BSRN WORKSHOP 2022

Mathieu DELSAUT
mathieu.delsaut@univ-reunion.fr

ENERGY LAB

WHAT ABOUT COEFFS ?

An empirical approach for determination of coefficients from Long & Shi automated quality assessment, applied on RUN and PAL stations.

JOURNEY

1. Quality control method
    - Reminder of the Long & Shi Article
    - Empirical approach to estimate coefficients
2. Coefficient study tools : pybsrnqc
    - Different plot and displays
    - QCrad
    - Interactive tool to determine coefficient
    - Perspectives
3. Application on PAL and RUN Station

INTRODUCTION

The laboratory EnergyLab becomes BSRN station candidate in 2018.

We had to implement quality control assessment.

We decide to use Long & Shi article but some level depends of coefficients.

We hired Maelle Baronet, Ecole des Mines post-graduate Engineering School to work on the suject.

The report with full details is acceccible on BSRN Website

QUALITY CONTROL METHOD

In 2008, Long & Shi define a ways to controle BSRN station. The metodology is called QCrad and implemented into the BSRN toolbox.

The Open Atmospheric Science Journal, 2008, *2*, 23-37

An Automated Quality Assessment and Control Algorithm for Surface Radiation Measurements

C.N. Long* and Y. Shi

Pacific Northwest National Laboratory, Richland, WA, USA

Abstract: We present an automated algorithm for testing surface broadband radiation measurements to detect erroneous data. The methodology has been developed using data from the US Department of Energy Atmospheric Radiation Measurement (ARM) Program. The testing includes physically possible limits as determined by the World Meteorological Organization (WMO) Baseline Surface Radiation Network, as well as user configurable limits based on climatological analysis of data collected at the measurement site. The algorithm can be run in near real time, or more typically on a daily basis. Additionally, longer monthly or yearly runs can be used to assess more subtle tendencies and problems in the data through evaluation of daily summaries of quality flagging.

INTRODUCTION

Our surface radiation quality testing methodology (hereafter called QCRad) in essence uses climatological analyses of the surface radiation measurements themselves to define reasonable limits for testing the data for unusual data values. The main assumption here is that the majority of the climatological data are "good" data, which for field sites operated with nominal care such as those of the US Department of

- AU = Earth-Sun distance in Astronomical Units [1 AU = mean E-S distance]
- $S_a = S_0/AU^2$ = solar constant adjusted for Earth-Sun distance
- σ = Stephan-Boltzman constant = 5.67×10^{-8} Wm^{-2} K^{-4}
- T_a = air temperature in Kelvin [must be in range 170K < T_a < 350K]

Long & Shi define different levels of data quality with flags.

QC flag describe the severity and direction (too low or too hight) :

Flag Value:	Related to Type test:
5-6	Global Physical Limits (PP)
3-4	User configurable (UC2) 2nd level tests, also LW Tc and Td tests
1-2	User configurable (UC1) 1st level tests and non-definitive tests

Long & Shi define different levels of data quality with flags.

QC flag describe the severity and direction (too low or too hight) :

QC1-QC6 (GSW, Diffuse SW, Direct SW, SWup, LWdn, and LWup basic limits tests)

-1-missing data or test not possible

0-No test failures

1-too low (UC1)

2-too high (UC1)

3-too low (UC2)

4-too high (UC2)

5-too low (PP)

6-too high (PP)

9-“tracker off” (QC2 and QC3)

19 QC Rules explained, some coefficient dependent. Example here for global :

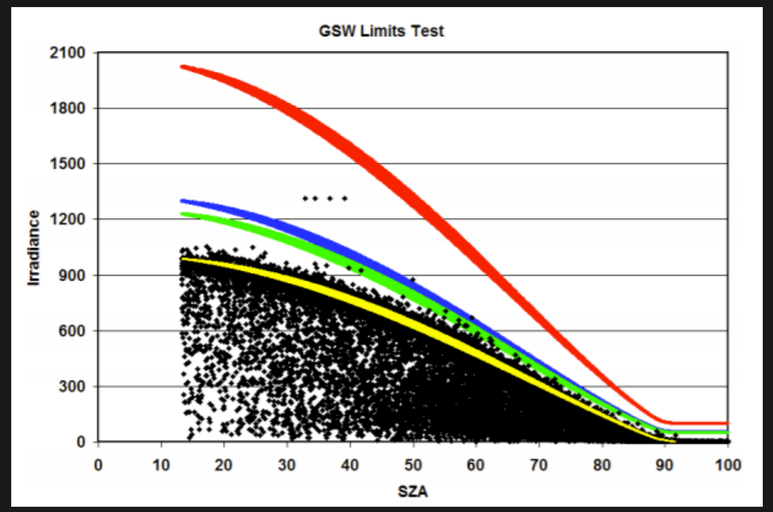
19 QC Rules explained, some coefficient dependent. Example here for global :

$$S_a \times C_1 \times \mu_0^{1.2} + 50 W m^{-2} (1^{st} level)$$

$$S_a \times D_1 \times \mu_0^{1.2} + 55 W m^{-2} (2^{nd} level)$$

$$S_a \times 1.5 \times \mu_0^{1.2} + 100 W m^{-2} (Physical\ limits)$$

19 QC Rules explained, some coefficient dependent. Example here for global :



"The question which arises now is the following : how can one determine these coefficients ? Apparently there is no precise method or approach to be followed to determine the coefficients of a given station."

EMPIRICAL APPROACH TO ESTIMATE COEFFICIENTS

WHICH POINTS ARE OUTLIERS?

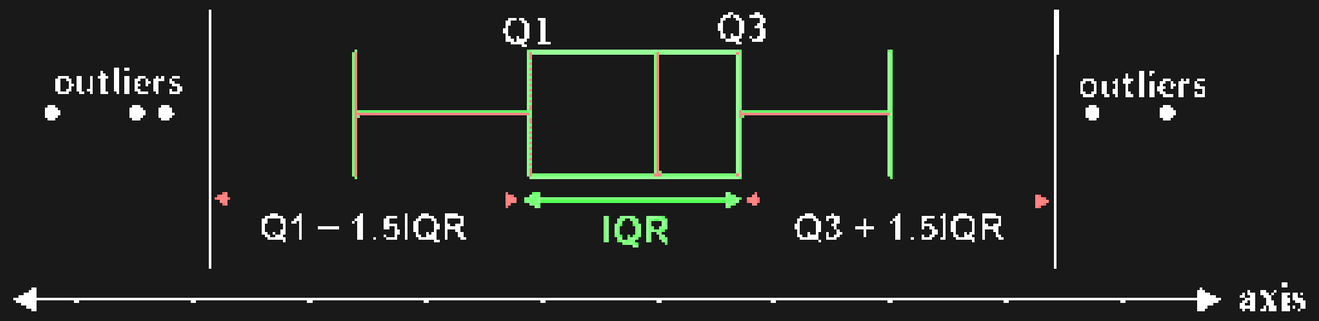
Our definition :

"Outliers are by definition infrequent observations, i.e. points that do not follow the characteristic distribution of the rest of the data"

SOME STATISTIC METHODS TO FIND OUTLIERS

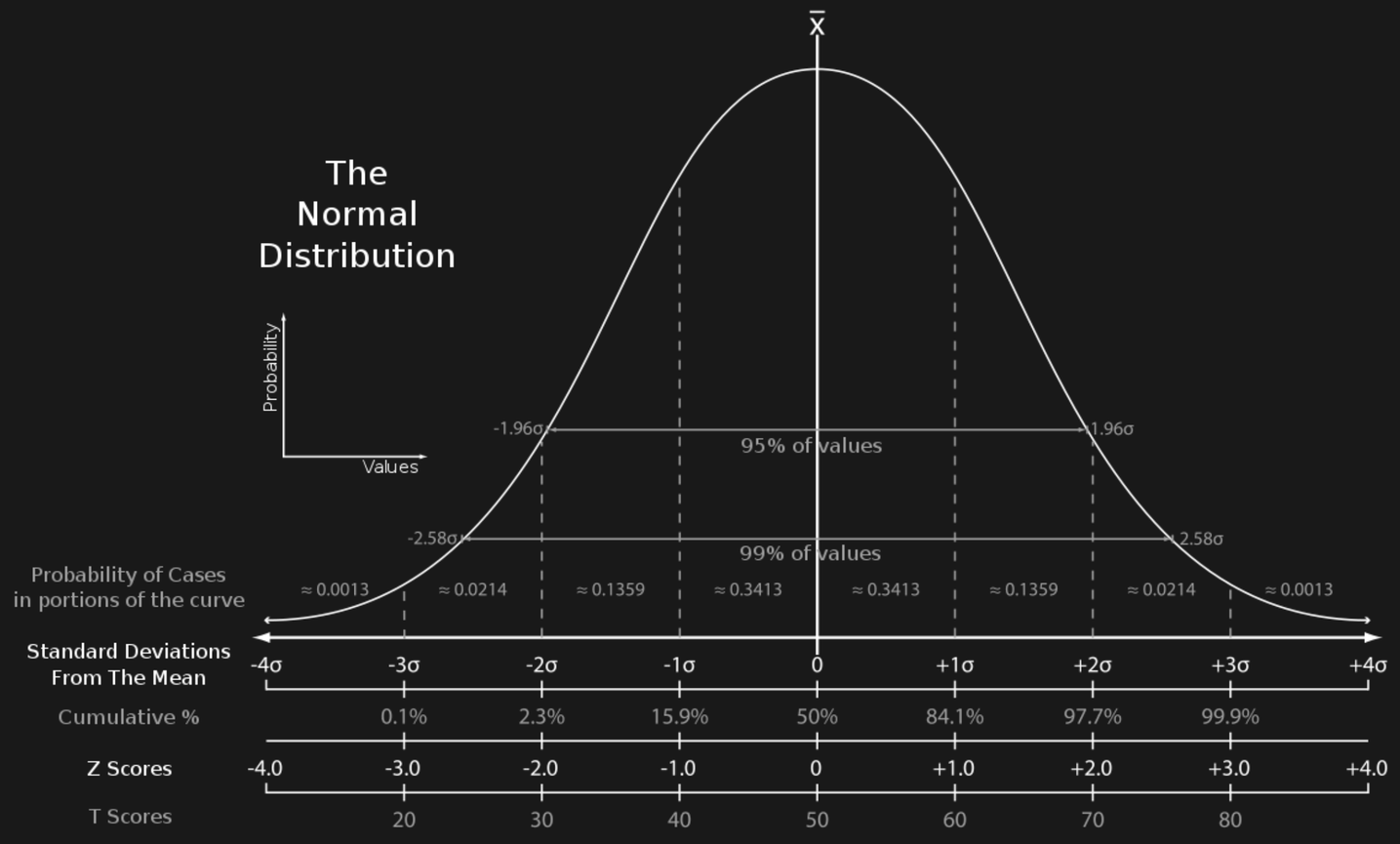
SOME STATISTIC METHODS TO FIND OUTLIERS

Inter quartile range (IQR)



