PYKU's DATA Analysis Frame

이준호

2018.05.06

1. Outline

- 1. Data analysis frame
  - (1). SPSS
  - (2). ROOT
  - (3). PYTHON
- 2. To Do & Discussion

1.1 Data analysis frame :: “RAW DATA ”

DATE	AQI	PM2p5	PM10	SO2	CO	NO2	O3_8h	DAYS	HOLI
20131202	142	109	138	61	2.6	88	11	1	0
20131203	86	64	86	38	1.6	54	45	2	0
20131204	109	82	101	42	2	62	23	3	0
20131205	56	39	56	30	1.2	38	52	4	0
20131206	169	128	162	48	2.5	78	15	5	0
20131207	291	241	285	64	4.2	98	6	6	1
20131208	223	173	189	47	2.9	60	41	7	1
20131209	26	11	16	10	0.6	22	51	1	0
20131210	45	21	45	14	1	29	52	2	0
20131211	30	19	30	15	0.7	30	45	3	0
20131212	29	16	29	11	0.8	25	56	4	0
20131213	66	48	63	29	1.3	45	29	5	0
20131214	56	40	48	29	1.2	41	46	6	1
20131215	64	46	55	31	1.5	49	31	7	1
20131216	134	102	126	59	2.5	70	10	1	0
20131217	80	58	41	25	1.4	29	42	2	0

We have BEIJING air pollution data from 2013.12.

Everything starts from “RAW DATA”

Importance of collecting data

1.2 Data analysis frame :: Analysis Branches

- (1). 3 Branches for perform data analysis what we have collected.
  - 1. SPSS
  - 2. ROOT
  - 3. Python

- (2). Format of the "RAW DATA", is perfectly compatible with "SPSS"
  - Do whatever with SPSS, if you want.

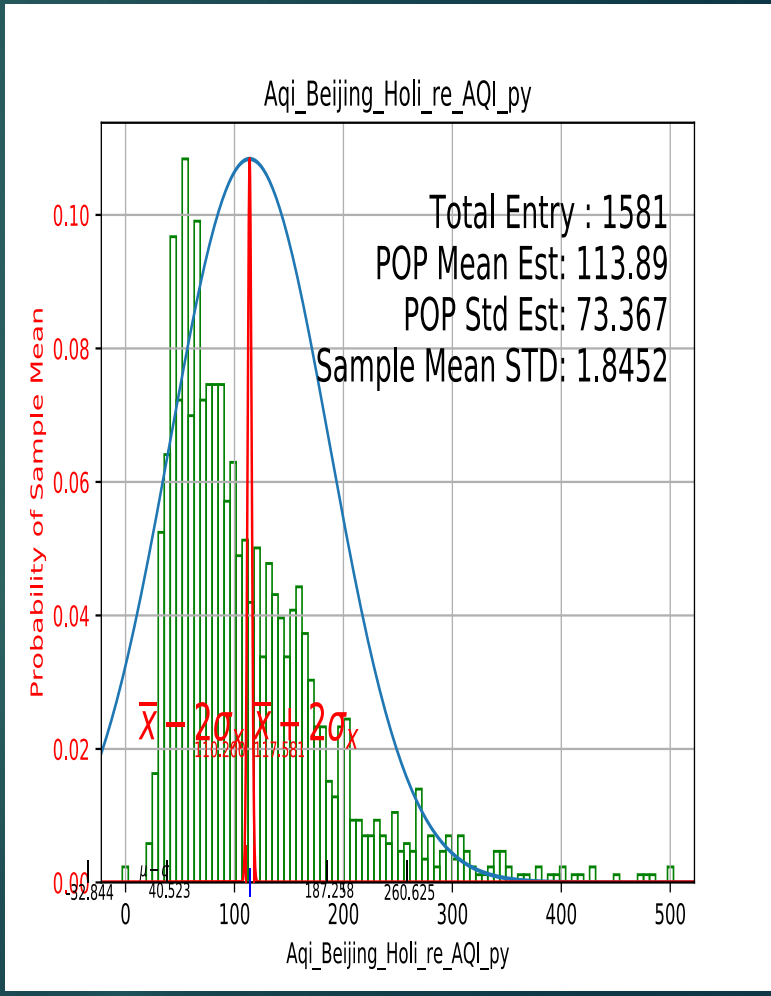
1.3 Data analysis frame :: ROOT & PYTHON. (Regenerate "Data" text file)

- We don't separate analysis frame at the first level :: Handling the "RAW Data" by PYTHON

1. Decoding the Encoded "RAW Data" (e.g. the "RAW Data" is opened, written, and saved by Excel. This will make everything decoded..)

2. (Optional) Removing out-of-valid-ranged Entry, by picking reasonable "N" * Standard deviation. ("project_pre/func/d0_Nsigma_skiming.py". Say, outside of 4*Std region can be excluded.)

3. (Optional, On work) Adding aditional cut to intentionally remove/select region. (Say, we want to see one specific distribution of Monday's)



1.4 Data analysis frame :: Why we separate?

- Having the selected events data text file ("DATA"), we separate analysis frame into 2 branches :: ROOT & PYTHON
  - 1. ROOT has advantage of visualize DATA. Very useful for Big DATA analysis, also provides Quick & Strong MC generator.
  - 2. PYTHON has lots of library for analysis collected DATA, including numpy, scipy, matploblib, etc.
  - 3. But for ROOT, one need Linux/Mac machine to get it installed. Not available for all of us… So, "PYTHON Frame" would be used for us, while ROOT will used for optimizing visualization of DATA.

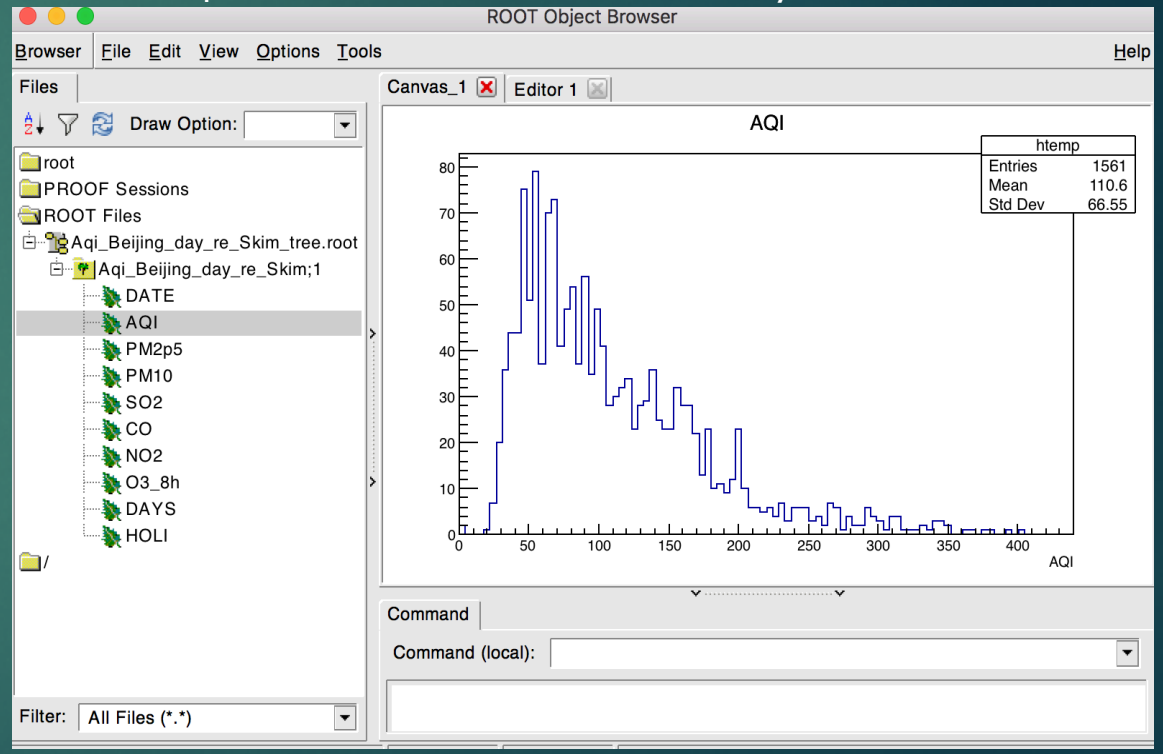
1.4 Data analysis frame :: ROOT

ROOT will be using both PYTHON and C++, So if only you have ROOT & PyROOT, you can perform this.

Now, we have "DATA"

```
DATE AQI PM2p5 PM10 SO2 CO NO2 O3_8h DAYS HOLI
20131202 142 109 138 61 2.6 88 11 1 0
20131203 86 64 86 38 1.6 54 45 2 0
20131204 109 82 101 42 2 62 23 3 0
20131205 56 39 56 30 1.2 38 52 4 0
20131206 169 128 162 48 2.5 78 15 5 0
20131207 291 241 285 64 4.2 98 6 6 1
20131208 223 173 189 47 2.9 60 41 7 1
20131209 26 11 16 10 0.6 22 51 1 0
20131210 45 21 45 14 1 29 52 2 0
20131211 30 19 30 15 0.7 30 45 3 0
20131212 29 16 29 11 0.8 25 56 4 0
20131213 66 48 63 29 1.3 45 29 5 0
20131214 56 40 48 29 1.2 41 46 6 1
20131215 64 46 55 31 1.5 49 31 7 1
20131216 134 102 126 59 2.5 70 10 1 0
```

