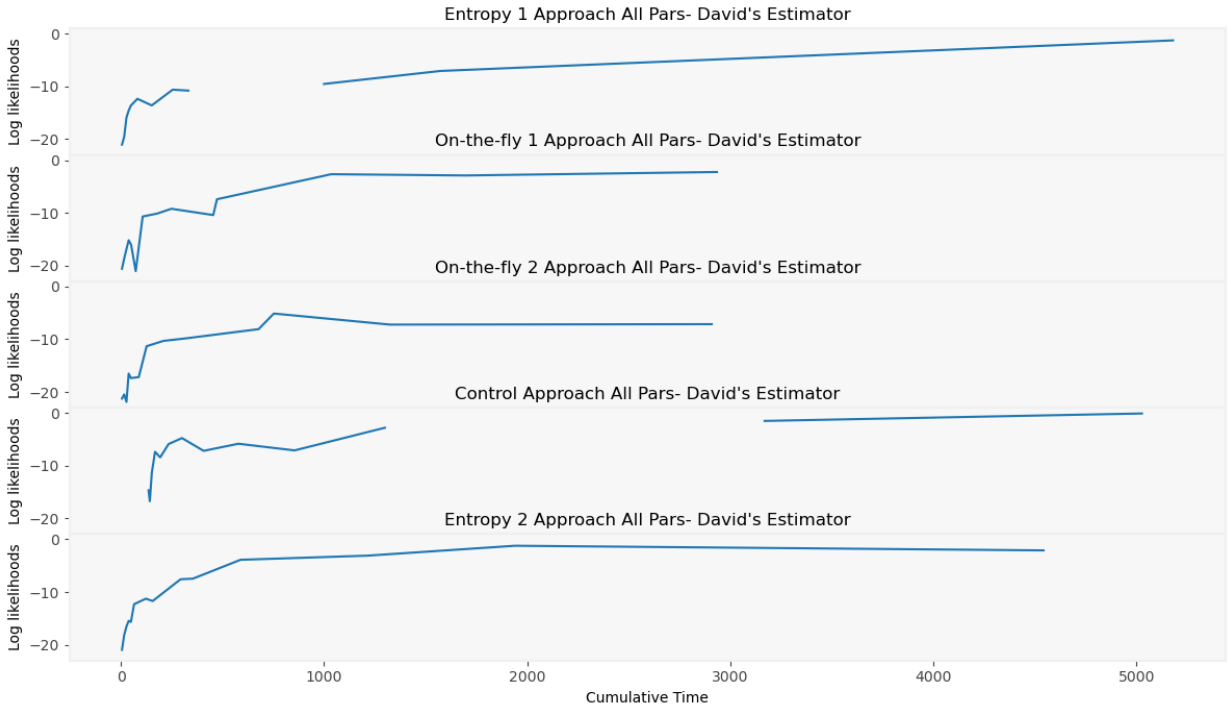
```
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-3
this is rigth index
2
ERROR: the ground truth to estimate is too close to one extreme. So, we do not have e
nought samples eitheron the left or right of the ground truth
This is the left index
-1
this is rigth index
4
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
C:\Users\rober\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3432: RuntimeWar
ning: Mean of empty slice.
  return _methods._mean(a, axis=axis, dtype=dtype,
C:\Users\rober\anaconda3\lib\site-packages\numpy\core\_methods.py:190: RuntimeWarnin
g: invalid value encountered in double_scalars
  ret = ret.dtype.type(ret / rcount)
```

In [30]:
```
MyPlots.plot_likelihood_iterations(5,1,
                           [Sum_likelihoods_at_each_iter_entropy1_for_pars,
                            Sum_likelihoods_at_each_iter_on_the_fly1_for_pars,
                            Sum_likelihoods_at_each_iter_on_the_fly2_for_pars,
                            Sum_likelihoods_at_each_iter_control_for_pars,
                            Sum_likelihoods_at_each_iter_entropy2_for_pars,
                           ],
                          ["Entropy 1 Approach All Pars- David's Estimator",
```

```
                          "On-the-fly 1 Approach All Pars- David's Estimator",
                          "On-the-fly 2 Approach All Pars- David's Estimator",
                          "Control Approach All Pars- David's Estimator",
                          "Entropy 2 Approach All Pars- David's Estimator"
                         ])
#help(plot_likelihood_iterations)
```



In [31]:
```
MyPlots.plot_likelihood_time(5,1,
                          [Sum_likelihoods_at_each_iter_entropy1_for_pars,
                           Sum_likelihoods_at_each_iter_on_the_fly1_for_pars,
                           Sum_likelihoods_at_each_iter_on_the_fly2_for_pars,
                           Sum_likelihoods_at_each_iter_control_for_pars,
                           Sum_likelihoods_at_each_iter_entropy2_for_pars
                          ],
                        [ exp_entropy1.totaltimes(),
                          exp_on_the_fly1.totaltimes(),
                          exp_on_the_fly2.totaltimes(),
                          exp_control.totaltimes(),
                         exp_entropy2.totaltimes()
                         ],
                          ["Entropy 1 Approach All Pars- David's Estimator",
                           "On-the-fly 1 Approach All Pars- David's Estimator",
                           "On-the-fly 2 Approach All Pars- David's Estimator",
                           "Control Approach All Pars- David's Estimator",
                           "Entropy 2 Approach All Pars- David's Estimator"
                          ])
```



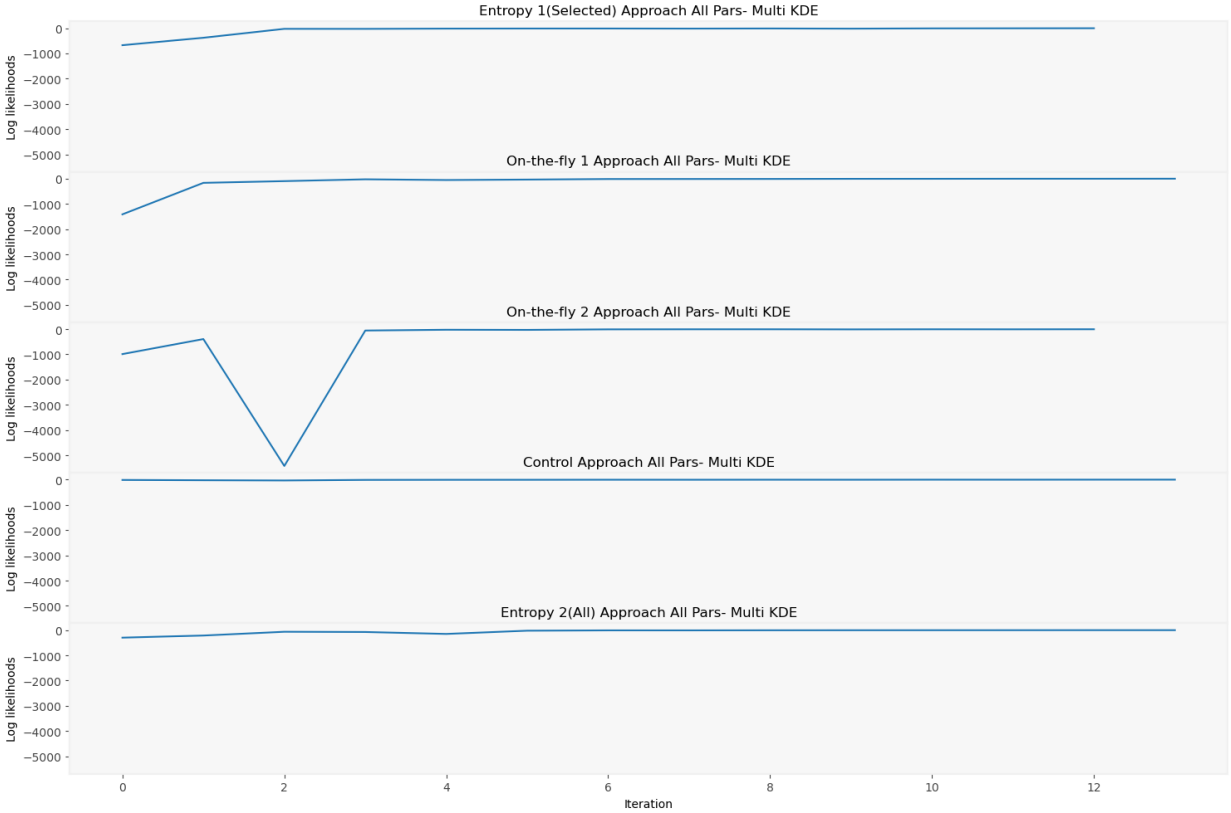
Multidimentional KDE Estimator for Parameters

Combined Parameters

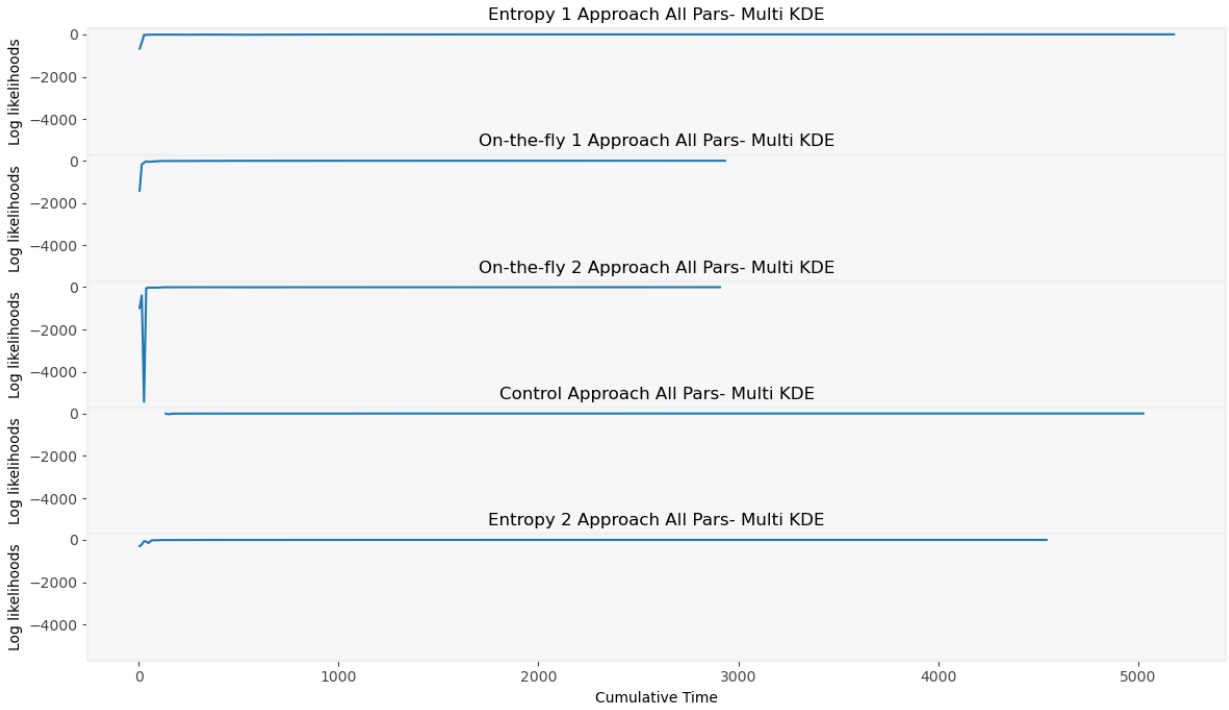
In [32]:
```
#all_total_likelihoods
Sum_likelihoods_at_each_iter_entropy1_for_pars_kde = MyPlots.likelihood_helper_kde(exp
                                                                                   exp_entropy1.get_t
Sum_likelihoods_at_each_iter_entropy2_for_pars_kde = MyPlots.likelihood_helper_kde(exp
                                                                                   exp_

Sum_likelihoods_at_each_iter_on_the_fly1_for_pars_kde = MyPlots.likelihood_helper_kde(
                                                                                   exp_on_the

Sum_likelihoods_at_each_iter_on_the_fly2_for_pars_kde = MyPlots.likelihood_helper_kde(
                                                                                   exp_on_the

Sum_likelihoods_at_each_iter_control_for_pars_kde = MyPlots.likelihood_helper_kde(exp_
                                                                                   exp_on_the
```

```
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
```

In [33]:
```
MyPlots.plot_likelihood_iterations(5,1,
                          [Sum_likelihoods_at_each_iter_entropy1_for_pars_kde,
                           Sum_likelihoods_at_each_iter_on_the_fly1_for_pars_kde,
                           Sum_likelihoods_at_each_iter_on_the_fly2_for_pars_kde,
                           Sum_likelihoods_at_each_iter_control_for_pars_kde,
                           Sum_likelihoods_at_each_iter_entropy2_for_pars_kde
                          ]
                        ,["Entropy 1(Selected) Approach All Pars- Multi KDE",
                          "On-the-fly 1 Approach All Pars- Multi KDE",
                          "On-the-fly 2 Approach All Pars- Multi KDE",
                          "Control Approach All Pars- Multi KDE",
                          "Entropy 2(All) Approach All Pars- Multi KDE"
                         ])
```



In [34]:
```
MyPlots.plot_likelihood_time(5,1,
                             [Sum_likelihoods_at_each_iter_entropy1_for_pars_kde,
                              Sum_likelihoods_at_each_iter_on_the_fly1_for_pars_kde,
                              Sum_likelihoods_at_each_iter_on_the_fly2_for_pars_kde,
                              Sum_likelihoods_at_each_iter_control_for_pars_kde,
                              Sum_likelihoods_at_each_iter_entropy2_for_pars_kde
                             ],
                            [exp_entropy1.totaltimes(),
                             exp_on_the_fly1.totaltimes(),
                             exp_on_the_fly2.totaltimes(),
                             exp_control.totaltimes(),
                             exp_entropy2.totaltimes()
                            ],
                           ["Entropy 1 Approach All Pars- Multi KDE",
                            "On-the-fly 1 Approach All Pars- Multi KDE",
                            "On-the-fly 2 Approach All Pars- Multi KDE",
                            "Control Approach All Pars- Multi KDE",
                            "Entropy 2 Approach All Pars- Multi KDE"
                           ])
```



Separated for each parameter. We will use the kde in the data for each parameter to build independent pdf's for each parameter

In [35]:
```
#Example with entropy approach
kdes_entropy1, log_pars_entropy1 = MyPlots.kdes_and_loglikelihoods_for_pars(exp_entrop
                                                                            exp_entropy1.get_tru

kdes_entropy2, log_pars_entropy2 = MyPlots.kdes_and_loglikelihoods_for_pars(exp_entrop
                                                                            exp_entropy2.get_tru
#Example with on-the-fly apporach
kdes_on_the_fly1, log_pars_on_the_fly1 = MyPlots.kdes_and_loglikelihoods_for_pars(exp_
                                                                                exp_on_the_fly1.
kdes_on_the_fly2, log_pars_on_the_fly2 = MyPlots.kdes_and_loglikelihoods_for_pars(exp_
                                                                                exp_on_the_fly2.

kdes_control, log_pars_control = MyPlots.kdes_and_loglikelihoods_for_pars(exp_control.
                                                                          exp_control.get_true

###############################################################################
#Comment out the section below. It makes sense only if you have a GMM version
###############################################################################

# #Example with entropy approach
# kdes_entropy1_gmm, log_pars_entropy1_gmm = MyPlots.kdes_and_loglikelihoods_for_pars(
#                                                                               exp_entropy1_gmm.g

# kdes_entropy2_gmm, log_pars_entropy2_gmm = MyPlots.kdes_and_loglikelihoods_for_pars(
#                                                                               exp_entropy2_gmm.g
# #Example with on-the-fly apporach
# kdes_on_the_fly1_gmm, log_pars_on_the_fly1_gmm = MyPlots.kdes_and_loglikelihoods_for
#                                                                                   exp_on_the_fly
# kdes_on_the_fly2_gmm, log_pars_on_the_fly2_gmm = MyPlots.kdes_and_loglikelihoods_for
#                                                                               exp_on_the_fly

# kdes_control_gmm, log_pars_control_gmm = MyPlots.kdes_and_loglikelihoods_for_pars(ex
#                                                                             exp_control_gmm.ge
```

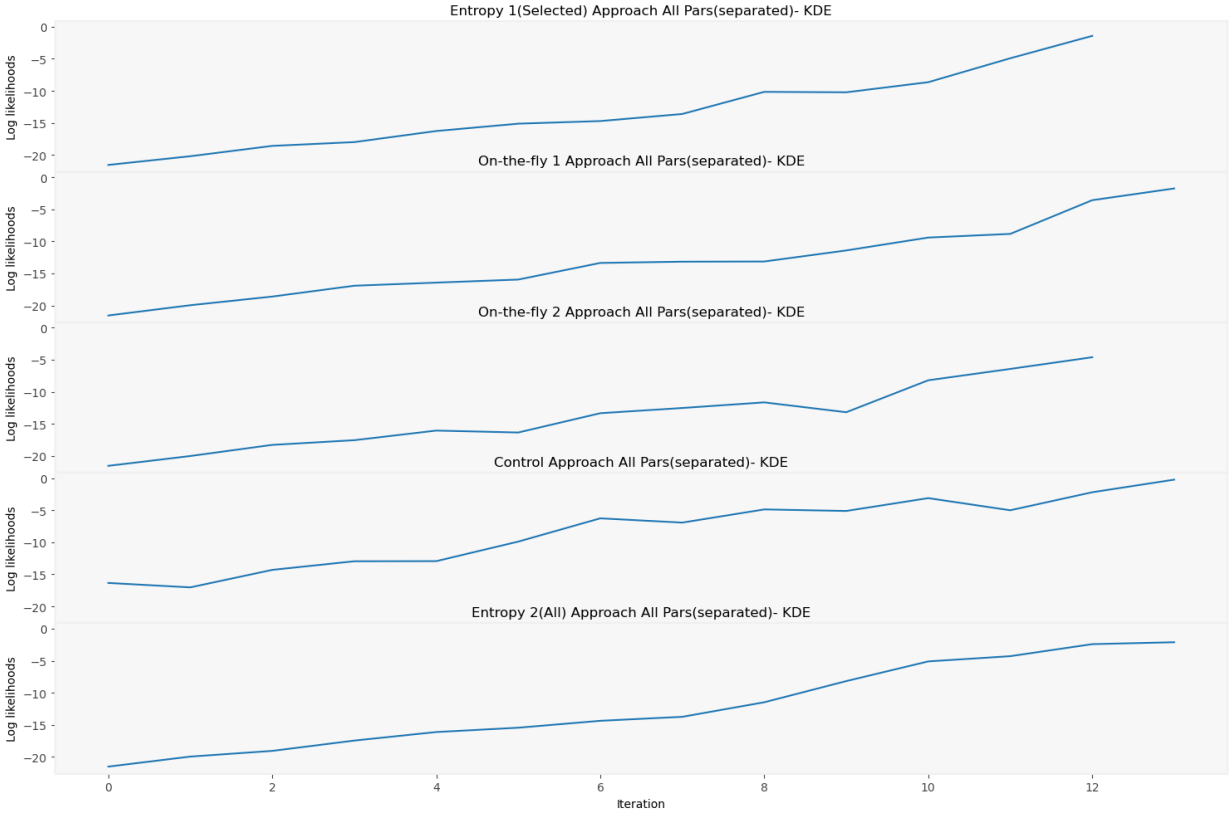
```
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
C:\Users\rober\FINAL NIST PROJECT\datastruct.py:353: VisibleDeprecationWarning: Creat
ing an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tup
les-or ndarrays with different lengths or shapes) is deprecated. If you meant to do t
his, you must specify 'dtype=object' when creating the ndarray.
  return np.array(all_pts)
```

In [36]:

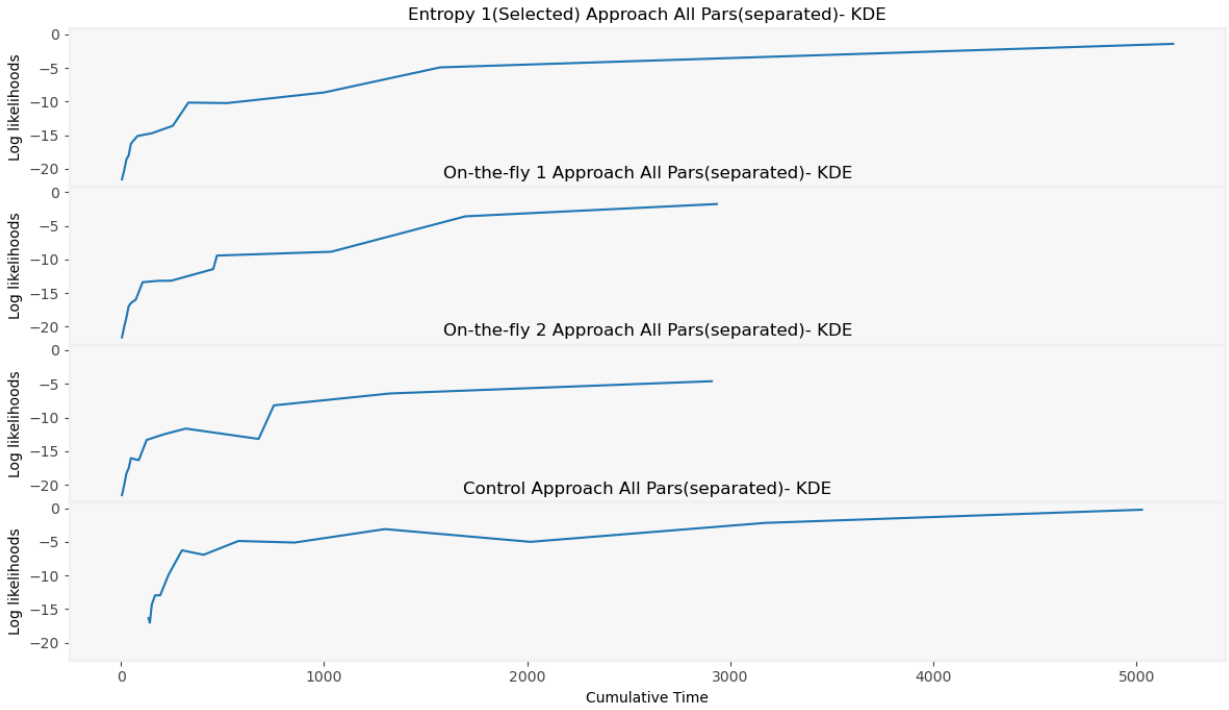
```
#Adding the loglikelihhods of the parameters at a given iteration
log_sums_kde_separated_iters_entropy1 = np.sum(log_pars_entropy1,axis= 1)
log_sums_kde_separated_iters_entropy2 = np.sum(log_pars_entropy2,axis= 1)
log_sums_kde_sepataed_iters_on_the_fly1 = np.sum(log_pars_on_the_fly1,axis= 1)
log_sums_kde_sepataed_iters_on_the_fly2 = np.sum(log_pars_on_the_fly2,axis= 1)
log_sums_kde_separated_iters_control = np.sum(log_pars_control,axis= 1)
```

In [37]:

```
MyPlots.plot_likelihood_iterations(5,1,
                          [log_sums_kde_separated_iters_entropy1,
                           log_sums_kde_sepataed_iters_on_the_fly1,
                          log_sums_kde_sepataed_iters_on_the_fly2 ,
                          log_sums_kde_separated_iters_control,
                          log_sums_kde_separated_iters_entropy2],
                          ["Entropy 1(Selected) Approach All Pars(separated)- KDE",
                           "On-the-fly 1 Approach All Pars(separated)- KDE",
                          "On-the-fly 2 Approach All Pars(separated)- KDE",
                          "Control Approach All Pars(separated)- KDE",
                          "Entropy 2(All) Approach All Pars(separated)- KDE"])
```



In [38]:
```
MyPlots.plot_likelihood_time(4,1,
                             [log_sums_kde_separated_iters_entropy1,
                              log_sums_kde_sepataed_iters_on_the_fly1,
                             log_sums_kde_sepataed_iters_on_the_fly2,
                             log_sums_kde_separated_iters_control,
                             log_sums_kde_separated_iters_entropy2],
                              [exp_entropy1.totaltimes(),
                               exp_on_the_fly1.totaltimes(),
                              exp_on_the_fly2.totaltimes(),
                              exp_control.totaltimes(),
                              exp_entropy2.totaltimes()],
                             ["Entropy 1(Selected) Approach All Pars(separated)- KDE",
                              "On-the-fly 1 Approach All Pars(separated)- KDE",
                              "On-the-fly 2 Approach All Pars(separated)- KDE",
                              "Control Approach All Pars(separated)- KDE",
                              "Entropy 2(All) Approach All Pars(separated)- KDE"
                             ])
```