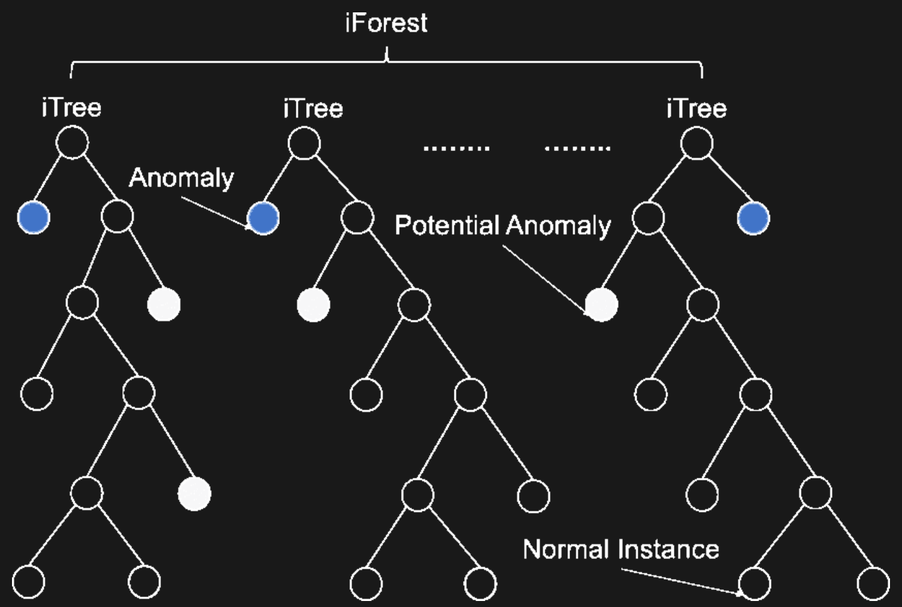
SOME STATISTIC METHODS TO FIND OUTLIERS

Z-Score



SOME STATISTIC METHODS TO FIND OUTLIERS

Isolation forest



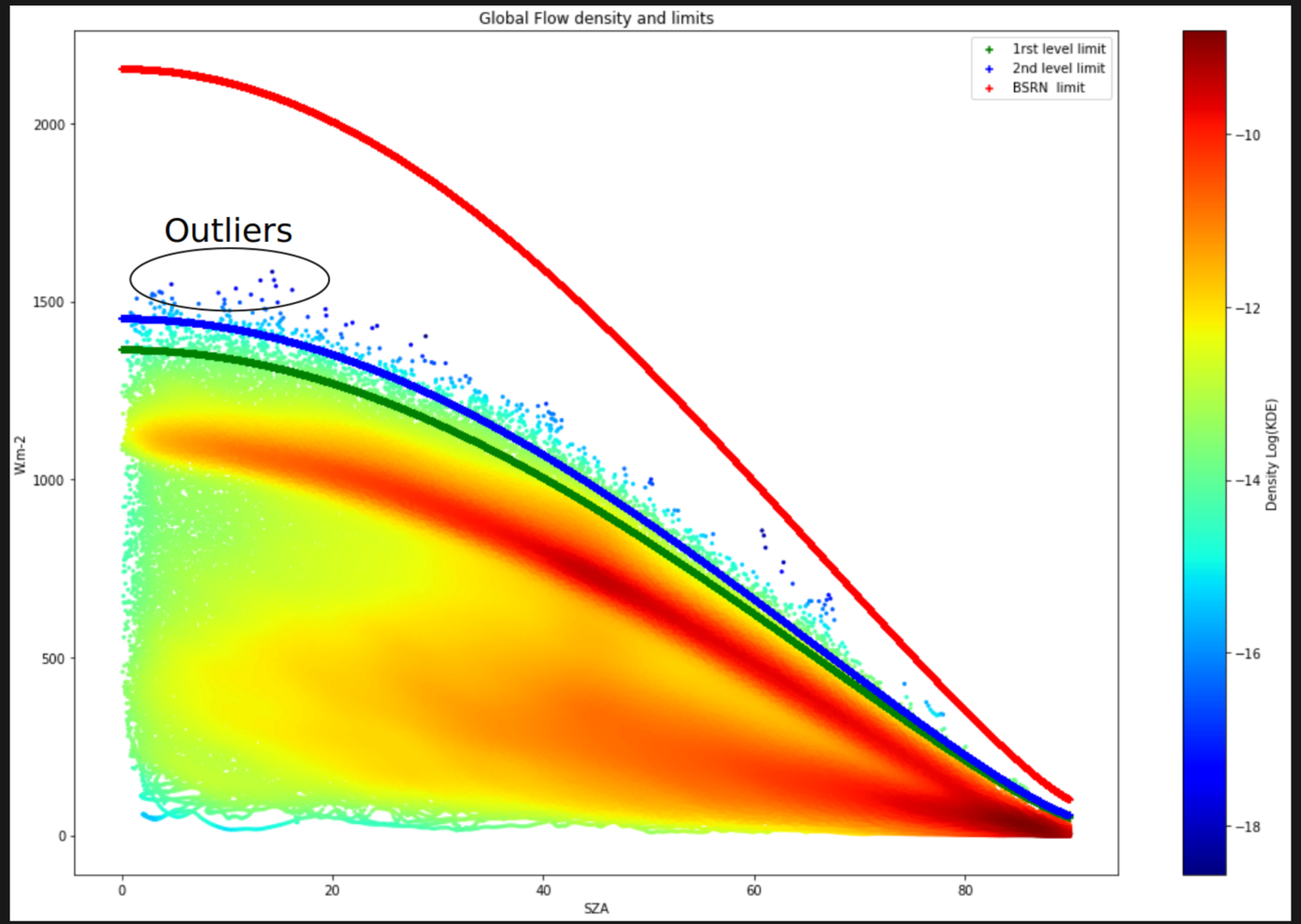
KERNEL DENSITY ESTIMATION (KDE)

let $(x_1, ..., x_n)$ independent and identically distributed and $h > 0$ a smoothing parameter. The KDE is

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

with a gaussian kernel $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$

The more neighbors a point has in its surroundings, the higher its density value.



Here, KDE on a two year datastet for Global.

If we choose a KDE threshold, and fix a coefficient, we can define the accuracy score of the QC Rule respecting the KDE threshold as

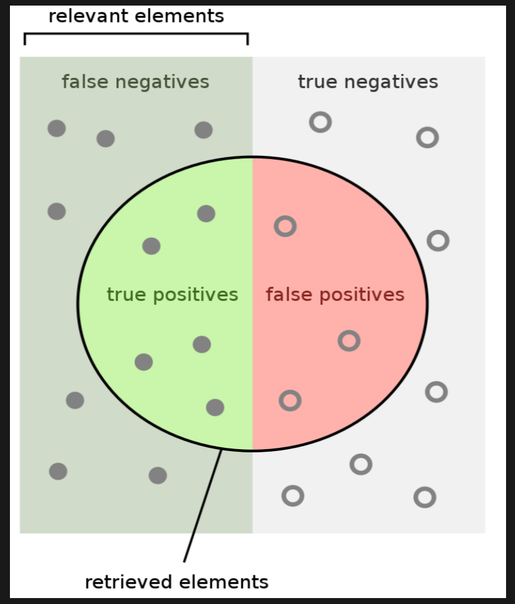
If we choose a KDE threshold, and fix a coefficient, we can define the accuracy score of the QC Rule respecting the KDE threshold as

- Counting the outliers detected by the curve
- Comparing to outliers identified by the KDE

If we choose a KDE threshold, and fix a coefficient, we can define the accuracy score of the QC Rule respecting the KDE threshold as

$$Accuracy = \frac{Correct\ Predictions}{All\ Predictions}$$

If we choose a KDE threshold, and fix a coefficient, we can define the accuracy score of the QC Rule respecting the KDE threshold as



If we choose a KDE threshold, and fix a coefficient, we can define the accuracy score of the QC Rule respecting the KDE threshold as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

If we choose a KDE threshold, and fix a coefficient, we can define the accuracy score of the QC Rule respecting the KDE threshold as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

BUT : it will not work because we want to eliminate a few points over a large amount.

Accuracy will grow with TN

So we define :

$Precision = \frac{TP}{TP+FP}$, tends toward 1 when no FP

So we define :

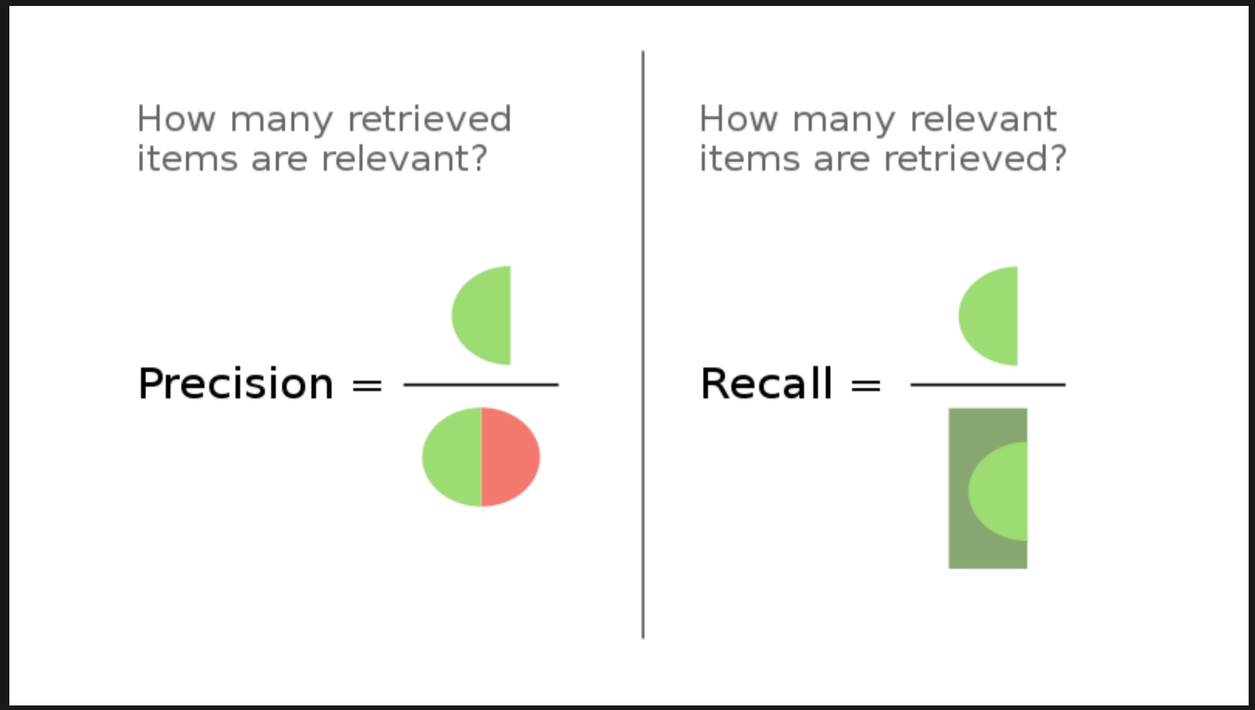
$Precision = \frac{TP}{TP+FP}$, tends toward 1 when no FP

$Recall = \frac{TP}{TP+FN}$, tends toward 1 when no FN

So we define :

$Precision = \frac{TP}{TP+FP}$, tends toward 1 when no FP

$Recall = \frac{TP}{TP+FN}$, tends toward 1 when no FN



So we define :

$Precision = \frac{TP}{TP+FP}$, tends toward 1 when no FP

$Recall = \frac{TP}{TP+FN}$, tends toward 1 when no FN

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

with a fix threshold, we keep the QC rules coefficients with the highest F1-score.