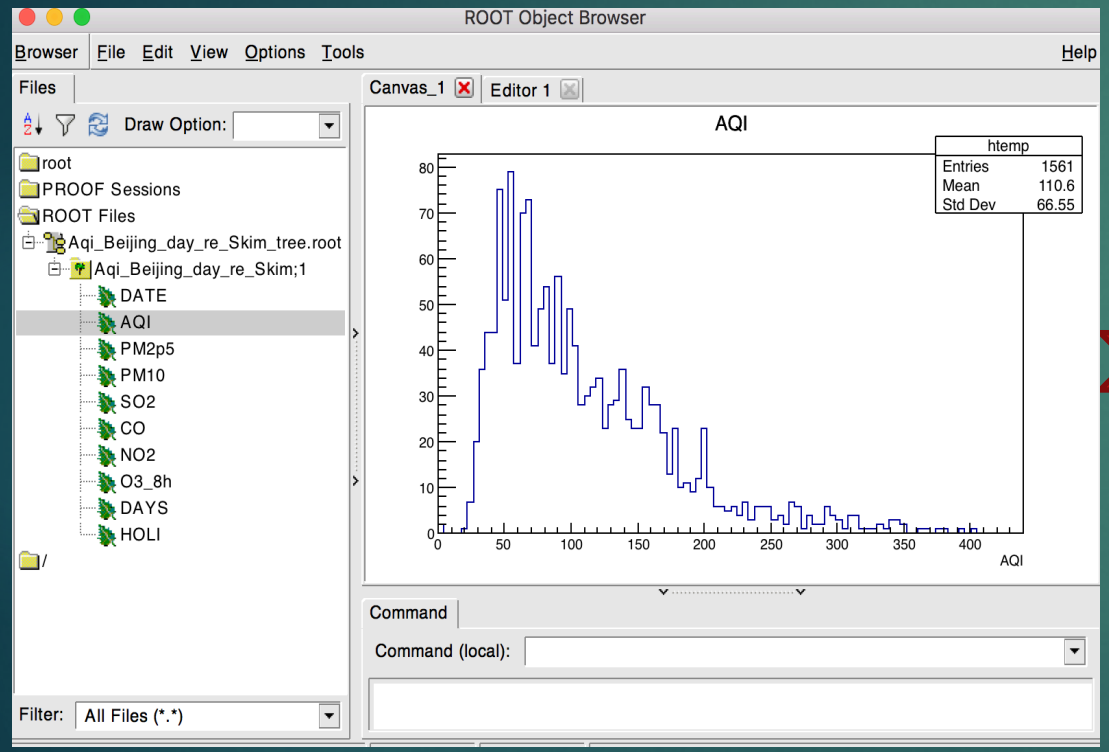
Transform the "DATA" into ROOT's Tree.
Each Entry -> One event in all of the branch
(Same expression, different format)



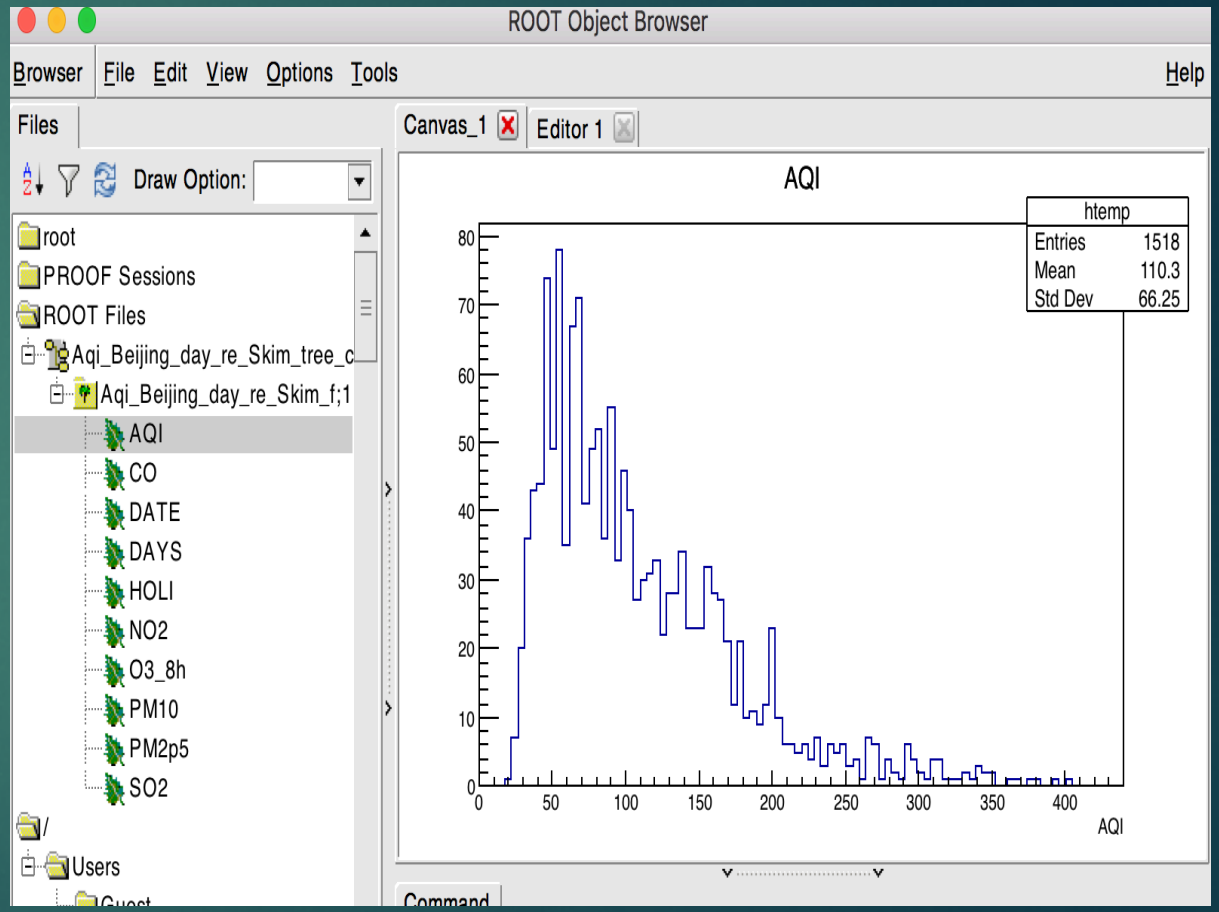
fruit_team/ROOT/Project/functions/rawTxt_Tree_root/Raw_text_to_Tree_root.py

1.4 Data analysis frame :: ROOT

The "DATA" corresponding ROOT's Tree ("Tree")



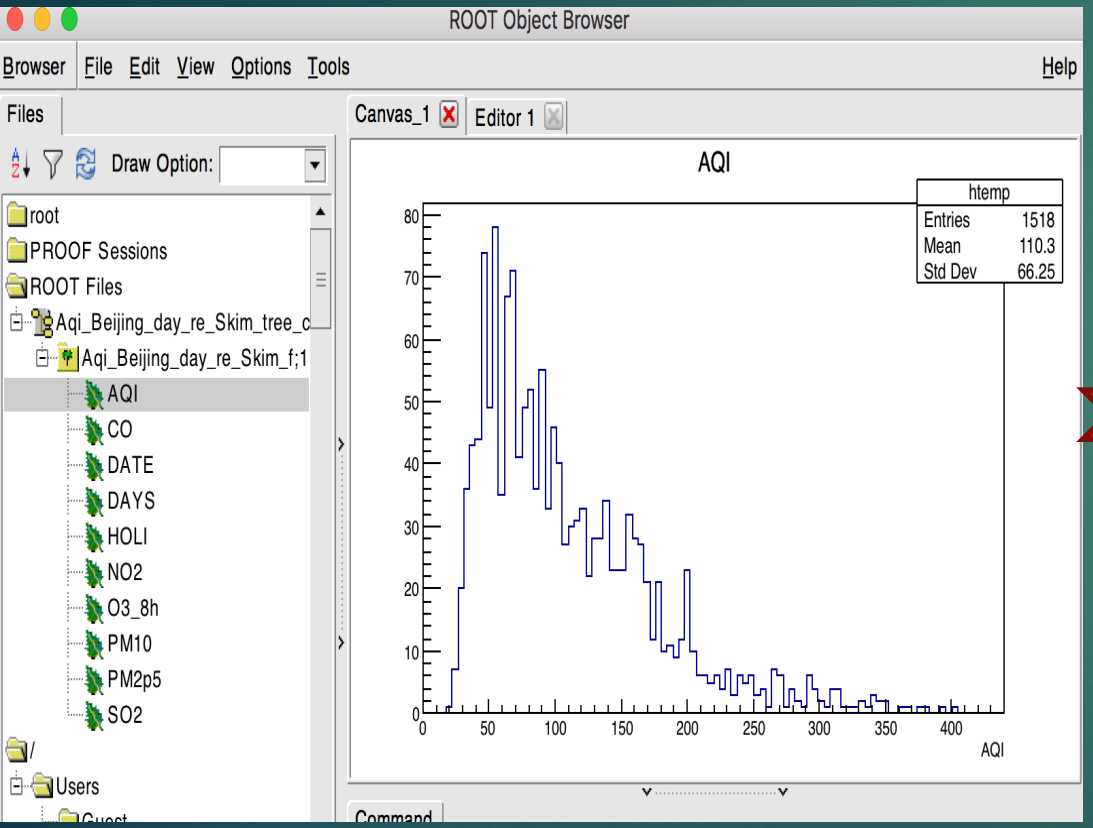
Further apply "Cut" on Tree, to select events we want. ("Cut Tree")



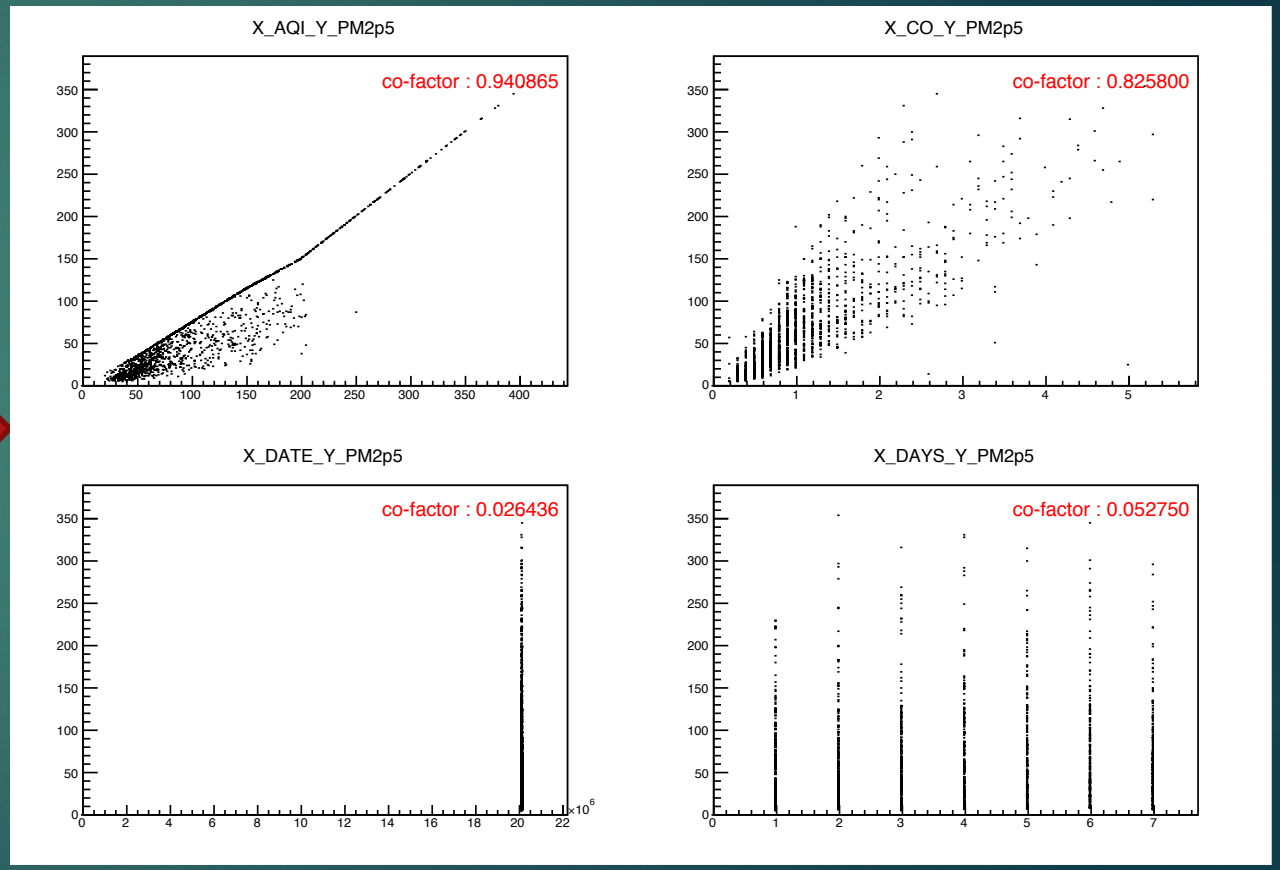
fruit_team/ROOT/Project/functions/rootTree_rootHist/func/Tree_to_D1H_CutnGenerate.py