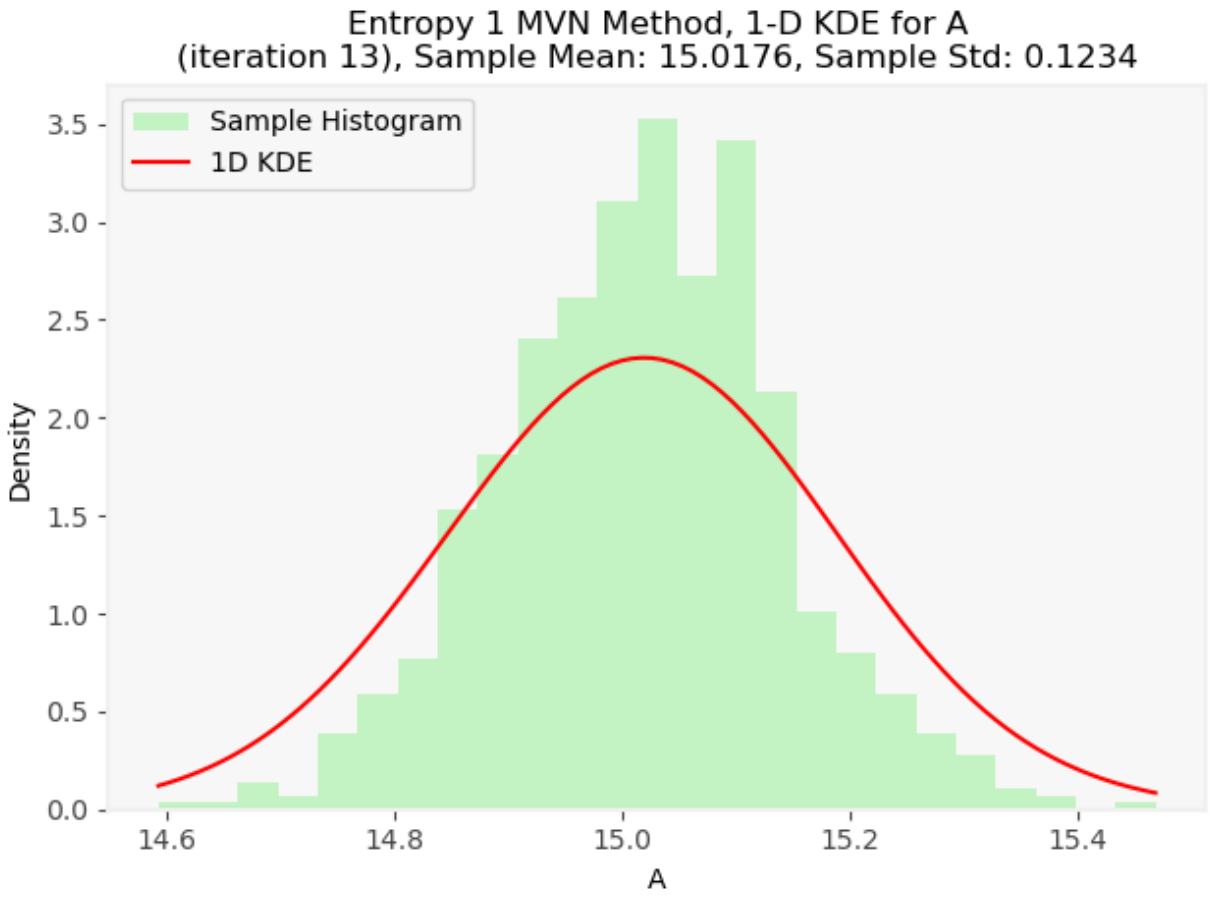


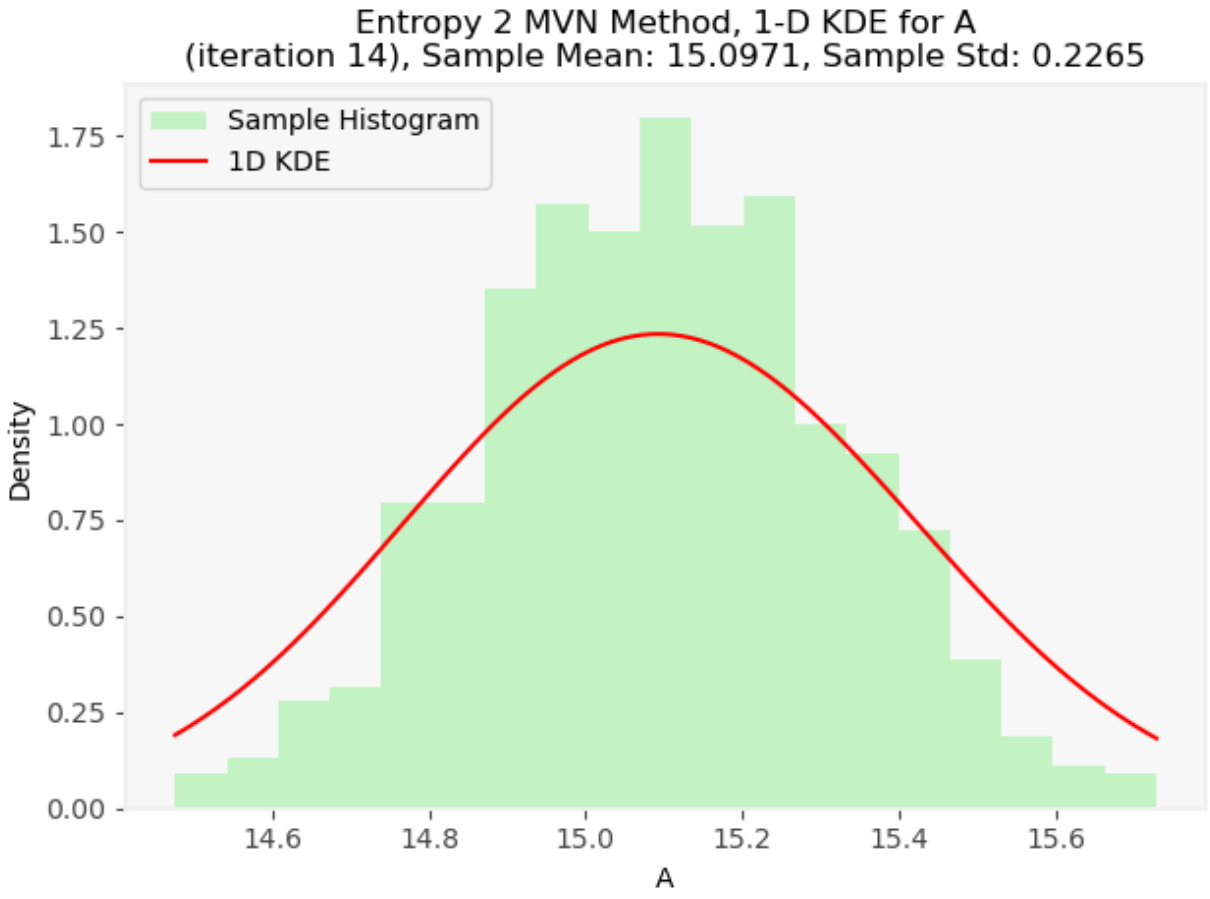
Histogram for Paramter Samples

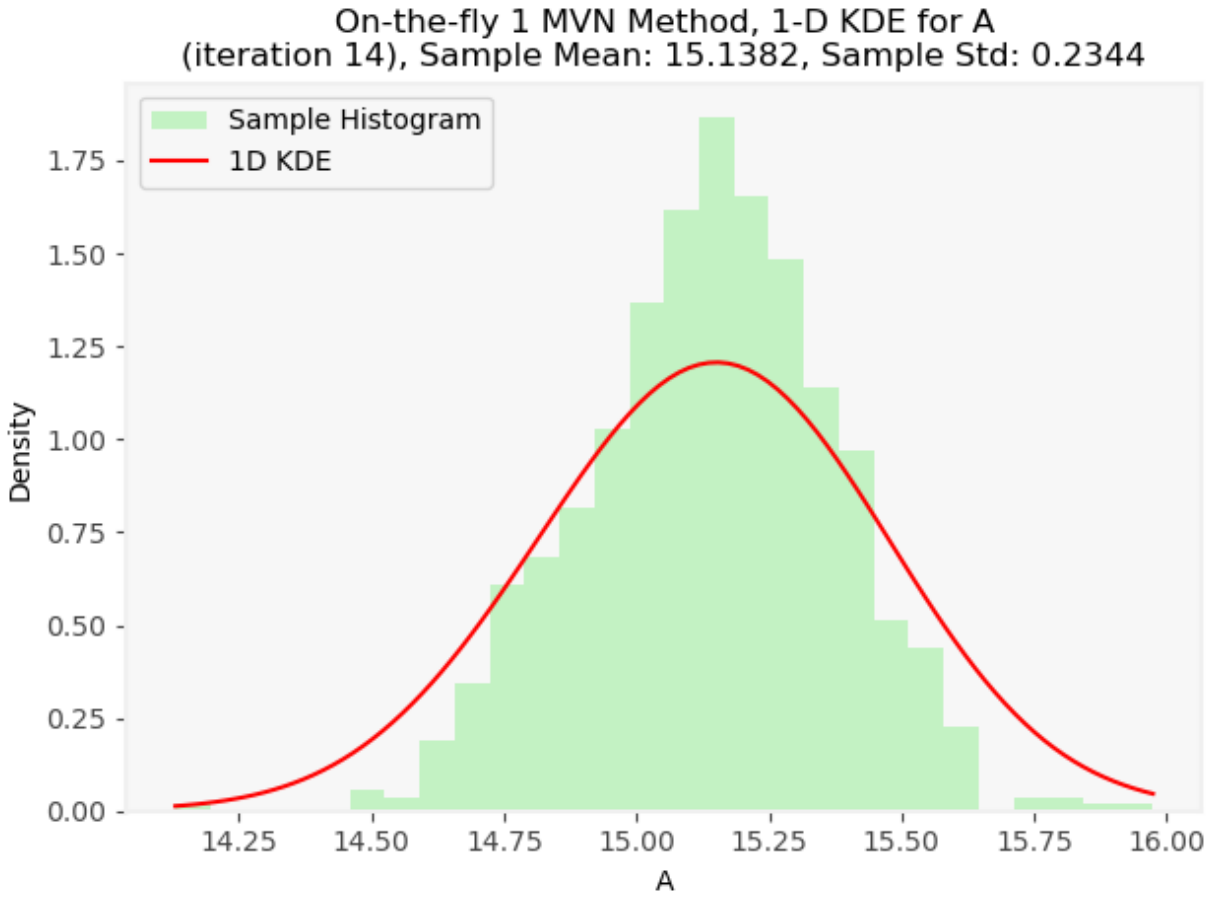
A

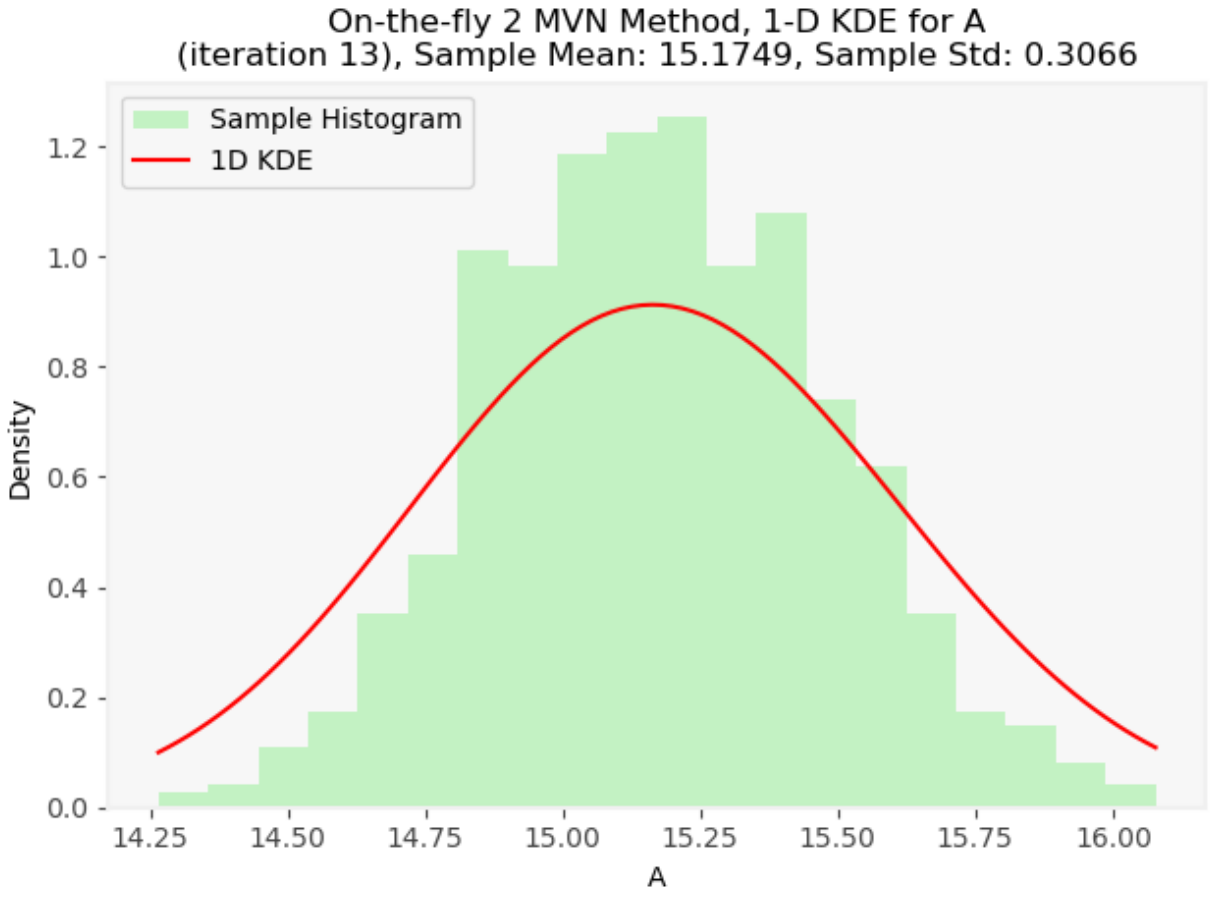
1.First we look at the last sample for all the autonomous approaches. 2.Second, we look at the result using GMM

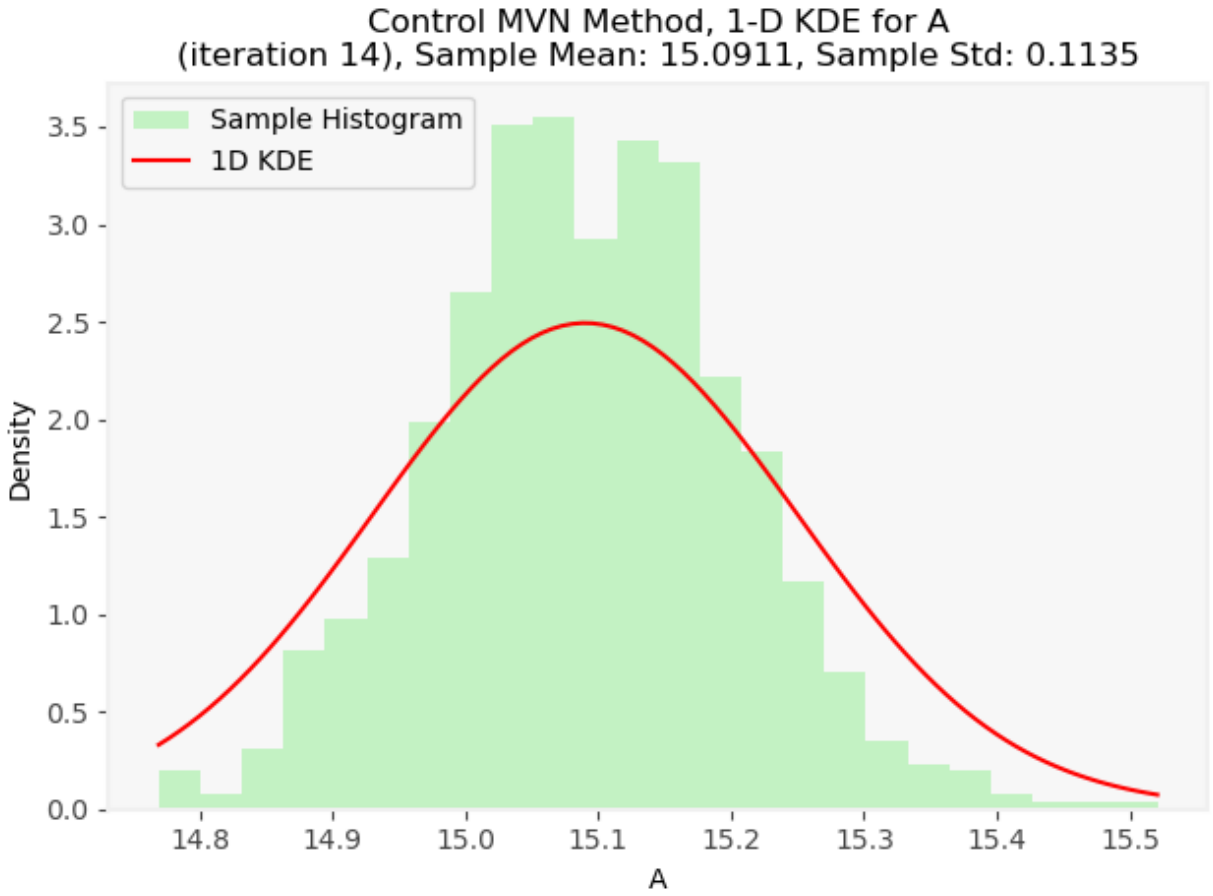
In [39]:
```
MyPlots.plot_hist_1d_kde(list_par_separated_e1[0][-1], kdes_entropy1[-1,0],"Entropy 1
plt.show()

MyPlots.plot_hist_1d_kde(list_par_separated_e2[0][-1], kdes_entropy2[-1,0],"Entropy 2
                         len(exp_entropy2.totaltimes()))
plt.show()

MyPlots.plot_hist_1d_kde(list_par_separated_o1[0][-1], kdes_on_the_fly1[-1,0],"On-the-
                          len(exp_on_the_fly1.totaltimes()))
plt.show()

MyPlots.plot_hist_1d_kde(list_par_separated_o2[0][-1], kdes_on_the_fly2[-1,0],"On-the-
                         len(exp_on_the_fly2.totaltimes()))
plt.show()

MyPlots.plot_hist_1d_kde(list_par_separated_c[0][-1], kdes_control[-1,0],"Control MVN'
                         len(exp_control.totaltimes()))
plt.show()
```


Entropy 1 MVN Method, 1-D KDE for A
(iteration 13), Sample Mean: 15.0176, Sample Std: 0.1234
Sample Histogram
1D KDE
Density
A
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
14.6
14.8
15.0
15.2
15.4


Entropy 2 MVN Method, 1-D KDE for A
(iteration 14), Sample Mean: 15.0971, Sample Std: 0.2265
Sample Histogram
1D KDE
Density
A
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
14.6
14.8
15.0
15.2
15.4
15.6


On-the-fly 1 MVN Method, 1-D KDE for A
(iteration 14), Sample Mean: 15.1382, Sample Std: 0.2344
Sample Histogram
1D KDE
Density
A
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
14.25
14.50
14.75
15.00
15.25
15.50
15.75
16.00


On-the-fly 2 MVN Method, 1-D KDE for A
(iteration 13), Sample Mean: 15.1749, Sample Std: 0.3066
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
1.2
14.25
14.50
14.75
15.00
15.25
15.50
15.75
16.00



In [40]:
```
#DELETE THIS LATER THIS MAKES SENSE ONLY IF YOU HAVE ALSO A GMM VERSION ##############


# MyPlots.plot_hist_1d_kde(list_par_separated_e1_gmm[0][-1],
#                          kdes_entropy1_gmm[-1,0],"Entropy 1 GMM", "A", len(exp_entro
# plt.show()


# MyPlots.plot_hist_1d_kde(list_par_separated_e2_gmm[0][-1], kdes_entropy2_gmm[-1,0],'
#                         len(exp_entropy2_gmm.totaltimes()))
# plt.show()


# MyPlots.plot_hist_1d_kde(list_par_separated_o1_gmm[0][-1], kdes_on_the_fly1_gmm[-1,0
#                          len(exp_on_the_fly1_gmm.totaltimes()))
# plt.show()


# MyPlots.plot_hist_1d_kde(list_par_separated_o2_gmm[0][-1], kdes_on_the_fly2_gmm[-1,0
#                         len(exp_on_the_fly2_gmm.totaltimes()))
# plt.show()

# MyPlots.plot_hist_1d_kde(list_par_separated_c_gmm[0][-1], kdes_control_gmm[-1,0],"Co
#                         len(exp_control_gmm.totaltimes()))
```

```
##########################################################################################
```

Recalculating Entropies for the cases using the GMM

In [41]:
```
# entropy1_H_total_gmm = recalculating_entropy(exp_entropy1, None, gmm_setting)
# entropy1_H_marg_gmm = recalculating_entropy(exp_entropy1, exp_entropy1.settings.sel,

# entropy2_H_total_gmm = recalculating_entropy(exp_entropy2, None,gmm_setting)
# entropy2_H_marg_gmm = recalculating_entropy(exp_entropy2, exp_on_the_fly1.settings.s
#

# on_the_fly1_H_total_gmm = recalculating_entropy(exp_on_the_fly1, None, gmm_setting)
# on_the_fly1_H_marg_gmm = recalculating_entropy(exp_on_the_fly1, exp_on_the_fly1.sett

# on_the_fly2_H_total_gmm = recalculating_entropy(exp_on_the_fly2, None,gmm_setting)
# on_the_fly2_H_marg_gmm = recalculating_entropy(exp_on_the_fly2, exp_on_the_fly2.sett

# control_H_total_gmm = recalculating_entropy(exp_control, None, gmm_setting)
# control_H_marg_gmm = recalculating_entropy(exp_control, exp_control.settings.sel, gm
```

In [42]:
```
#Notice we could also print the total entropy, instead, of the marginalized entropy.
#On-thefly always deals with the total entropy. You cannot choose a parameter of inter

#Using the recalcualed entropies using GMM


# times = [exp_entropy1.totaltimes(), exp_on_the_fly1.totaltimes(),
#          exp_on_the_fly2.totaltimes(),exp_control.totaltimes(), exp_entropy2.totalti
# entropies = [entropy1_H_total_gmm, on_the_fly1_H_total_gmm,
#              on_the_fly2_H_total_gmm, control_H_total_gmm, entropy2_H_total_gmm]
# MyPlots.plot_entropy_times(times, entropies)



#C1 blue entrop
#C2 yellow on the
#C3 green on the 2
#C4 red control
#C5 Purple entropy2

#Total Entropy
```

In [43]:
```
##########This works only if you have a GMM version

# MyPlots.plot_entropy_times([exp_entropy1.totaltimes(),exp_entropy1.totaltimes()],
#                            [exp_entropy1.entropy(), entropy1_H_total_gmm])

#Blue MVN
#Orange GMM
```

In [44]:
```
# MyPlots.plot_entropy_times([exp_entropy2.totaltimes(),exp_entropy2.totaltimes()],
#                            [exp_entropy2.entropy(), entropy2_H_total_gmm])
```

```
#Again undistingishable
```

In [45]:
```
# MyPlots.plot_entropy_times([exp_on_the_fly1.totaltimes(),exp_on_the_fly1.totaltimes(
#                           [exp_on_the_fly1.entropy(), on_the_fly1_H_total_gmm])


#Again undistingishable
```

In [46]:
```
# MyPlots.plot_entropy_times([exp_on_the_fly2.totaltimes(),exp_on_the_fly2.totaltimes(
#                           [exp_on_the_fly2.entropy(), on_the_fly2_H_total_gmm])


#Again undistingishable
```

In [47]:
```
#Notice we could also print the total entropy, instead, of the marginalized entropy.
#On-thefly always deals with the total entropy. You cannot choose a parameter of inter


# times = [exp_entropy1.totaltimes(), exp_on_the_fly1.totaltimes(),
#          exp_on_the_fly2.totaltimes(),exp_control.totaltimes(), exp_entropy2.totalti
# entropies = [entropy1_H_marg_gmm, on_the_fly1_H_marg_gmm,
#              on_the_fly2_H_marg_gmm, control_H_marg_gmm, entropy2_H_marg_gmm]
# MyPlots.plot_entropy_times(times, entropies)
```

In [48]:
```
# MyPlots.plot_entropy_times([exp_entropy1.totaltimes(),exp_entropy1.totaltimes()],
#                           [exp_entropy1.entropy_marg(),entropy1_H_marg_gmm ])
```

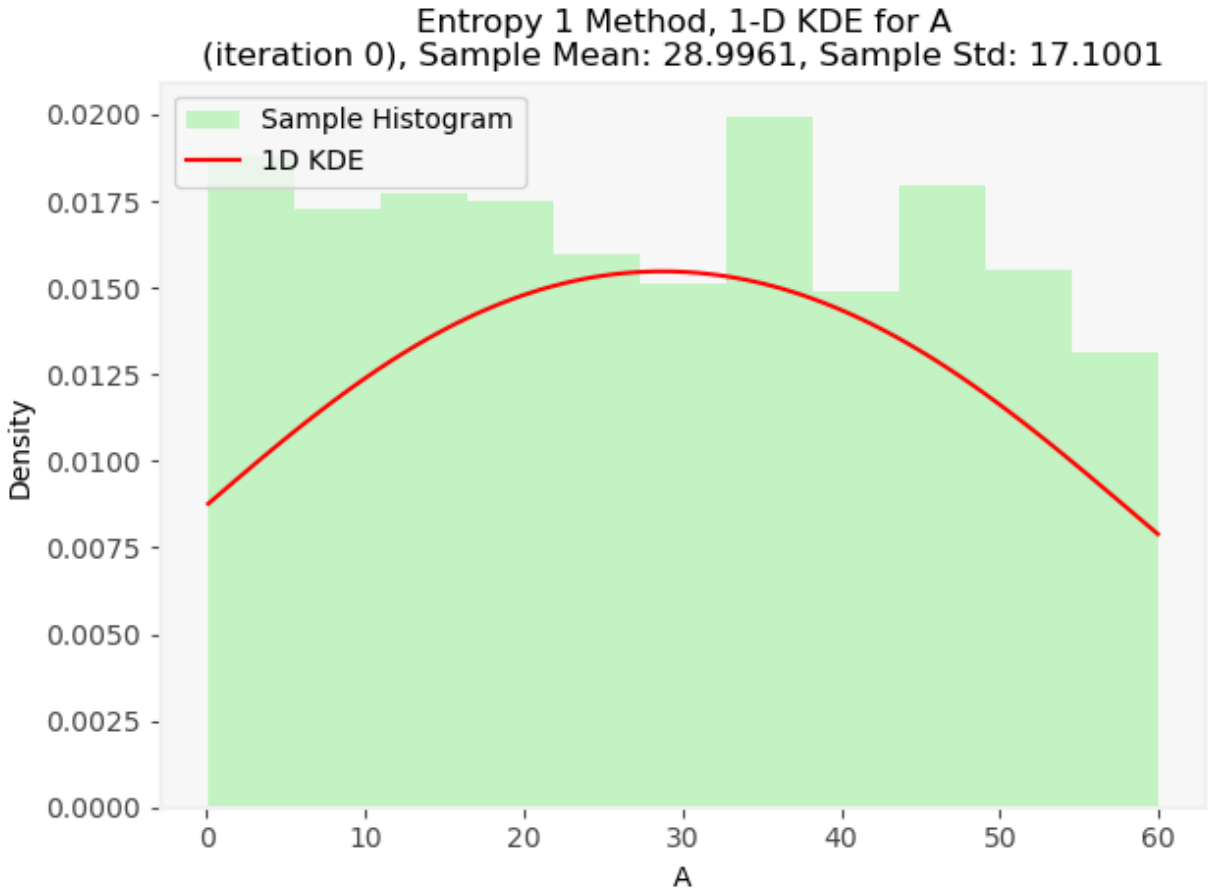
In [49]:
```
# MyPlots.plot_entropy_times([exp_entropy2.totaltimes(),exp_entropy2.totaltimes()],
#                           [exp_entropy2.entropy_marg(),entropy2_H_marg_gmm ])
```

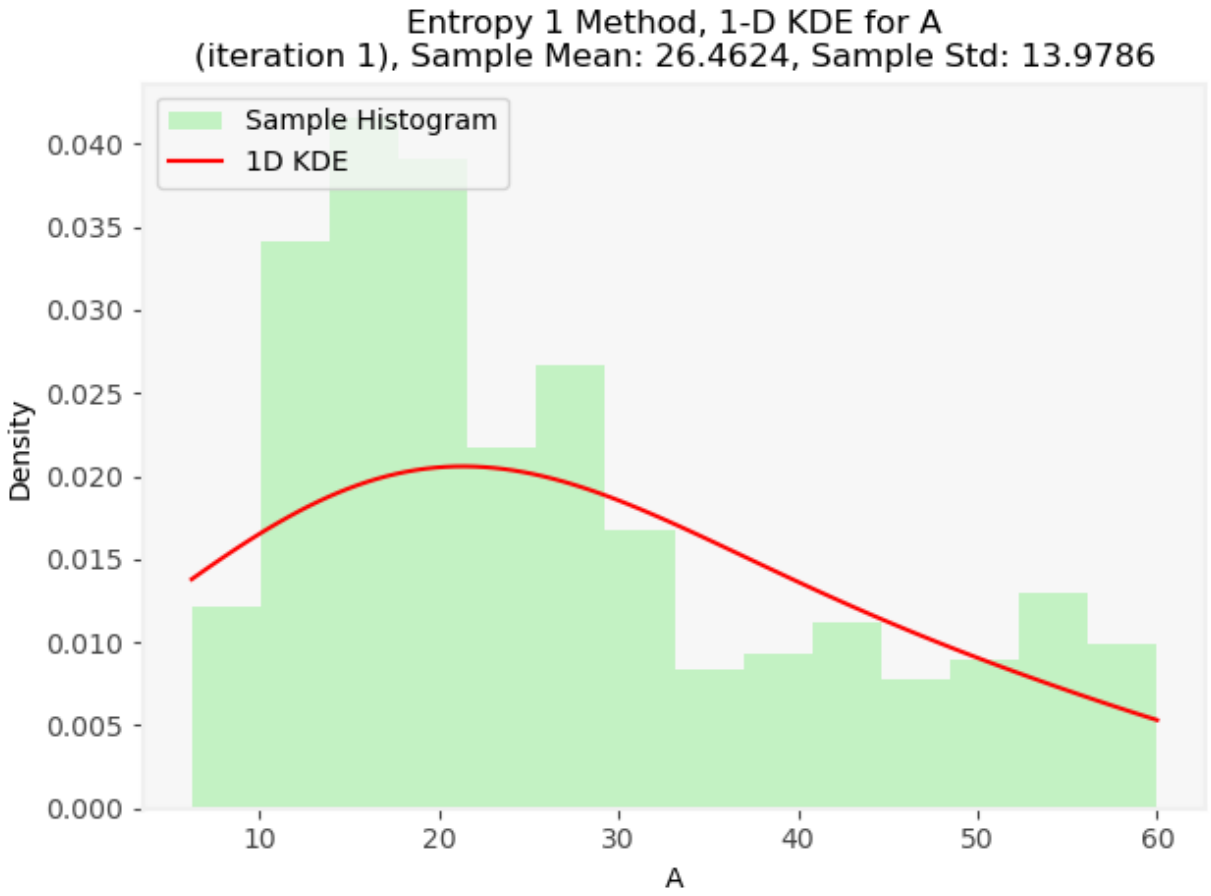
In [50]:
```
# MyPlots.plot_entropy_times([exp_on_the_fly1.totaltimes(),exp_on_the_fly1.totaltimes(
#                           [exp_on_the_fly1.entropy_marg(),on_the_fly1_H_marg_gmm])
```

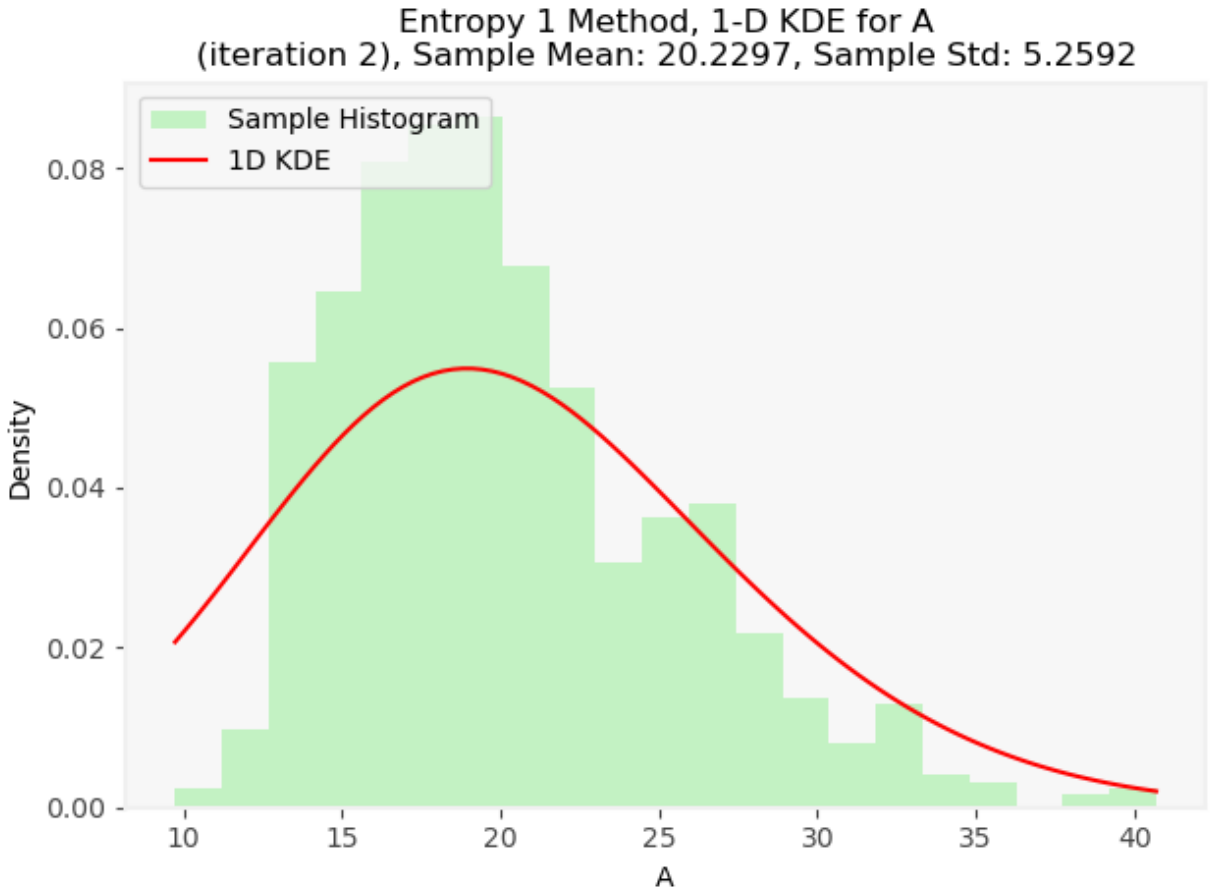
In [51]:
```
# MyPlots.plot_entropy_times([exp_on_the_fly2.totaltimes(),exp_on_the_fly2.totaltimes(
#                           [exp_on_the_fly2.entropy_marg(),on_the_fly2_H_marg_gmm])
```