METHOD SUMMARY

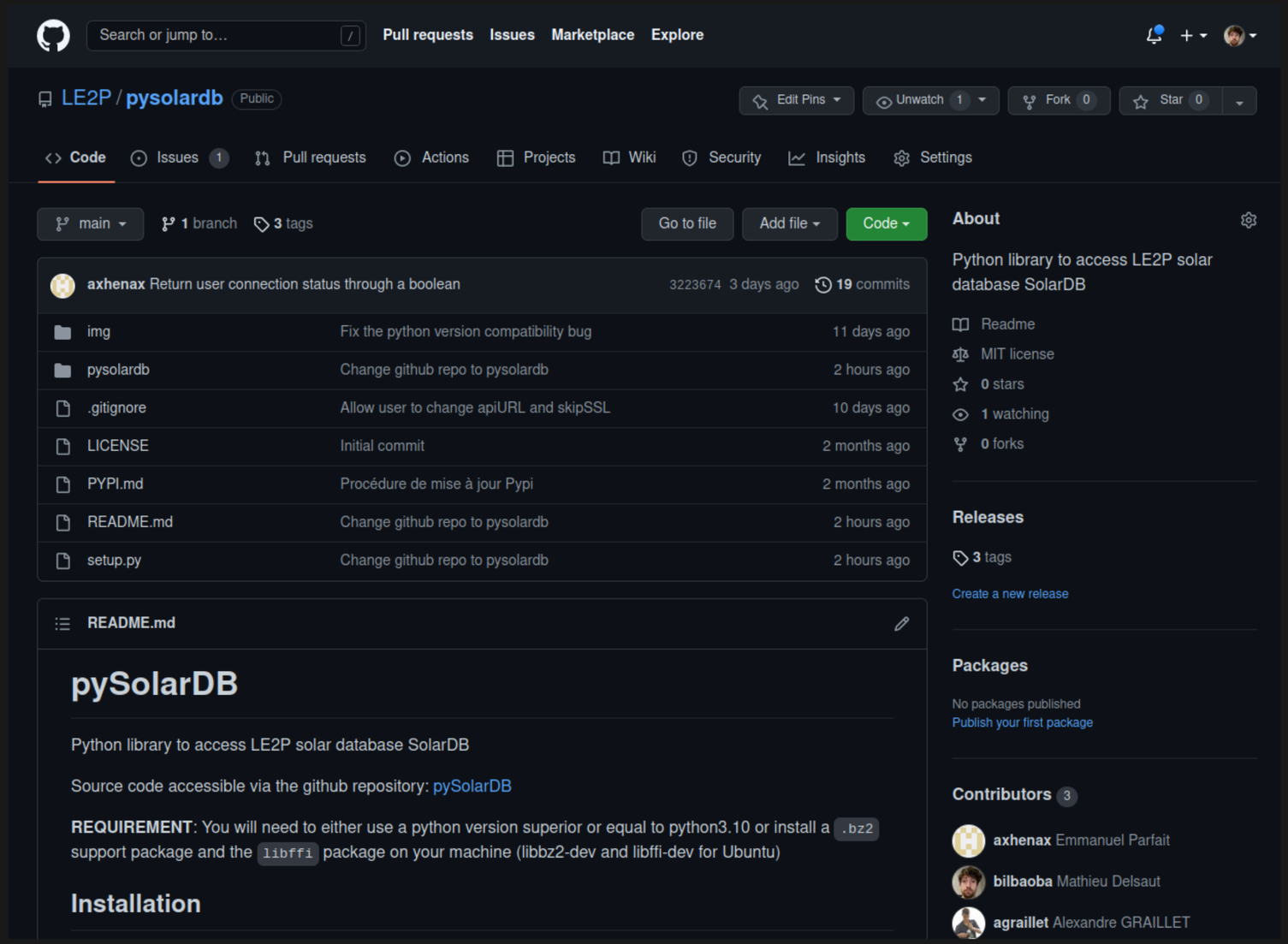
1. Choice of the QC and the level studied
2. Computation on the Gaussian KDE on the dataset
3. Choice of the density threshold for outliers
4. Test of a range of coeffs and calculation of F1 score
5. Selection of the best coefficient

ADVANTAGES / LIMITATIONS :

- (+) Easy to compute
- (+) Usable on other stations
- (-) Empirical aspect on the choice of the threshold

THE PYBSRNQC LIBRARY

This module allows you to study the data and the limits given by the coefficients thanks to visualization and calculation tools.



Installation

```
// using pip
pip install pybsrnqc
```

It operates on a repository containing monthly csv datafiles :

timestamp	global2_avg	direct_avg	diffuse_avg	downward_avg	temperature
2019-08-01 00:00:00	145.1	665.300	69.340	365.65	25
[...]	[...]	[...]	[...]	[...]	[...]
2019-08-31 23:59:00	145.1	665.300	69.340	365.65	25

Python imports

```
from pybsrnqc import plot_limits as pl
from pybsrnqc import open_data as od
from pybsrnqc import qc_functions as qcf
from pybsrnqc import coef_study as cs
from pybsrnqc.config import Coef, Station
```

Python imports

```
from pybsrnqc import plot_limits as pl
from pybsrnqc import open_data as od
from pybsrnqc import qc_functions as qcf
from pybsrnqc import coef_study as cs
from pybsrnqc.config import Coef, Station
```

You can define the location of your station,

```
# Coefficients values and station location
station = Station()
station.LAT = 48.711951760391926
station.LON = 2.207638279957924
station.ALT = 159.0
station.TZ = "Europe/Paris"
```

Python imports

```
from pybsrnqc import plot_limits as pl
from pybsrnqc import open_data as od
from pybsrnqc import qc_functions as qcf
from pybsrnqc import coef_study as cs
from pybsrnqc.config import Coef, Station
```

You can define the location of your station,

```
# Coefficients values and station location
station = Station()
station.LAT = 48.711951760391926
station.LON = 2.207638279957924
station.ALT = 159.0
station.TZ = "Europe/Paris"
```

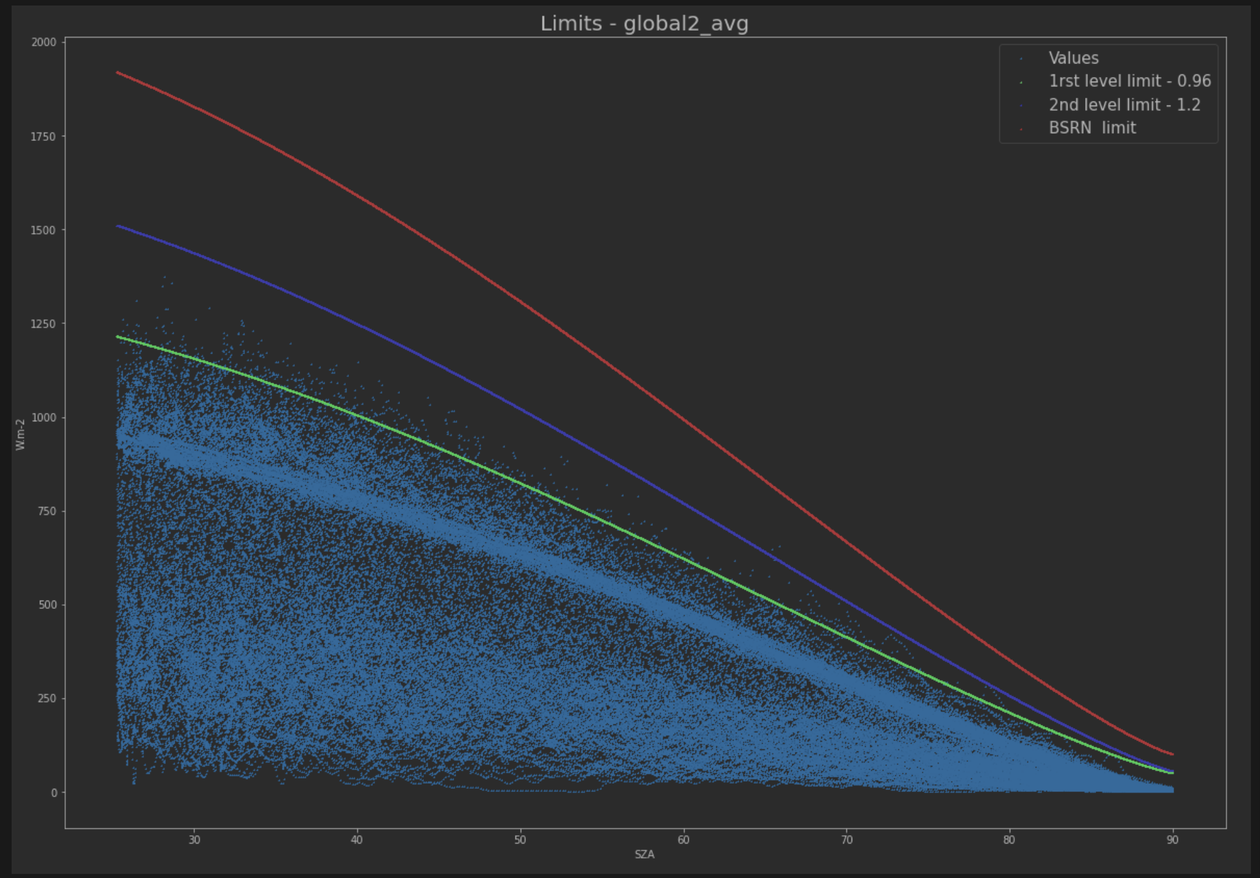
and import your data.

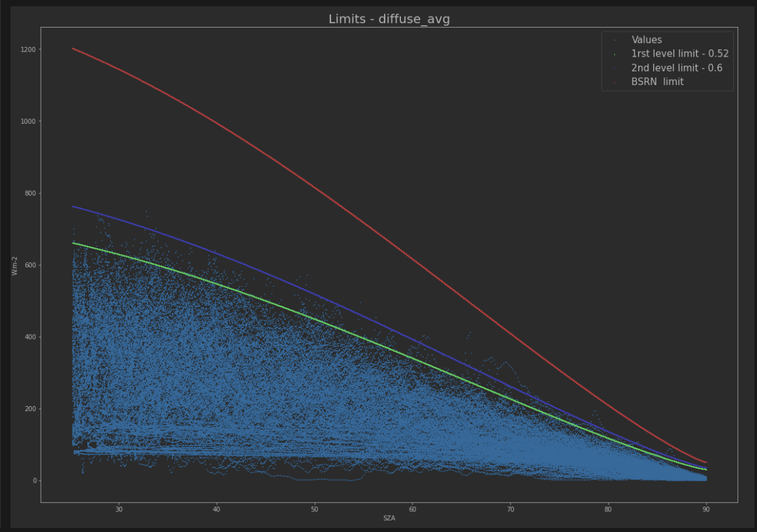
```
# Get data
path = './dataset/'
df = od.open_all(path,period=['202105','202108'], station=station)
```

Function compute the SZA. Result into a dataframe (df).