1.4 Data analysis frame :: ROOT

"Cut Tree"



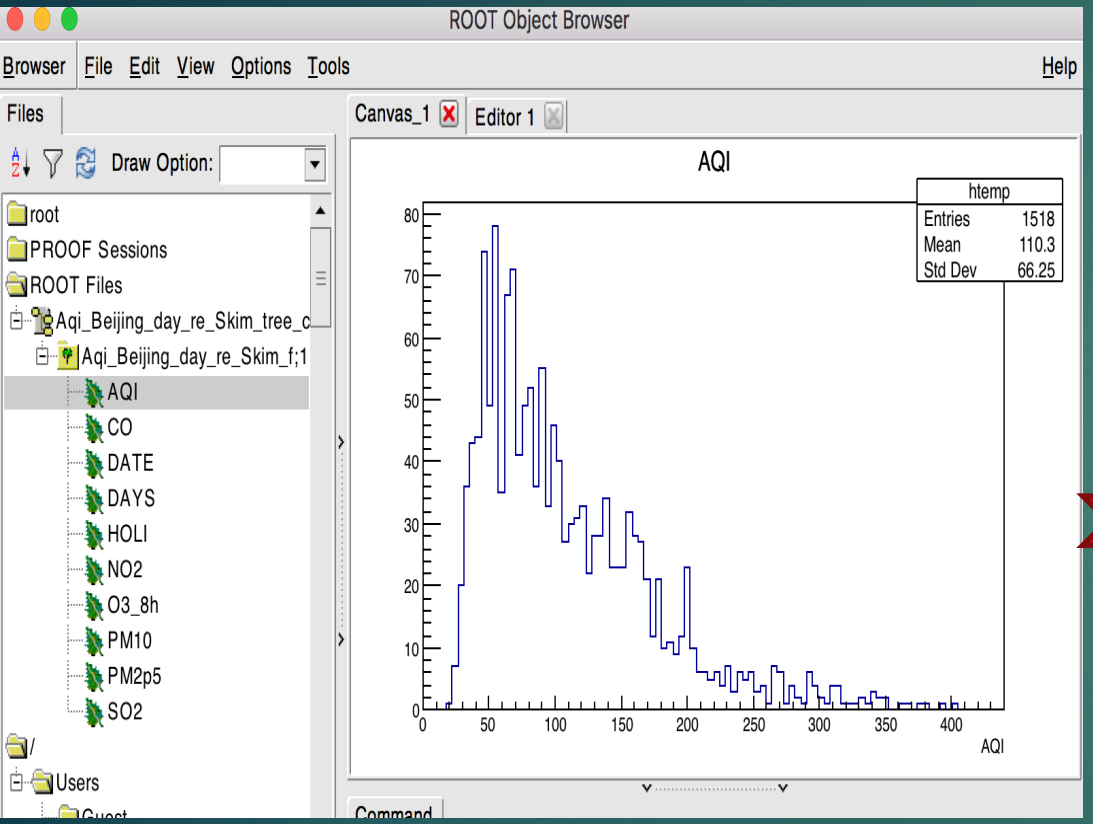
Perform 2D scattering, for all of possible pair Combination ("2D no bin")



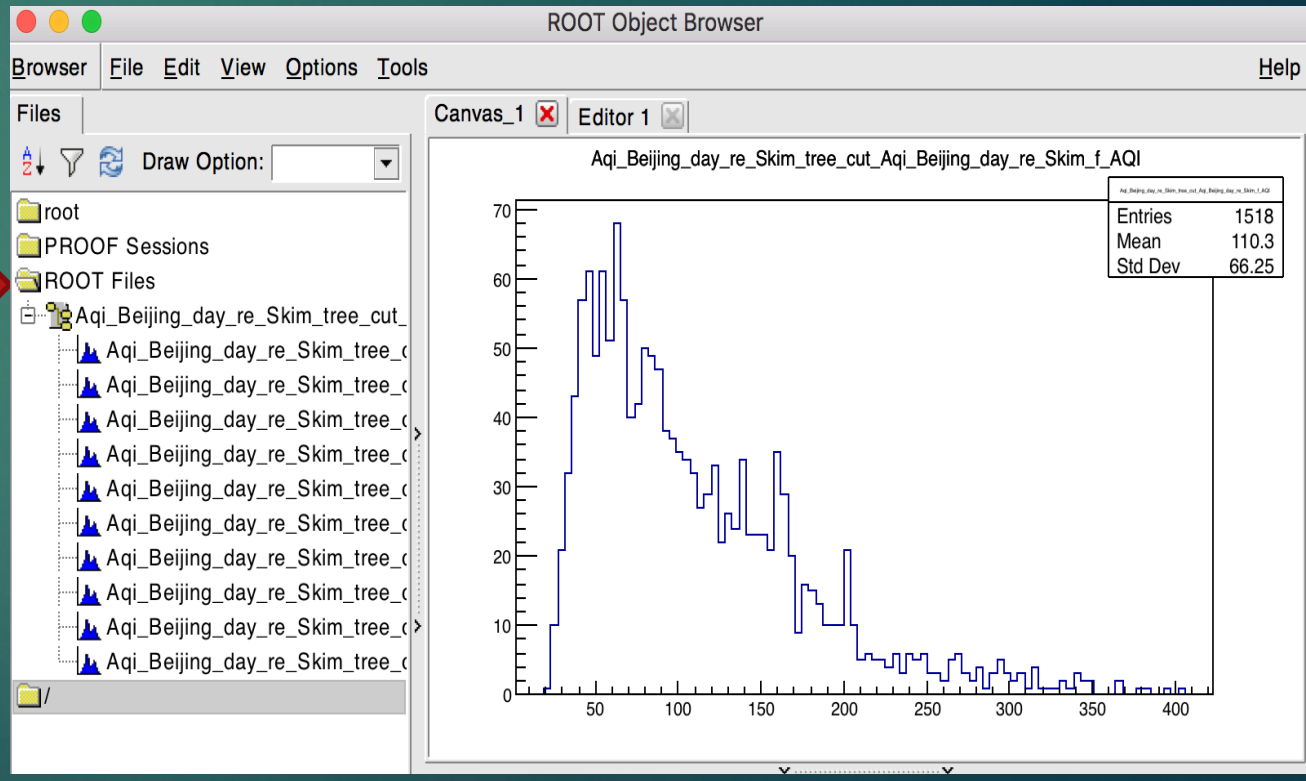
fruit_team/ROOT/Project/functions/TGraph_saver/Tree_branches_to_vector.C

1.4 Data analysis frame :: ROOT

Cut Tree



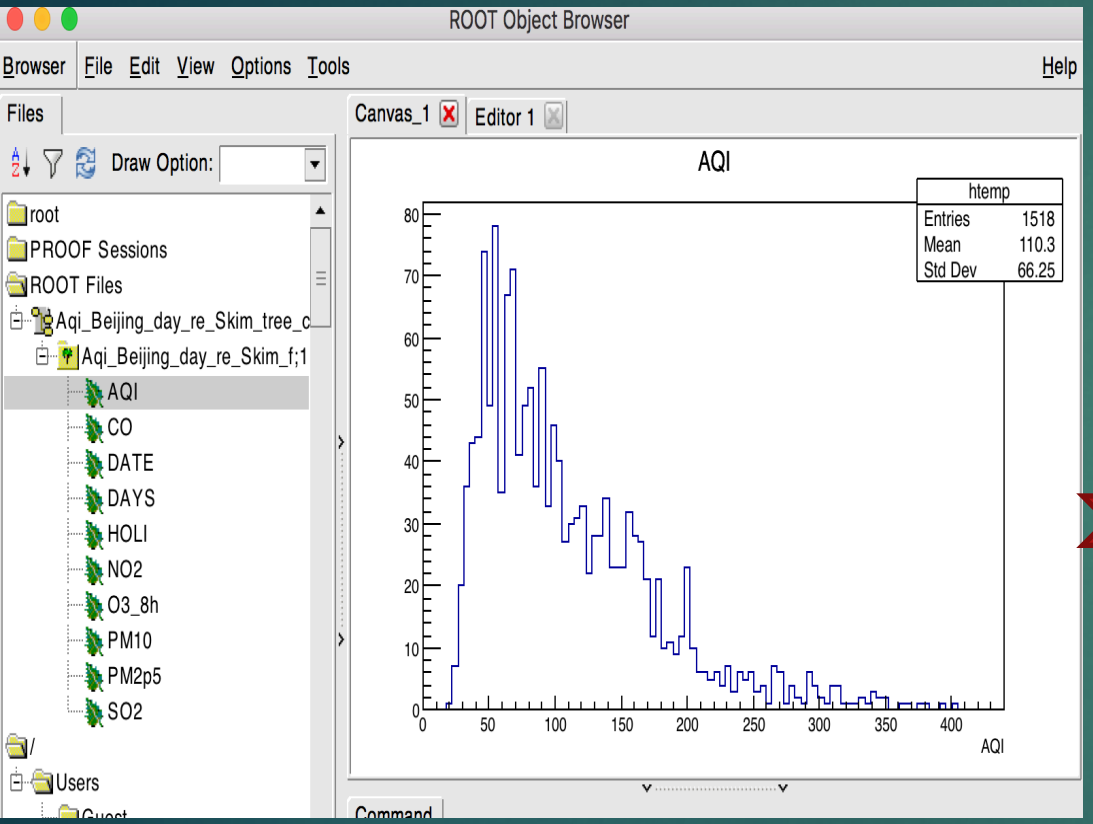
Perform D1 histogram transfromation for further "binned" study. (This can be interfaced with python later on. I will come back to this later). ("TH1D")