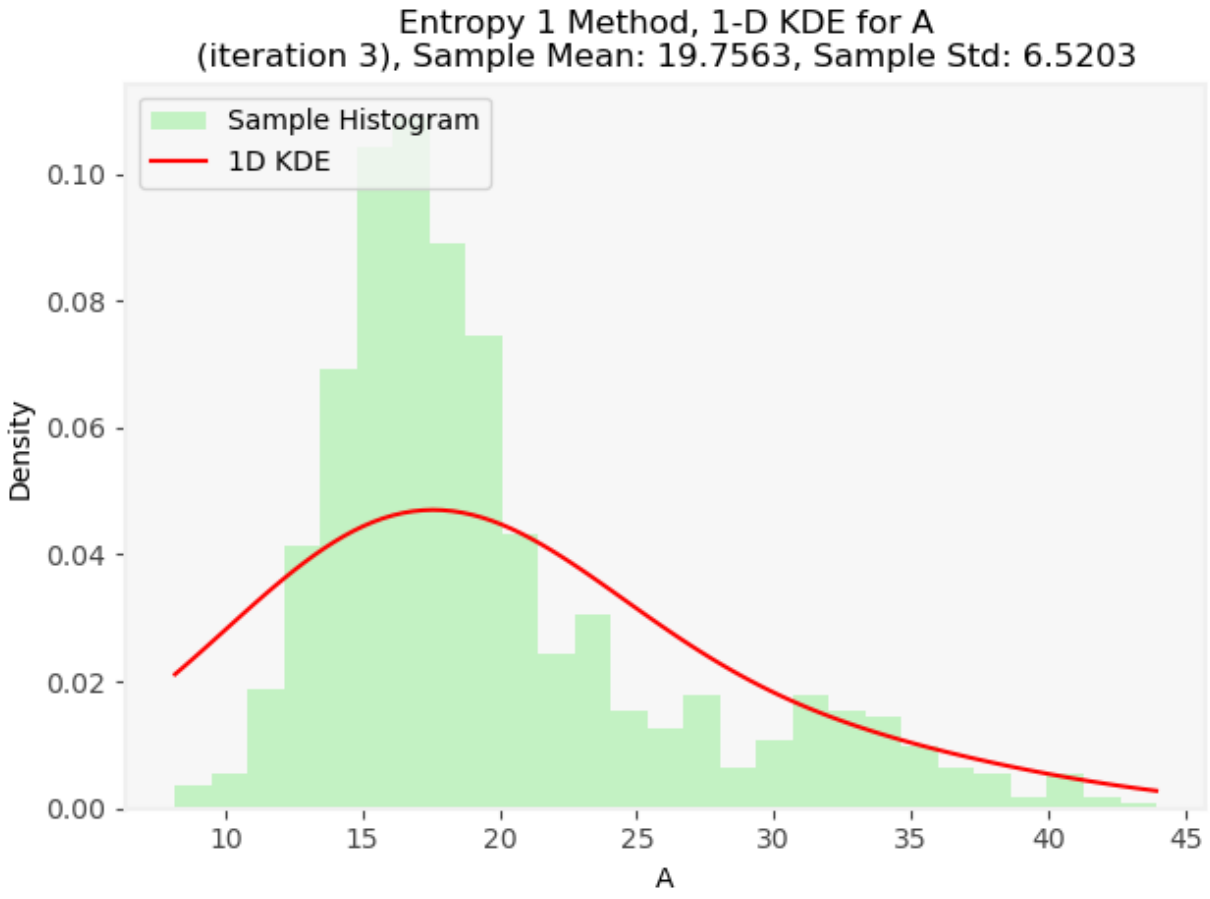
Enf of the Recalculation section

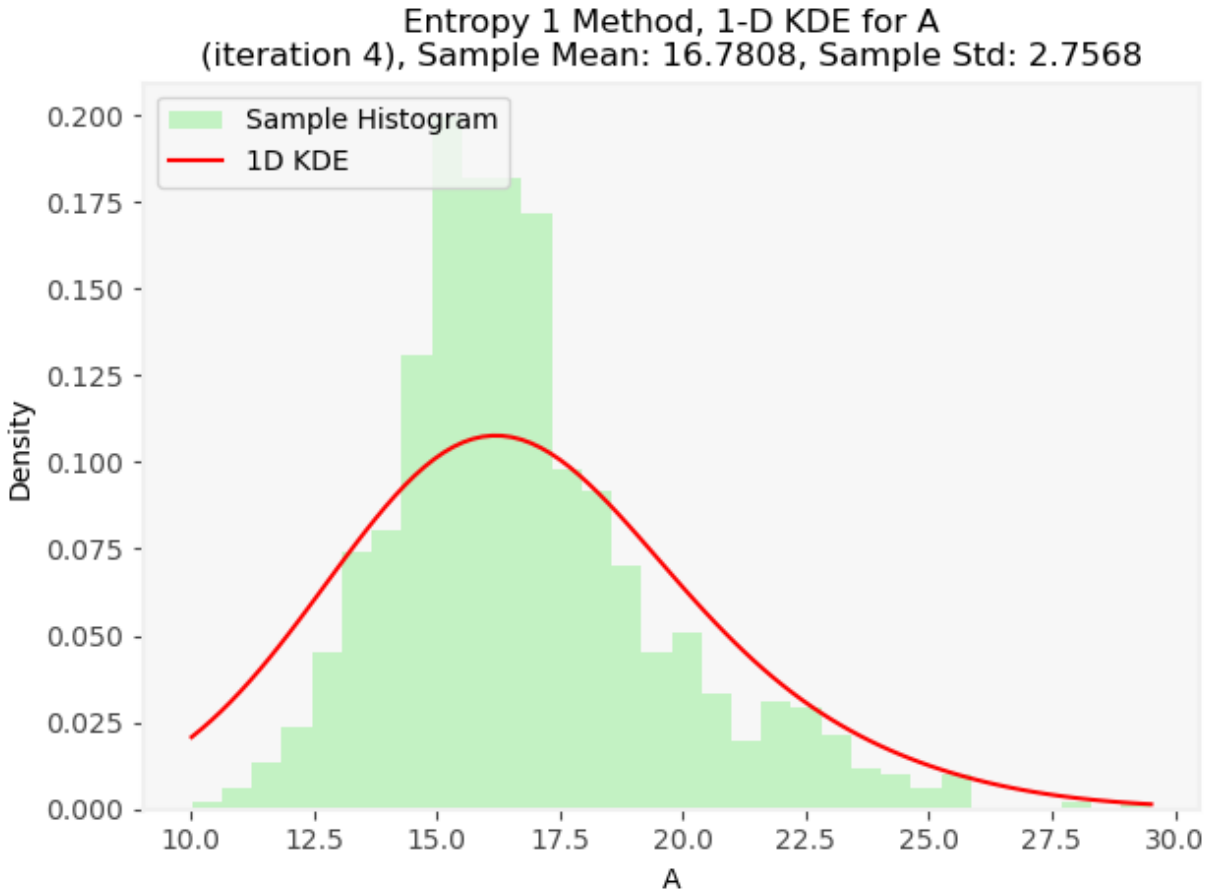
In [52]:
```
#Printing the evolution of parameter I for Entropy
#later do the same for parameter A since that was the parameter selected by entropy


for i in range(len(exp_entropy1.totaltimes())):
    MyPlots.plot_hist_1d_kde(list_par_separated_e1[0][i], kdes_entropy1[i,0],"Entropy
```

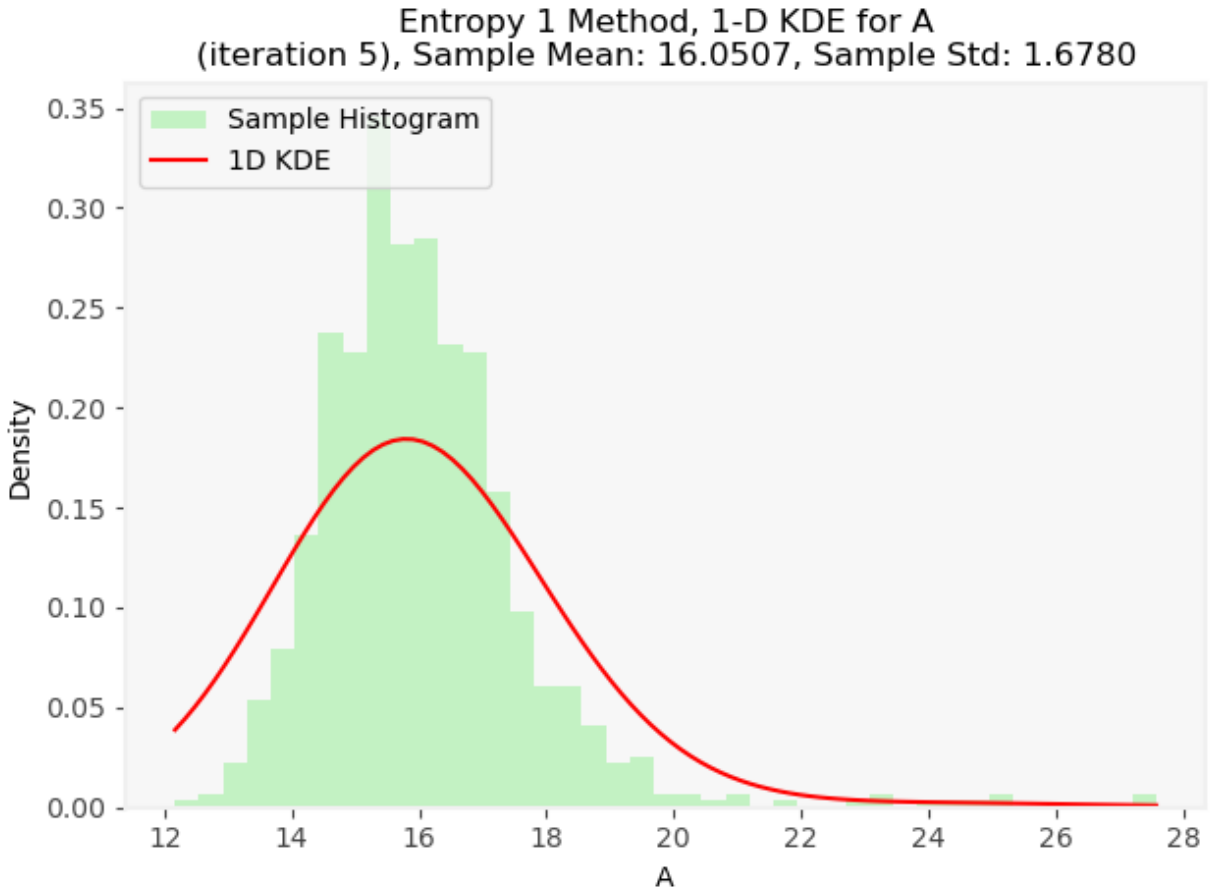

Entropy 1 Method, 1-D KDE for A
(iteration 0), Sample Mean: 28.9961, Sample Std: 17.1001
Sample Histogram
1D KDE
Density
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
10
20
30
40
50
60
A

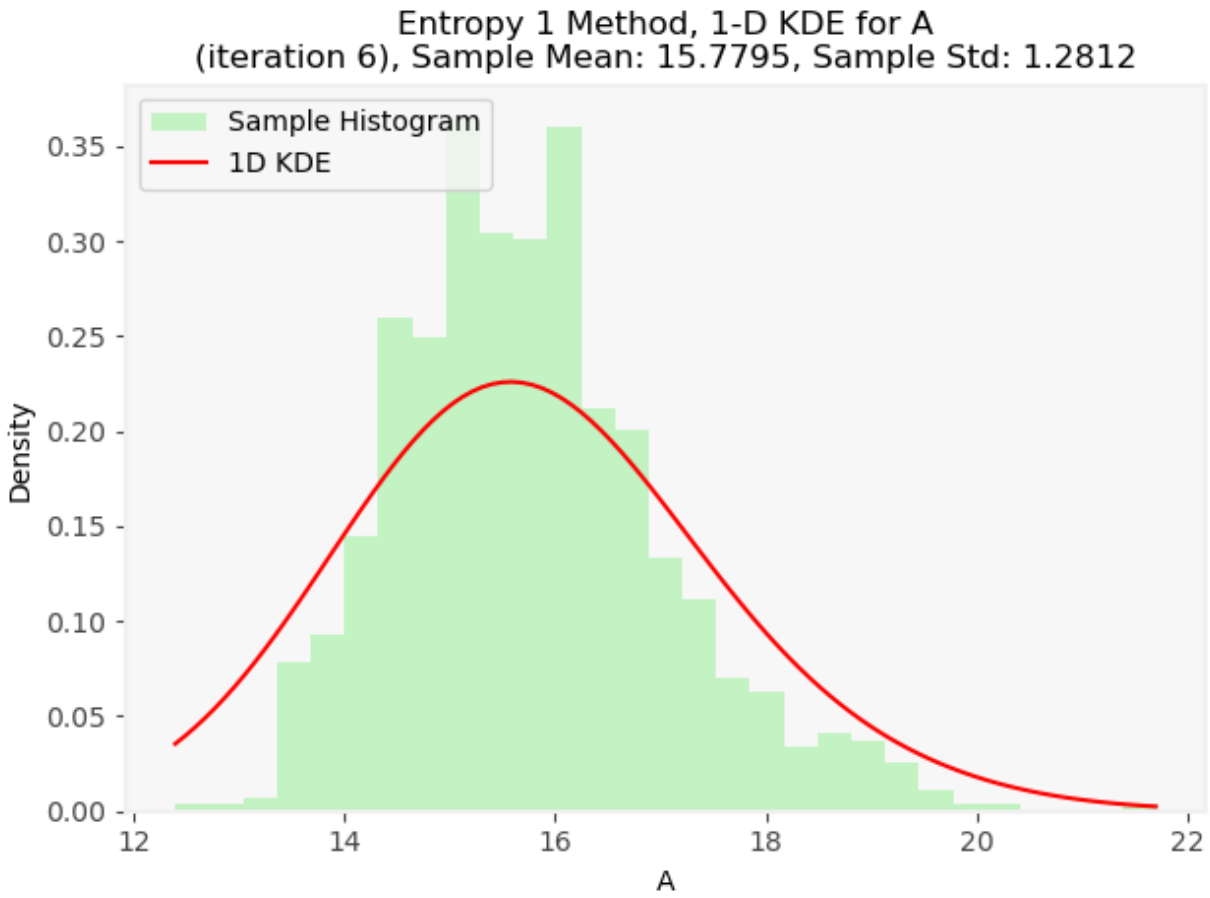

Entropy 1 Method, 1-D KDE for A
(iteration 1), Sample Mean: 26.4624, Sample Std: 13.9786
Sample Histogram
1D KDE
Density
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
10
20
30
40
50
60
A

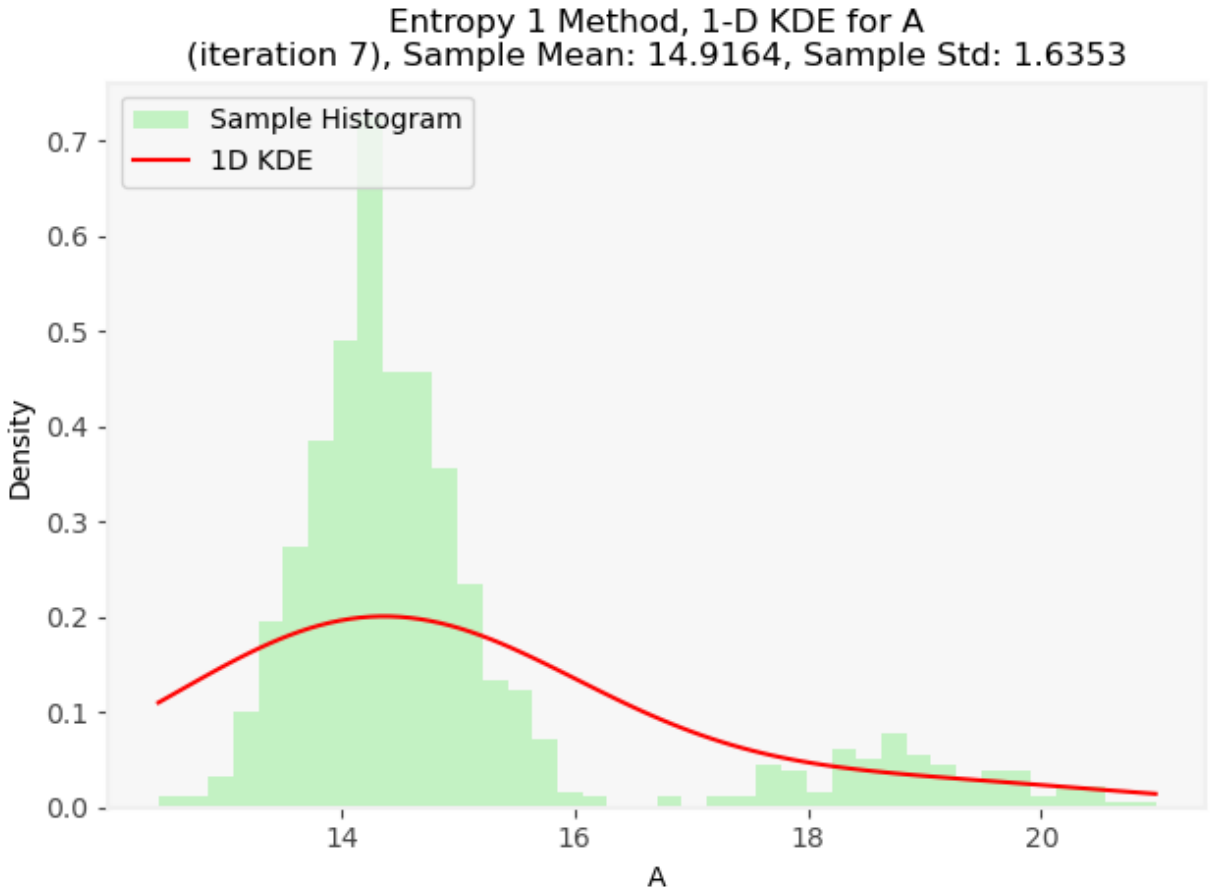

Entropy 1 Method, 1-D KDE for A
(iteration 2), Sample Mean: 20.2297, Sample Std: 5.2592
Sample Histogram
1D KDE
Density
0.00
0.02
0.04
0.06
0.08
10
15
20
25
30
35
40
A

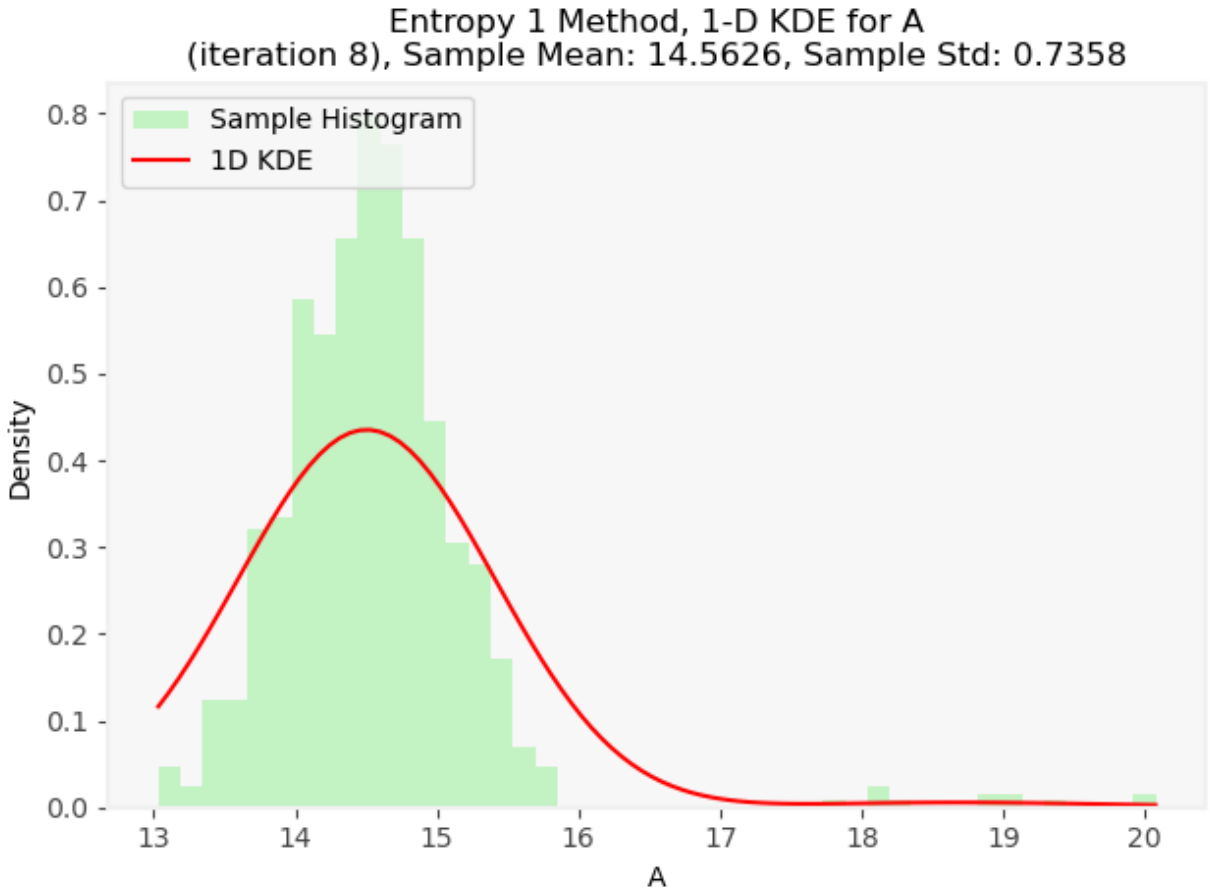

Entropy 1 Method, 1-D KDE for A
(iteration 3), Sample Mean: 19.7563, Sample Std: 6.5203
Sample Histogram
1D KDE
Density
0.00
0.02
0.04
0.06
0.08
0.10
10
15
20
25
30
35
40
45
A

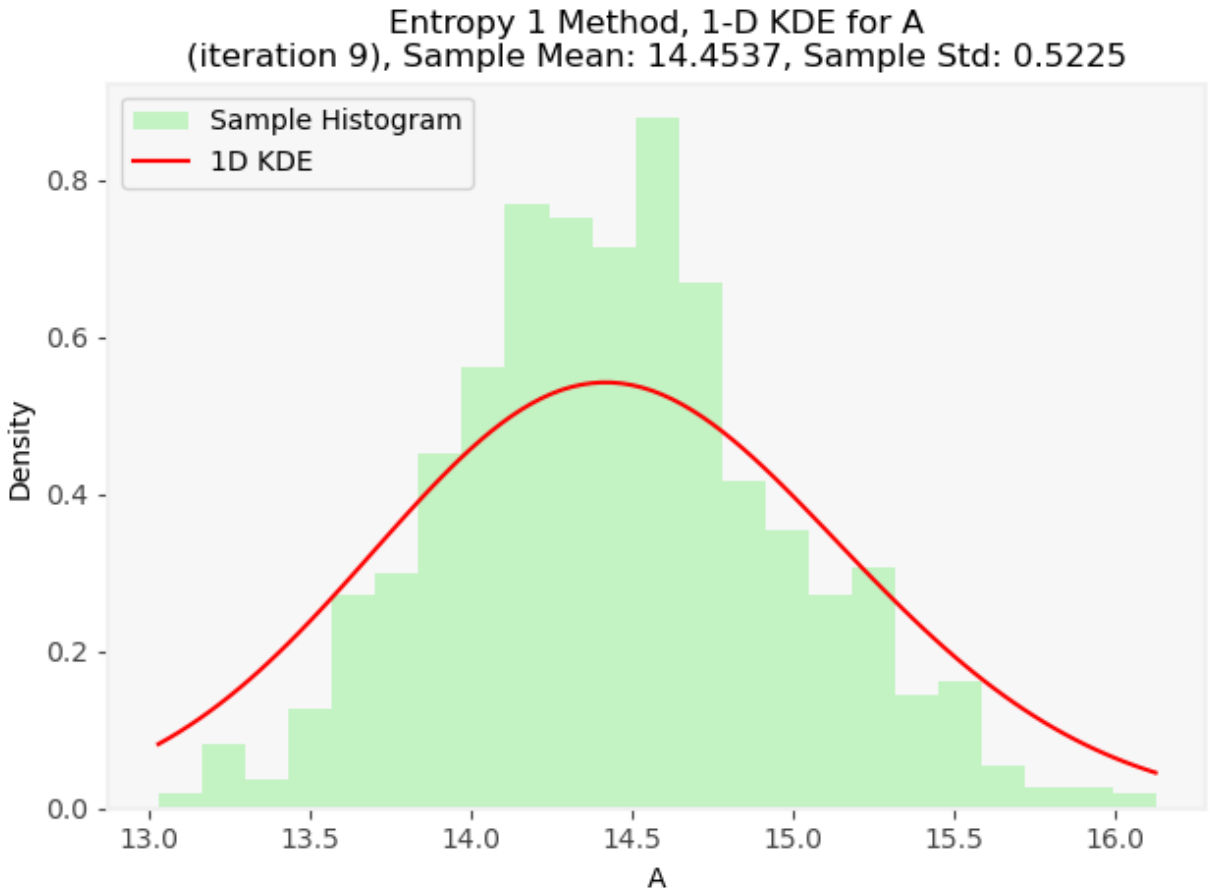

Entropy 1 Method, 1-D KDE for A
(iteration 4), Sample Mean: 16.7808, Sample Std: 2.7568
Sample Histogram
1D KDE
Density
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0
A

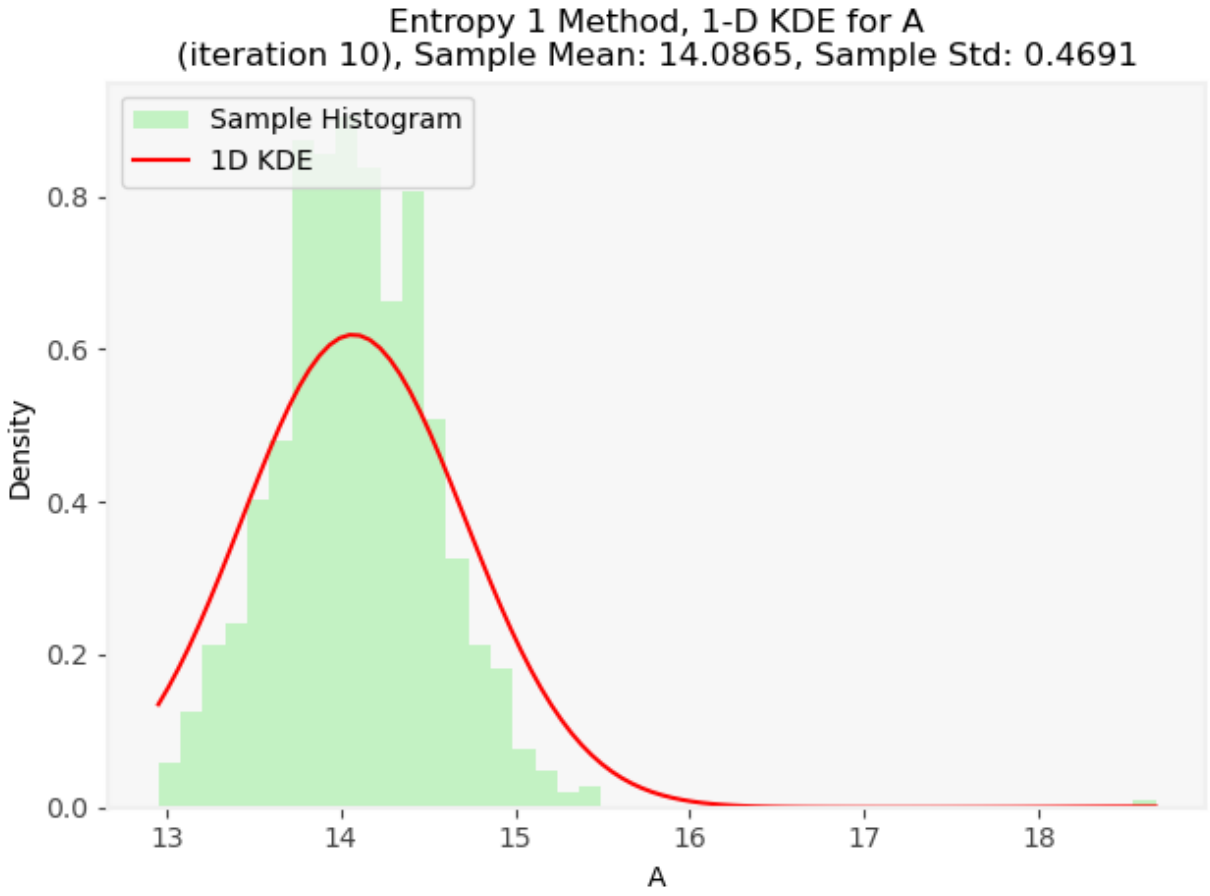

Entropy 1 Method, 1-D KDE for A
(iteration 5), Sample Mean: 16.0507, Sample Std: 1.6780
Sample Histogram
1D KDE
Density
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
12
14
16
18
20
22
24
26
28
A

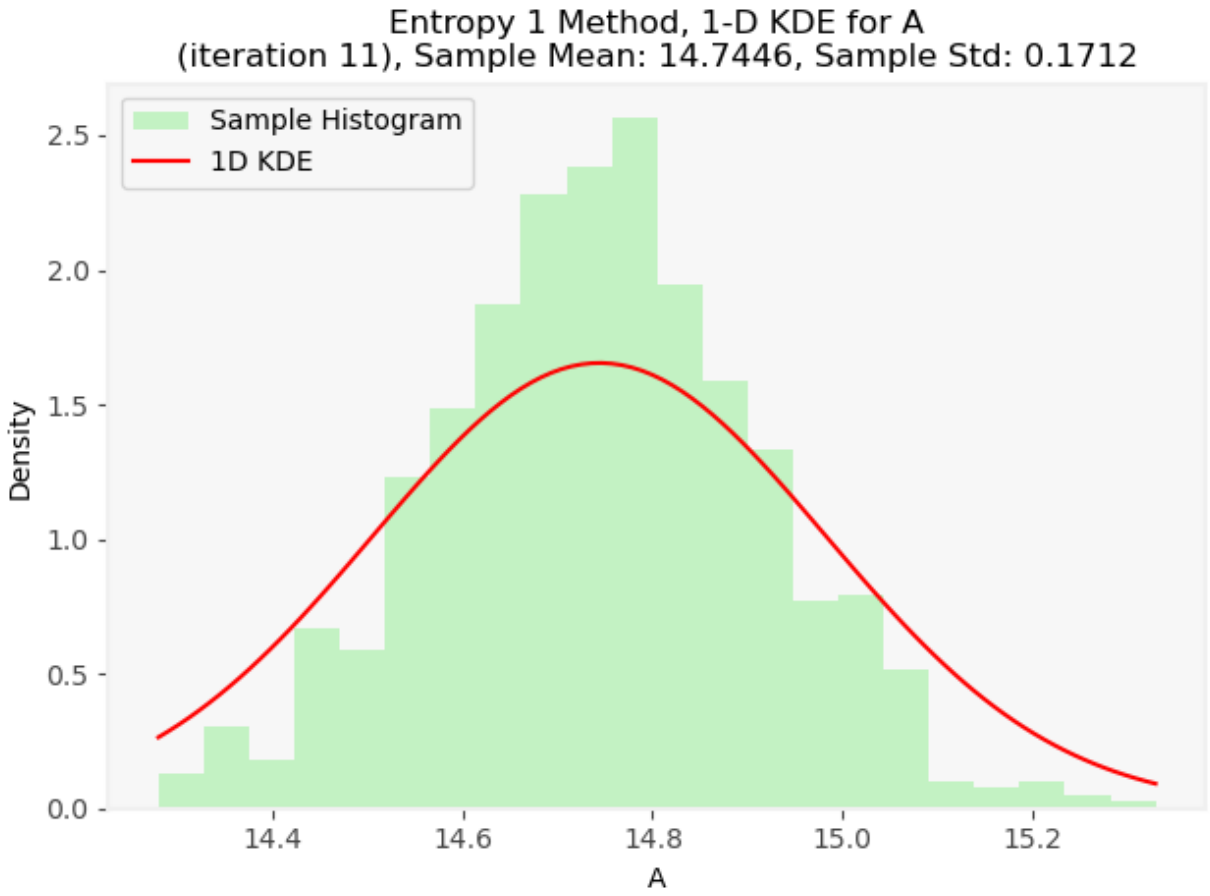

Entropy 1 Method, 1-D KDE for A
(iteration 6), Sample Mean: 15.7795, Sample Std: 1.2812
Sample Histogram
1D KDE
Density
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
12
14
16
18
20
22
A