And then plot standard limits on the loaded dataset

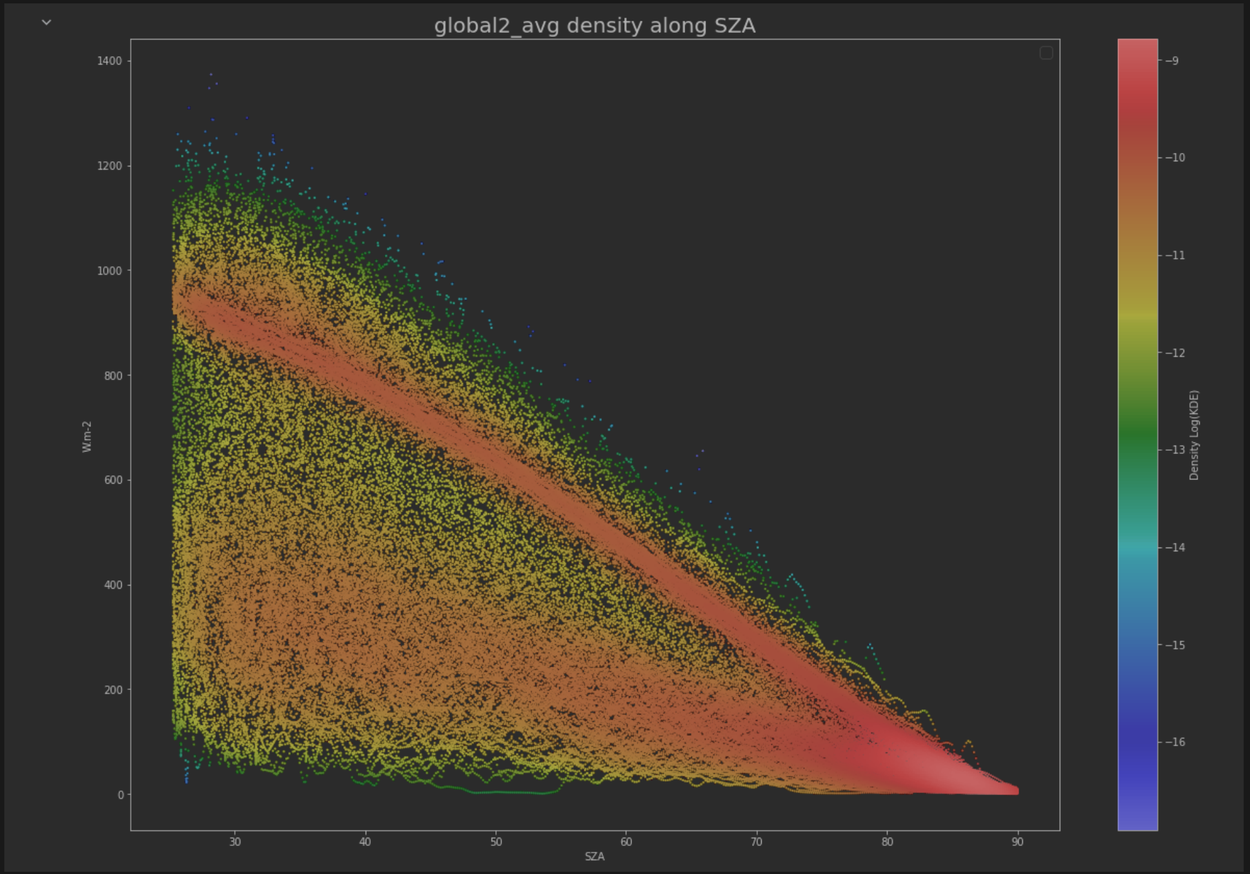
```
1 # Plotting data and BSRN limits
2 coefs = Coef()
3 pl.limit_plot(df, qcf.QC1(), coefs)
4 pl.limit_plot(df, qcf.QC2(), coefs)
```





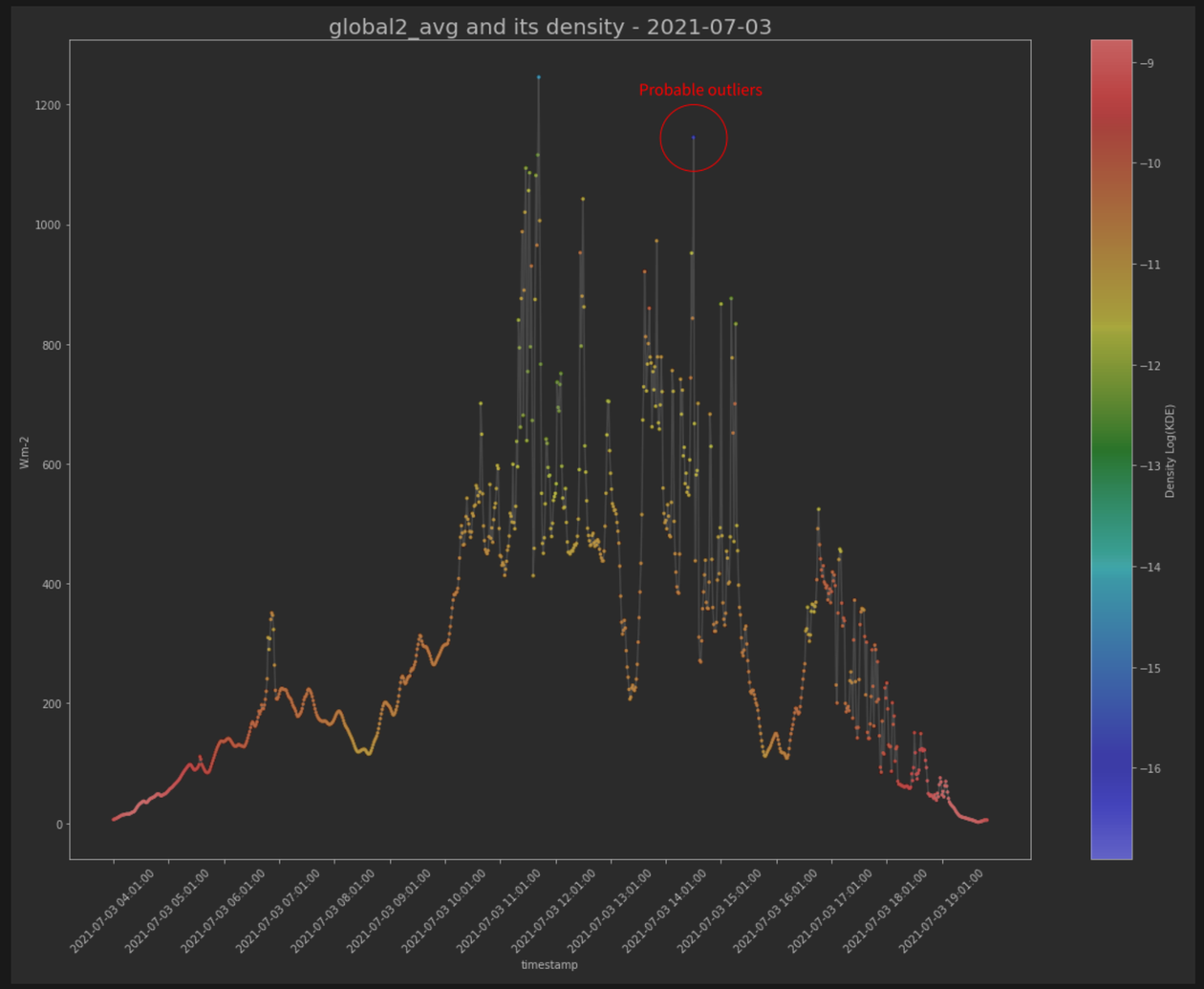
Compute the KDE for DF and choose a threshold

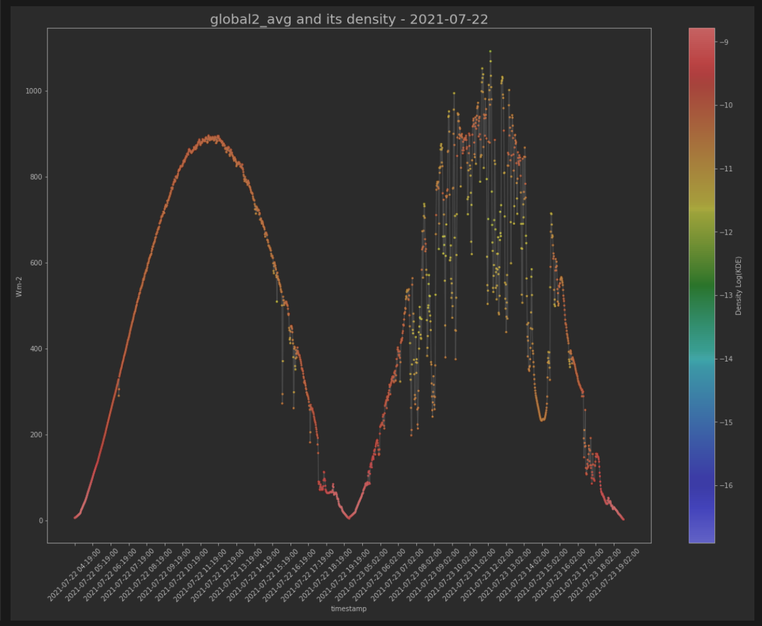
```
1 # Calculating kernel density for the dataset
2 log_kernel = pl.kde_computing(df, qcf.QC1())
```



You can even look at series over KDE

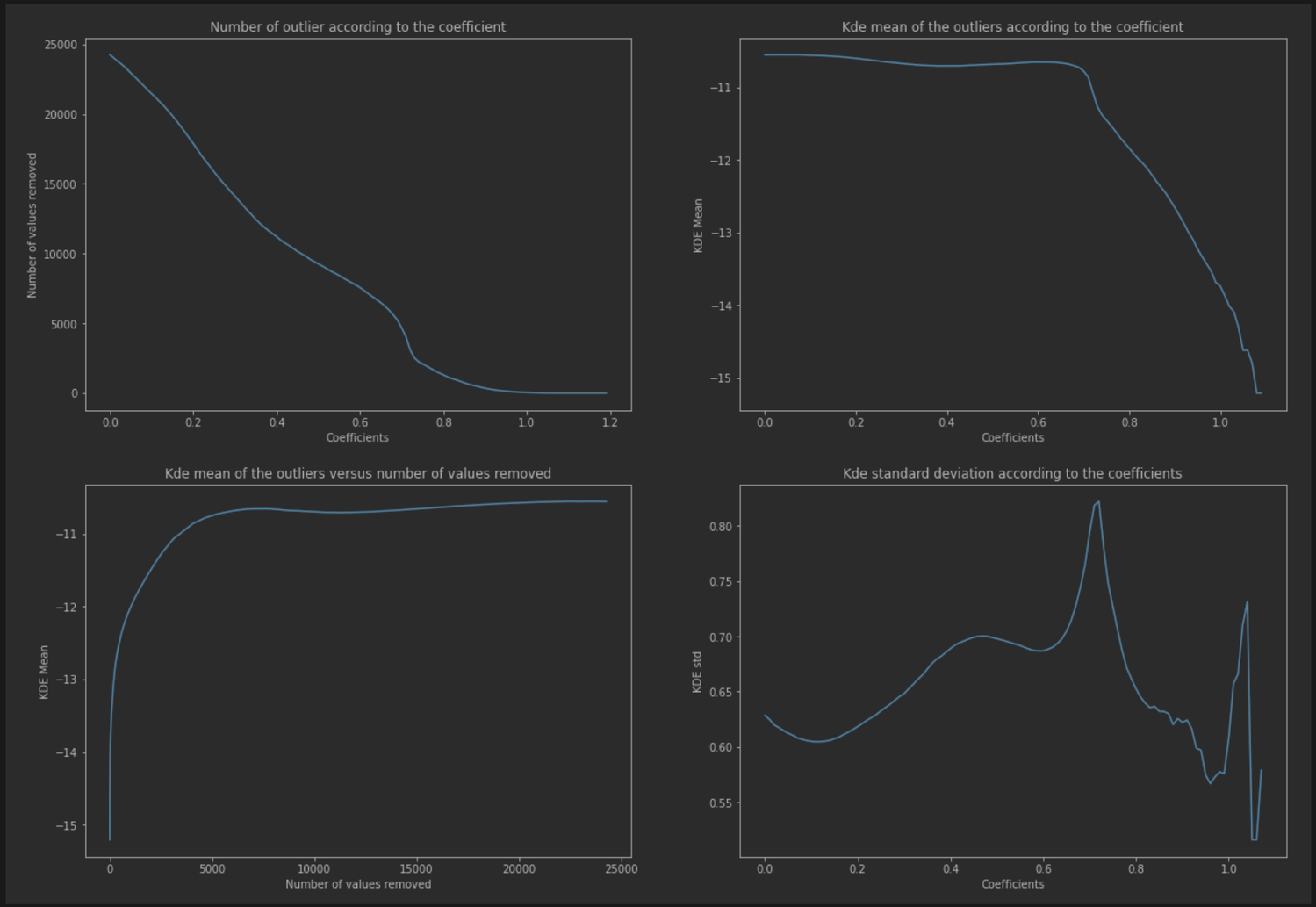
```
1 # Plotting time series on a specific date
2 pl.plot_series_kde(df, log_kernel, qcf.QC1(), begin='2021-07-03', end='2021-07-04')
3 pl.plot_series_kde(df, log_kernel, qcf.QC1(), begin='2021-07-22', end='2021-07-24')
```