fruit_team/ROOT/Project/functions/rootTree_rootHist/func/Tree_to_D1H_Convert.py

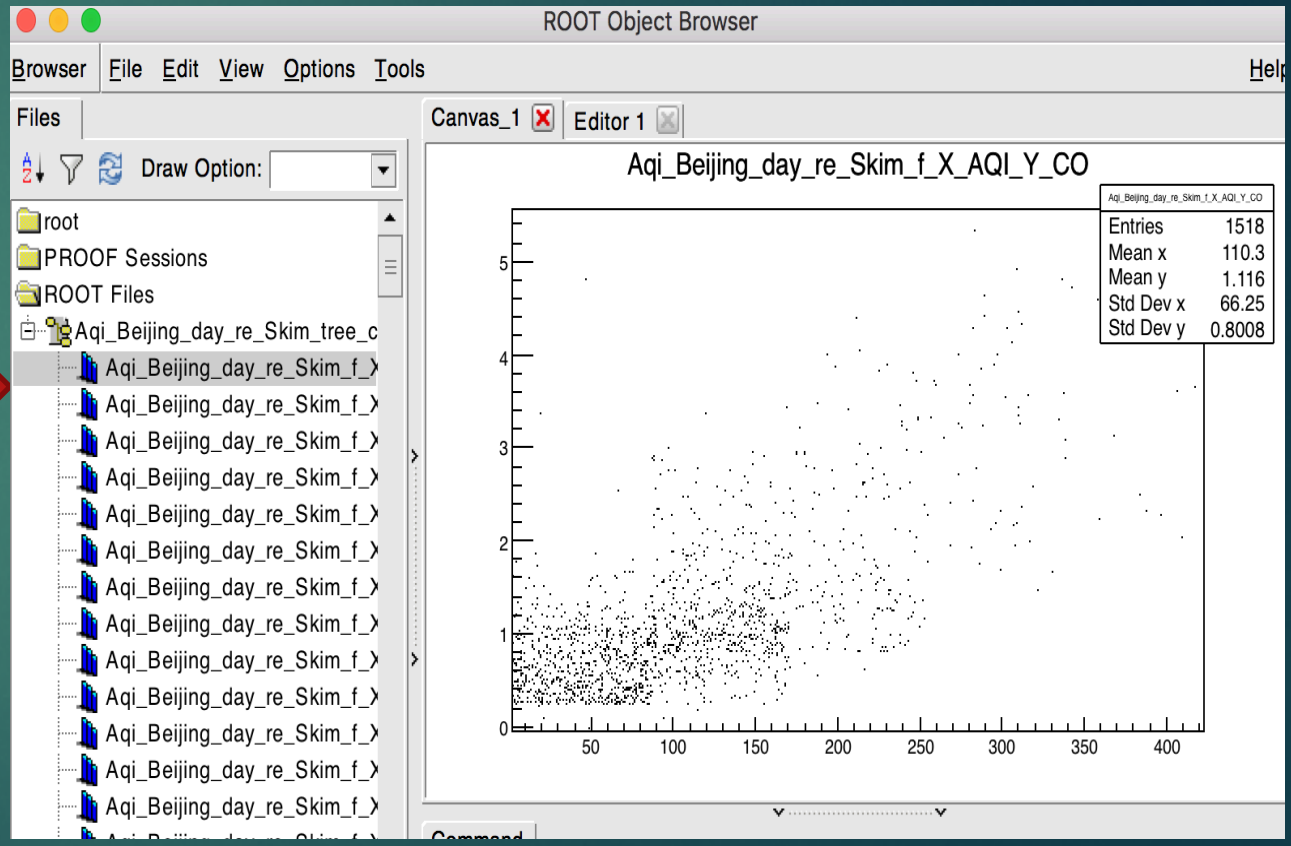
1.4 Data analysis frame :: ROOT

Cut Tree



Setting bin_number

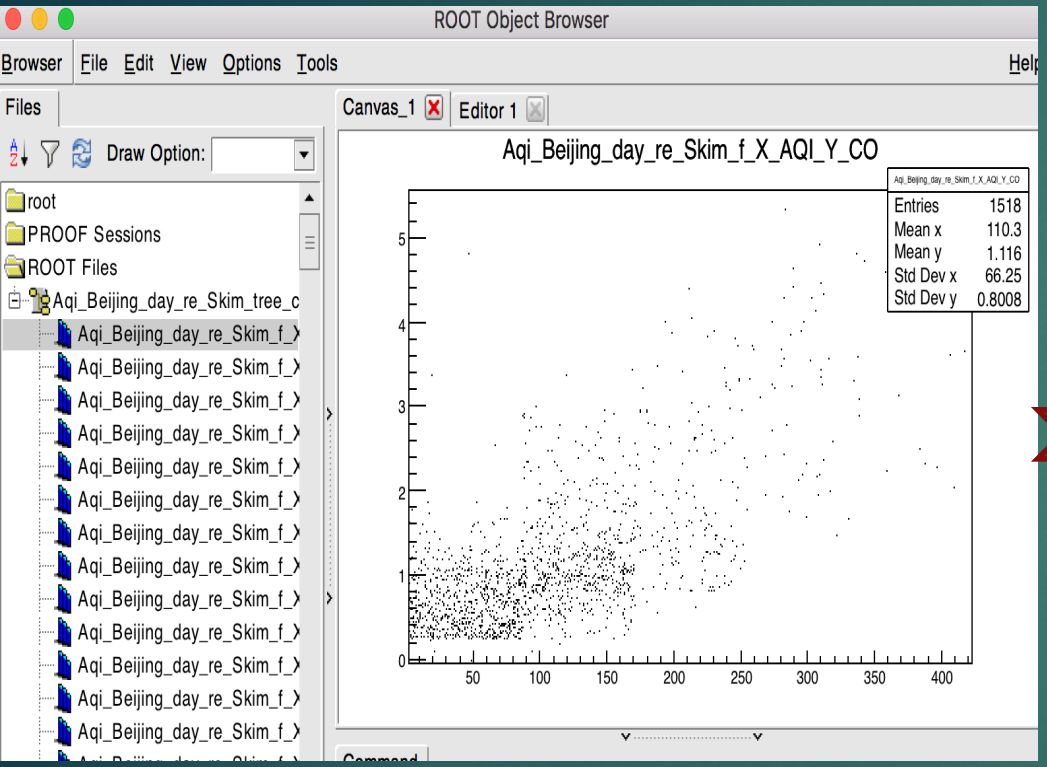
To 2D histogram with every possible combination.
This would be input for futher 2D plotting ("TH2D")



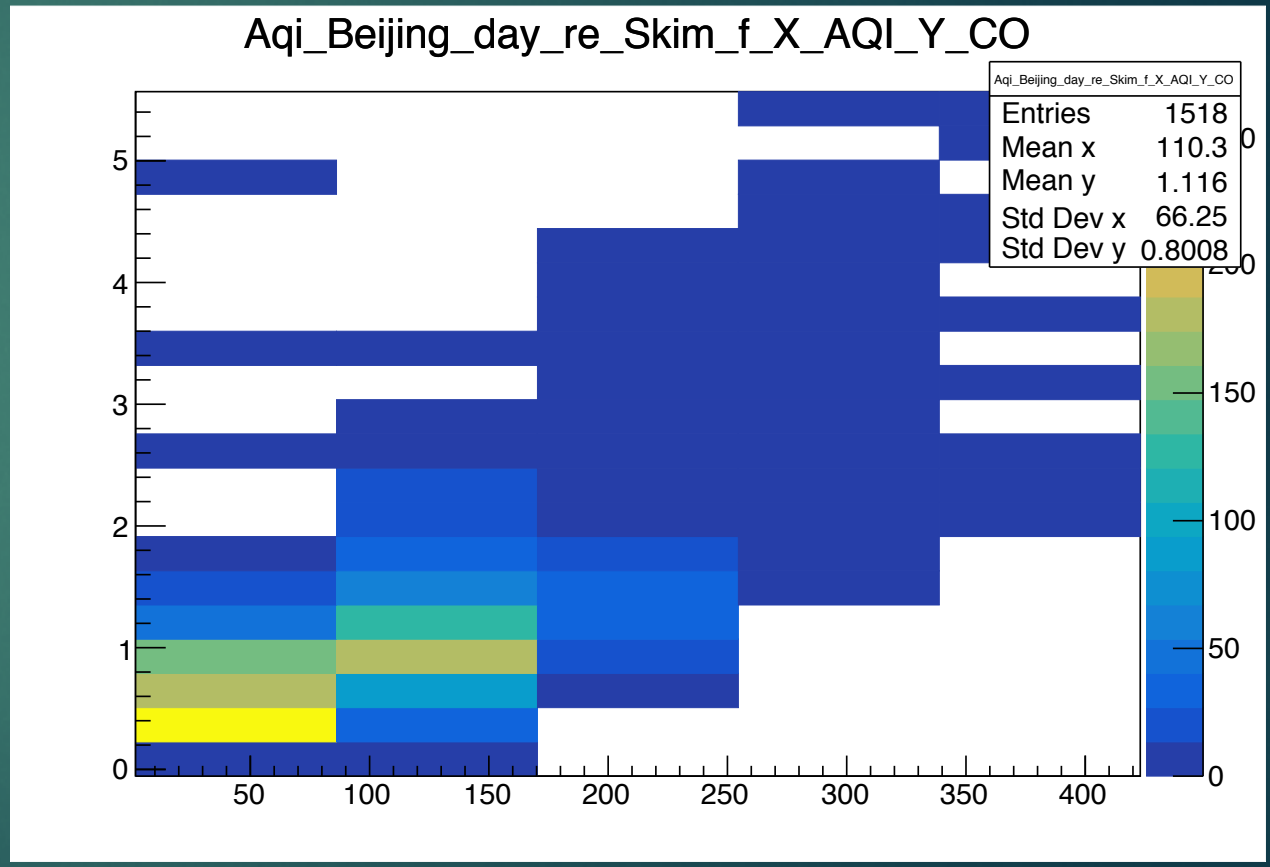
fruit_team/ROOT/Project/functions/rootTree_rootHist/func/Tree_to_D2H_Convert.py