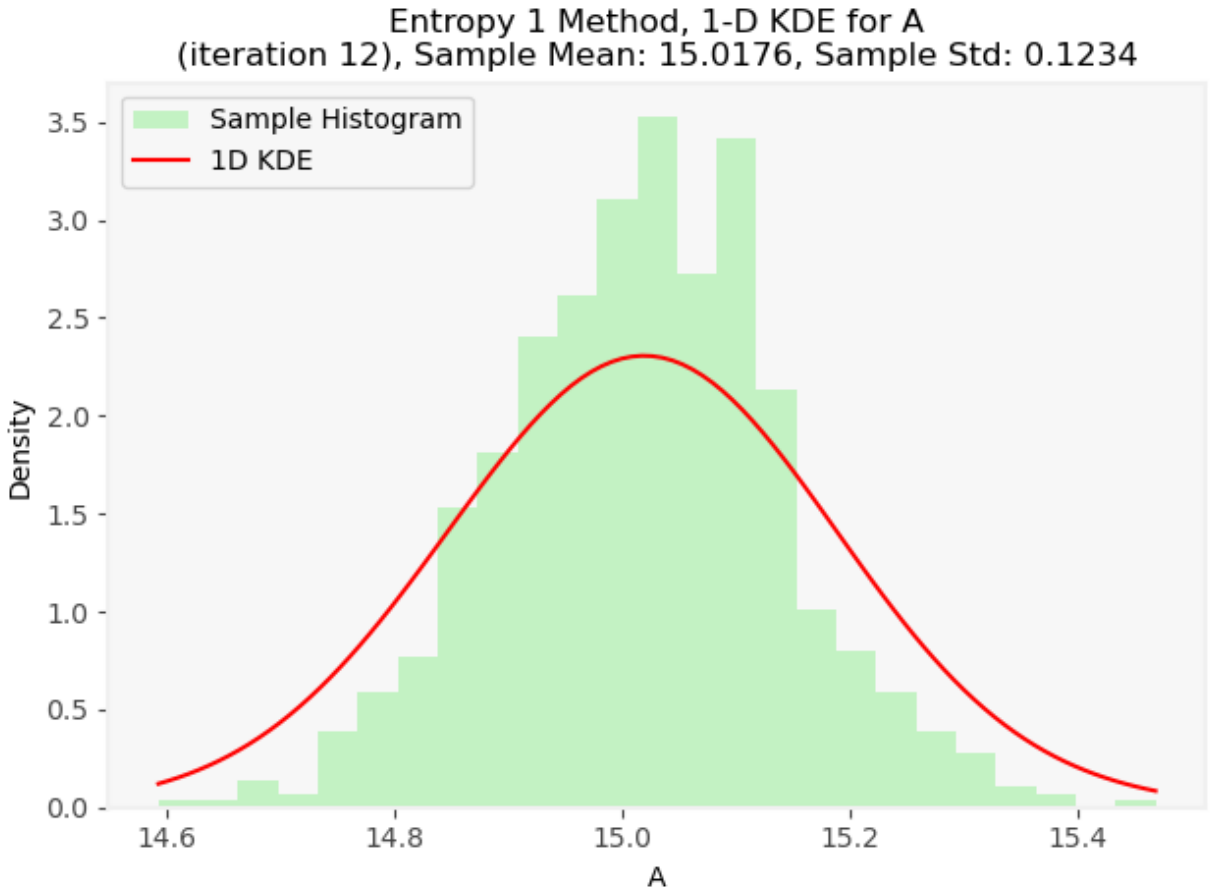

Entropy 1 Method, 1-D KDE for A
(iteration 7), Sample Mean: 14.9164, Sample Std: 1.6353
Sample Histogram
1D KDE
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
14
16
18
20
A


Entropy 1 Method, 1-D KDE for A
(iteration 8), Sample Mean: 14.5626, Sample Std: 0.7358
Sample Histogram
1D KDE
Density
A
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
13
14
15
16
17
18
19
20


Entropy 1 Method, 1-D KDE for A
(iteration 9), Sample Mean: 14.4537, Sample Std: 0.5225
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
13.0
13.5
14.0
14.5
15.0
15.5
16.0

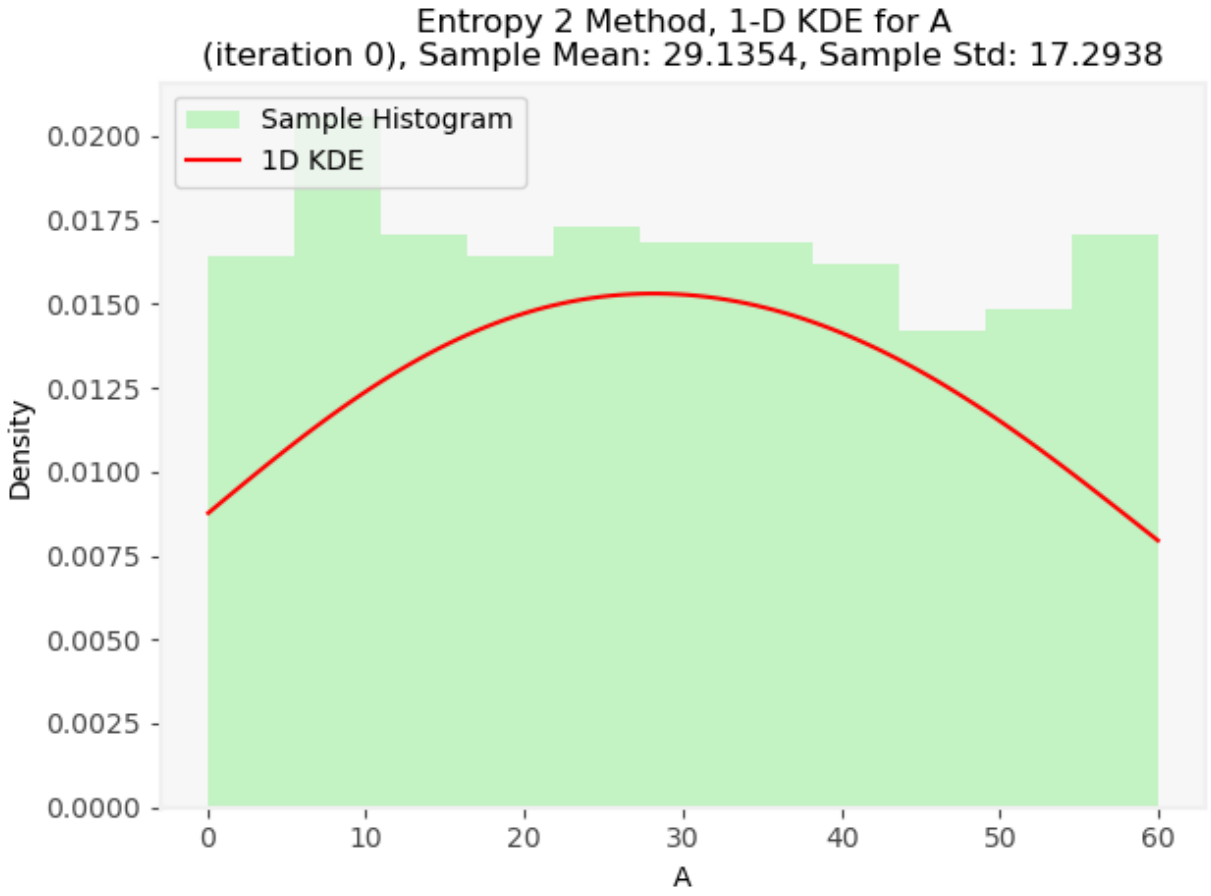

Entropy 1 Method, 1-D KDE for A
(iteration 10), Sample Mean: 14.0865, Sample Std: 0.4691
Sample Histogram
1D KDE
Density
0.0
0.2
0.4
0.6
0.8
13
14
15
16
17
18
A

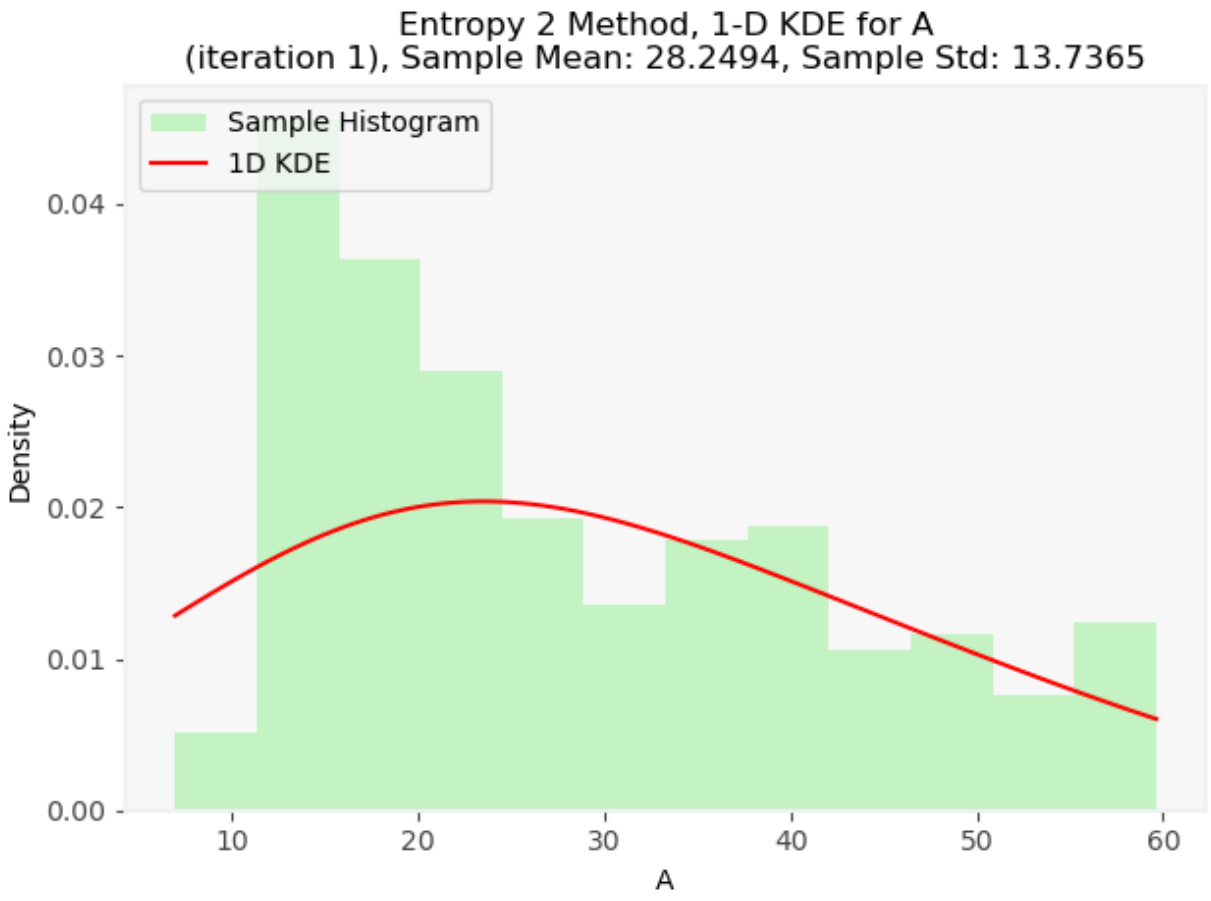

Entropy 1 Method, 1-D KDE for A
(iteration 11), Sample Mean: 14.7446, Sample Std: 0.1712
Sample Histogram
1D KDE
Density
0.0
0.5
1.0
1.5
2.0
2.5
14.4
14.6
14.8
15.0
15.2
A

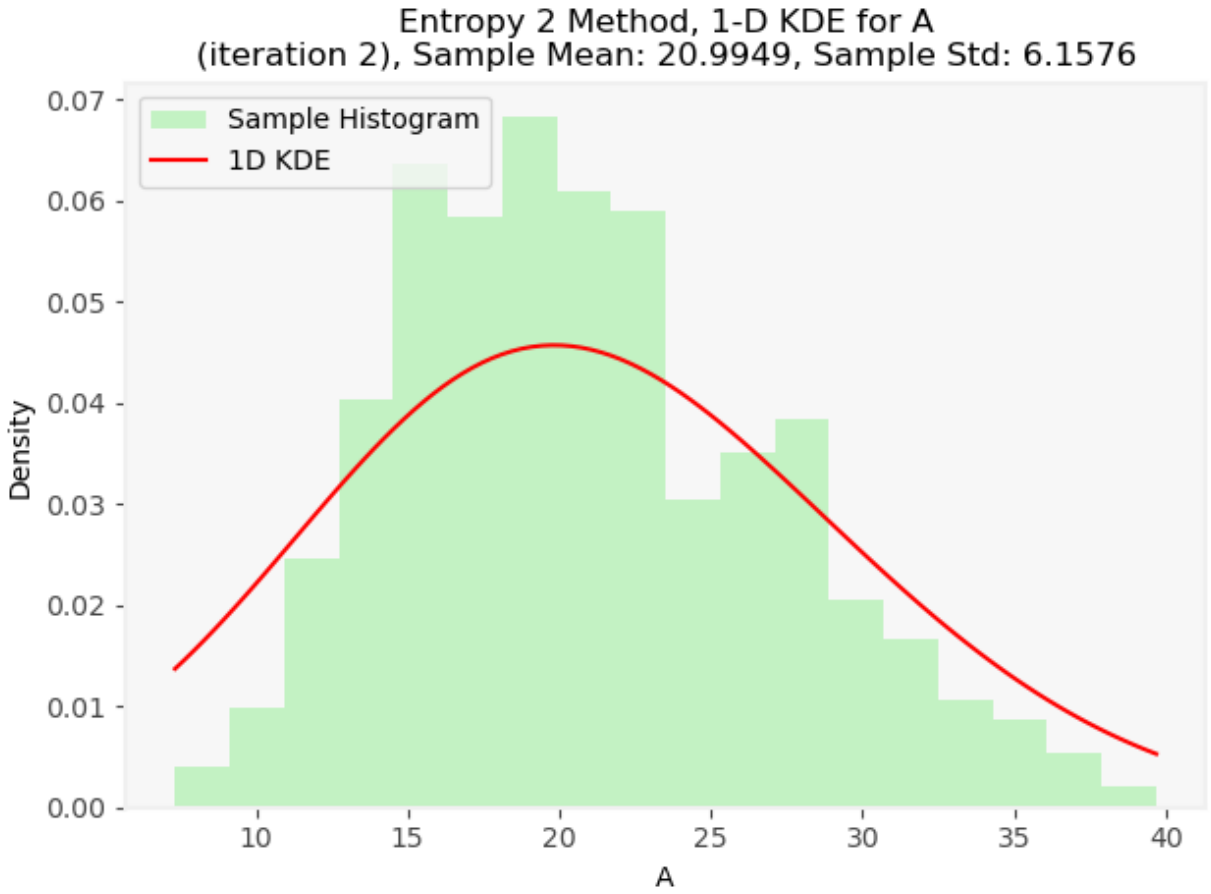


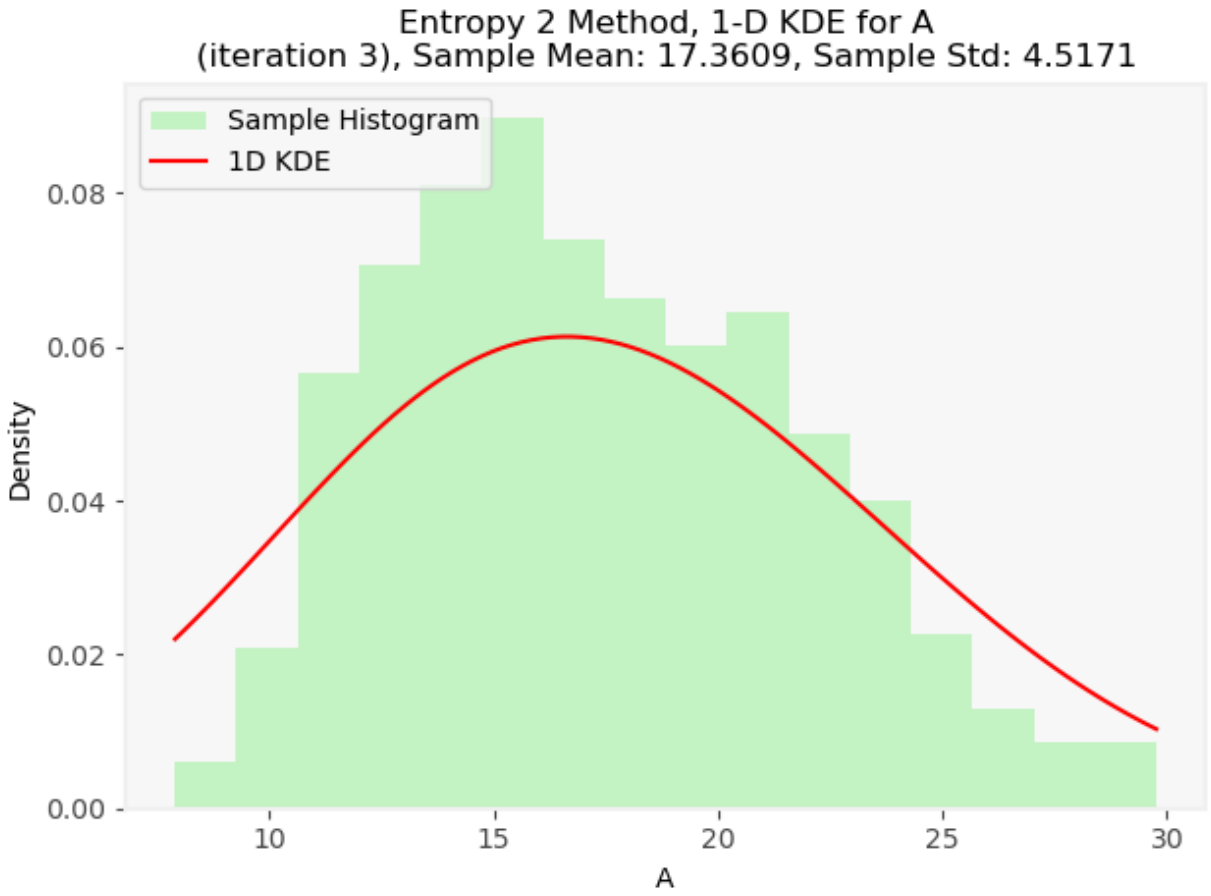
In [ ]:

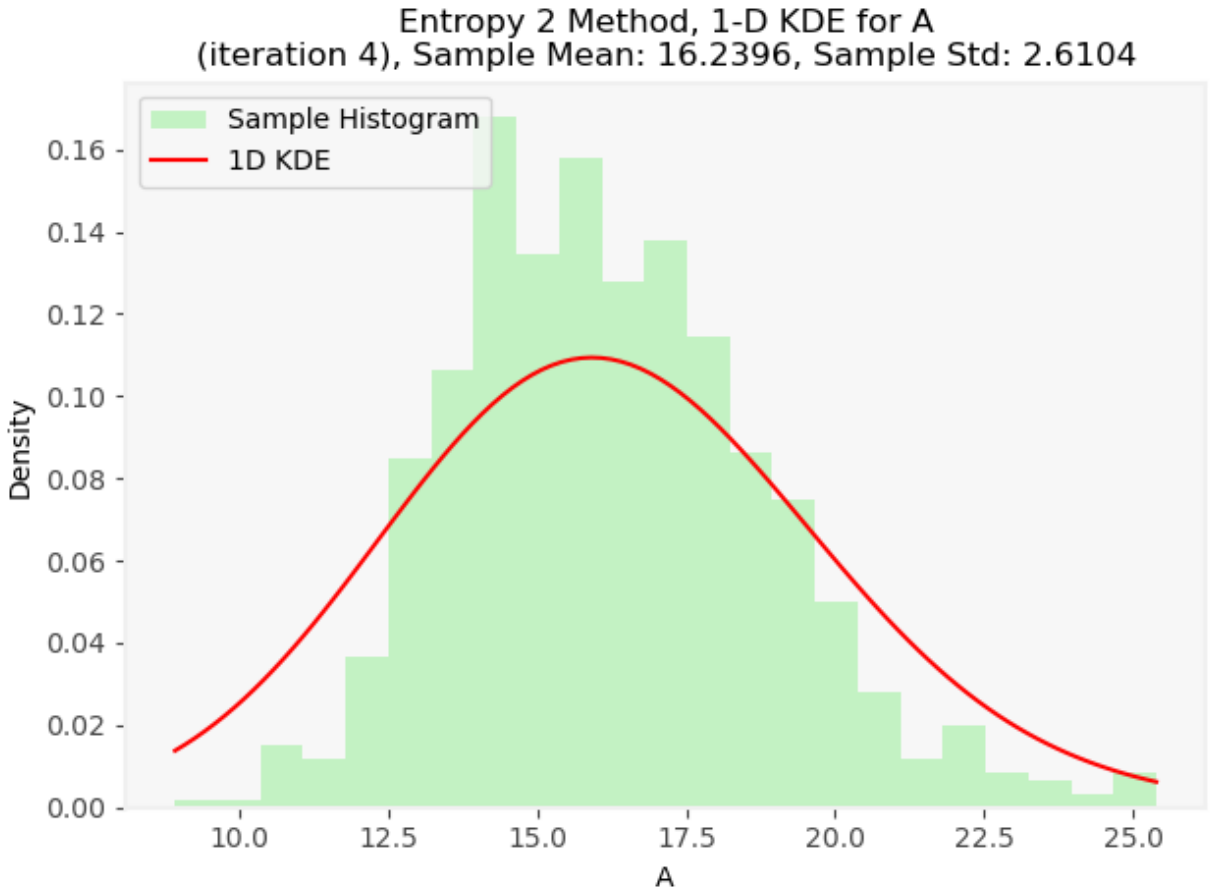
In [53]:
```
for i in range(len(exp_entropy2.totaltimes())):
    MyPlots.plot_hist_1d_kde(list_par_separated_e2[0][i], kdes_entropy2[i,0],"Entropy
```

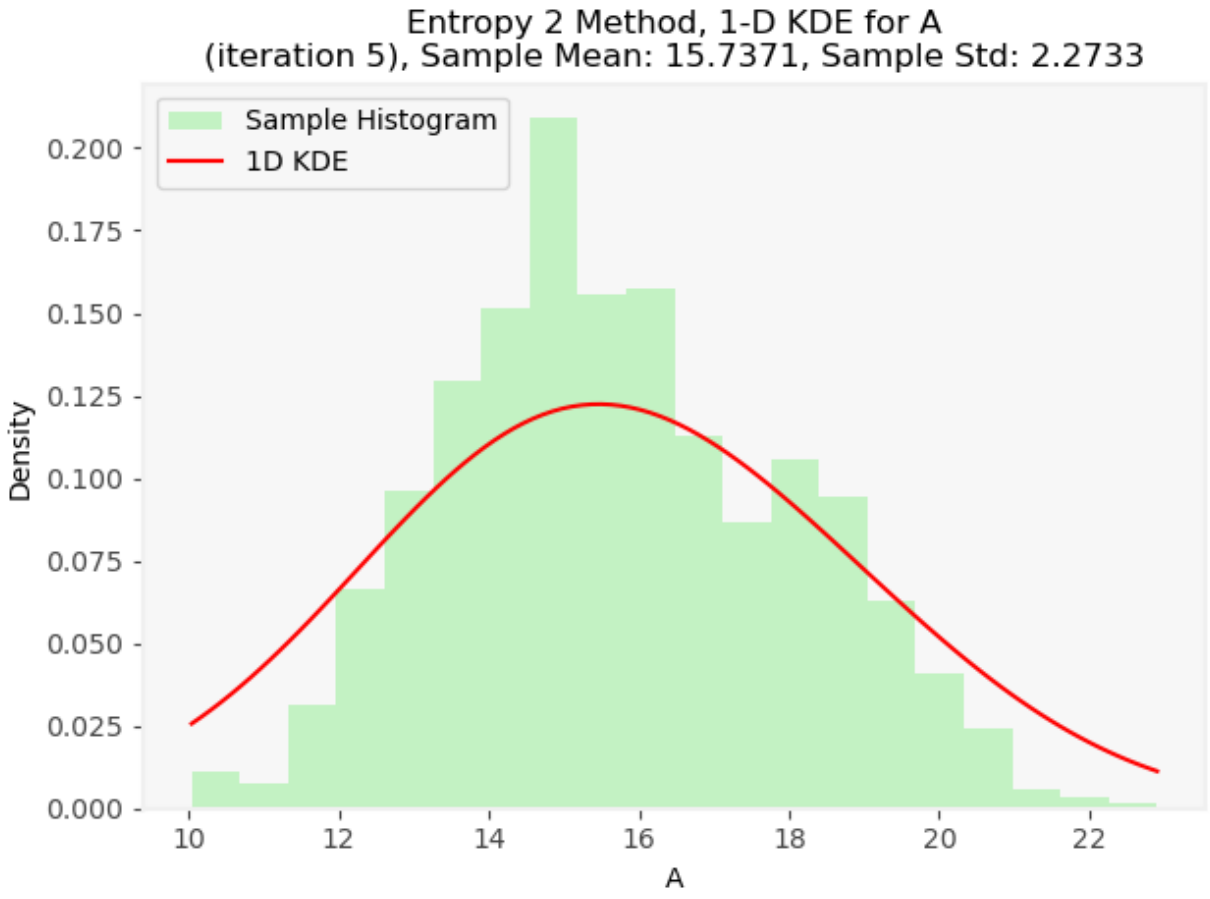

Entropy 2 Method, 1-D KDE for A
(iteration 0), Sample Mean: 29.1354, Sample Std: 17.2938
Sample Histogram
1D KDE
Density
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
10
20
30
40
50
60
A

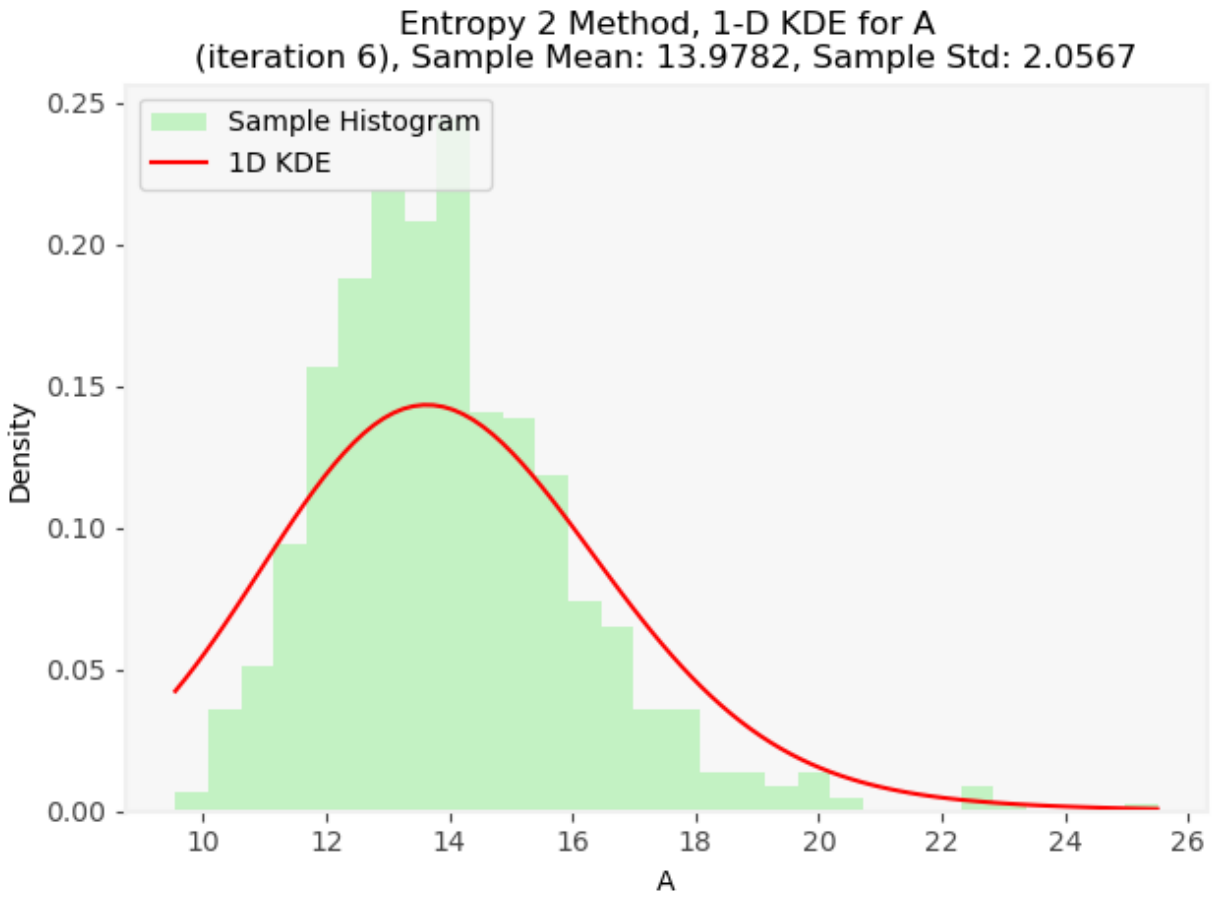

Entropy 2 Method, 1-D KDE for A
(iteration 1), Sample Mean: 28.2494, Sample Std: 13.7365
Sample Histogram
1D KDE
Density
0.04
0.03
0.02
0.01
0.00
10
20
30
40
50
60
A

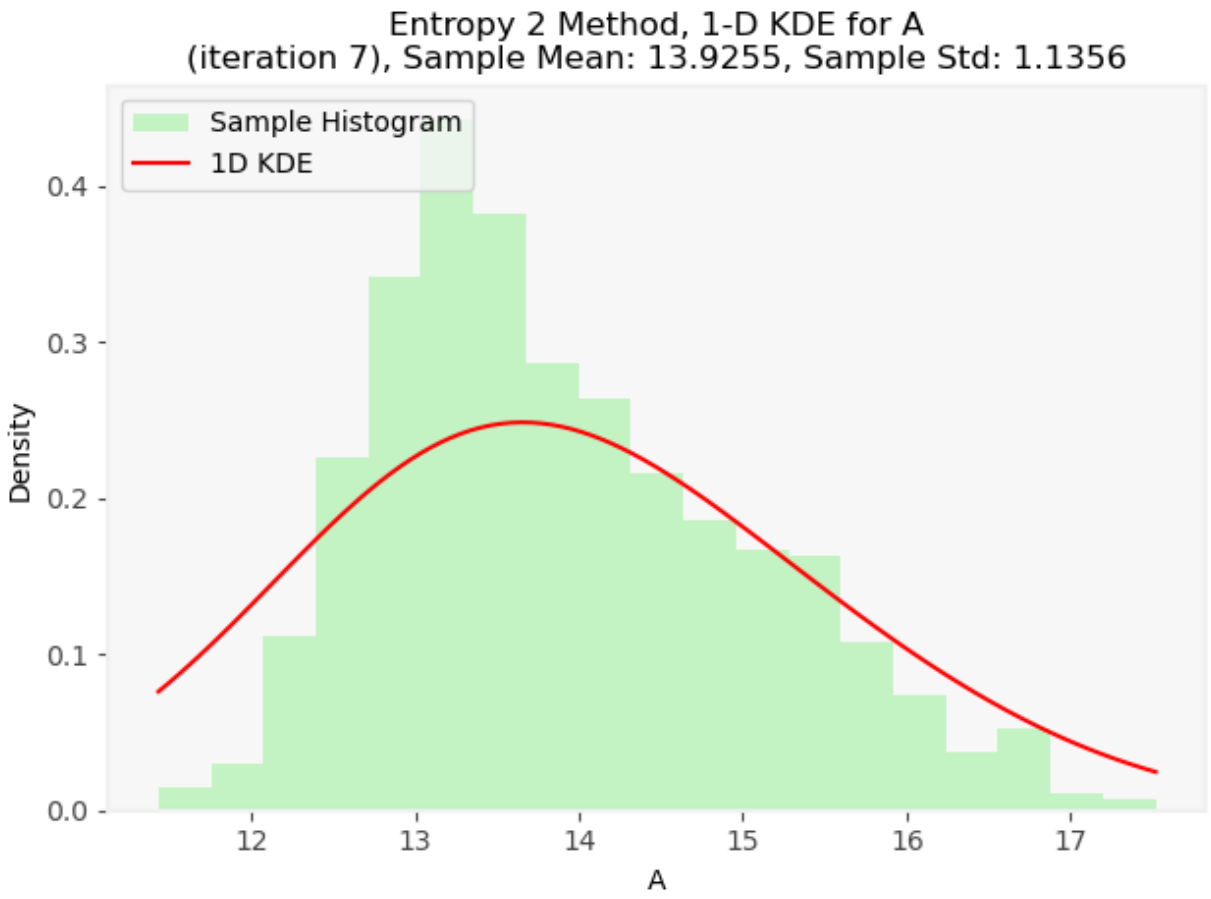

Entropy 2 Method, 1-D KDE for A
(iteration 2), Sample Mean: 20.9949, Sample Std: 6.1576
Sample Histogram
1D KDE
Density
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
10
15
20
25
30
35
40
A