Helpers to decide which threshold

```
1 # Observing the effect of the coefficient value over certain indicators
2 df_var = cs.coef_variation(df, log_kernel, qcf.QC1())
```



And then, you can finally compute the best coefficient respecting threshold

```
# Finding the best coefficient for a density threshold given
df_score, score = cs.calc_coef(df, log_kernel, qcf.QC1(), threshold=-14, level='level_2')
```

```
Upper limit
        D1  accuracy_score  precision_score  recall_score  f1_score
104   1.04        0.999722         0.545455      0.666667       0.6
```

And then, you can finally compute the best coefficient respecting threshold

```
# Finding the best coefficient for a density threshold given
df_score, score = cs.calc_coef(df, log_kernel, qcf.QC1(), threshold=-14, level='level_2')
```

```
Upper limit
        D1  accuracy_score  precision_score  recall_score  f1_score
104   1.04        0.999722         0.545455      0.666667       0.6
```

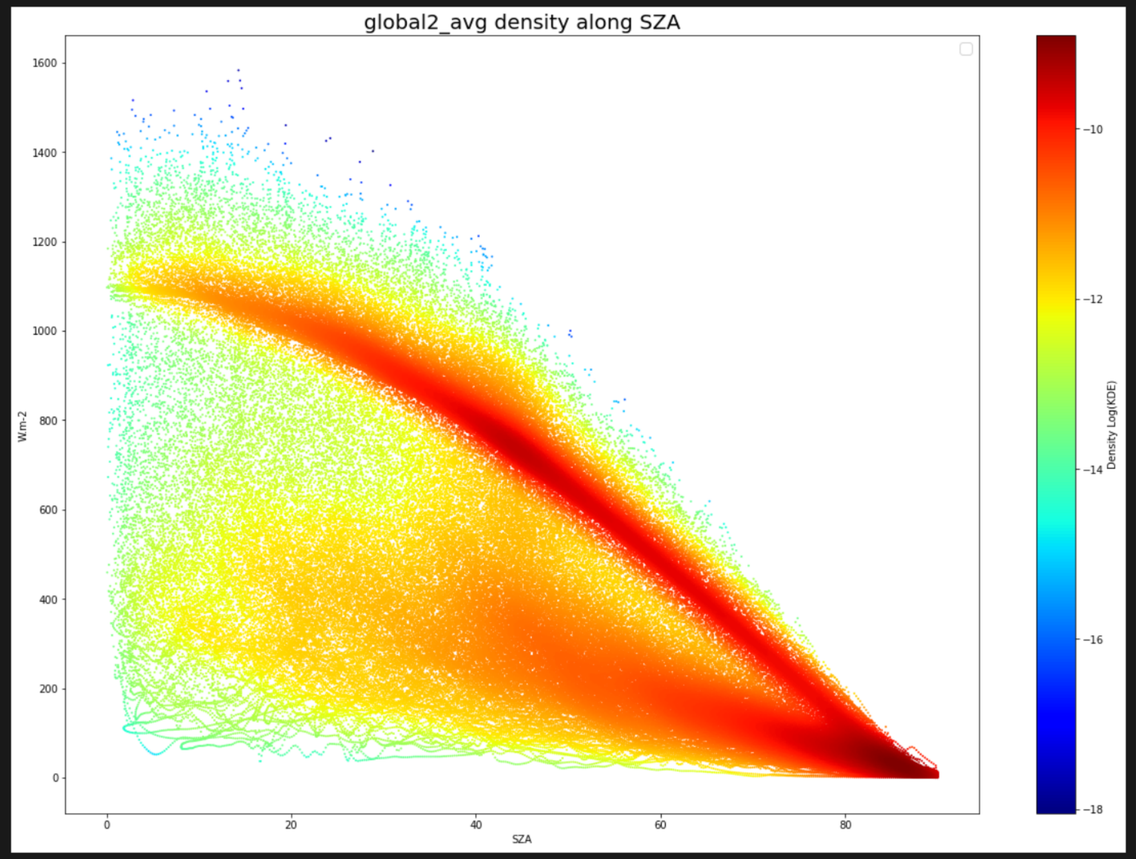
An automatic way exist

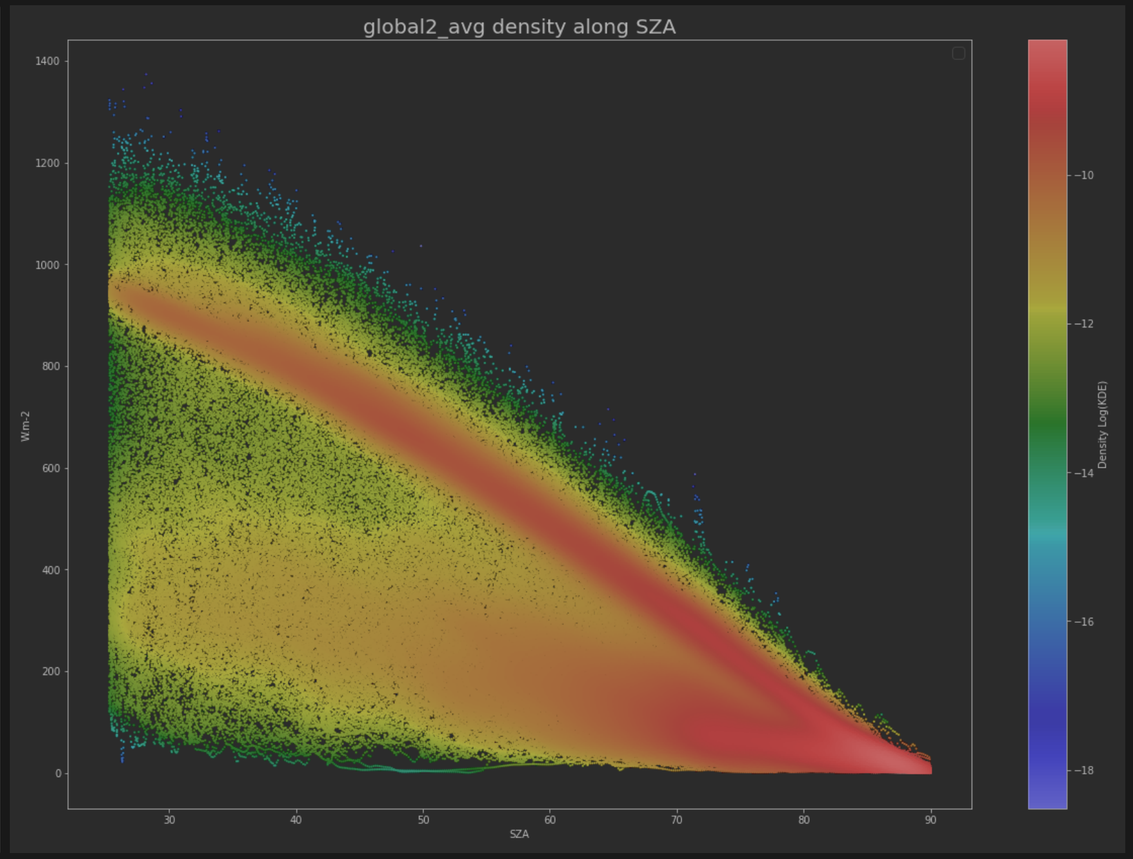
```
from pybsrnqc import coef_calculator as cc
name_coef, coef = cc.compute('./dataset')
```