1.4 Data analysis frame :: ROOT

TH2D



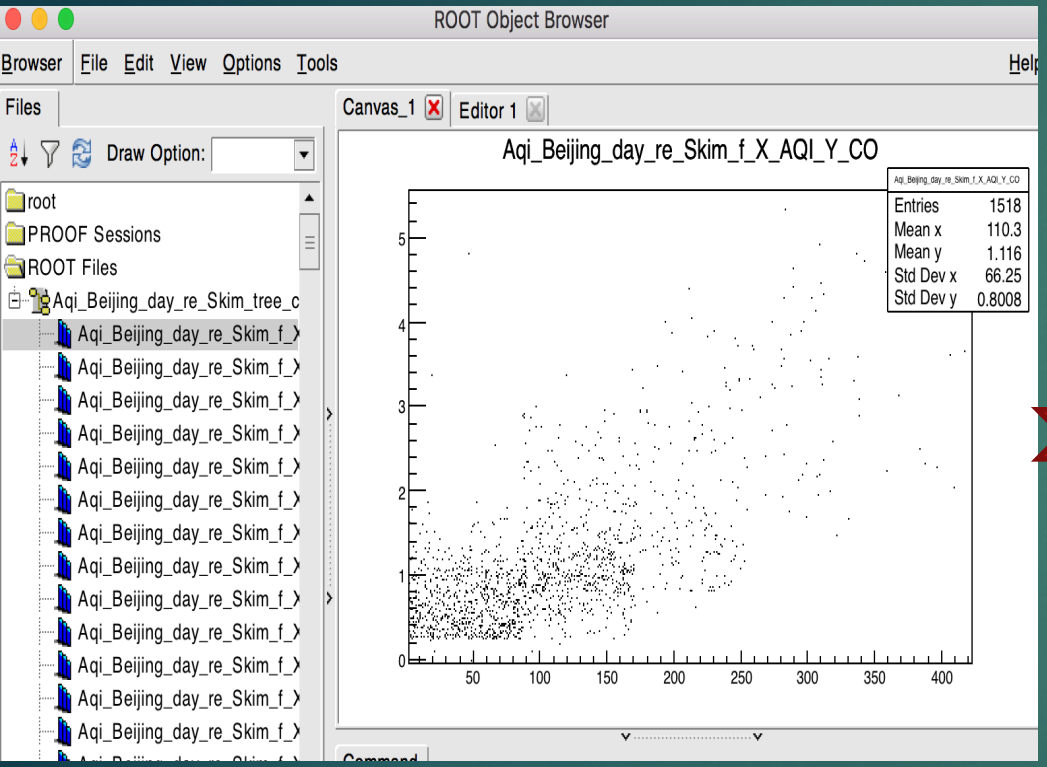
ColZ 2D



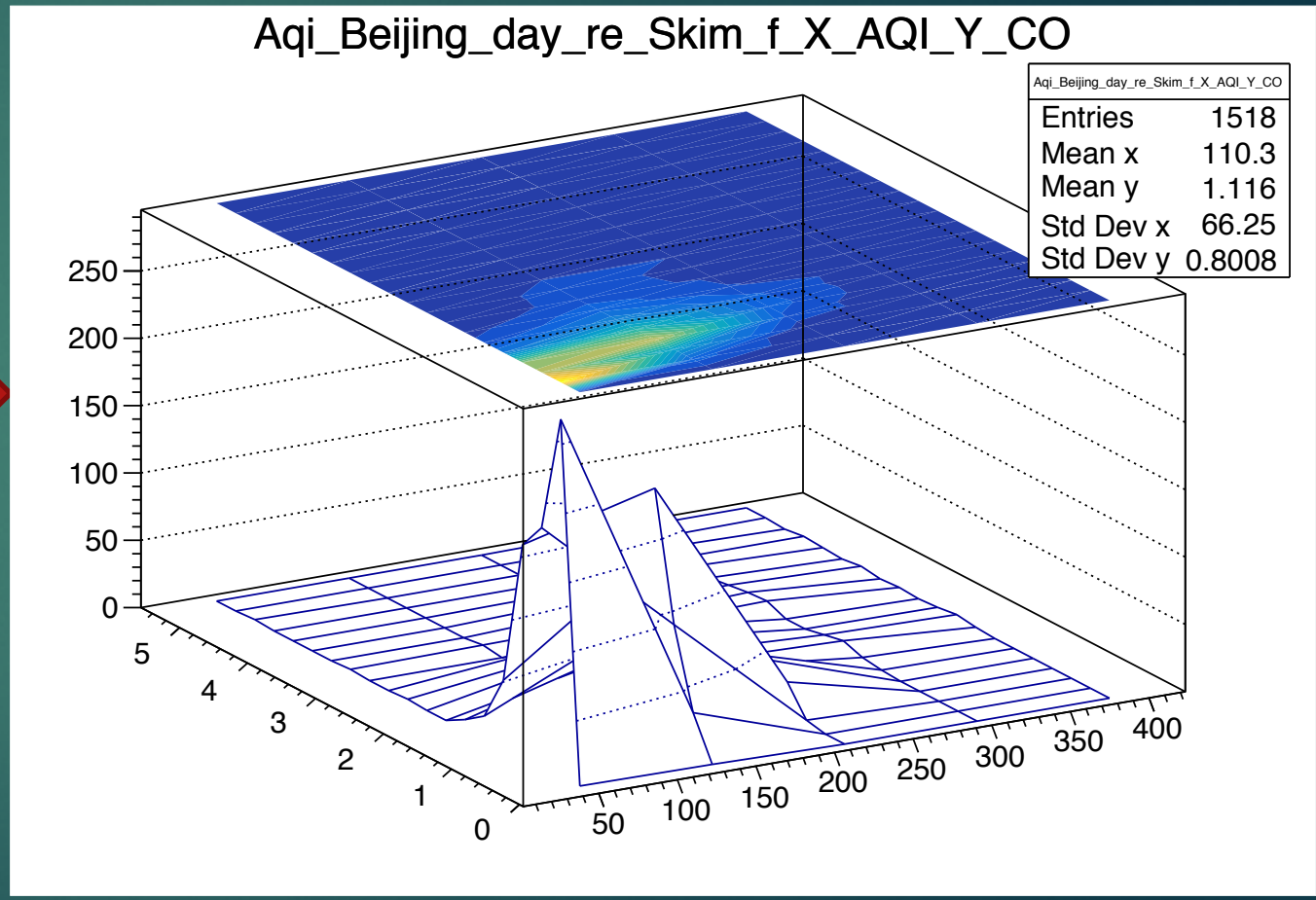
fruit_team/ROOT/Project/functions/2Dplots_Saver/TwoD_Plot_Saver_default.C