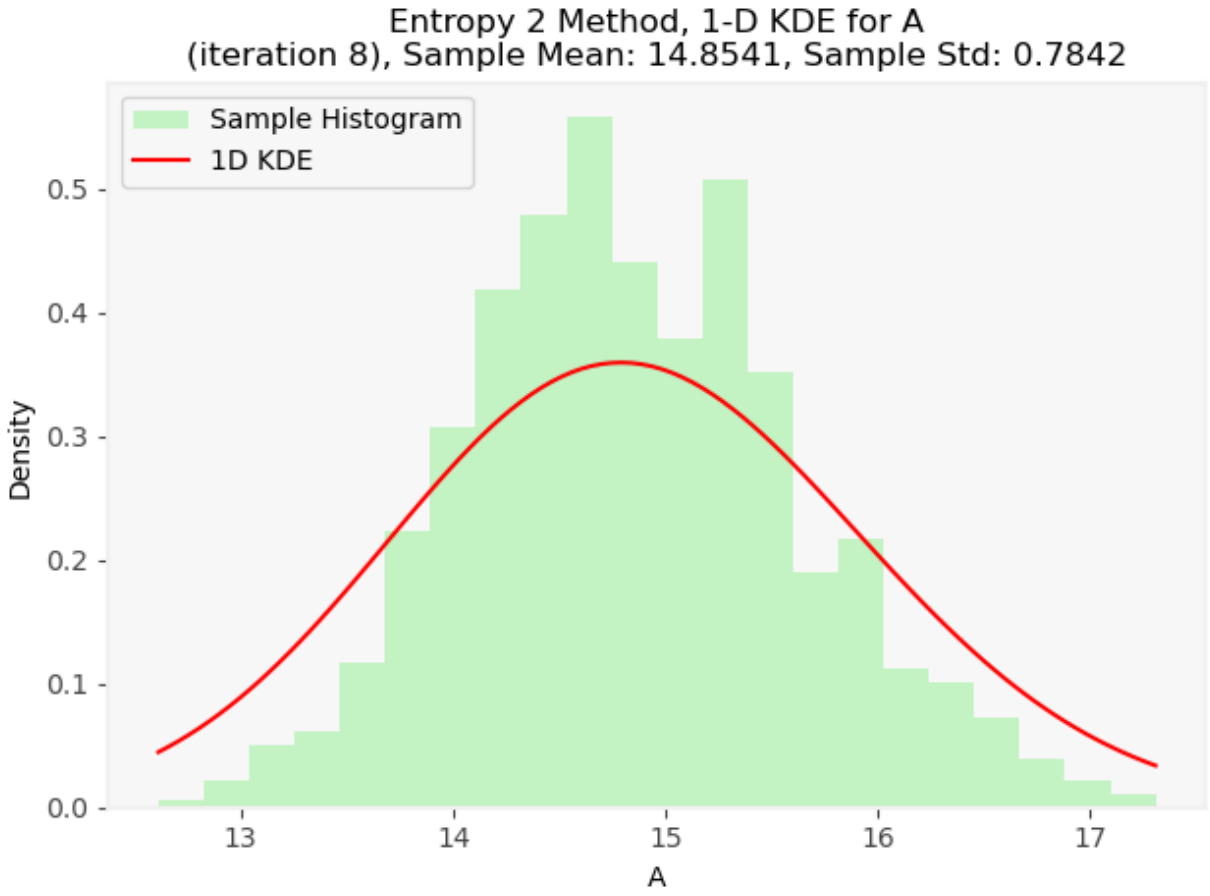

Entropy 2 Method, 1-D KDE for A
(iteration 3), Sample Mean: 17.3609, Sample Std: 4.5171
Sample Histogram
1D KDE
Density
0.00
0.02
0.04
0.06
0.08
10
15
20
25
30
A

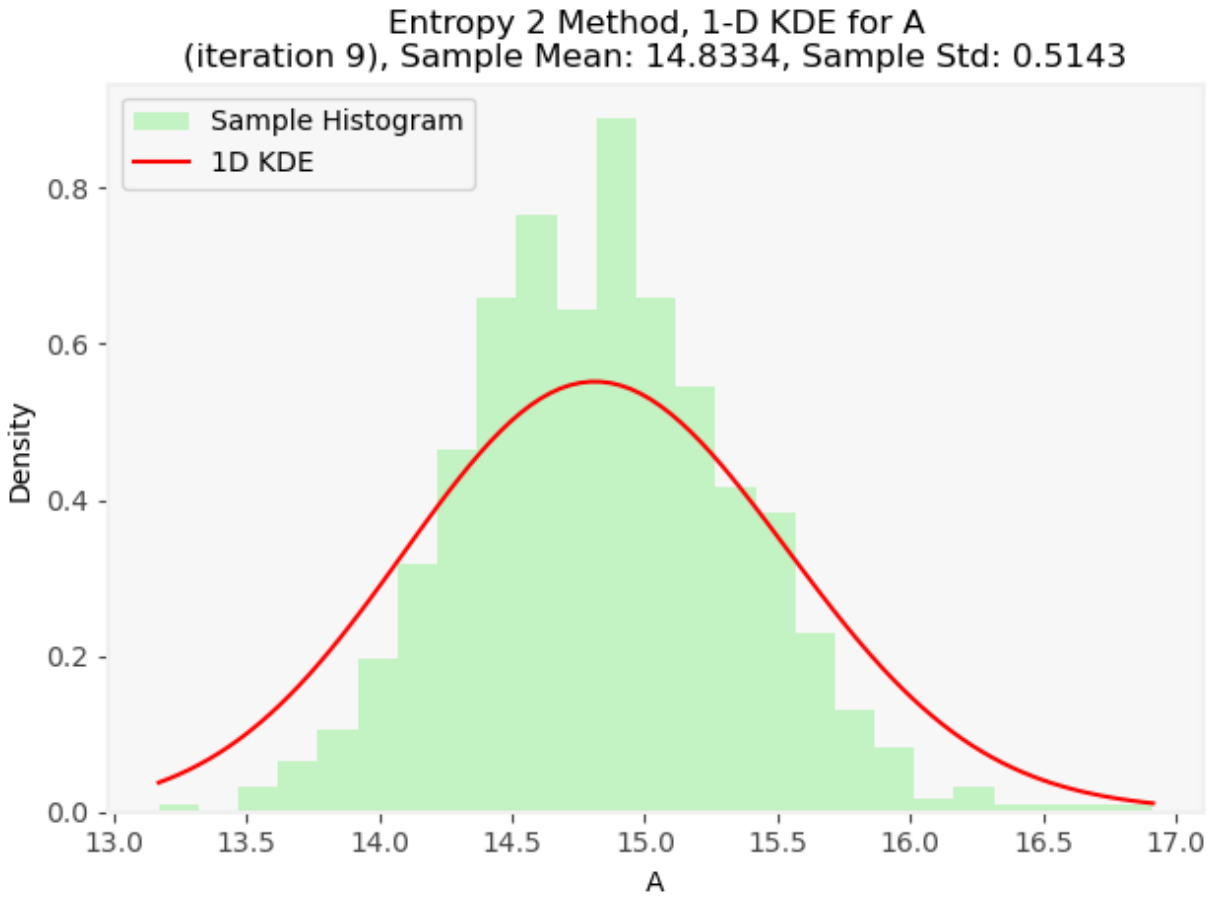

Entropy 2 Method, 1-D KDE for A
(iteration 4), Sample Mean: 16.2396, Sample Std: 2.6104
Sample Histogram
1D KDE
Density
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
10.0
12.5
15.0
17.5
20.0
22.5
25.0
A

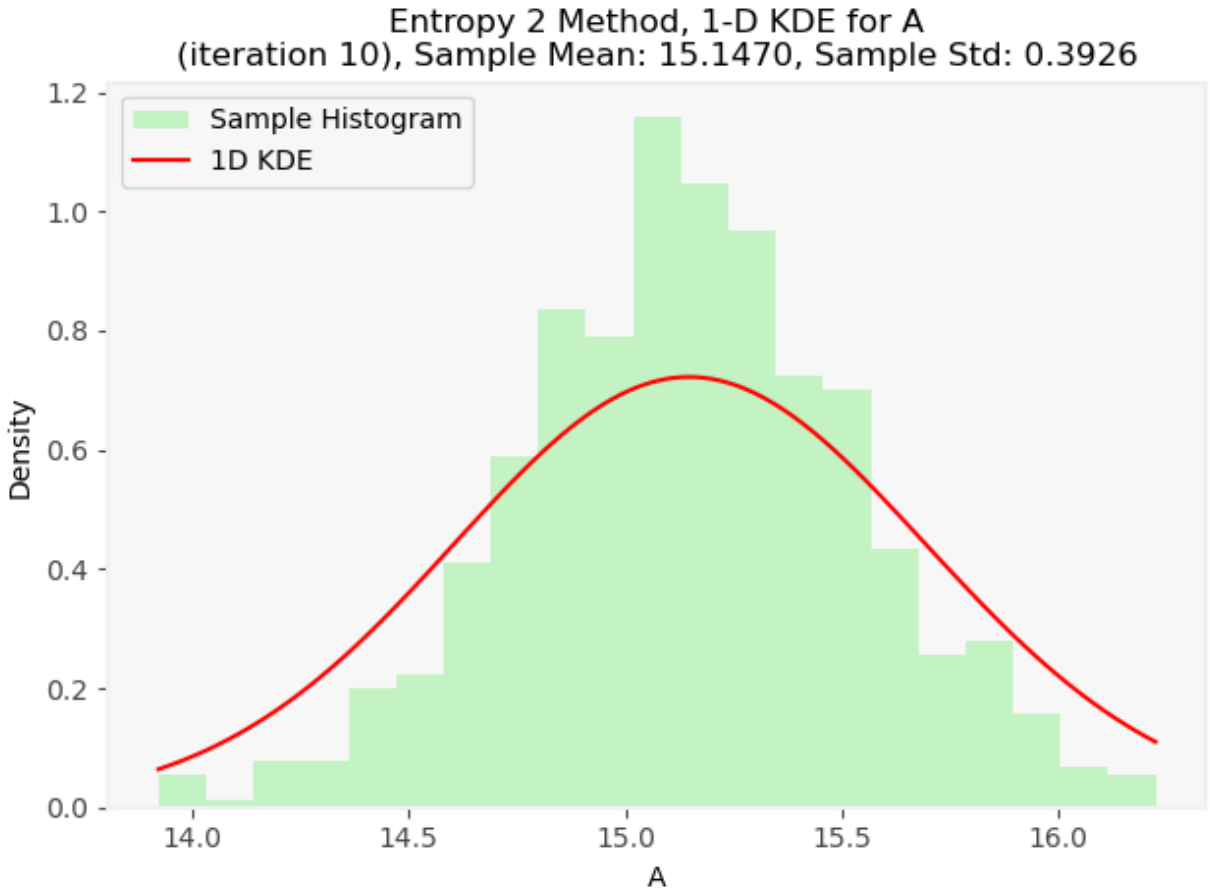

Entropy 2 Method, 1-D KDE for A
(iteration 5), Sample Mean: 15.7371, Sample Std: 2.2733
Sample Histogram
1D KDE
Density
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
10
12
14
16
18
20
22
A

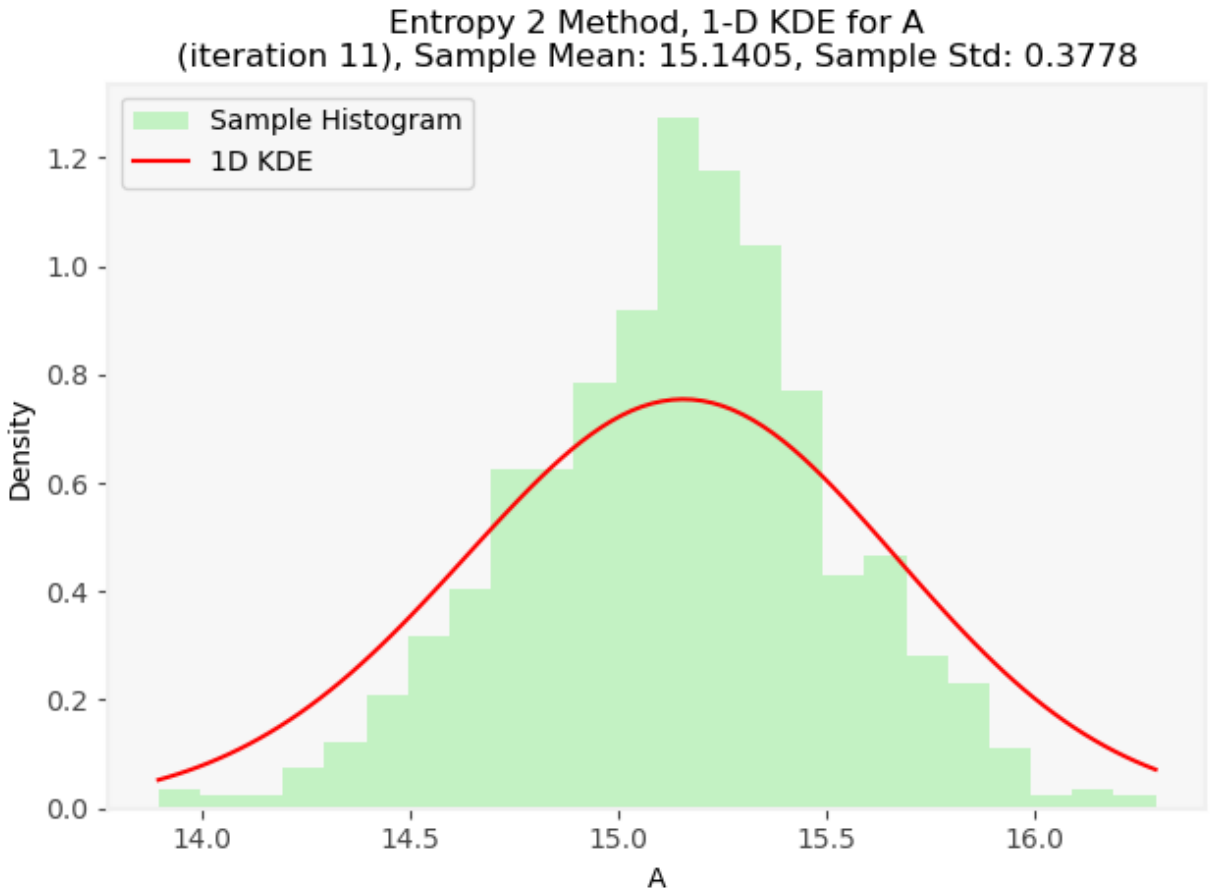

Entropy 2 Method, 1-D KDE for A
(iteration 6), Sample Mean: 13.9782, Sample Std: 2.0567
Sample Histogram
1D KDE
Density
A
0.25
0.20
0.15
0.10
0.05
0.00
10
12
14
16
18
20
22
24
26

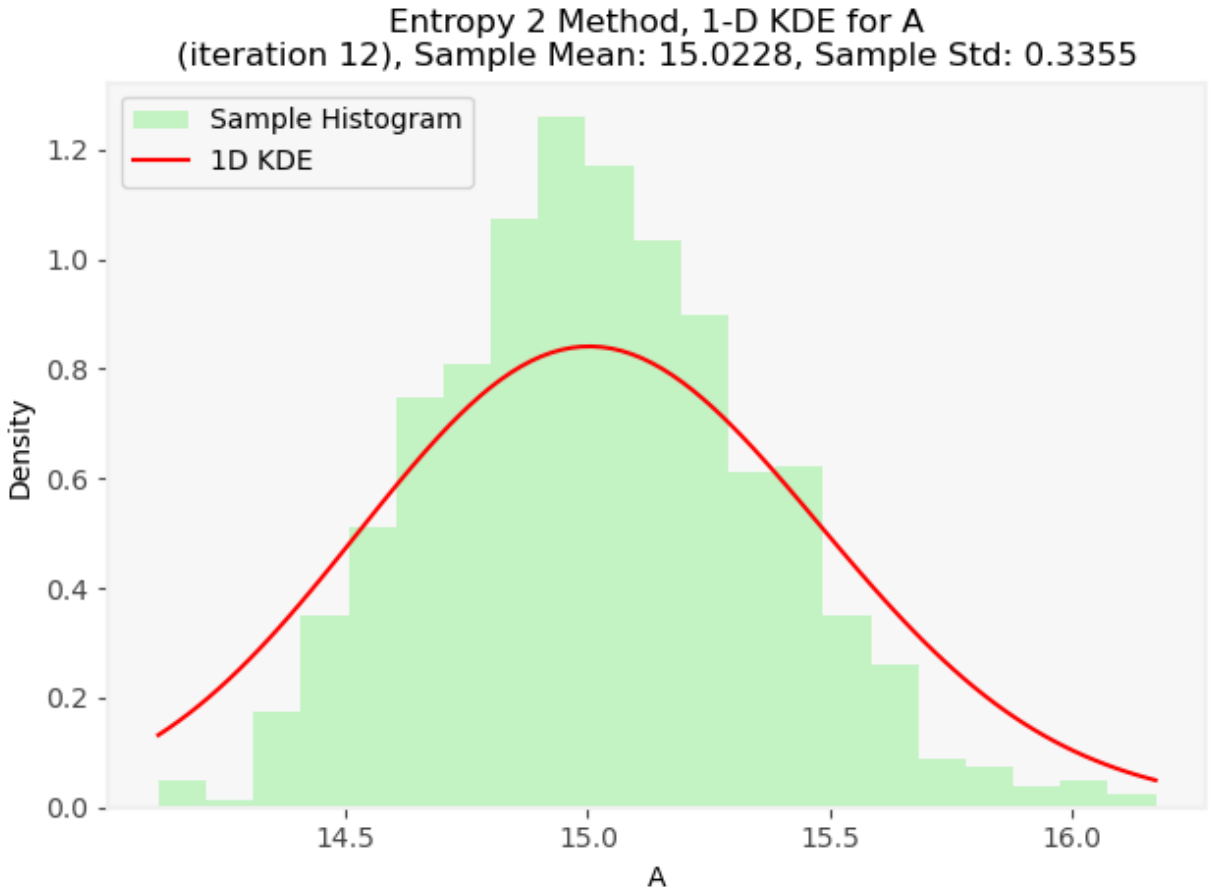

Entropy 2 Method, 1-D KDE for A
(iteration 7), Sample Mean: 13.9255, Sample Std: 1.1356
Sample Histogram
1D KDE
Density
A
0.4
0.3
0.2
0.1
0.0
12
13
14
15
16
17

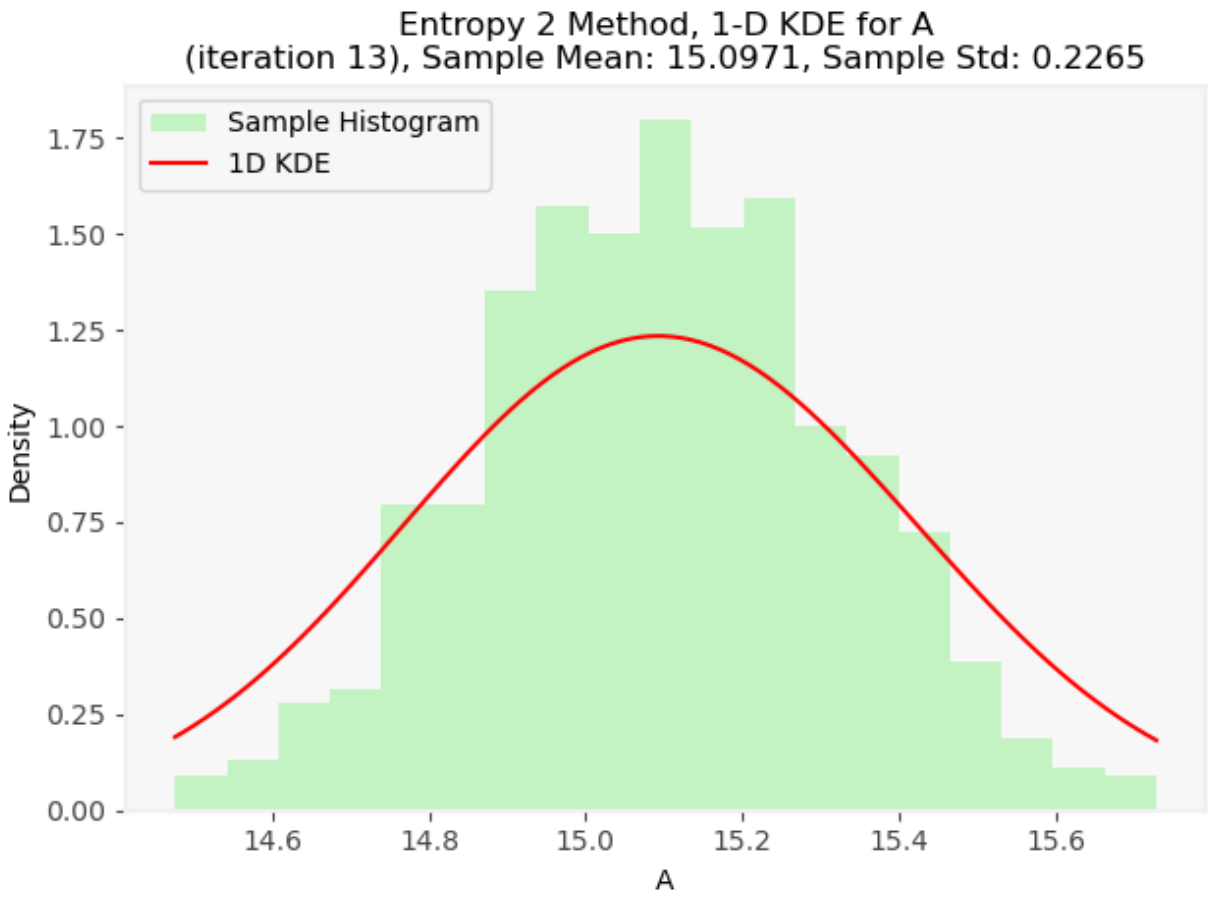

Entropy 2 Method, 1-D KDE for A
(iteration 8), Sample Mean: 14.8541, Sample Std: 0.7842
Sample Histogram
1D KDE
Density
0.0
0.1
0.2
0.3
0.4
0.5
13
14
15
16
17
A


Entropy 2 Method, 1-D KDE for A
(iteration 9), Sample Mean: 14.8334, Sample Std: 0.5143
Sample Histogram
1D KDE
Density
0.0
0.2
0.4
0.6
0.8
13.0
13.5
14.0
14.5
15.0
15.5
16.0
16.5
17.0
A


Entropy 2 Method, 1-D KDE for A
(iteration 10), Sample Mean: 15.1470, Sample Std: 0.3926
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
1.2
14.0
14.5
15.0
15.5
16.0

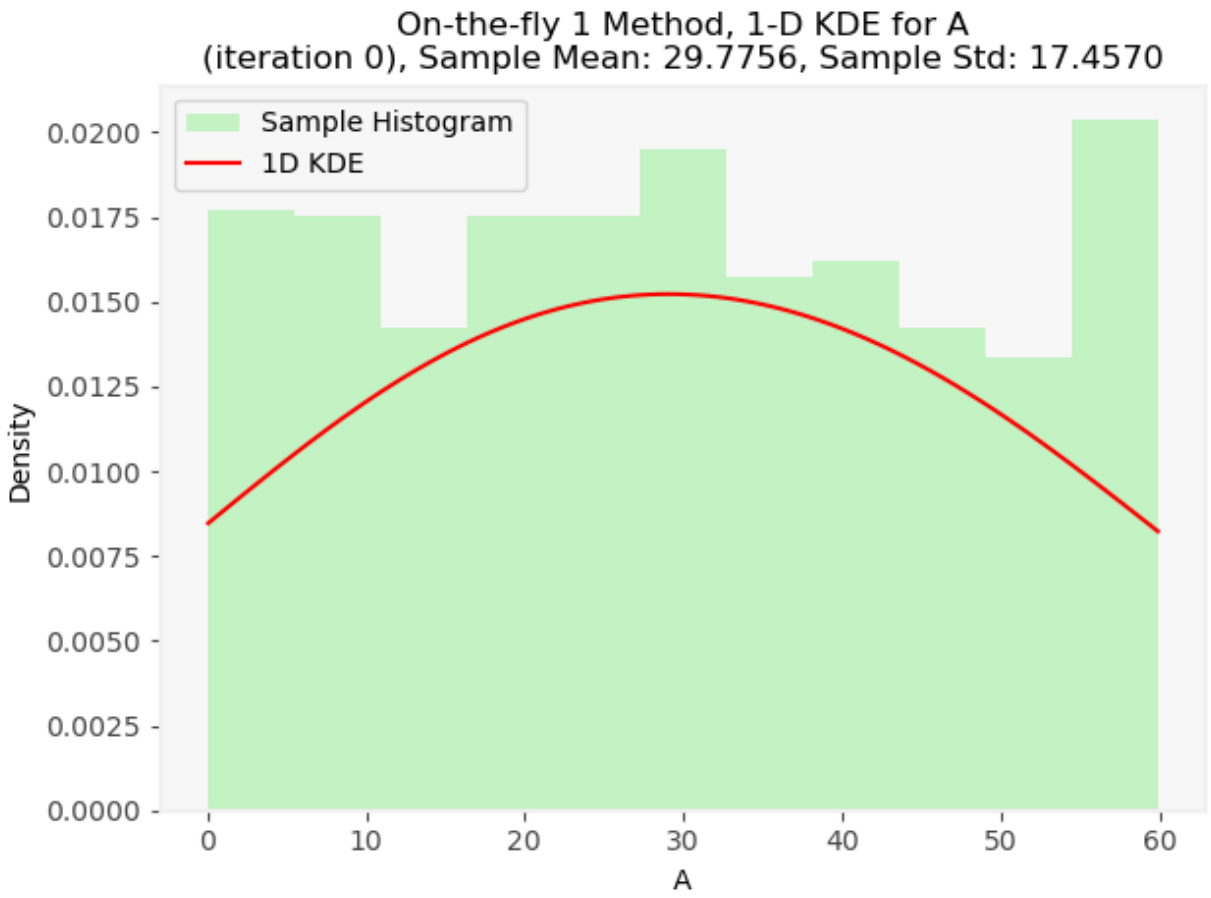

Entropy 2 Method, 1-D KDE for A
(iteration 11), Sample Mean: 15.1405, Sample Std: 0.3778
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
1.2
14.0
14.5
15.0
15.5
16.0