APPLICATION ON PAL AND RUN STATION

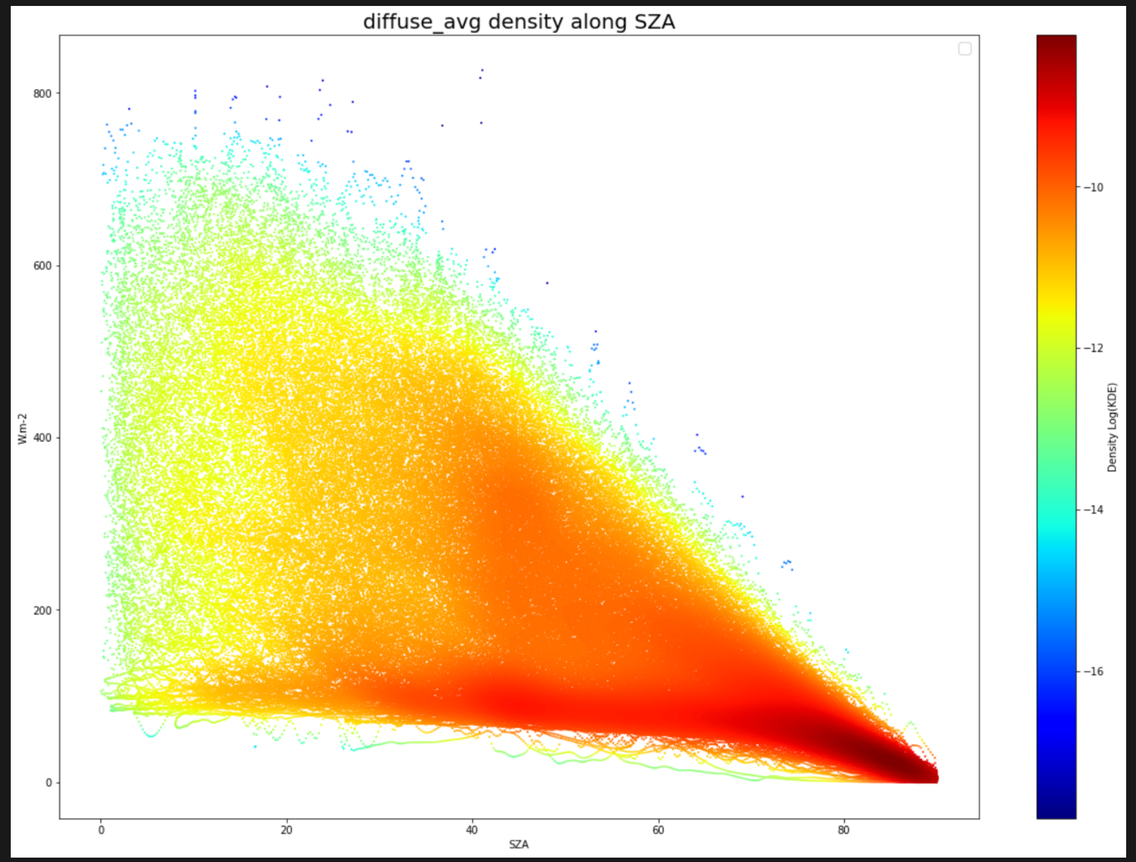
LET COMPARE KDE FOR BOTH SITE

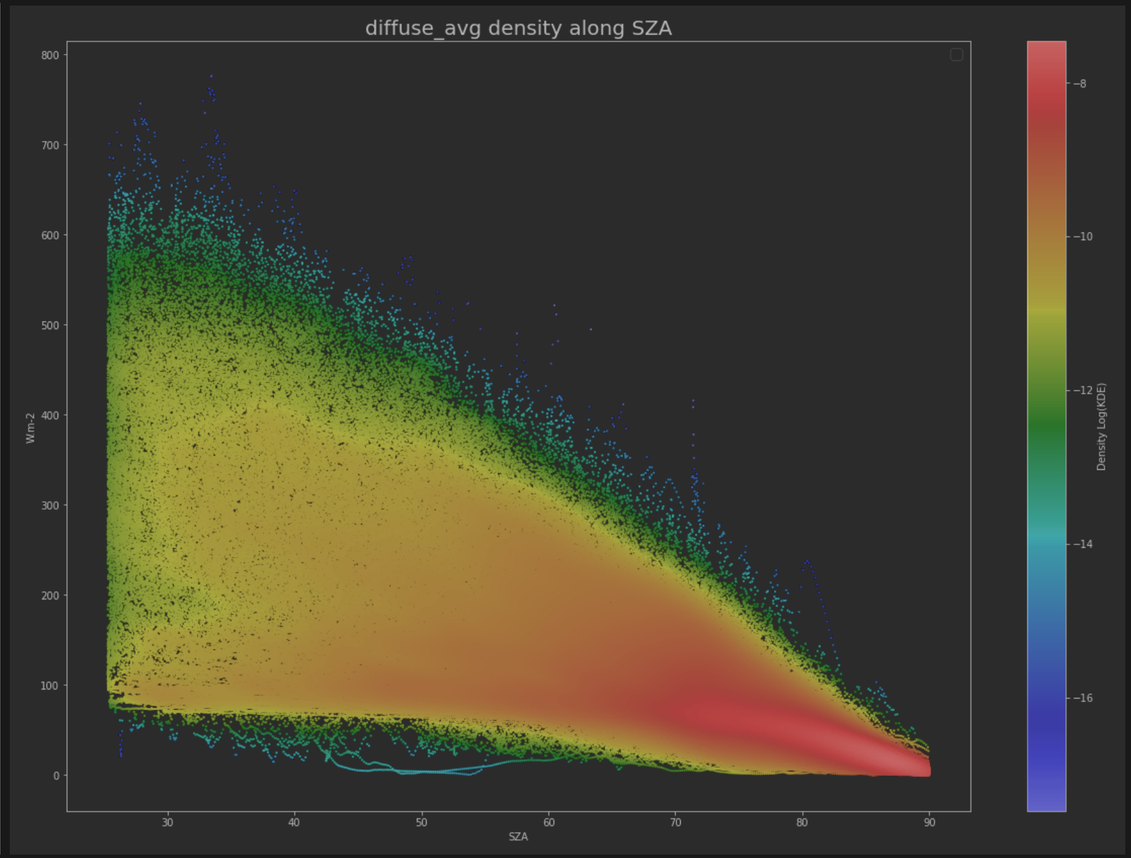
KDE QC1 : RUN (left) vs PAL (right)



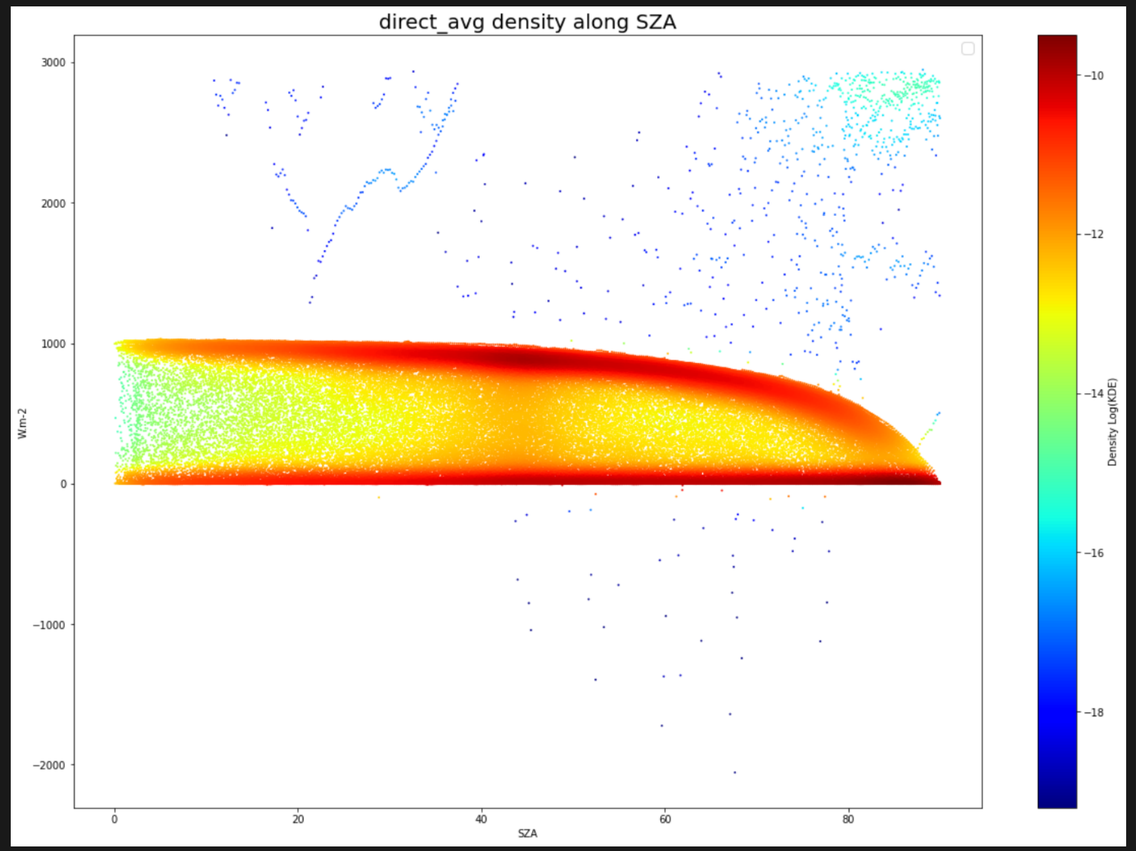


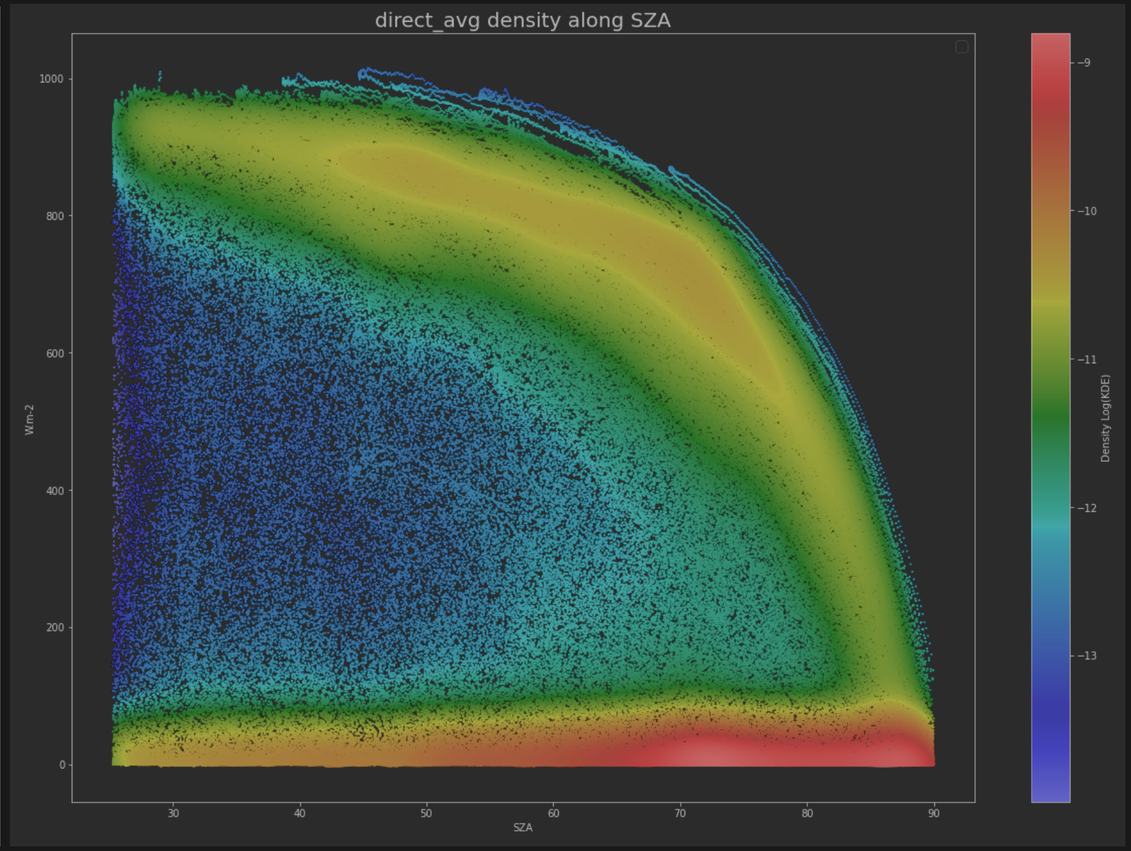
KDE QC2 : RUN (left) vs PAL (right)