1.4 Data analysis frame :: ROOT

TH2D



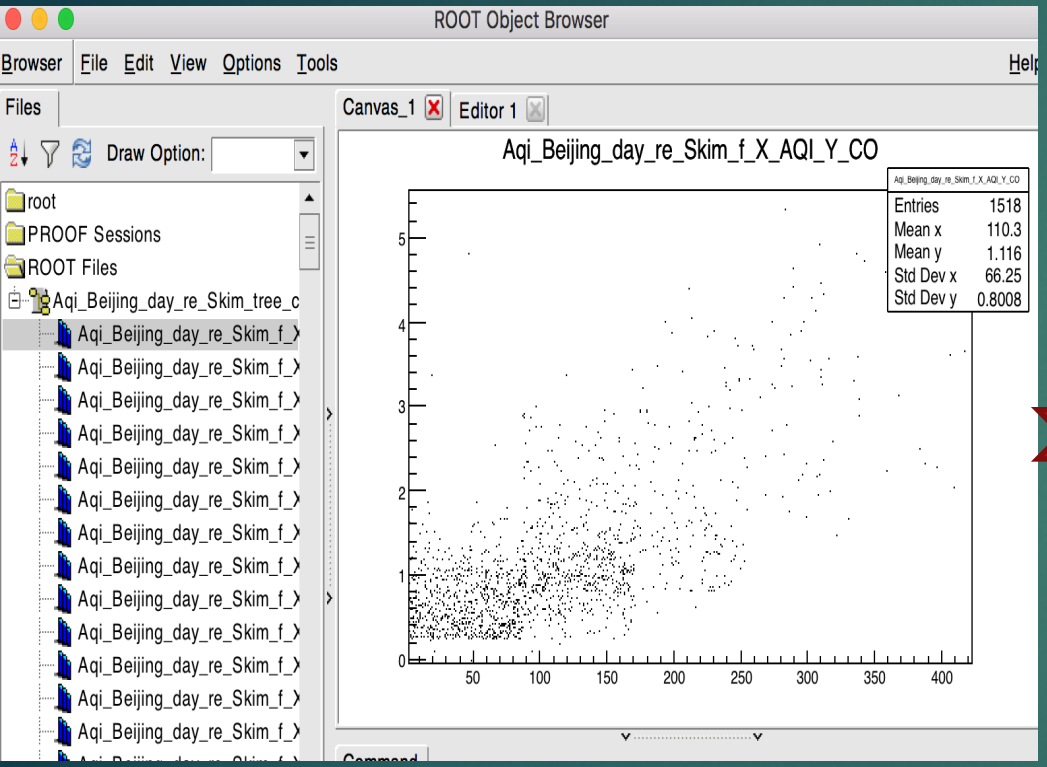
Surf 2D



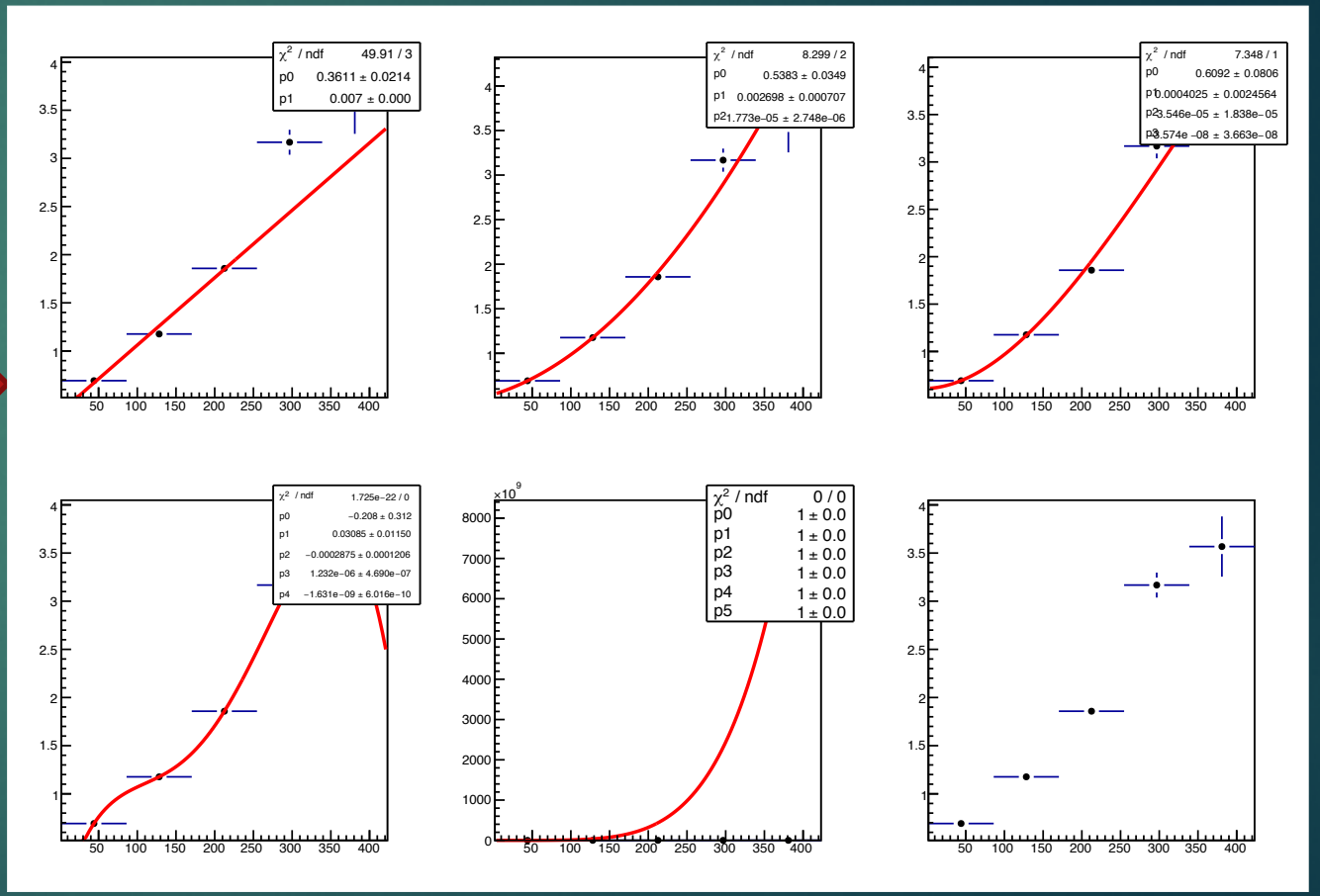
fruit_team/ROOT/Project/functions/2Dplots_Saver/TwoD_Plot_Saver_default.C

1.4 Data analysis frame :: ROOT

TH2D



ProfileX 2D



fruit_team/ROOT/Project/functions/2Dplots_Saver/TwoD_profileX_pol_fitter.C

1.4 Data analysis frame :: ROOT

Overview :: RAW DATA -> DATA -> Tree -> Cut_Tree

Cut_Tree -> TH1D -> Interfacing with python (come back later, without ROOT)

-> TH2D -> ColZ. 2D

-> . Surf 2D

-> profileX 2D

All of these codes could be invoked by simply lauching the python script :: "**project_pre/data_txt/BEIJING_Aqi/**execute_test_ROOT.py" (Let me show you!)

(This is what we have for ROOT, for now)

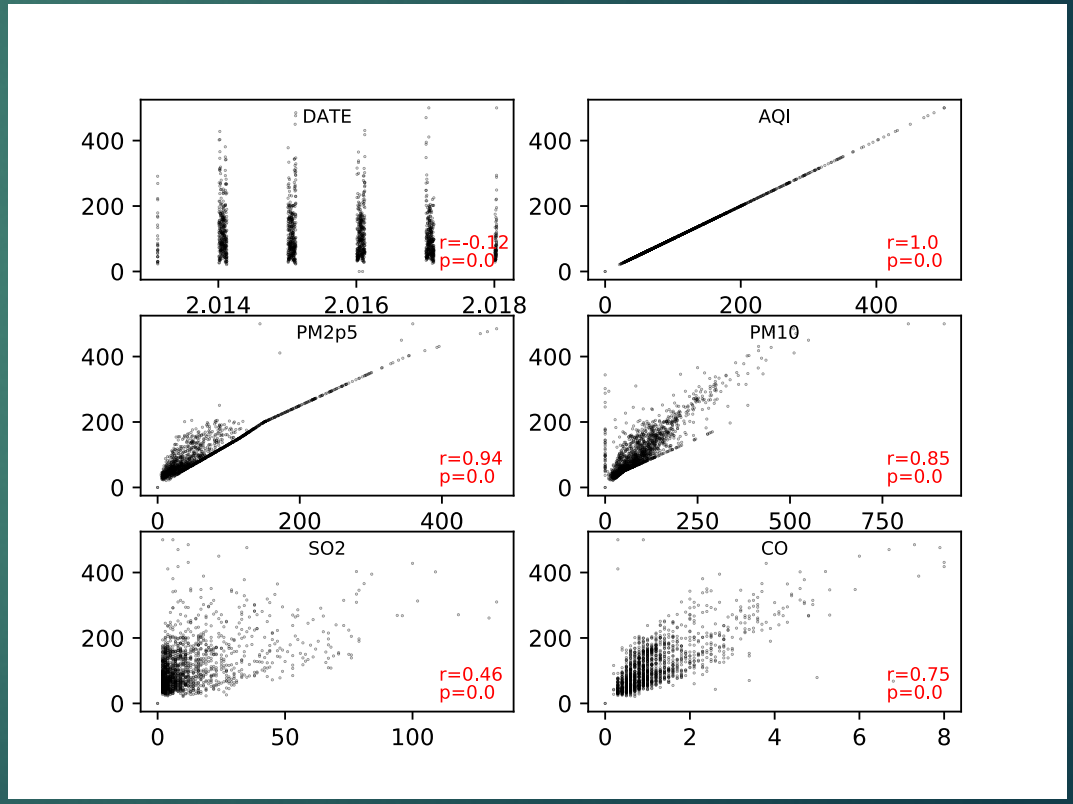
1.4 Data analysis frame :: PYTHON

Same procedure :: “RAW DATA” to “DATA”

“DATA”

```
DATE AQI PM2p5 PM10 SO2 CO NO2 O3_8h DAYS HOLI
20131202 142 109 138 61 2.6 88 11 1 0
20131203 86 64 86 38 1.6 54 45 2 0
20131204 109 82 101 42 2 62 23 3 0
20131205 56 39 56 30 1.2 38 52 4 0
20131206 169 128 162 48 2.5 78 15 5 0
20131207 291 241 285 64 4.2 98 6 6 1
20131208 223 173 189 47 2.9 60 41 7 1
20131209 26 11 16 10 0.6 22 51 1 0
20131210 45 21 45 14 1 29 52 2 0
20131211 30 19 30 15 0.7 30 45 3 0
20131212 29 16 29 11 0.8 25 56 4 0
20131213 66 48 63 29 1.3 45 29 5 0
20131214 56 40 48 29 1.2 41 46 6 1
20131215 64 46 55 31 1.5 49 31 7 1
20131216 134 102 126 59 2.5 70 10 1 0
```

2D scattering plot, directly using “DATA”
(Again, all of possible combination) (“MAN 2D”)



project_pre/func/man_pyscat.py

1.4 Data analysis frame :: PYTHON, interfacing with ROOT's TH1D

DATA

```
DATE AQI PM2p5 PM10 SO2 CO NO2 O3_8h DAYS HOLI
20131202 142 109 138 61 2.6 88 11 1 0
20131203 86 64 86 38 1.6 54 45 2 0
20131204 109 82 101 42 2 62 23 3 0
20131205 56 39 56 30 1.2 38 52 4 0
20131206 169 128 162 48 2.5 78 15 5 0
20131207 291 241 285 64 4.2 98 6 6 1
20131208 223 173 189 47 2.9 60 41 7 1
20131209 26 11 16 10 0.6 22 51 1 0
20131210 45 21 45 14 1 29 52 2 0
20131211 30 19 30 15 0.7 30 45 3 0
20131212 29 16 29 11 0.8 25 56 4 0
20131213 66 48 63 29 1.3 45 29 5 0
20131214 56 40 48 29 1.2 41 46 6 1
20131215 64 46 55 31 1.5 49 31 7 1
20131216 134 102 126 59 2.5 70 10 1 0
```

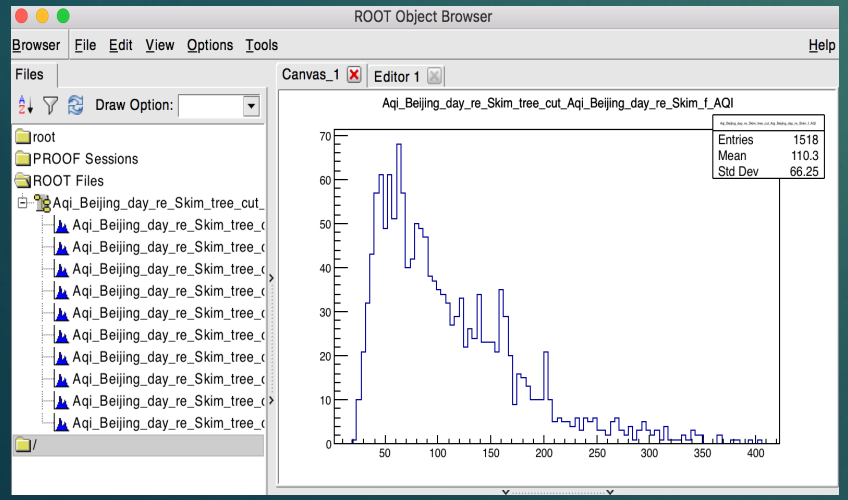
```
1 -25.000000 -19.500000 0.000000
2 -19.500000 -14.000000 0.000000
3 -14.000000 -8.500000 0.000000
4 -8.500000 -3.000000 0.000000
5 -3.000000 2.500000 2.000000
6 2.500000 8.000000 0.000000
7 8.000000 13.500000 0.000000
8 13.500000 19.000000 0.000000
9 19.000000 24.500000 5.000000
10 24.500000 30.000000 14.000000
11 30.000000 35.500000 45.000000
12 35.500000 41.000000 55.000000
13 41.000000 46.500000 83.000000
14 46.500000 52.000000 62.000000
15 52.000000 57.500000 93.000000
16 57.500000 63.000000 60.000000
```