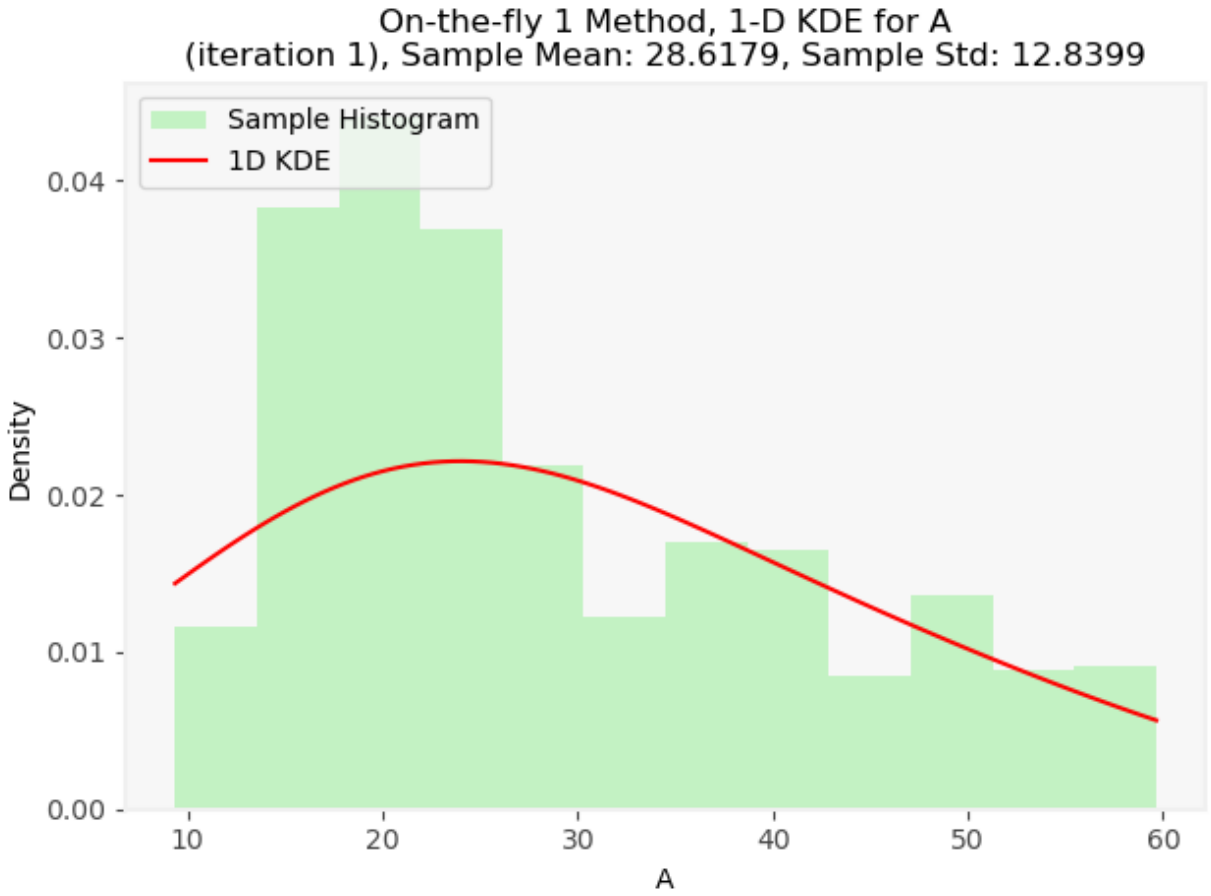


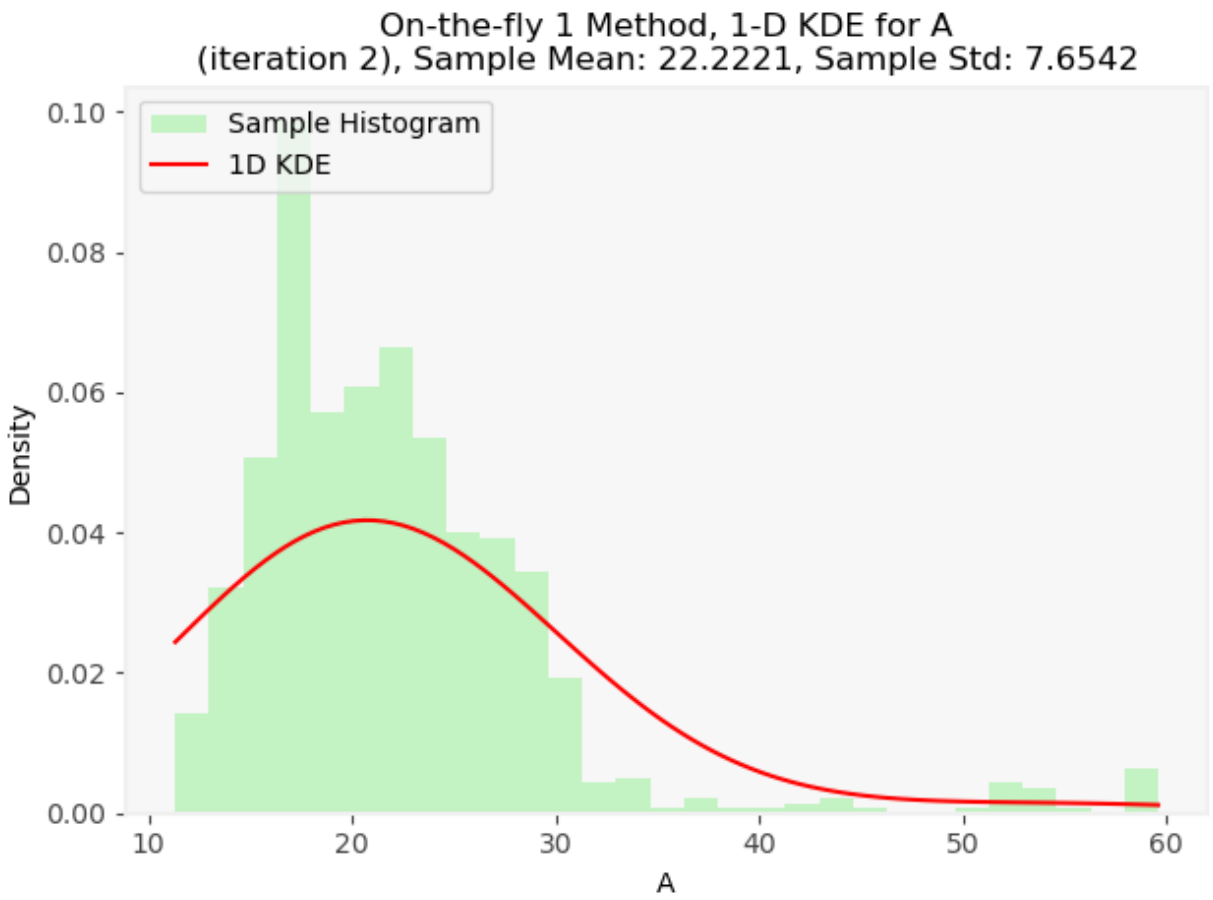


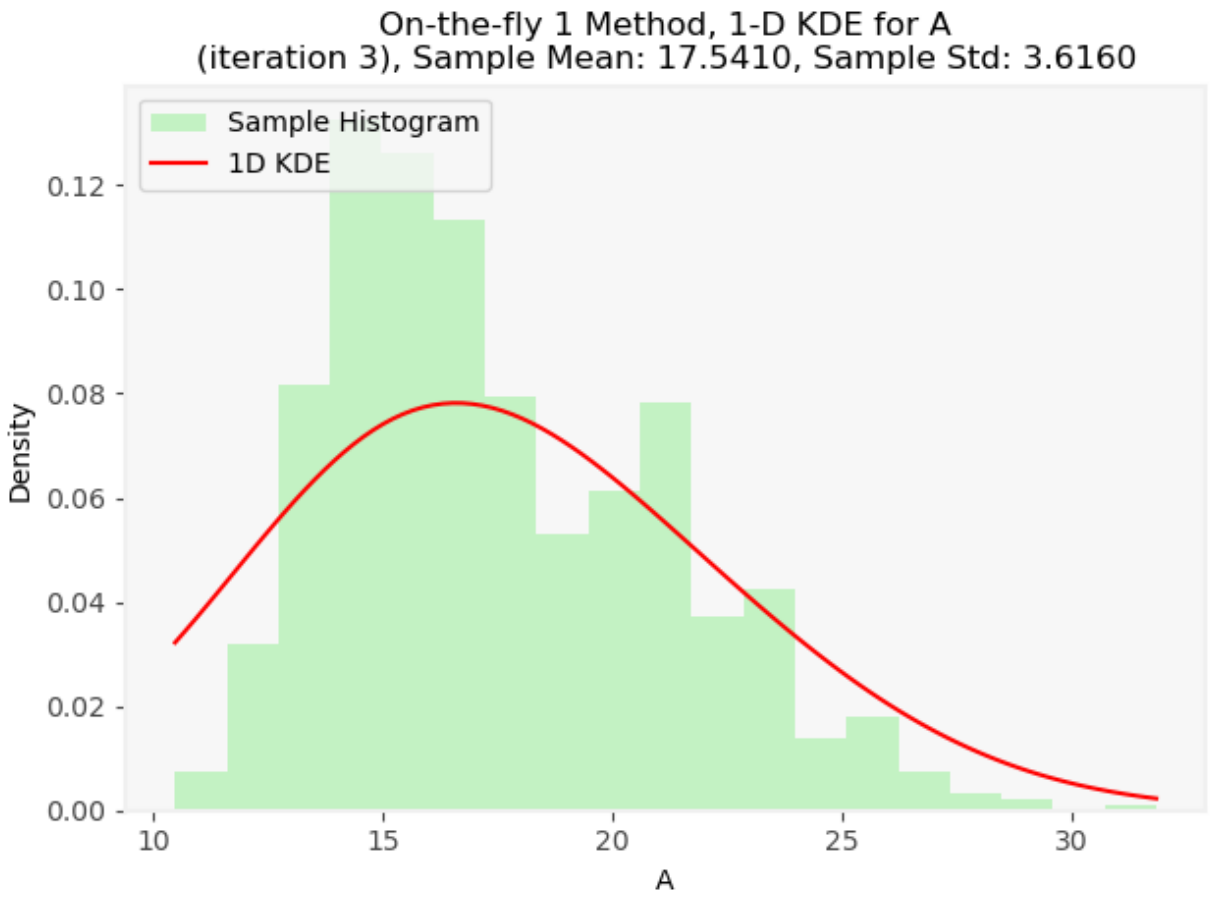
In [54]:
```
#Printing the evolution of parameter A for On-the-fly  1
for i in range(len(exp_on_the_fly1.totaltimes())):
```

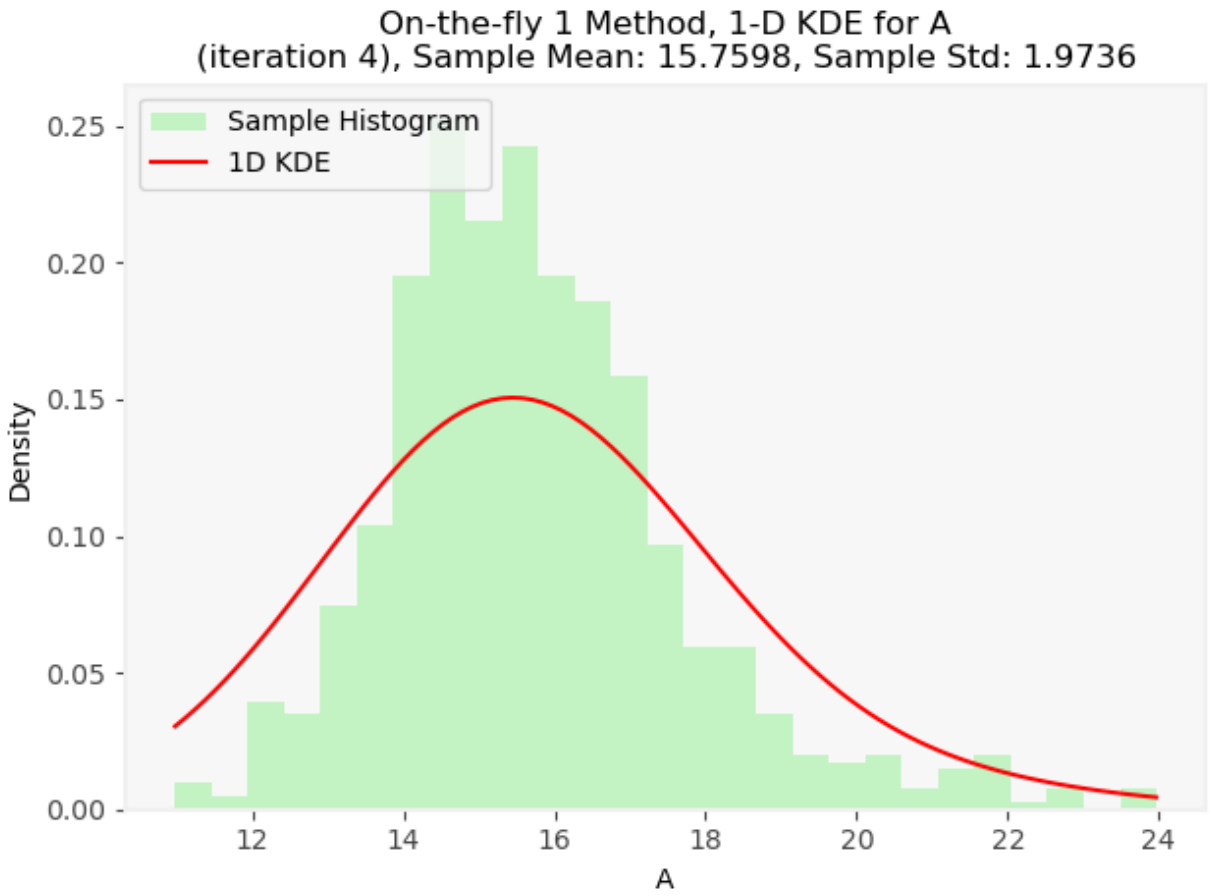
```
    MyPlots.plot_hist_1d_kde(list_par_separated_o1[0][i], kdes_on_the_fly1[i,0],"On-th
```

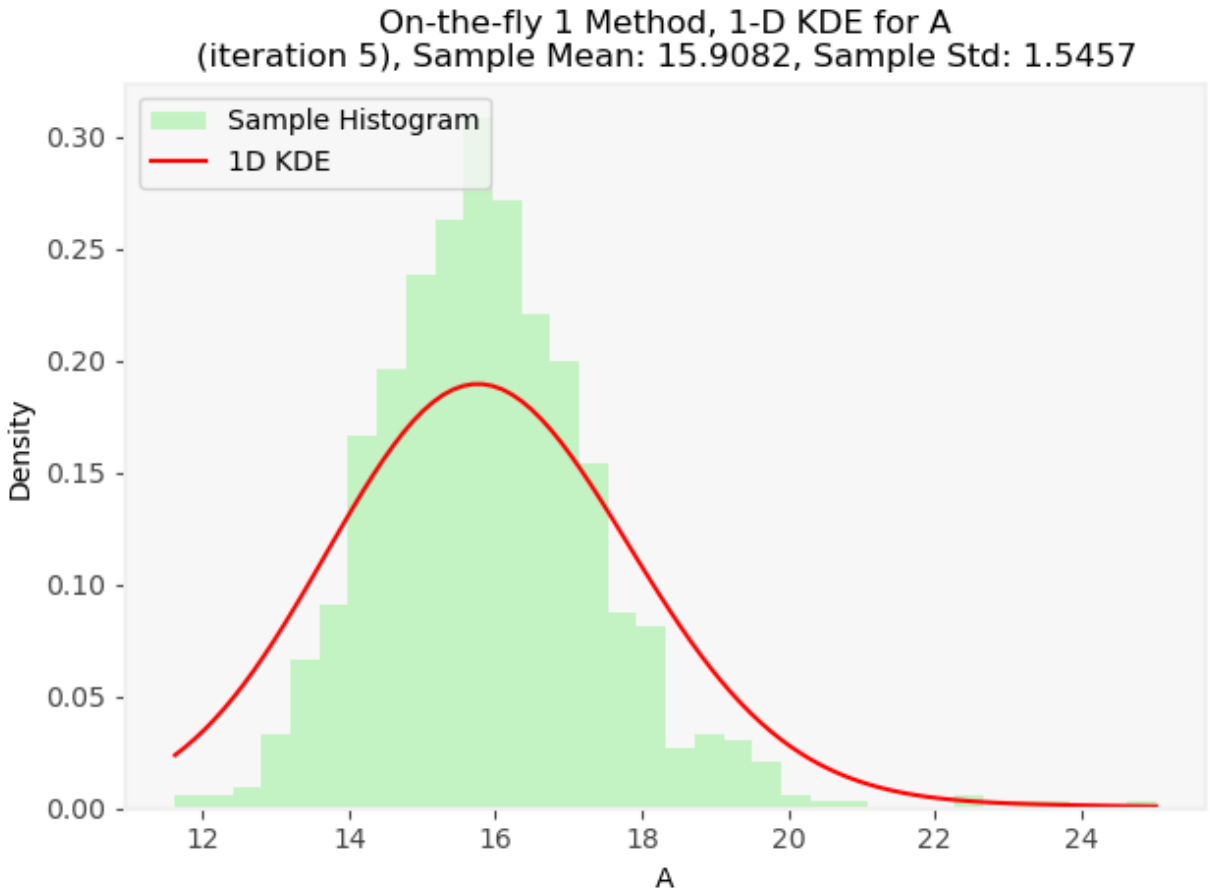


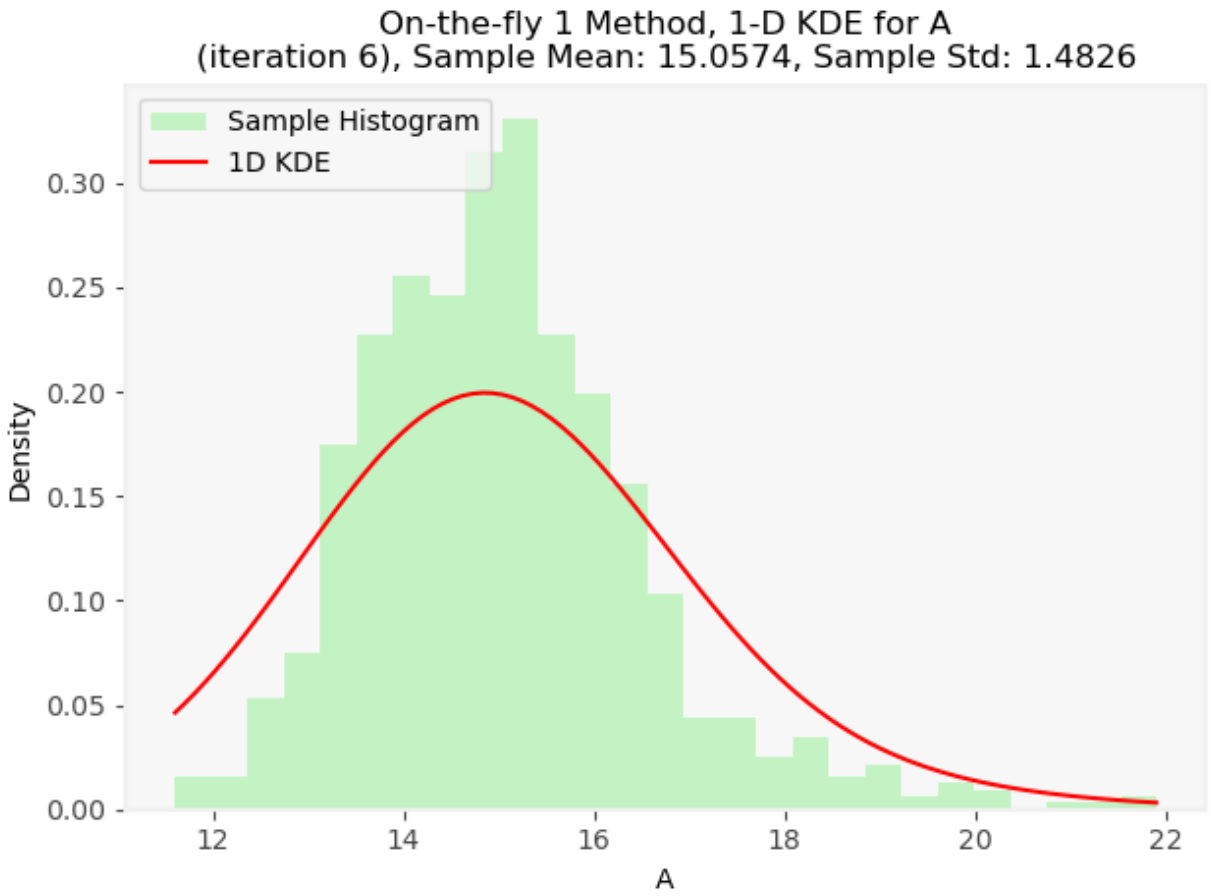


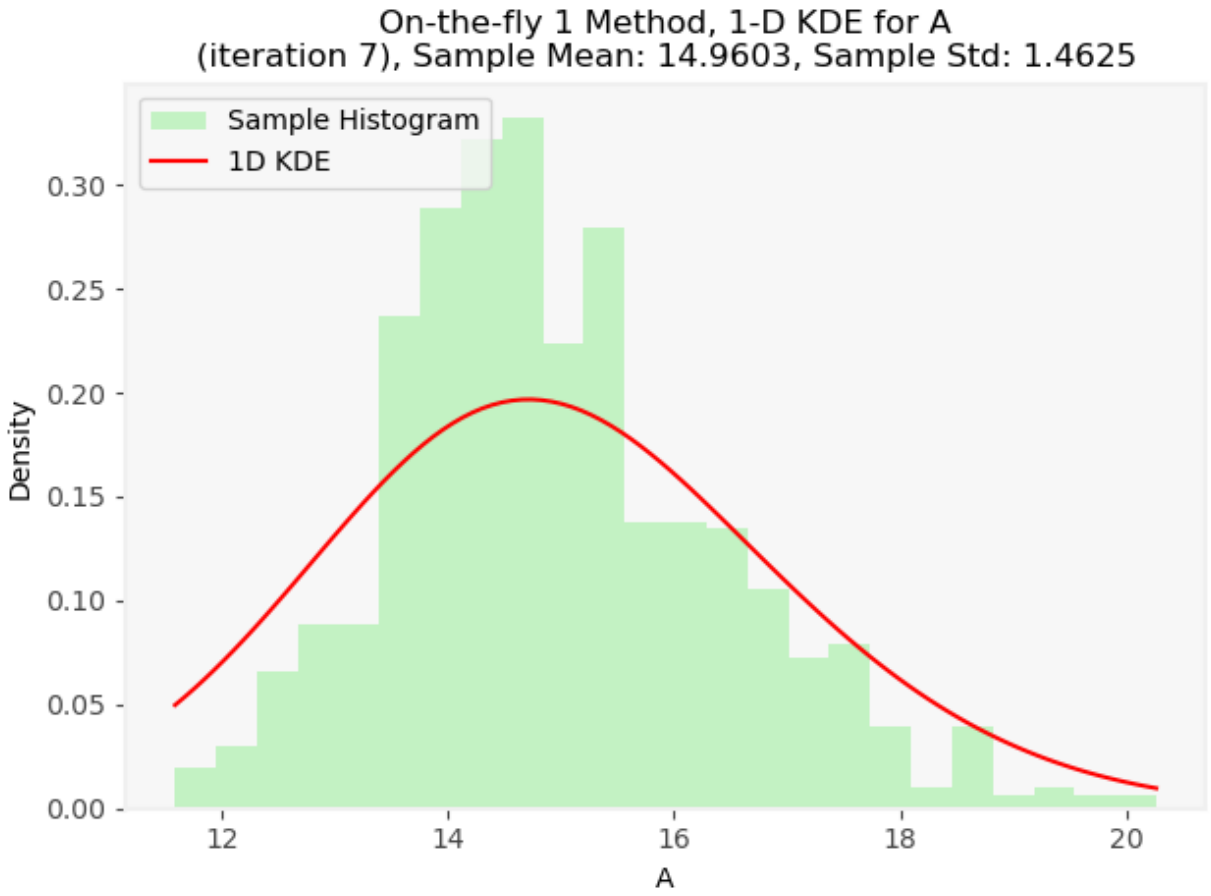

On-the-fly 1 Method, 1-D KDE for A
(iteration 2), Sample Mean: 22.2221, Sample Std: 7.6542
Sample Histogram
1D KDE
Density
0.10
0.08
0.06
0.04
0.02
0.00
10
20
30
40
50
60
A

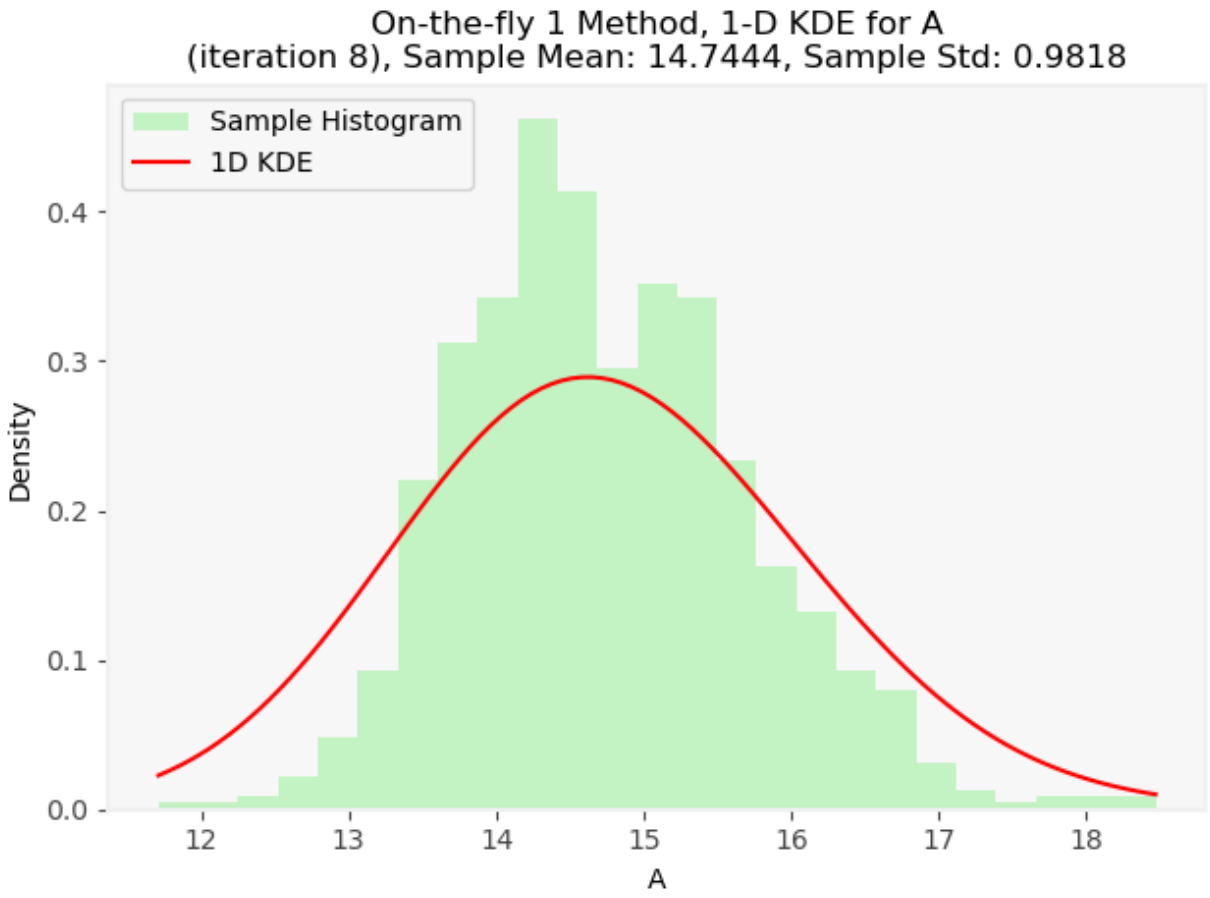

On-the-fly 1 Method, 1-D KDE for A
(iteration 3), Sample Mean: 17.5410, Sample Std: 3.6160
Sample Histogram
1D KDE
Density
0.12
0.10
0.08
0.06
0.04
0.02
0.00
10
15
20
25
30
A

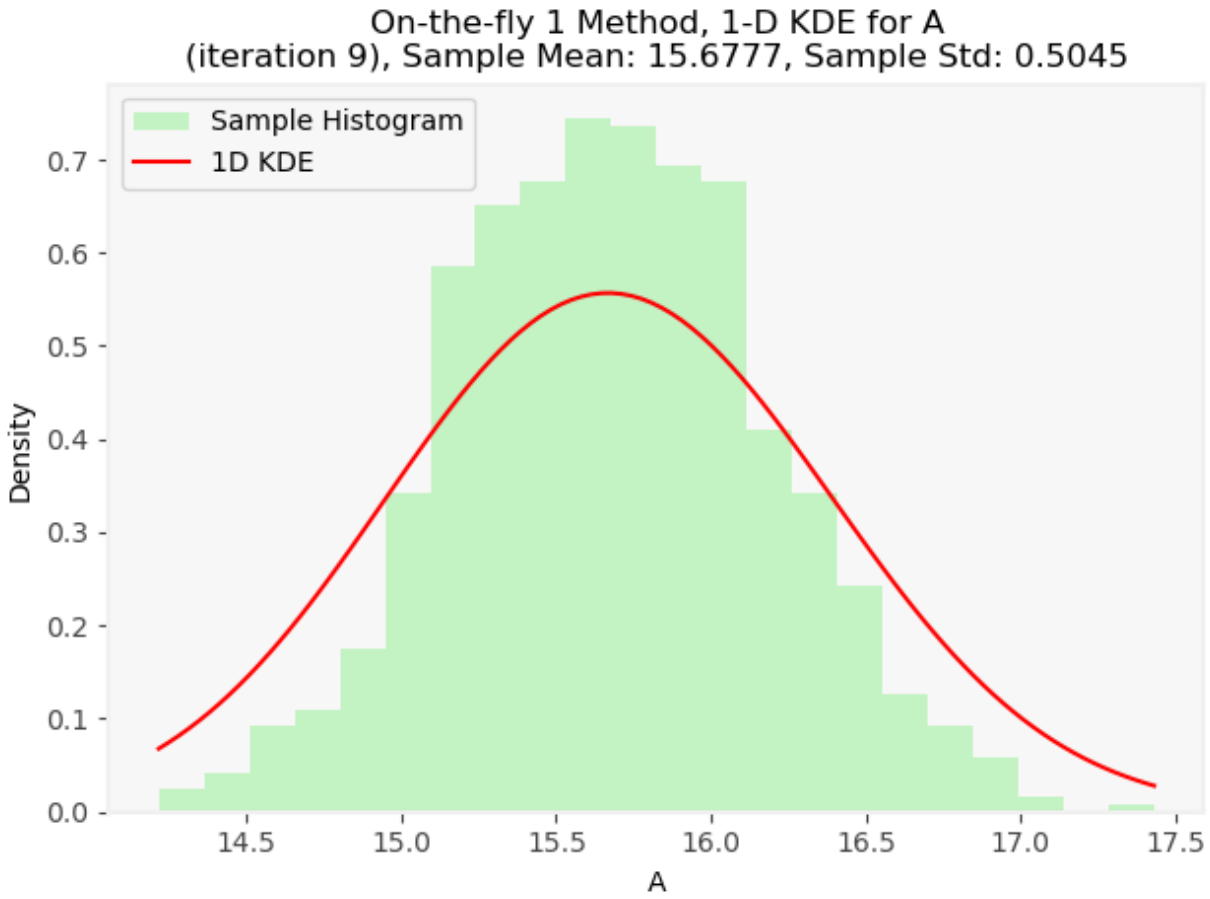

On-the-fly 1 Method, 1-D KDE for A
(iteration 4), Sample Mean: 15.7598, Sample Std: 1.9736
Sample Histogram
1D KDE
Density
0.25
0.20
0.15
0.10
0.05
0.00
12
14
16
18
20
22
24
A

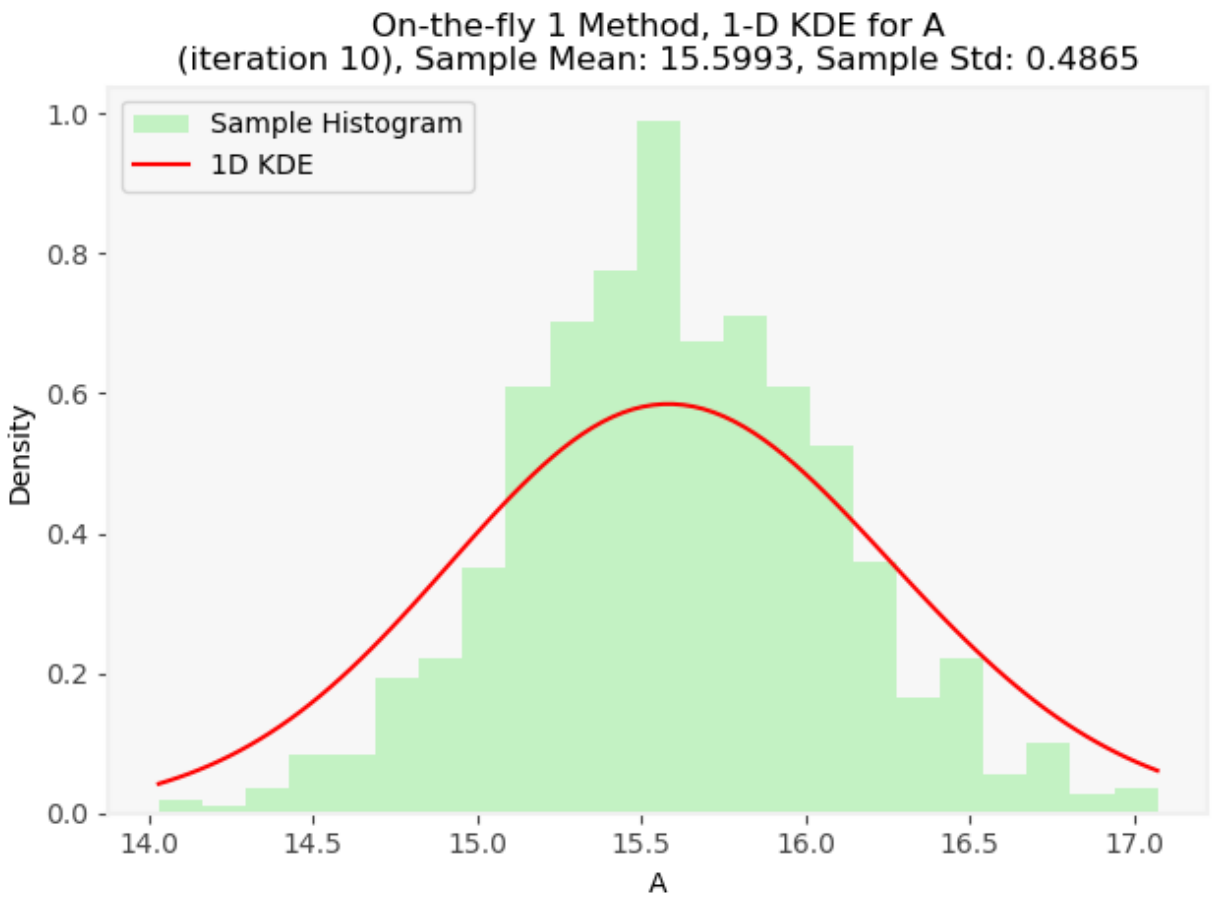

On-the-fly 1 Method, 1-D KDE for A
(iteration 5), Sample Mean: 15.9082, Sample Std: 1.5457
Sample Histogram
1D KDE
Density
0.30
0.25
0.20
0.15
0.10
0.05
0.00
12
14
16
18
20
22
24
A