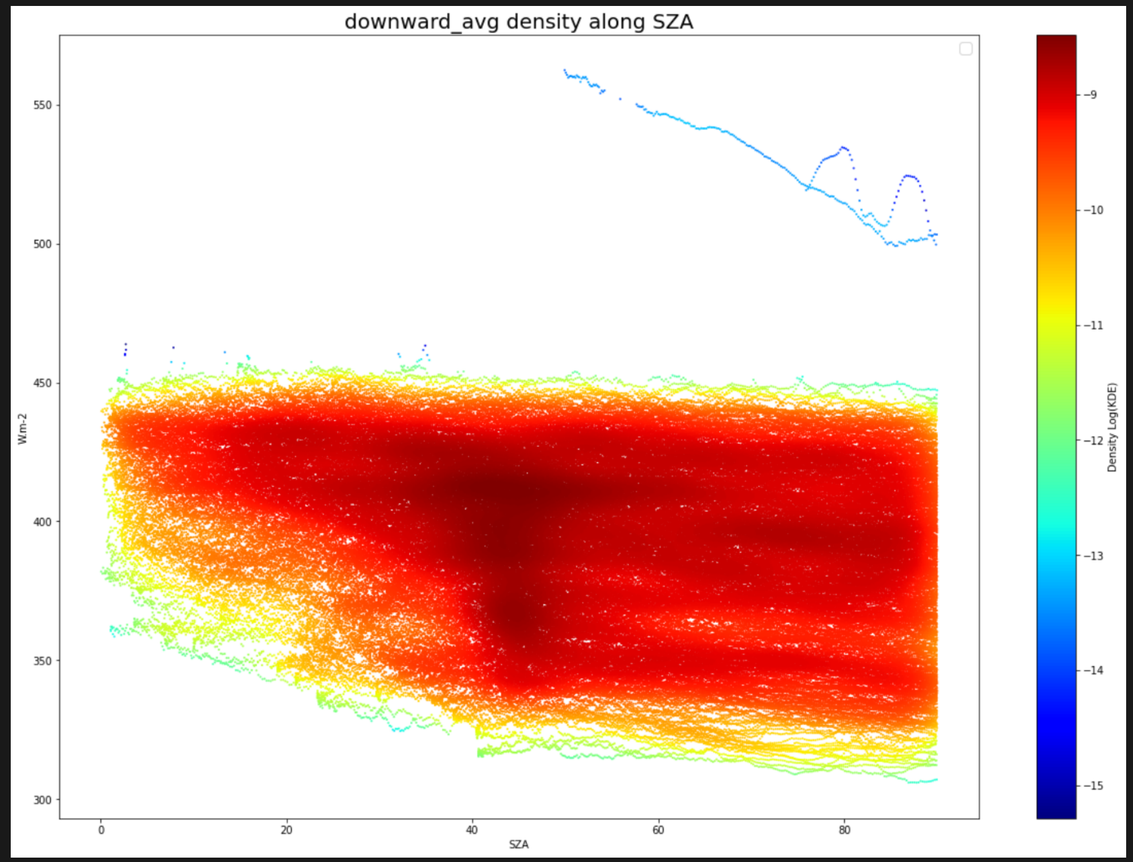


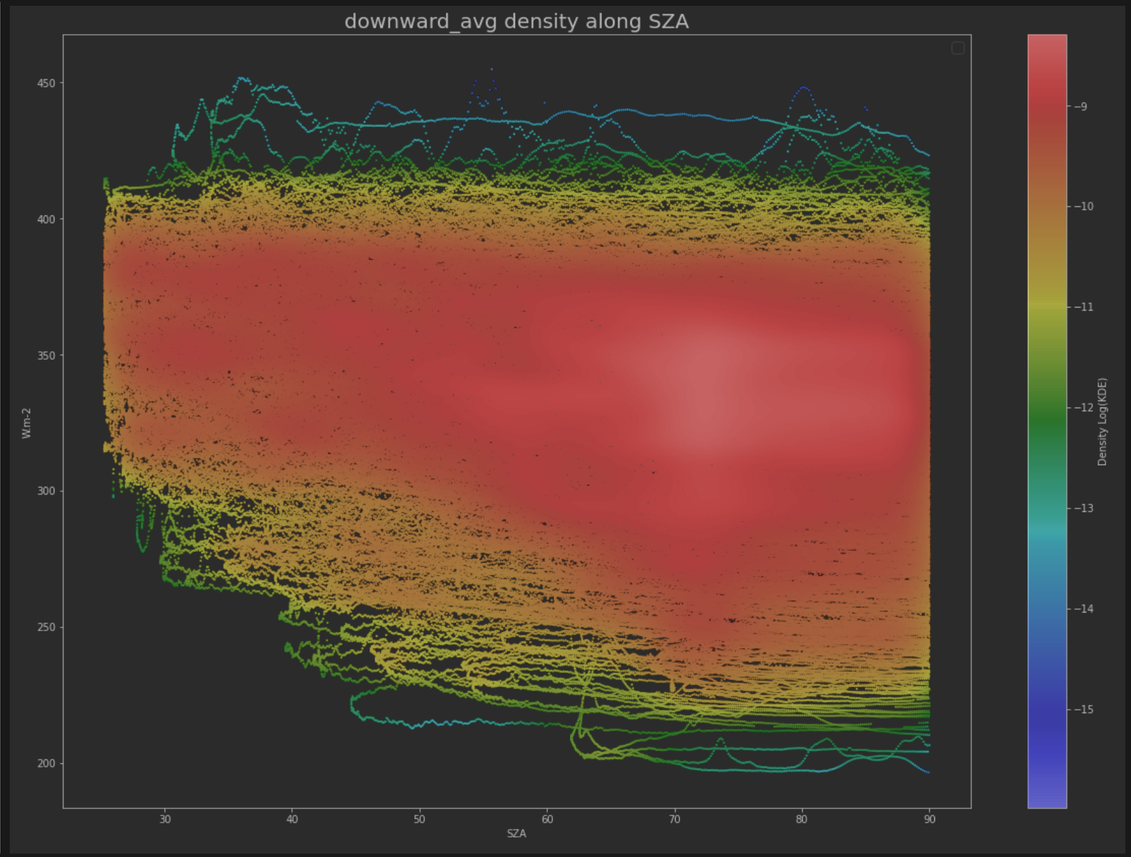
KDE QC3 : RUN (left) vs PAL (right)



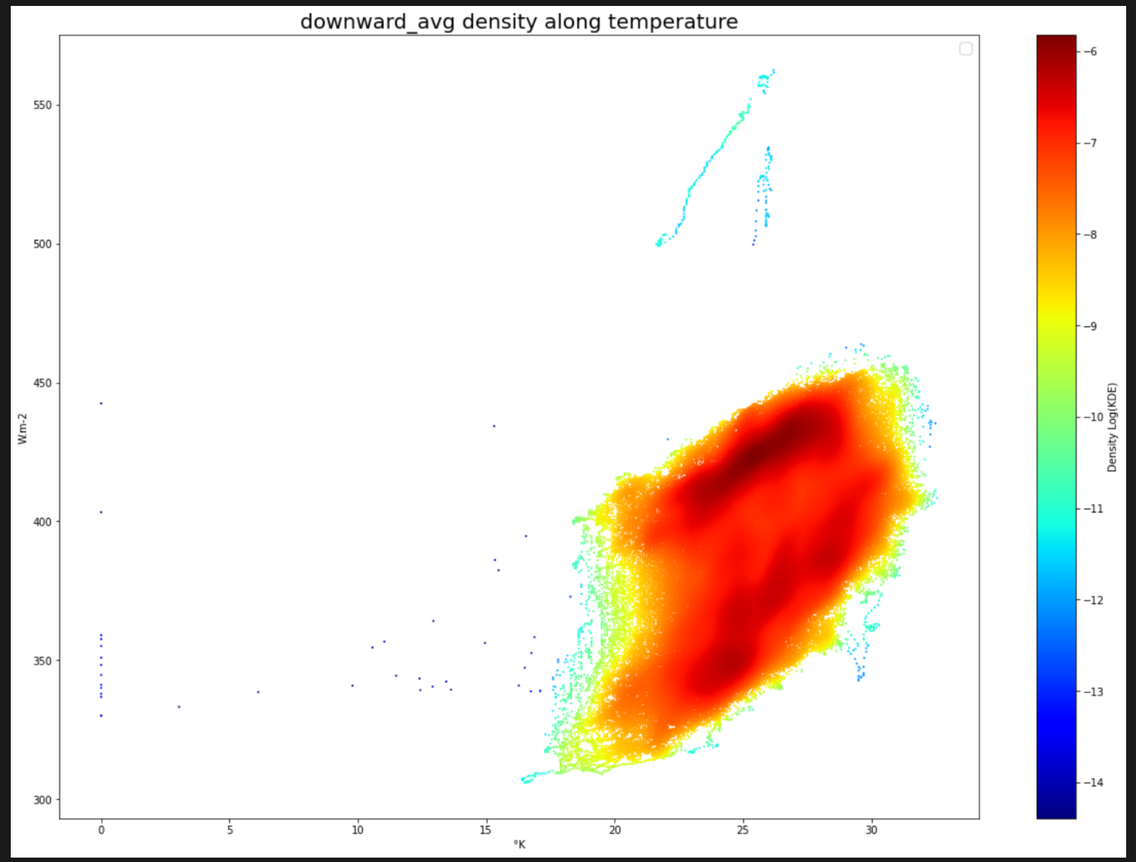


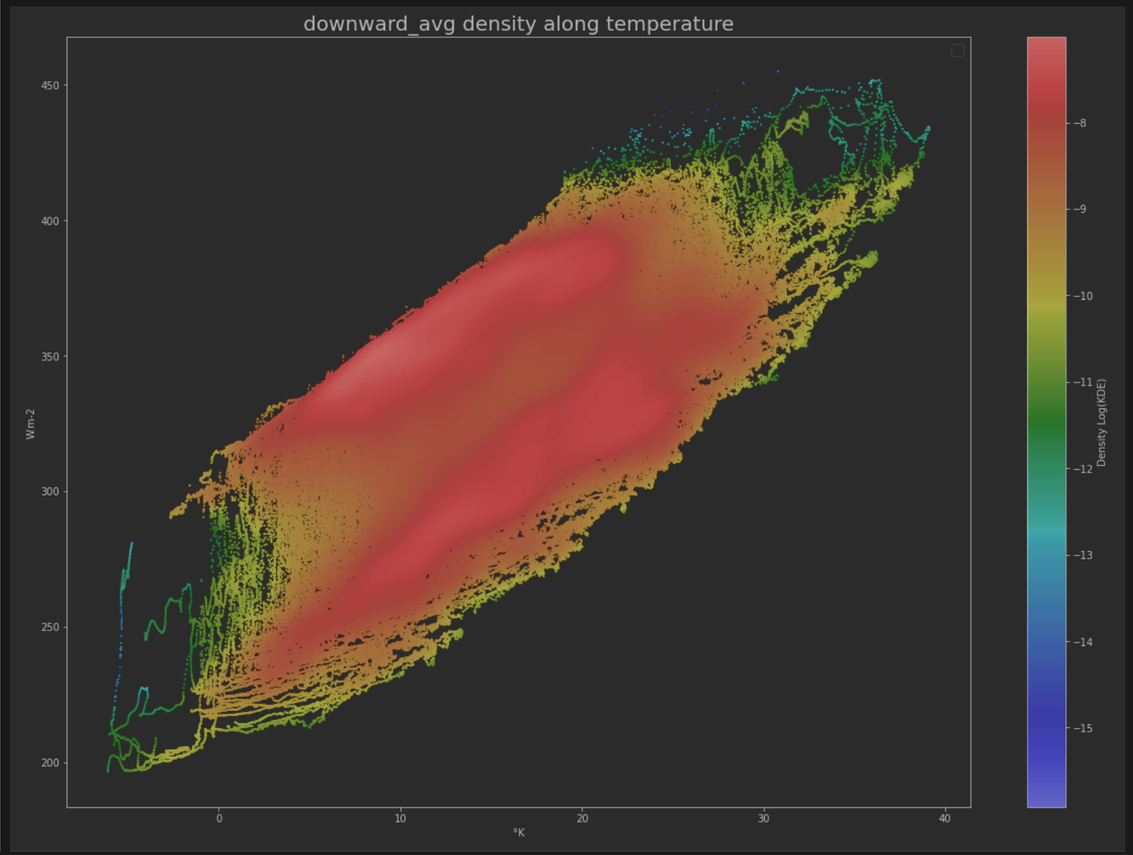
KDE QC5 : RUN (left) vs PAL (right)





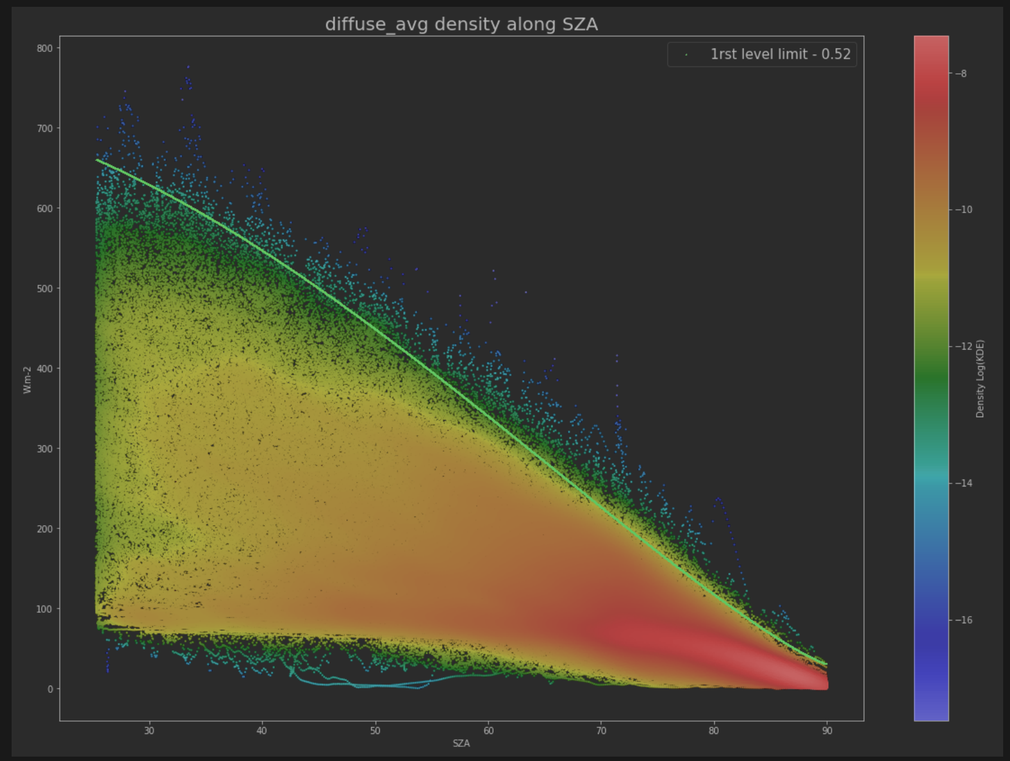
KDE QC10 : RUN (left) vs PAL (right)