"D1 Hist_DATA"

project_pre/func/RAW_to_HIST_txt.py

These two are almost identical

TH1D



```
1 -25.000000 -19.500000 0.000000
2 -19.500000 -14.000000 0.000000
3 -14.000000 -8.500000 0.000000
4 -8.500000 -3.000000 0.000000
5 -3.000000 2.500000 2.000000
6 2.500000 8.000000 0.000000
7 8.000000 13.500000 0.000000
8 13.500000 19.000000 0.000000
9 19.000000 24.500000 5.000000
10 24.500000 30.000000 14.000000
11 30.000000 35.500000 45.000000
12 35.500000 41.000000 55.000000
13 41.000000 46.500000 83.000000
14 46.500000 52.000000 62.000000
15 52.000000 57.500000 93.000000
16 57.500000 63.000000 60.000000
```

"D1 Hist_DATA"

fruit_team/ROOT/Project/functions/rootHist_TXT/func/D1H_roothist_to_txt.py

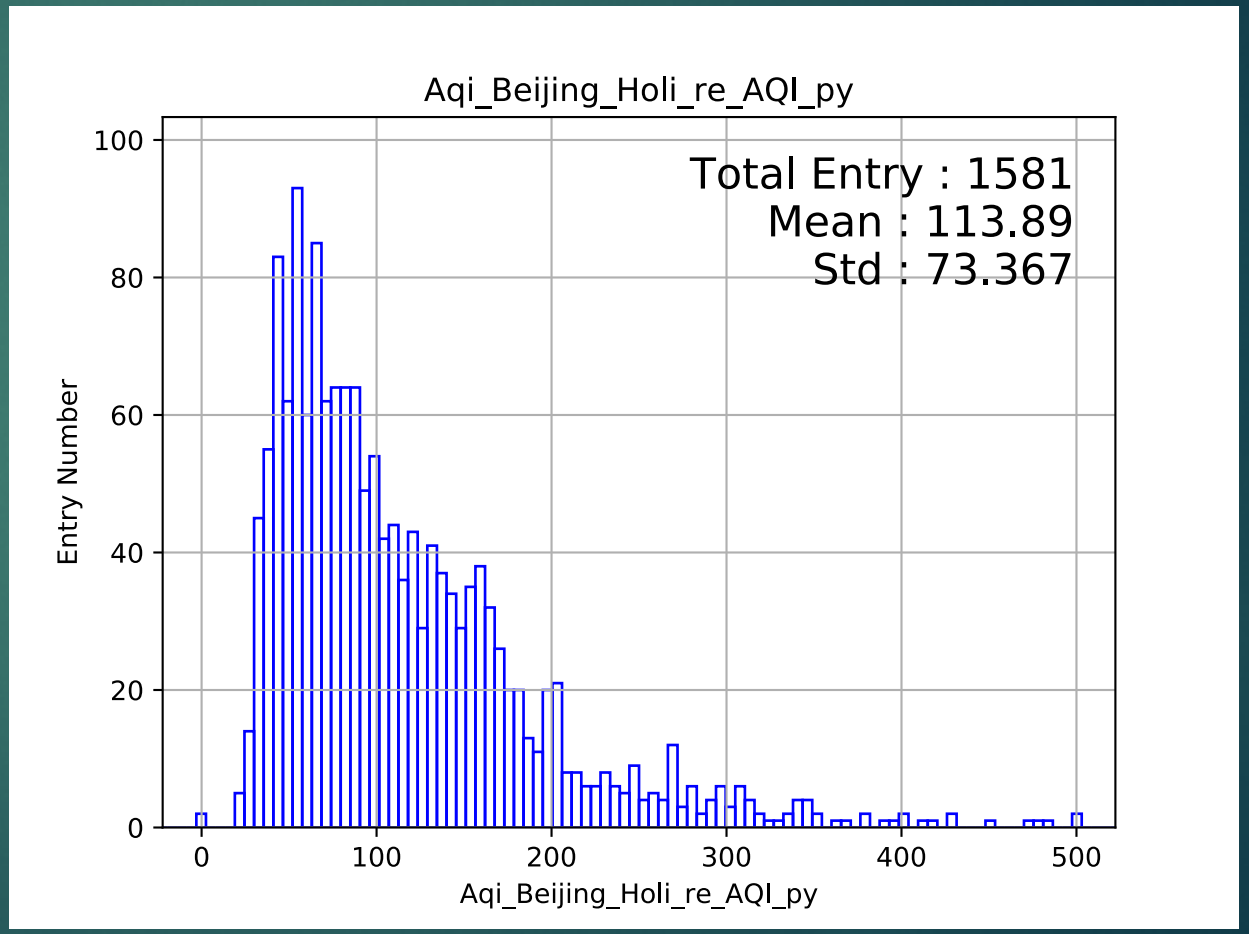
1.4 Data analysis frame :: PYTHON

So now, we have same output for ROOT & Python, which would be the inputs of one dimention tests

Hist DATA

```
1 -25.000000 -19.500000 0.000000
2 -19.500000 -14.000000 0.000000
3 -14.000000 -8.500000 0.000000
4 -8.500000 -3.000000 0.000000
5 -3.000000 2.500000 2.000000
6 2.500000 8.000000 0.000000
7 8.000000 13.500000 0.000000
8 13.500000 19.000000 0.000000
9 19.000000 24.500000 5.000000
10 24.500000 30.000000 14.000000
11 30.000000 35.500000 45.000000
12 35.500000 41.000000 55.000000
13 41.000000 46.500000 83.000000
14 46.500000 52.000000 62.000000
15 52.000000 57.500000 93.000000
16 57.500000 63.000000 60.000000
```

PY_TH1D



project_pre/func/c2_basic_histo_plotting.py

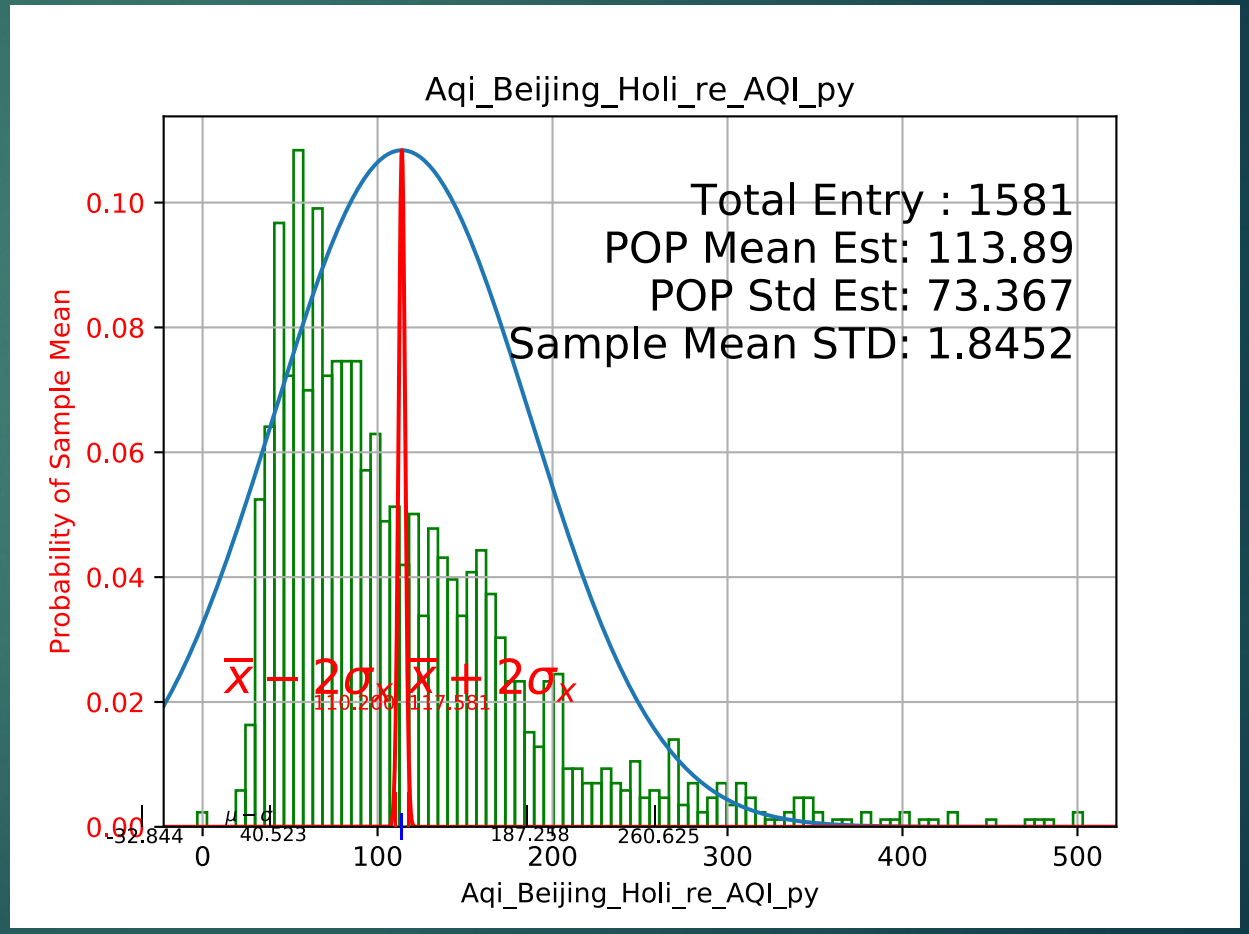
1.4 Data analysis frame :: PYTHON

So now, we have same output for ROOT & Python, which would be the inputs of one dimentional tests

Hist DATA

```
1 -25.000000 -19.500000 0.000000
2 -19.500000 -14.000000 0.000000
3 -14.000000 -8.500000 0.000000
4 -8.500000 -3.000000 0.000000
5 -3.000000 2.500000 2.000000
6 2.500000 8.000000 0.000000
7 8.000000 13.500000 0.000000
8 13.500000 19.000000 0.000000
9 19.000000 24.500000 5.000000
10 24.500000 30.000000 14.000000
11 30.000000 35.500000 45.000000
12 35.500000 41.000000 55.000000
13 41.000000 46.500000 83.000000
14 46.500000 52.000000 62.000000
15 52.000000 57.500000 93.000000
16 57.500000 63.000000 60.000000
```