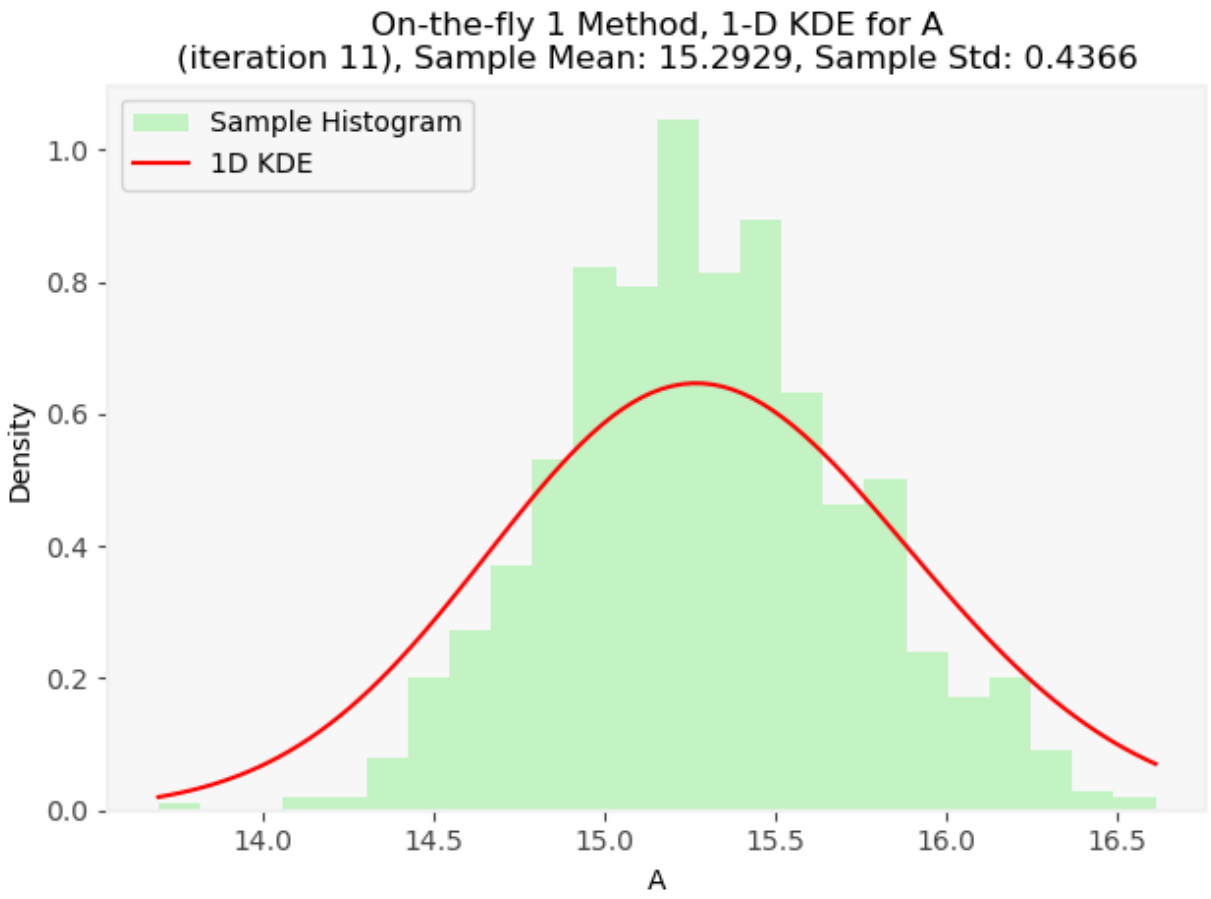

On-the-fly 1 Method, 1-D KDE for A
(iteration 6), Sample Mean: 15.0574, Sample Std: 1.4826
Sample Histogram
1D KDE
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
12
14
16
18
20
22
A

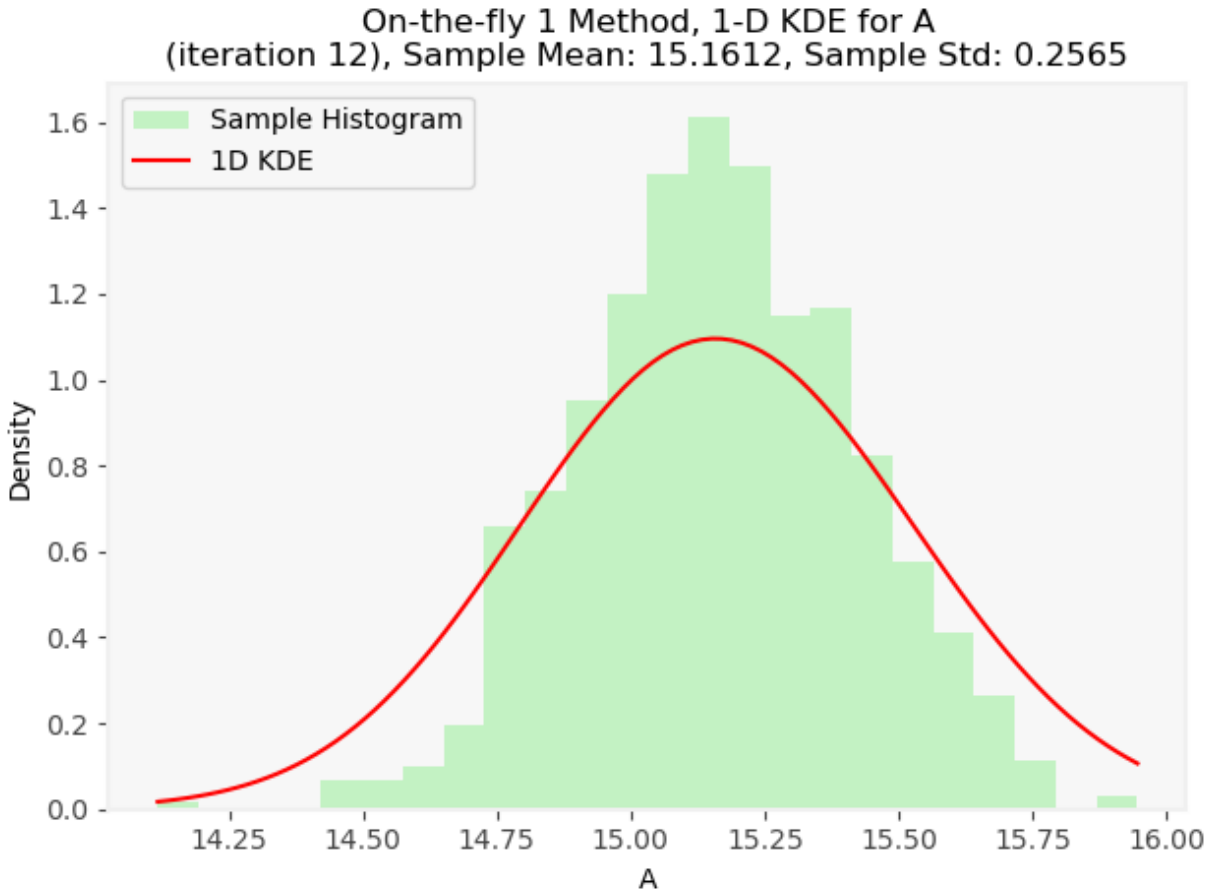

On-the-fly 1 Method, 1-D KDE for A
(iteration 7), Sample Mean: 14.9603, Sample Std: 1.4625
Sample Histogram
1D KDE
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
12
14
16
18
20
A

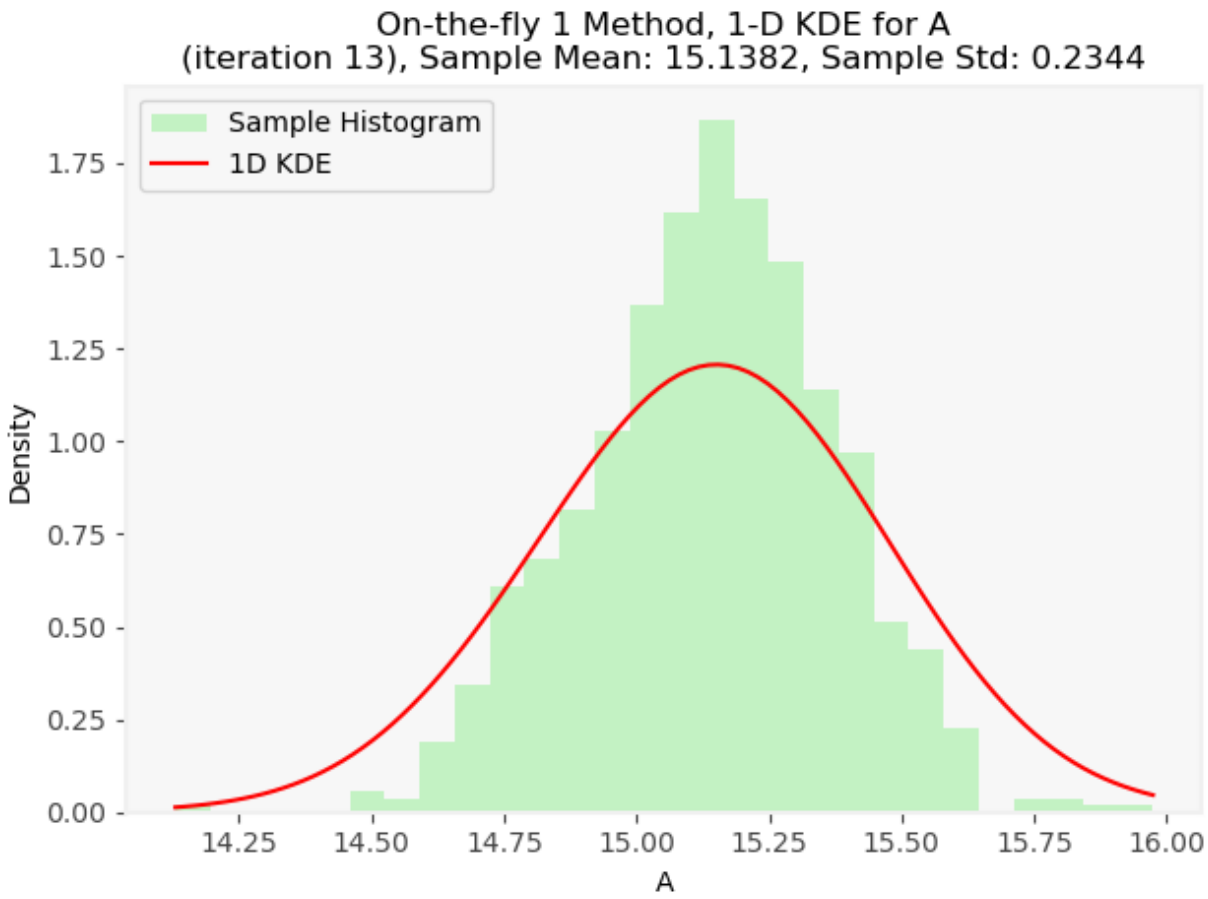

On-the-fly 1 Method, 1-D KDE for A
(iteration 8), Sample Mean: 14.7444, Sample Std: 0.9818
Sample Histogram
1D KDE
Density
0.0
0.1
0.2
0.3
0.4
12
13
14
15
16
17
18
A


On-the-fly 1 Method, 1-D KDE for A
(iteration 9), Sample Mean: 15.6777, Sample Std: 0.5045
Sample Histogram
1D KDE
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
14.5
15.0
15.5
16.0
16.5
17.0
17.5
A

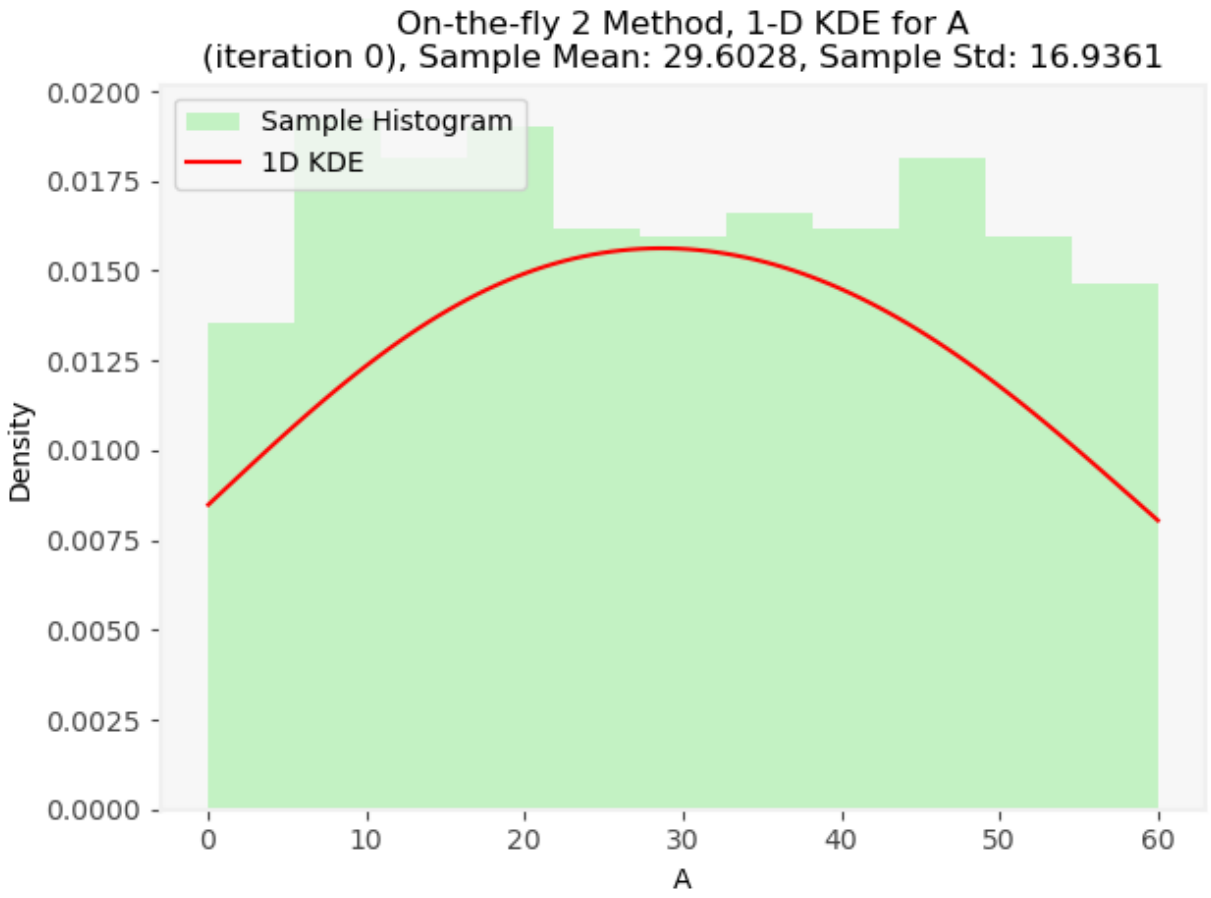

On-the-fly 1 Method, 1-D KDE for A
(iteration 10), Sample Mean: 15.5993, Sample Std: 0.4865
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
14.0
14.5
15.0
15.5
16.0
16.5
17.0

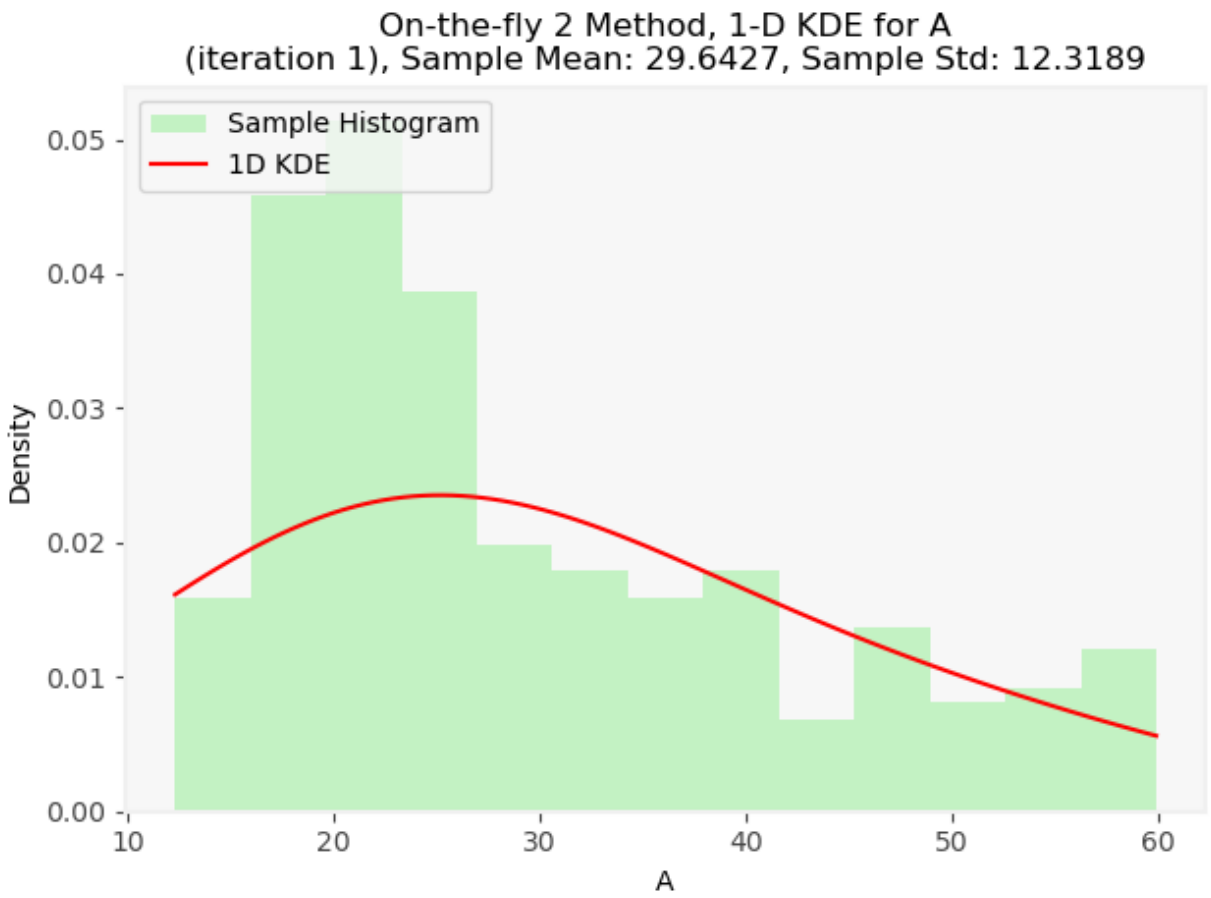

On-the-fly 1 Method, 1-D KDE for A
(iteration 11), Sample Mean: 15.2929, Sample Std: 0.4366
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
14.0
14.5
15.0
15.5
16.0
16.5

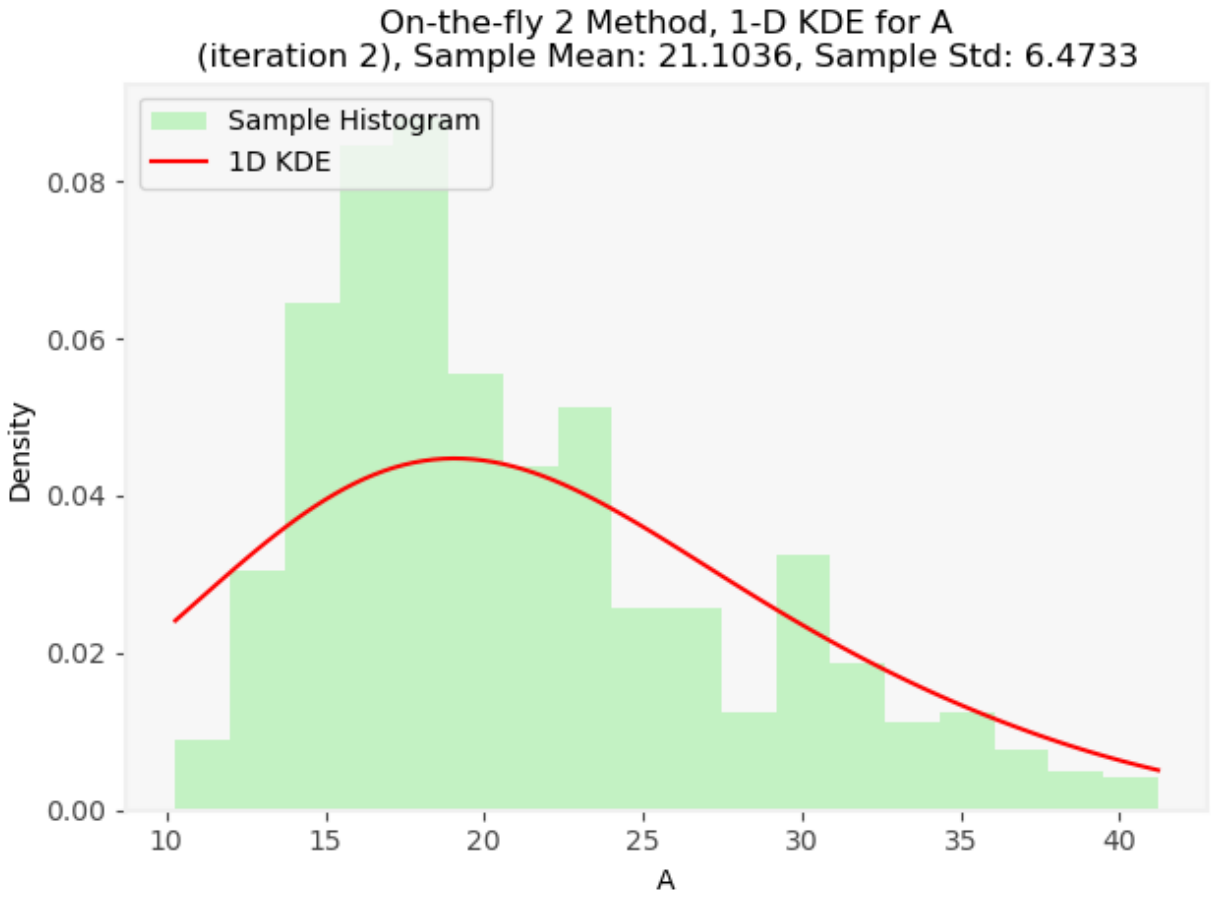

On-the-fly 1 Method, 1-D KDE for A
(iteration 12), Sample Mean: 15.1612, Sample Std: 0.2565
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
14.25
14.50
14.75
15.00
15.25
15.50
15.75
16.00

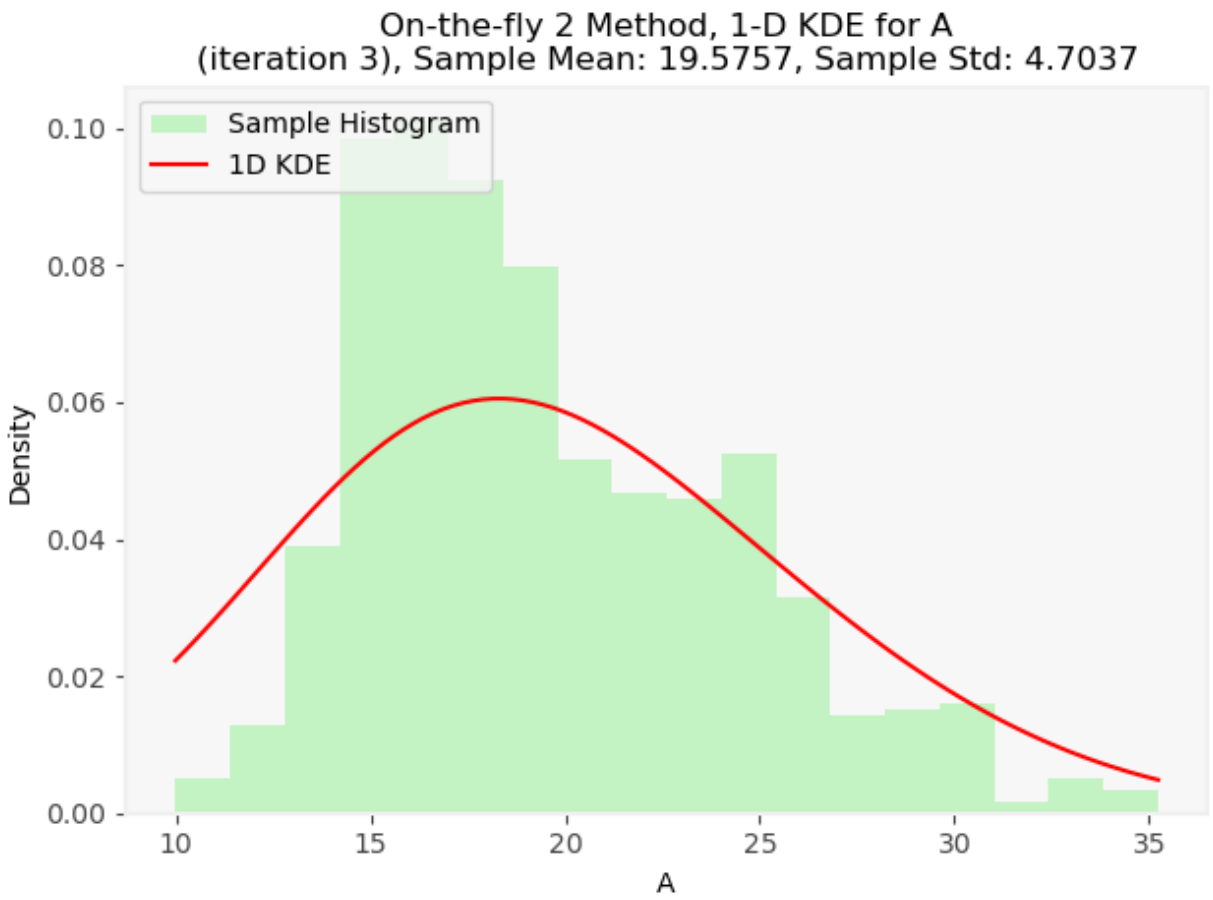

On-the-fly 1 Method, 1-D KDE for A
(iteration 13), Sample Mean: 15.1382, Sample Std: 0.2344
Sample Histogram
1D KDE
Density
A
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
14.25
14.50
14.75
15.00
15.25
15.50
15.75
16.00

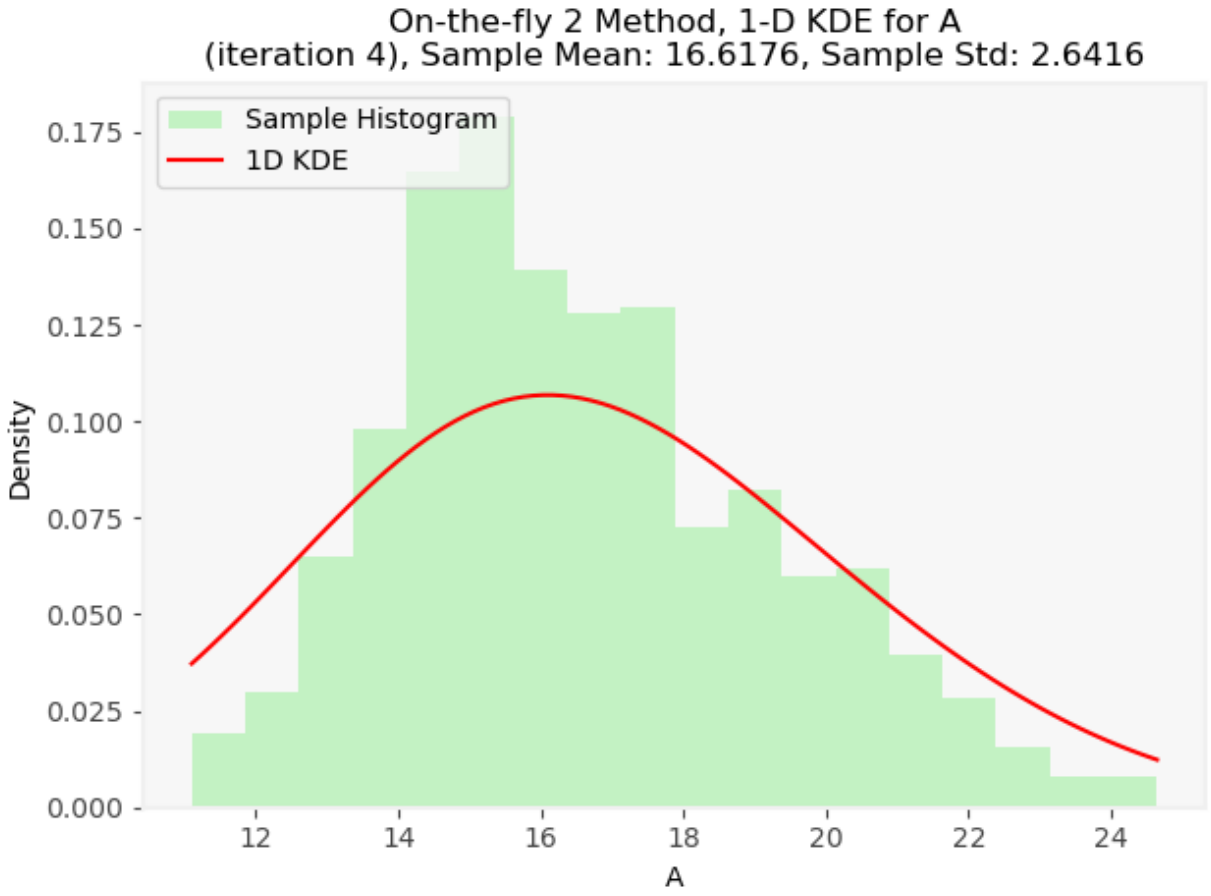
In [55]:
```
#Printing the evolution of parameter A for On-the-fly 2
for i in range(len(exp_on_the_fly2.totaltimes())):
    MyPlots.plot_hist_1d_kde(list_par_separated_o2[0][i], kdes_on_the_fly2[i,0],"On-th
```