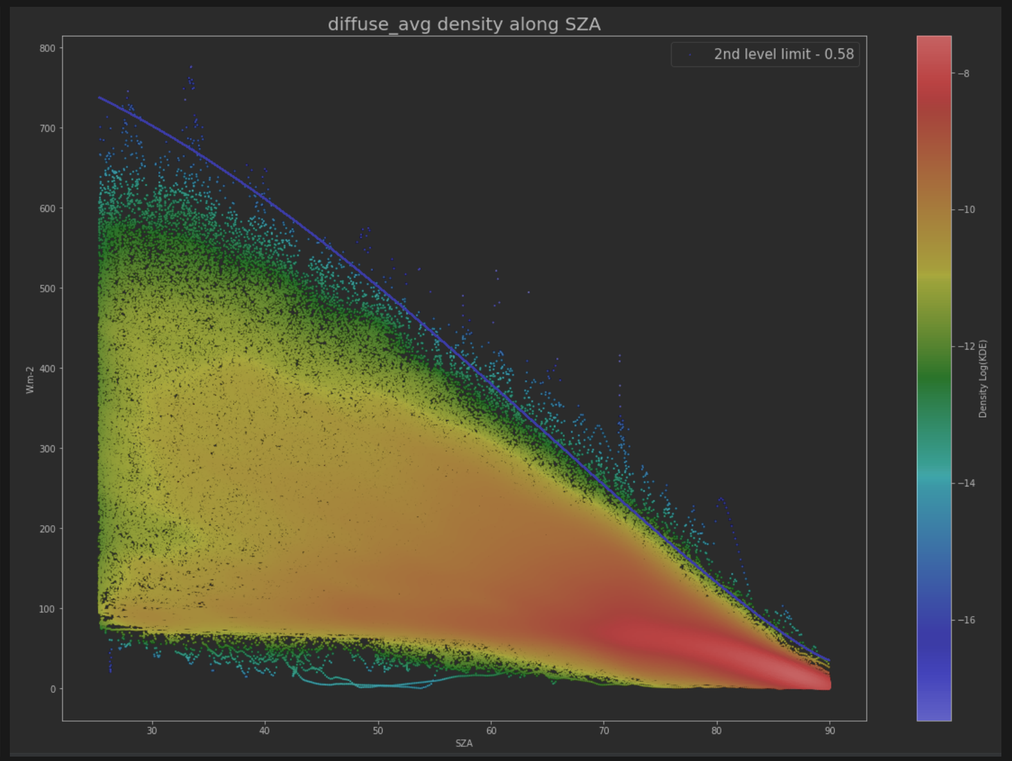




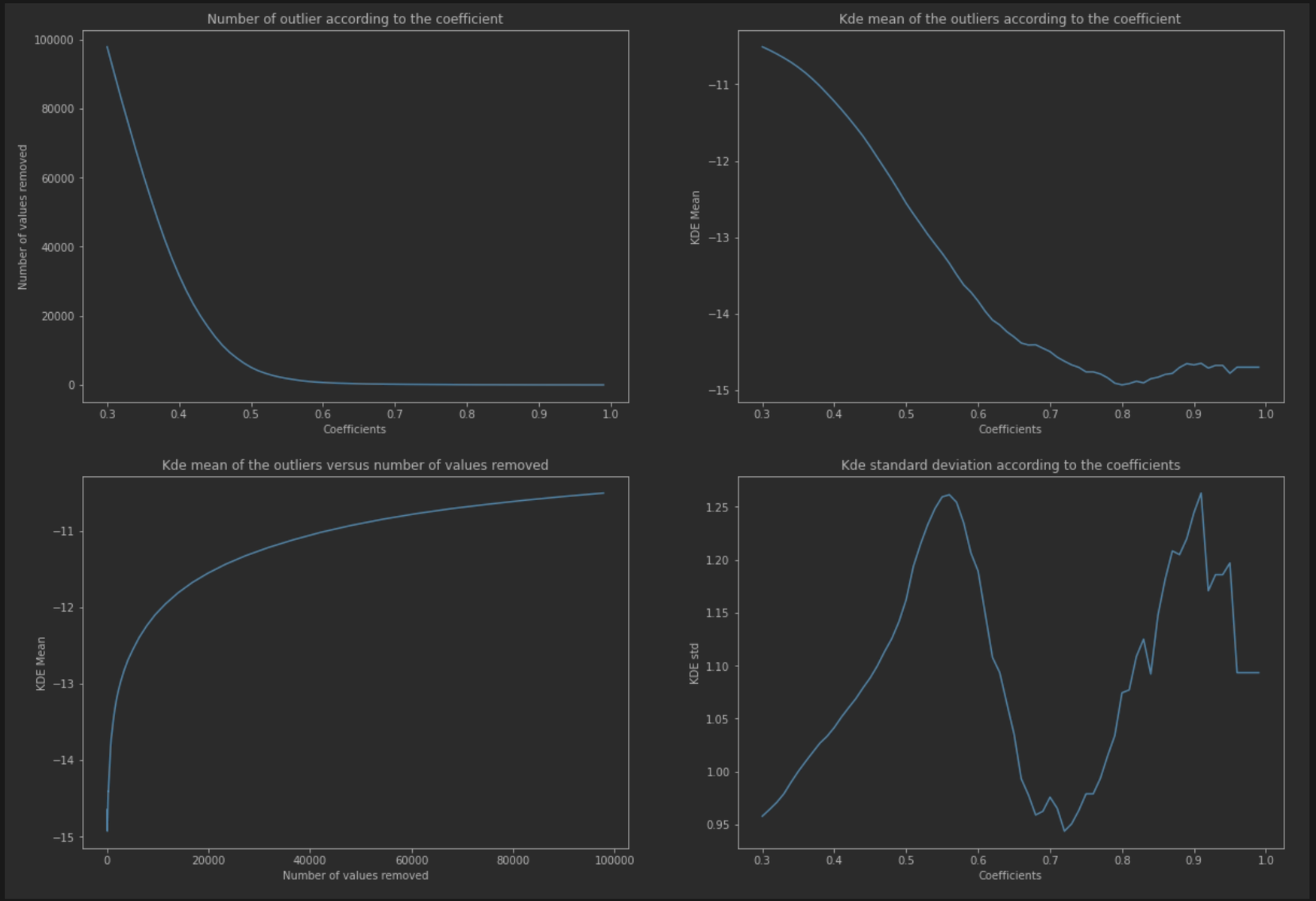
STUDIE OF QC2 FOR PAL

Lets plot classical limit for continetal data





Some stats to estimate threshold



What ?!

```python
# Finding the best coefficient for a density threshold given
df_score, score = cs.calc_coef(df, log_kernel, QC, threshold=-14.15, coef_range=[0.3, 0.6], step=0.02, level="level_1")
df_score2, score2 = cs.calc_coef(df, log_kernel, QC, threshold=-15,  coef_range=[0.3, 0.7], step=0.02, level="level_2")
```

```
.............
--------------
Upper limit
      C2  accuracy_score  precision_score  recall_score  f1_score
14  0.58        0.997641         0.252039      0.604982  0.355835
.................
-------------------
Upper limit
      D2  accuracy_score  precision_score  recall_score  f1_score
14  0.58        0.998225         0.143426      0.685714  0.237232
```

So it seems methods has quite of limitation but it help us to find set of cefficients for RUN :

QC Rule	1st level	2nd level
QC1	$C_1 = 0.96$	$D_1 = 1.09$
QC2	$C_2 = 0.52$	$D_2 = 0.60$
QC3	$C_3 = 0.76$	$D_3 = 0.80$
QC5 min	$C_5 = 315$	$D_5 = 308$
QC5 max	$C_6 = 450$	$D_6 = 457$
QC10 min	$C_{11} = 0.73$	$D_{11} = 0.70$
QC10 max	$C_{12} = 0.20$	$D_{12} = 3.30$

So it seems methods has quite of limitation but it help us to find set of cefficients for RUN :

QC Rule	1st level	2nd level
QC1	$C_1 = 0.96$	$D_1 = 1.09$
QC2	$C_2 = 0.52$	$D_2 = 0.60$
QC3	$C_3 = 0.76$	$D_3 = 0.80$
QC5 min	$C_5 = 315$	$D_5 = 308$
QC5 max	$C_6 = 450$	$D_6 = 457$
QC10 min	$C_{11} = 0.73$	$D_{11} = 0.70$
QC10 max	$C_{12} = 0.20$	$D_{12} = 3.30$

Eventually, the whole purpose of the method was to study our dataset and quality of our data.

So it seems methods has quite of limitation but it help us to find set of cefficients for RUN :

QC Rule	1st level	2nd level
QC1	$C_1 = 0.96$	$D_1 = 1.09$
QC2	$C_2 = 0.52$	$D_2 = 0.60$
QC3	$C_3 = 0.76$	$D_3 = 0.80$
QC5 min	$C_5 = 315$	$D_5 = 308$
QC5 max	$C_6 = 450$	$D_6 = 457$
QC10 min	$C_{11} = 0.73$	$D_{11} = 0.70$
QC10 max	$C_{12} = 0.20$	$D_{12} = 3.30$

Eventually, the whole purpose of the method was to study our dataset and quality of our data.

Said otherwise, the journey was more important than the tools we created

CONCLUSION

CONCLUSION

- Implementation of a method to find the user-configurable coeffs

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers
- Python tools downloadable on pypi

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers
- Python tools downloadable on pypi
- Not all QC implemented in the studies. But all QCRad accecccible

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers
- Python tools downloadable on pypi
- Not all QC implemented in the studies. But all QCRad accecccible
- Methods have some limits :

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers
- Python tools downloadable on pypi
- Not all QC implemented in the studies. But all QCRad accecccible
- Methods have some limits :
  - Arbitrary choice of threshold

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers
- Python tools downloadable on pypi
- Not all QC implemented in the studies. But all QCRad acceccible
- Methods have some limits :
  - Arbitrary choice of threshold
  - Need to clean data as for PAL station

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers
- Python tools downloadable on pypi
- Not all QC implemented in the studies. But all QCRad accecccible
- Methods have some limits :
  - Arbitrary choice of threshold
  - Need to clean data as for PAL station
  - Self imposed contingent constrain using Long & Shi Rules

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers
- Python tools downloadable on pypi
- Not all QC implemented in the studies. But all QCRad accecccible
- Methods have some limits :
  - Arbitrary choice of threshold
  - Need to clean data as for PAL station
  - Self imposed contingent constrain using Long & Shi Rules
- The project remain in development

CONCLUSION

- Implementation of a method to find the user-configurable coeffs
- Methods based on statistical definition of outliers
- Python tools downloadable on pypi
- Not all QC implemented in the studies. But all QCRad accecccible
- Methods have some limits :
  - Arbitrary choice of threshold
  - Need to clean data as for PAL station
  - Self imposed contingent constrain using Long & Shi Rules
- The project remain in development

"Apparently there is no precise method or approach to be followed to determine the coefficients of a given station."

Merci.