PY_TH1D_Z



project_pre/func/c5_single_sample_mean_Zdistribution.py

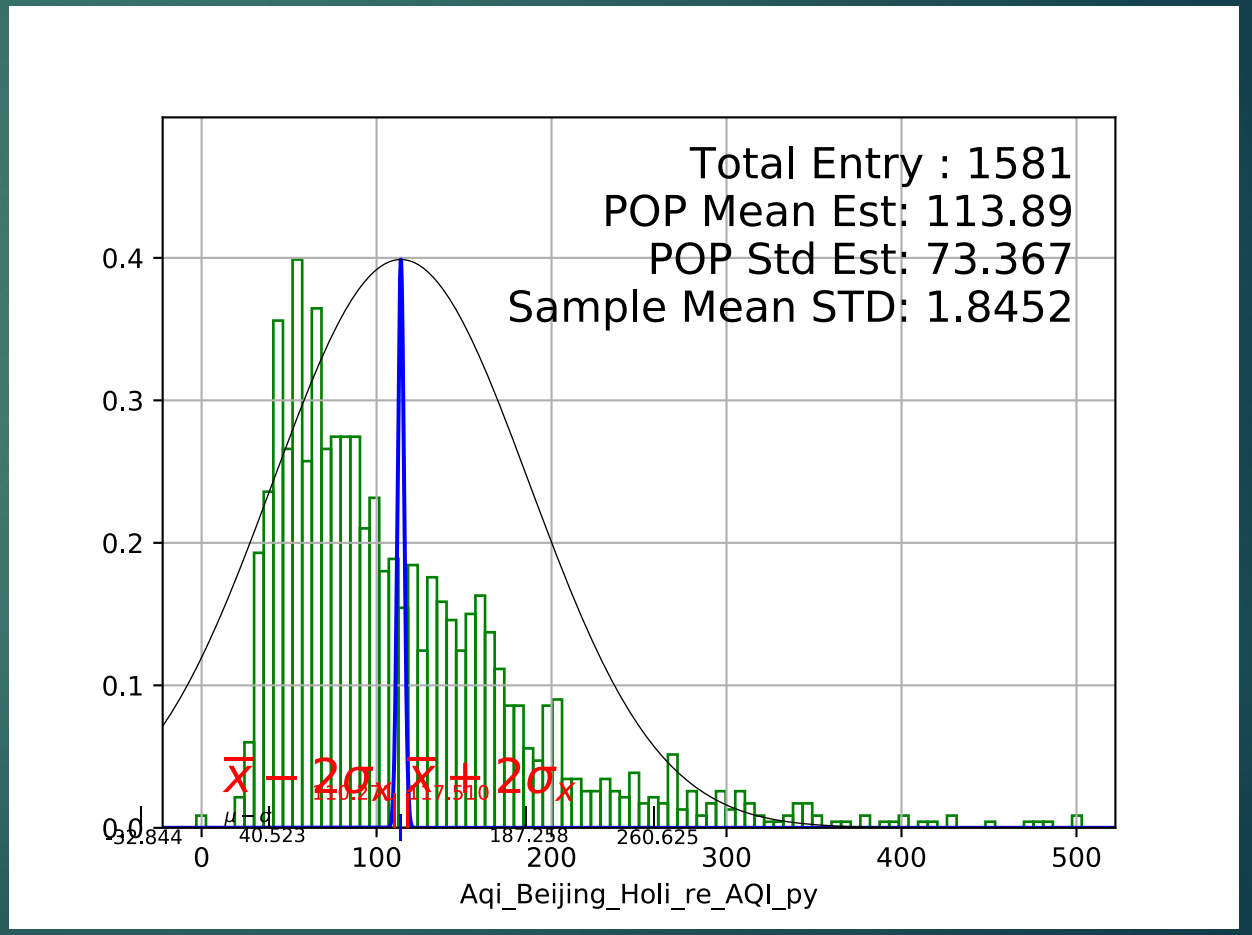
1.4 Data analysis frame :: PYTHON

So now, we have same output for ROOT & Python, which would be the inputs of one dimentional tests

Hist DATA

1	-25.000000	-19.500000	0.000000
2	-19.500000	-14.000000	0.000000
3	-14.000000	-8.500000	0.000000
4	-8.500000	-3.000000	0.000000
5	-3.000000	2.500000	2.000000
6	2.500000	8.000000	0.000000
7	8.000000	13.500000	0.000000
8	13.500000	19.000000	0.000000
9	19.000000	24.500000	5.000000
10	24.500000	30.000000	14.000000
11	30.000000	35.500000	45.000000
12	35.500000	41.000000	55.000000
13	41.000000	46.500000	83.000000
14	46.500000	52.000000	62.000000
15	52.000000	57.500000	93.000000
16	57.500000	63.000000	60.000000

PY_TH1D_T



project_pre/func/c5_single_sample_mean_Tdistribution.py

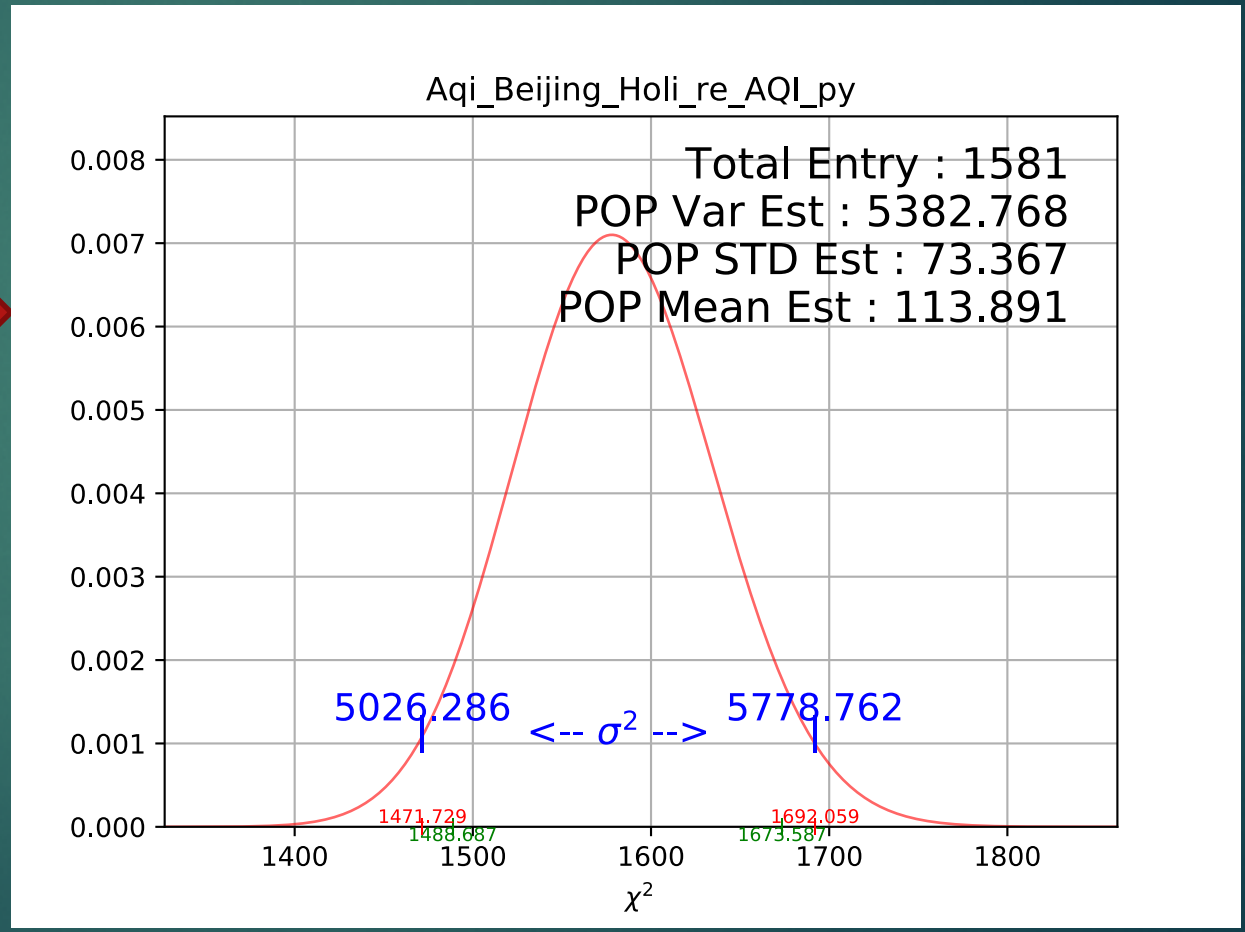
1.4 Data analysis frame :: PYTHON

So now, we have same output for ROOT & Python, which would be the inputs of one dimentional tests

Hist DATA

```
1 -25.000000 -19.500000 0.000000
2 -19.500000 -14.000000 0.000000
3 -14.000000 -8.500000 0.000000
4 -8.500000 -3.000000 0.000000
5 -3.000000 2.500000 2.000000
6 2.500000 8.000000 0.000000
7 8.000000 13.500000 0.000000
8 13.500000 19.000000 0.000000
9 19.000000 24.500000 5.000000
10 24.500000 30.000000 14.000000
11 30.000000 35.500000 45.000000
12 35.500000 41.000000 55.000000
13 41.000000 46.500000 83.000000
14 46.500000 52.000000 62.000000
15 52.000000 57.500000 93.000000
16 57.500000 63.000000 60.000000
```

PY_TH1D_Chi2



project_pre/func/c7_single_sample_variance_distribution.py

1.4 Data analysis frame :: PYTHON

Overview :: RAW DATA -> DATA

DATA -> MAN 2D

-> D1 Hist_DATA -> PY_TH1D

-> PY_TH1D_Z

-> PY_TH1D_T

-> PY_TH1D_Chi2

-> Additional_macro

All of these codes (excluding "Additional Macro") could be invoked by simply lauching the python script ::
"**project_pre/data_txt/BEIJING_Aqi/**execute_test_python.py", or
"**project_pre/data_txt/BEIJING_Aqi/**execute_test_python_windows.py" if you work on Windows OS. (Let me show you!)

1.4 Data analysis frame :: ROOT & PYTHON (Additional_macro)

Also , there are more macros includeing plotting and none plotting, for single & two sample ::

1. "project_pre/func/c4_Fit_Poisson_histo_plotting.py" (This is plotting Possion distribution code, but may not work correctly)
2. "project_pre/func/c5_double_sample_mean_Zdistribution.py" (as title)
3. "project_pre/func/c6_single_sample_proportion_hypothesis.py"
4. "project_pre/func/c6_single_sample_proportion_distribution.py"
5. "project_pre/func/c6_double_sample_proportion_hypothesis.py"
6. "project_pre/func/c6_double_sample_proportion_distribution.py"
7. "project_pre/func/c7_single_sample_variance_distribution.py"
8. "project_pre/func/c7_single_sample_variance_hypothesis.py"
9. "project_pre/func/c7_double_sample_variance_distribution.py"
10. "project_pre/func/c7_double_sample_variance_hypothesis.py"
11. "project_pre/func/c8_ANOVA_hypothesis.py"

These macros are not automatically invoked with "execute_test*.py",
One have to import them manually. (But most of these takes "Hist DATA" as one of its input!)

2. TO DO & Discussion

- 1. Expand DATA contents (Temperature, wind speed, humidity…)
- 2. Complementing macros. (“adding cut”… what else?)
- 3. What else do we need ?