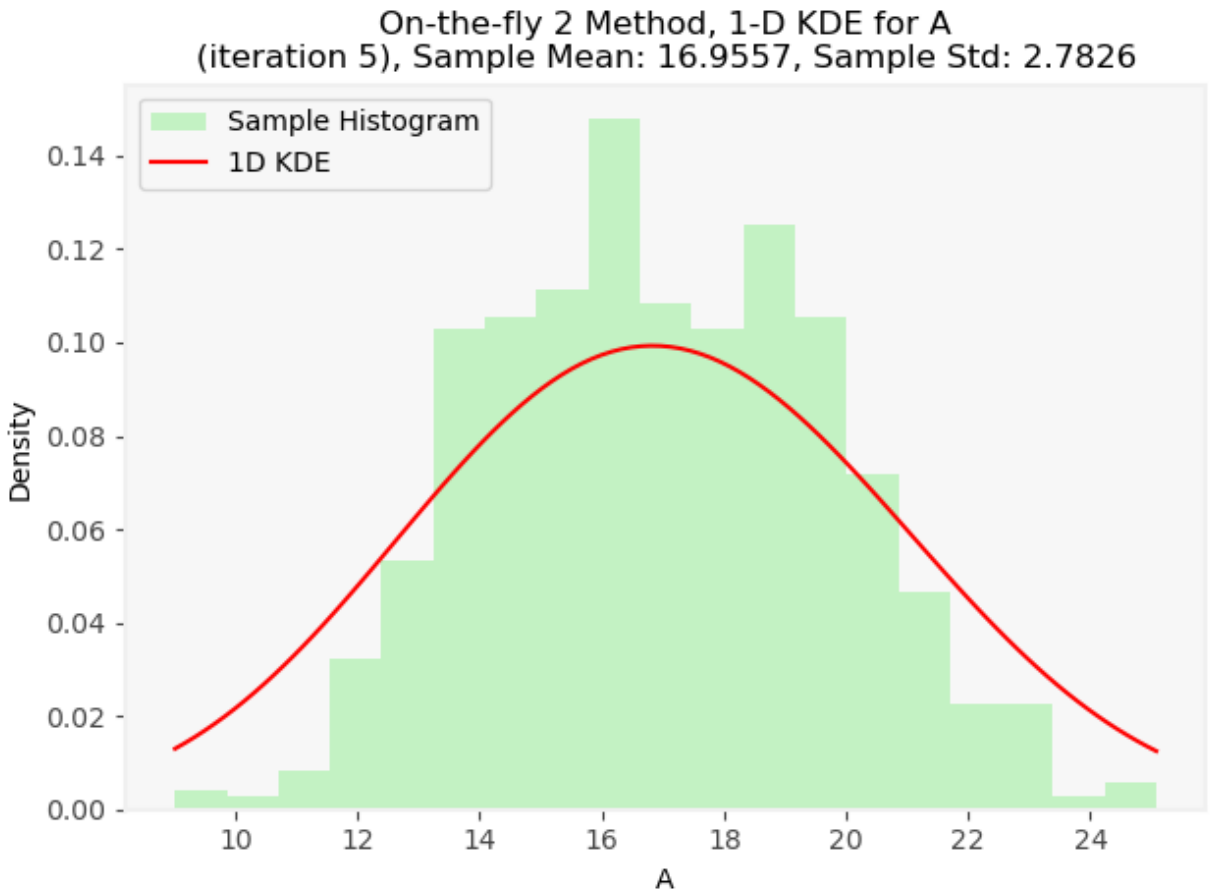


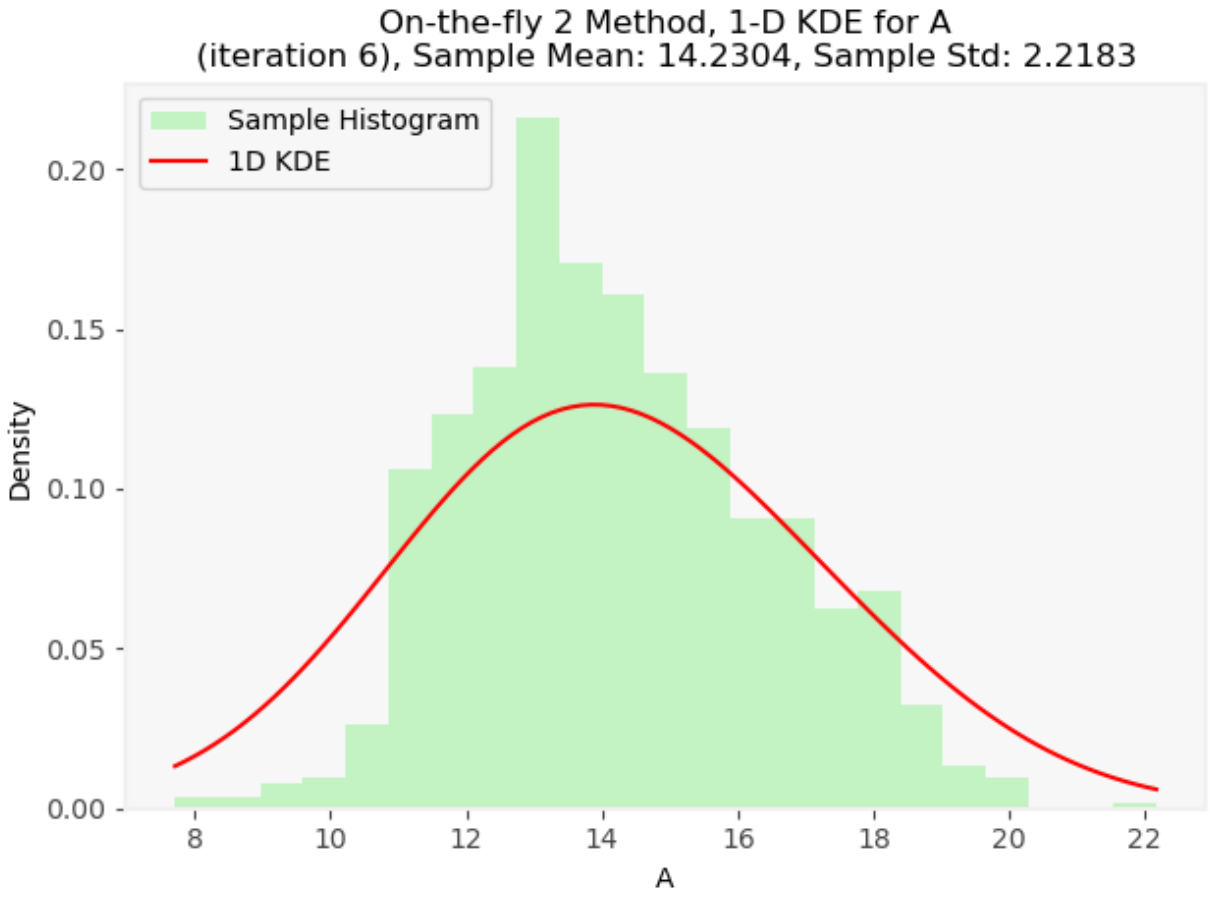

On-the-fly 2 Method, 1-D KDE for A
(iteration 1), Sample Mean: 29.6427, Sample Std: 12.3189
Sample Histogram
1D KDE
Density
0.00
0.01
0.02
0.03
0.04
0.05
10
20
30
40
50
60
A

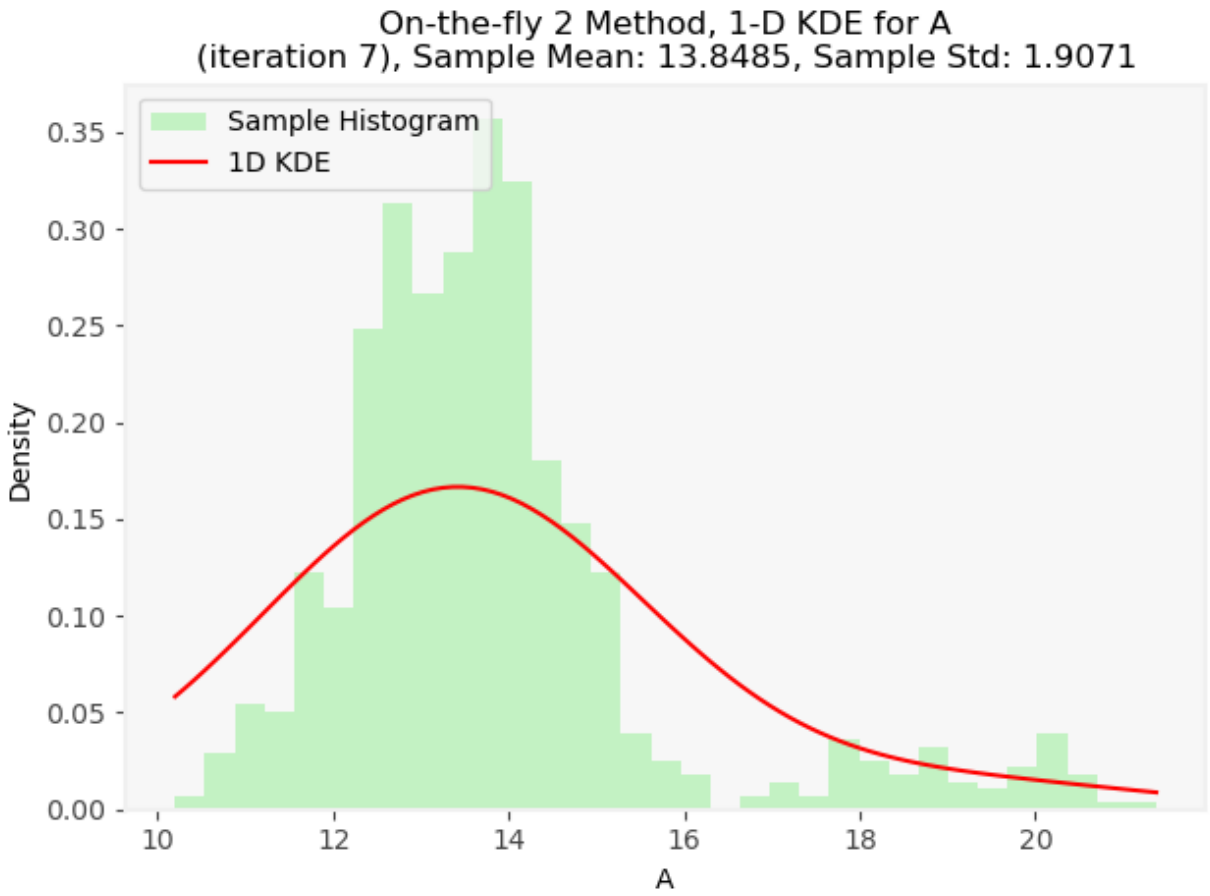

On-the-fly 2 Method, 1-D KDE for A
(iteration 2), Sample Mean: 21.1036, Sample Std: 6.4733
Sample Histogram
1D KDE
Density
0.00
0.02
0.04
0.06
0.08
10
15
20
25
30
35
40
A

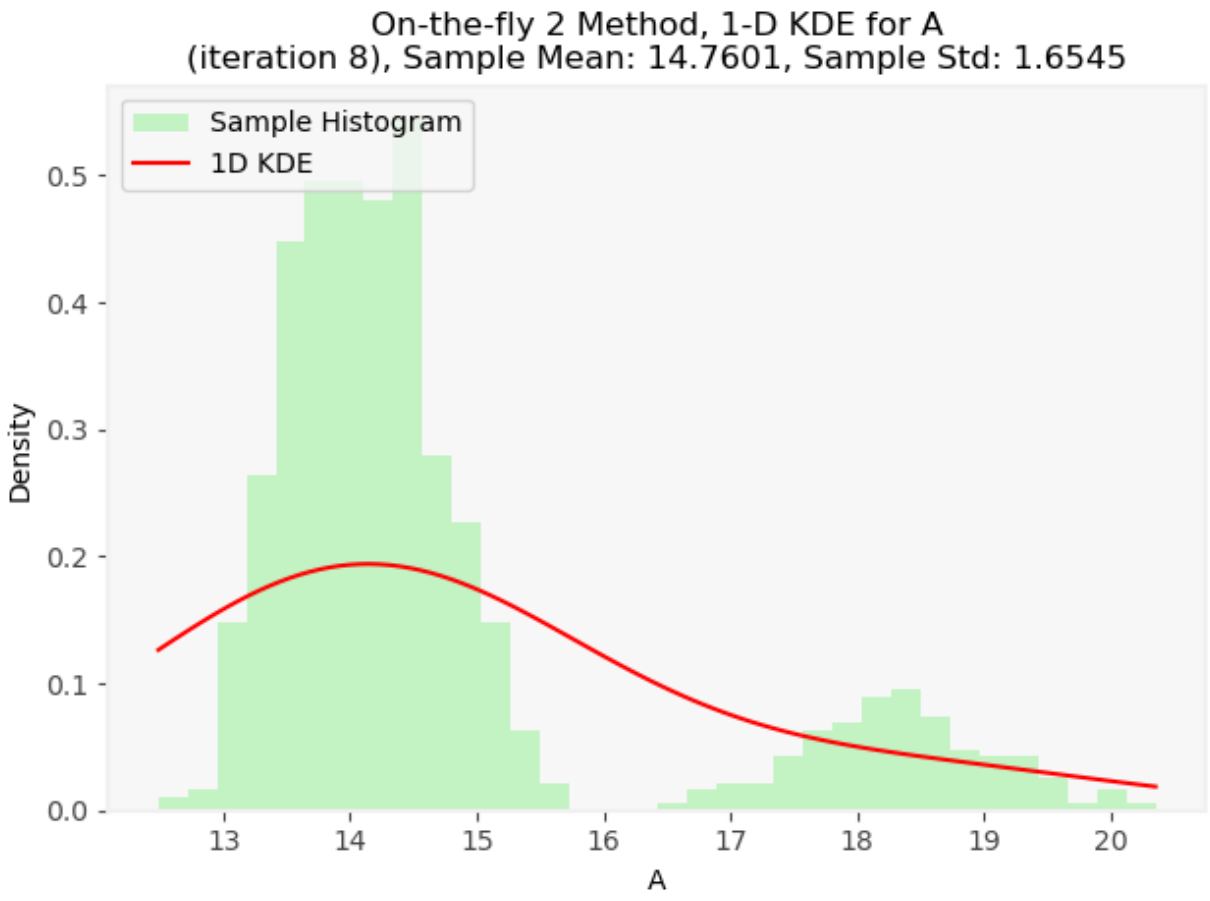

On-the-fly 2 Method, 1-D KDE for A
(iteration 3), Sample Mean: 19.5757, Sample Std: 4.7037
Sample Histogram
1D KDE
Density
0.10
0.08
0.06
0.04
0.02
0.00
10
15
20
25
30
35
A

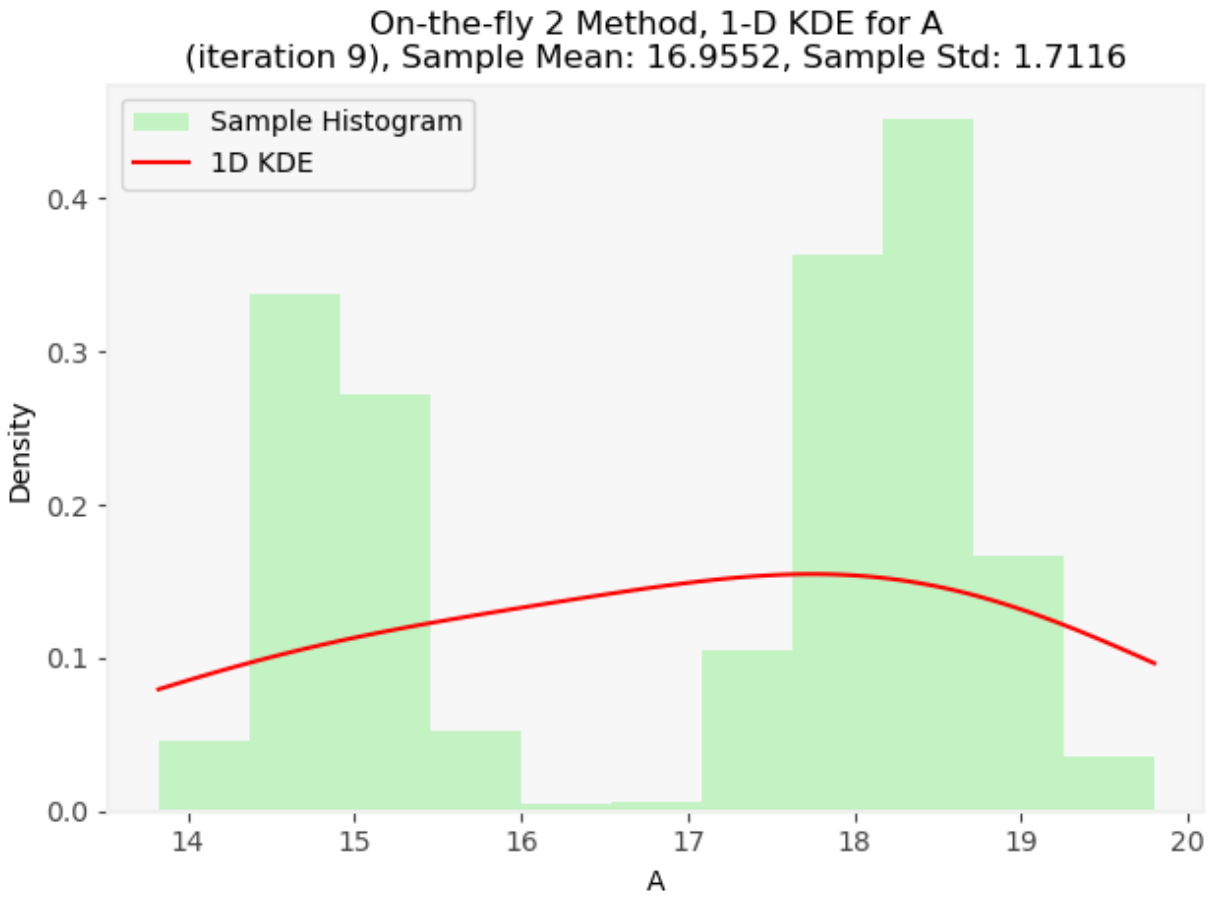

On-the-fly 2 Method, 1-D KDE for A
(iteration 4), Sample Mean: 16.6176, Sample Std: 2.6416
Sample Histogram
1D KDE
Density
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
12
14
16
18
20
22
24
A

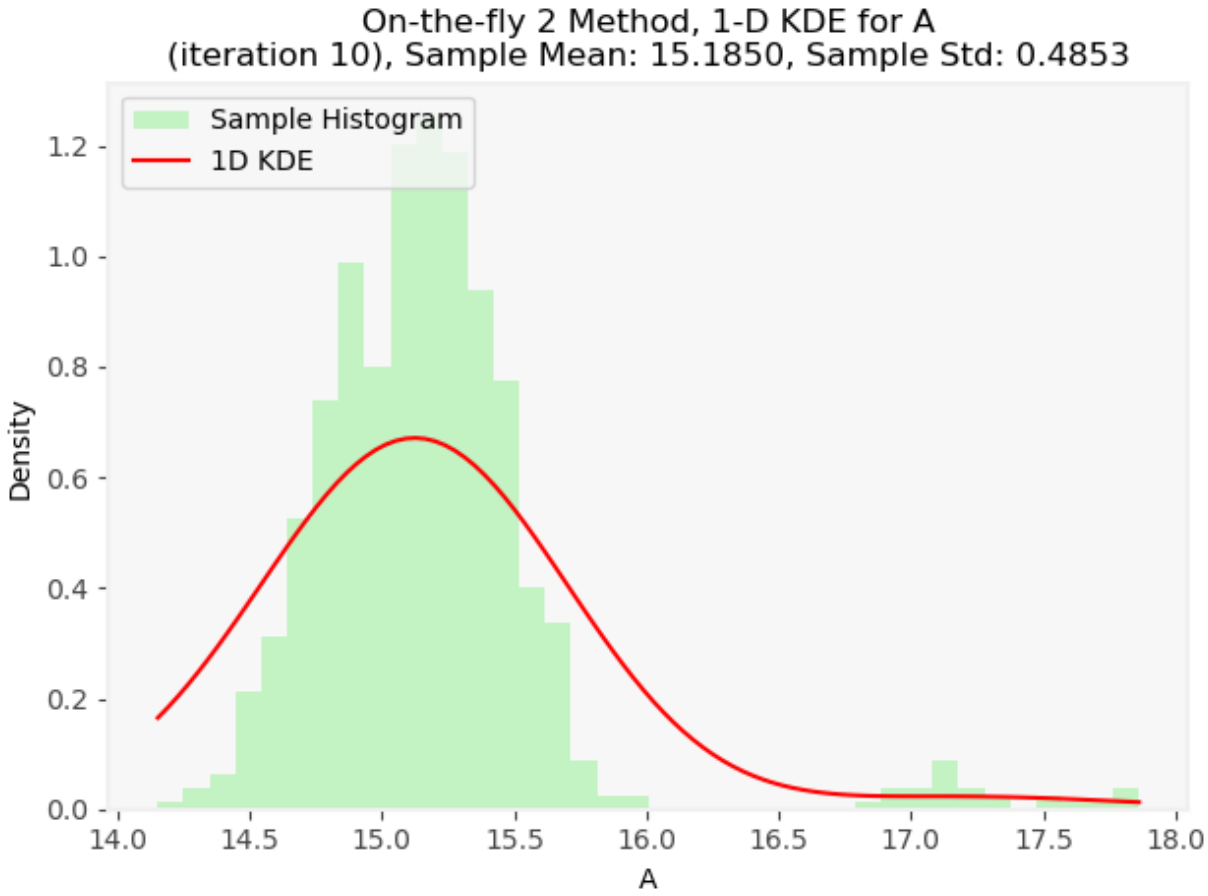

On-the-fly 2 Method, 1-D KDE for A
(iteration 5), Sample Mean: 16.9557, Sample Std: 2.7826
Sample Histogram
1D KDE
Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
10
12
14
16
18
20
22
24
A

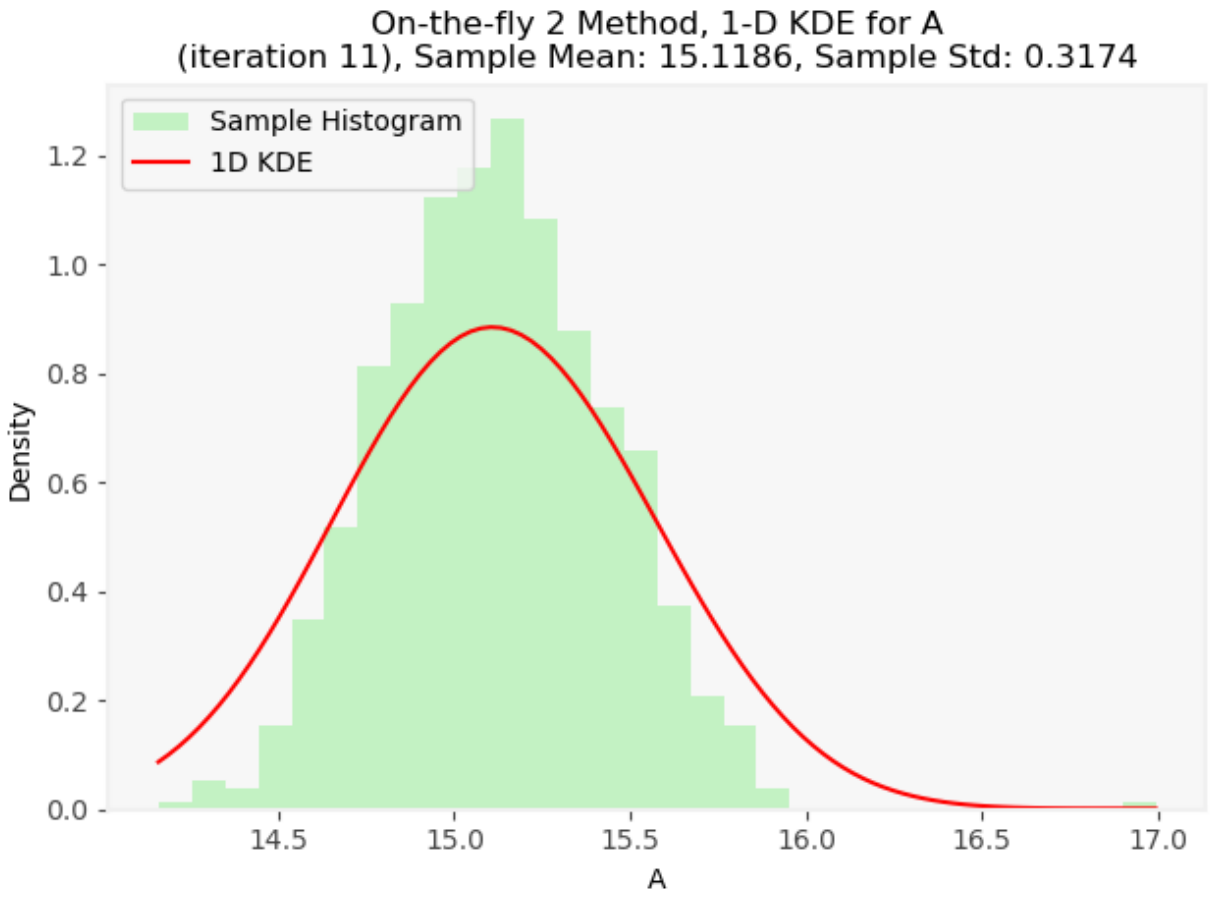

On-the-fly 2 Method, 1-D KDE for A
(iteration 6), Sample Mean: 14.2304, Sample Std: 2.2183
Sample Histogram
1D KDE
Density
0.00
0.05
0.10
0.15
0.20
8
10
12
14
16
18
20
22
A

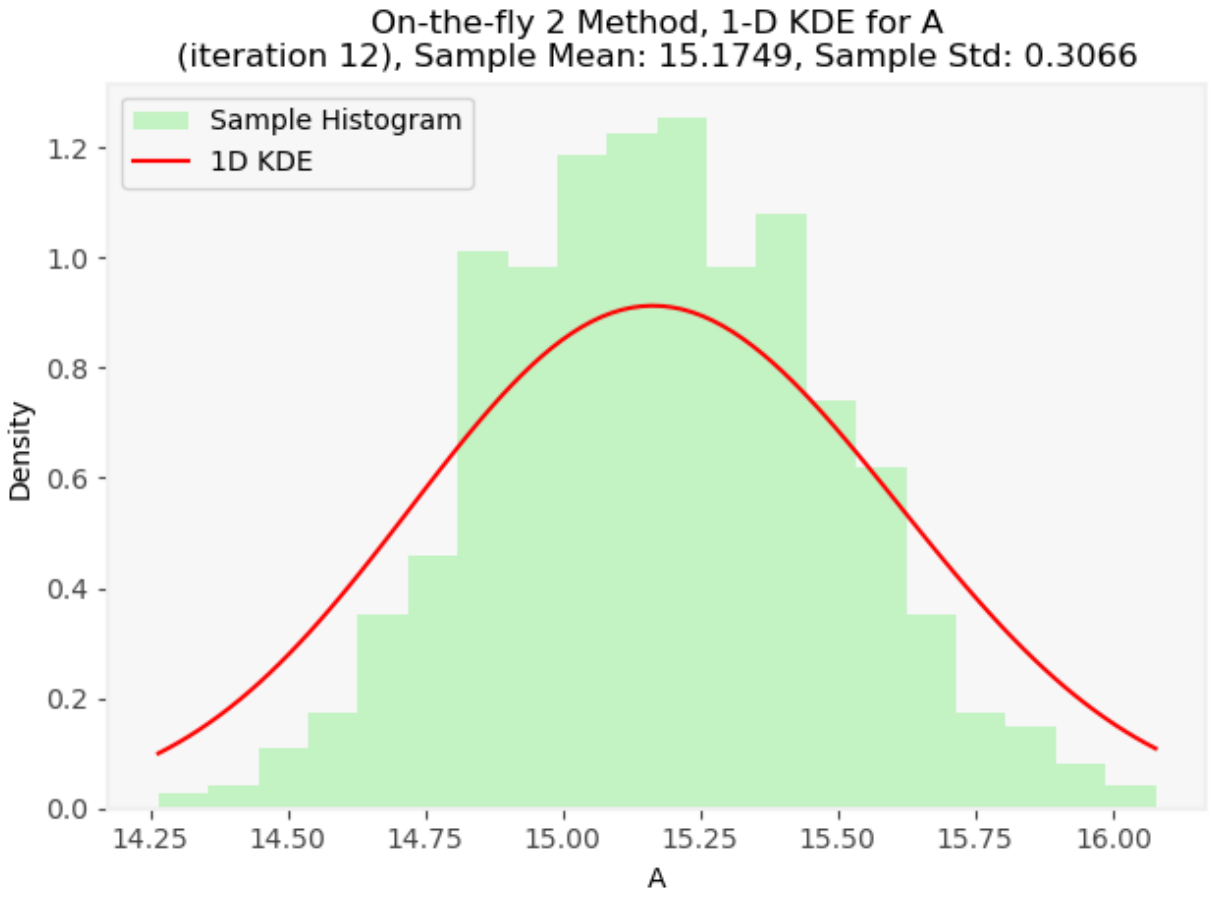

On-the-fly 2 Method, 1-D KDE for A
(iteration 7), Sample Mean: 13.8485, Sample Std: 1.9071
Sample Histogram
1D KDE
Density
A
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
10
12
14
16
18
20


On-the-fly 2 Method, 1-D KDE for A
(iteration 8), Sample Mean: 14.7601, Sample Std: 1.6545
Sample Histogram
1D KDE
Density
A
0.0
0.1
0.2
0.3
0.4
0.5
13
14
15
16
17
18
19
20


On-the-fly 2 Method, 1-D KDE for A
(iteration 9), Sample Mean: 16.9552, Sample Std: 1.7116
Sample Histogram
1D KDE
Density
A
0.0
0.1
0.2
0.3
0.4
14
15
16
17
18
19
20


On-the-fly 2 Method, 1-D KDE for A
(iteration 10), Sample Mean: 15.1850, Sample Std: 0.4853
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
1.2
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0


On-the-fly 2 Method, 1-D KDE for A
(iteration 11), Sample Mean: 15.1186, Sample Std: 0.3174
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
1.2
14.5
15.0
15.5
16.0
16.5
17.0


On-the-fly 2 Method, 1-D KDE for A
(iteration 12), Sample Mean: 15.1749, Sample Std: 0.3066
Sample Histogram
1D KDE
Density
A
0.0
0.2
0.4
0.6
0.8
1.0
1.2
14.25
14.50
14.75
15.00
15.25
15.50
15.75
16.00

Playground

In [ ]:

In [ ]:

In [ ]:

In [ ]: