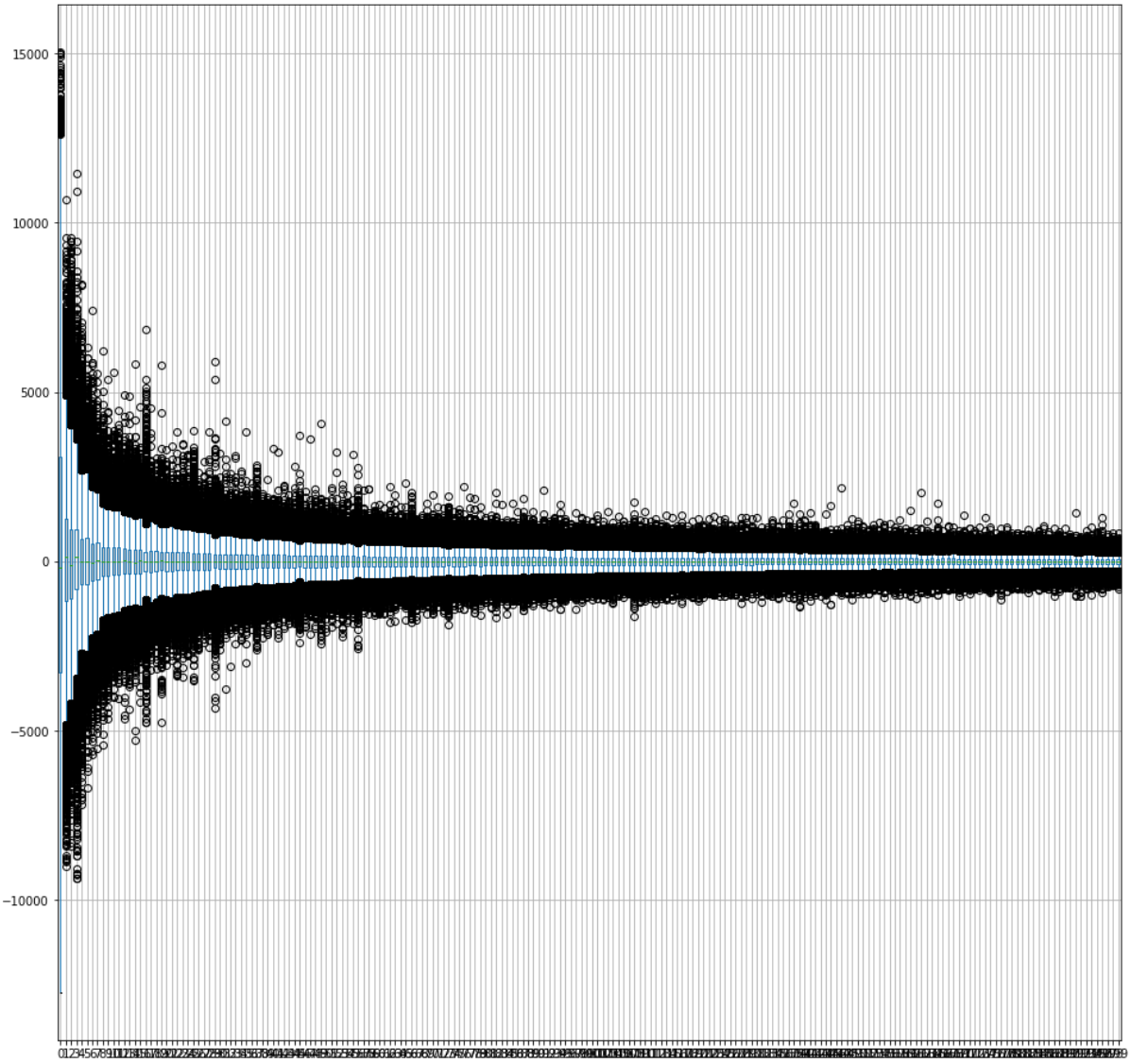
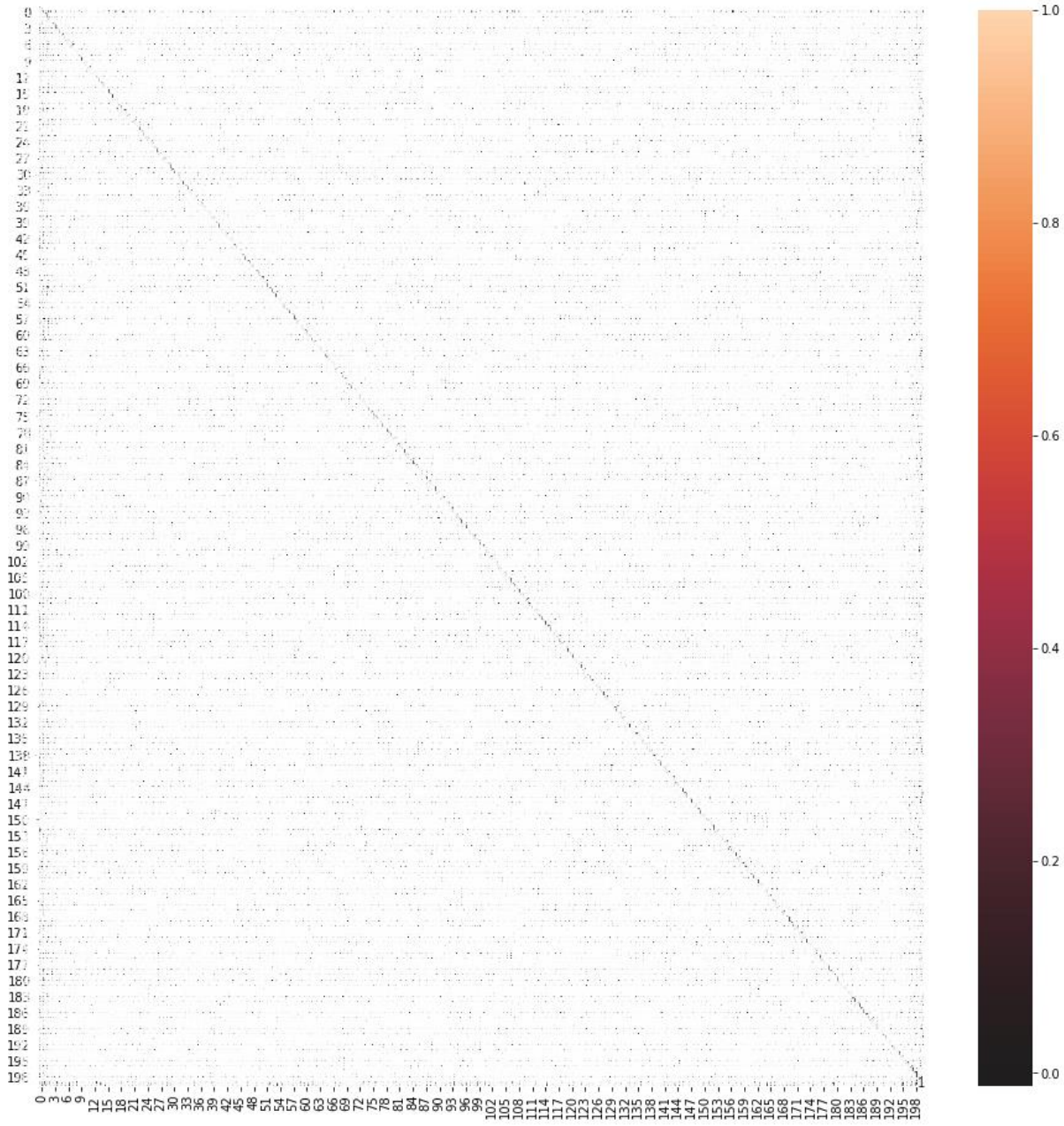
VERSION DU 25/04/19

SORTIE CONSOLE

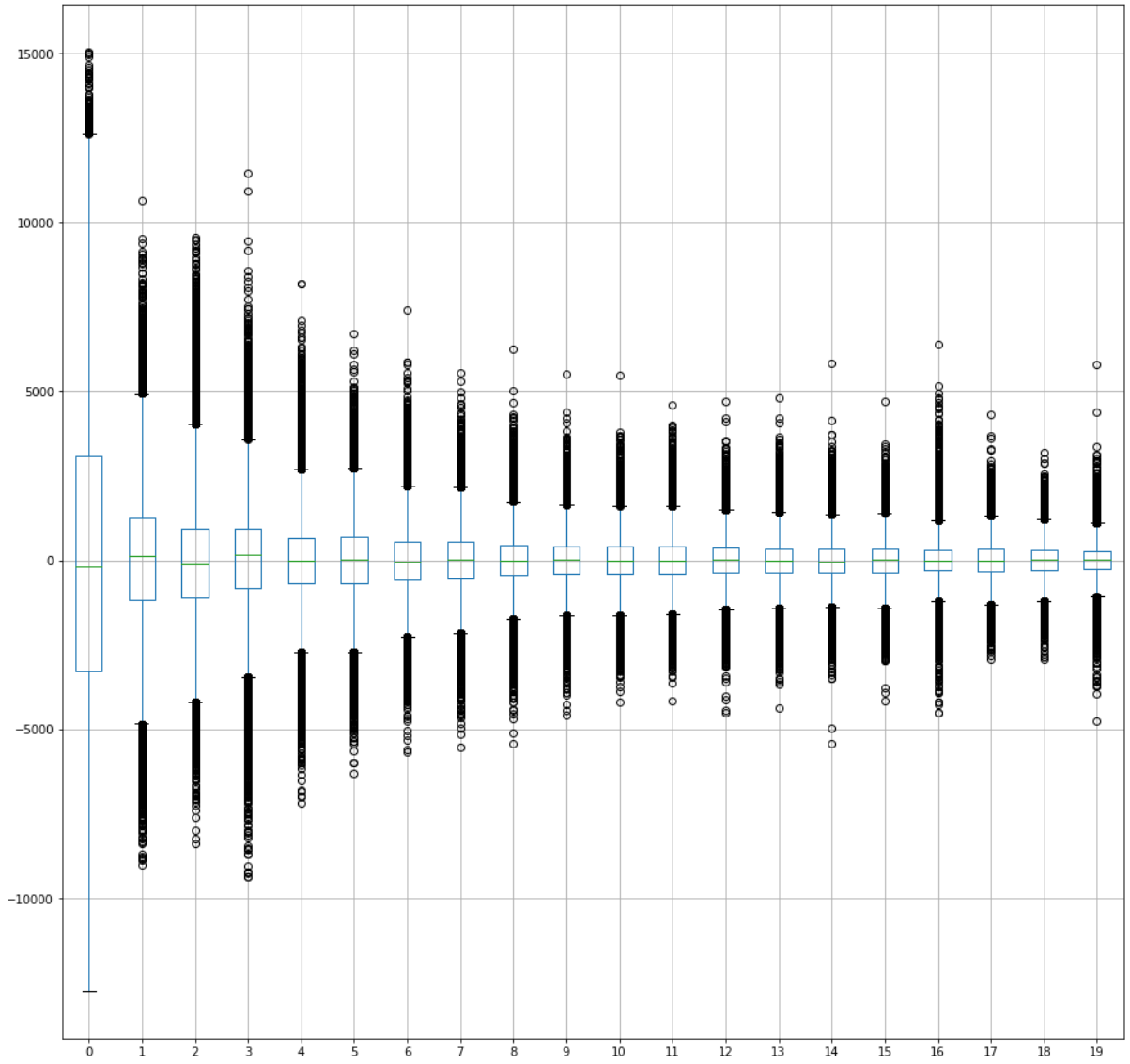
```
In [16]:
runfile('C:/Users/isabe/Pictures/monetFINAL/starting_kit/sampl
e_code_submission/preprocessingFinal.py',
wdir='C:/Users/isabe/Pictures/monetFINAL/starting_kit/sample_c
ode_submission')
Reloaded modules: visualisation, data_io, data_converter,
data_manager
Info file found : C:\Users\isabe\Downloads\monet-
master\starting_kit\c1_input_data\perso_public.info
*** Original data ***
(65856, 200)
[[ -720.579708 -330.571966 -2188.381016 ... -298.022503
268.679125
-107.863398]
[-5322.93398 -2089.062676 -380.988992 ... 23.010128 26.57474
169.194344]
[-4200.092388 -1871.126468 447.135529 ... 67.240814 130.350498
11.225183]
...
[ 656.854711 -104.34349 -1564.462215 ... 111.361633 114.283328
158.923886]
[ 5036.277568 413.900109 1060.588285 ... 87.56614 -157.06906
-174.150162]
[-1010.799311 3252.215798 -4499.276864 ... 6.165946 -70.050208
201.096046]]
```

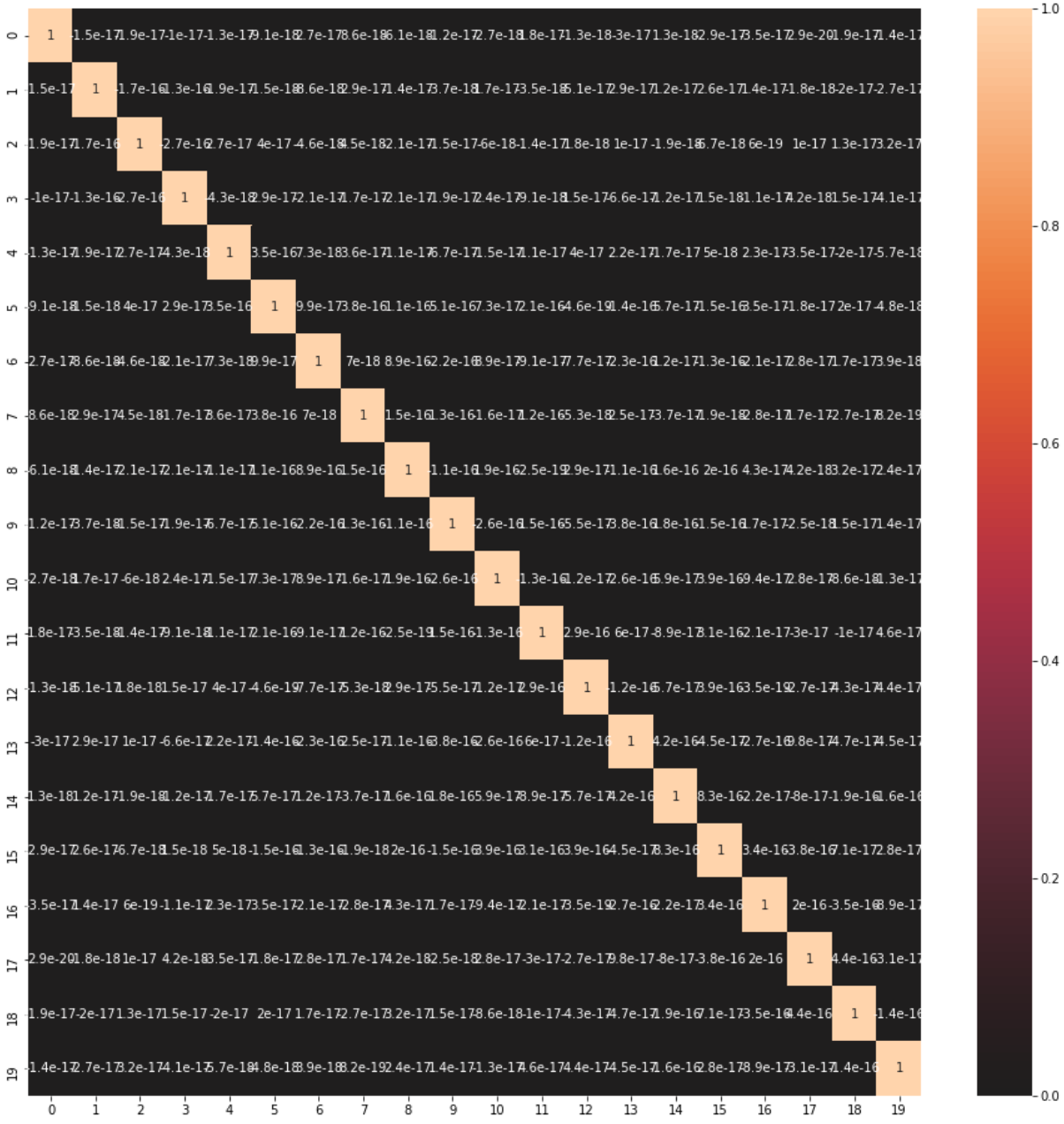

15000
10000
5000
0
−5000
−10000



```
*** Transformed data PCA ***
(65856, 20)
Test 0
[[ -729.63333332 -286.83019539 -2176.88897528 ... -
630.95885372
1178.52007039 -417.99158787]
[-5329.58124154 -2084.85358804 -416.23187307 ... 39.61729046
-307.46828362 -81.41623296]
[-4206.18969162 -1889.83503315 400.26579662 ... -83.93542792
-197.46716045 -222.54971241]
...
[ 649.75840187 -78.94781369 -1579.25349134 ... -619.58802834
-403.6140143 133.33331012]
[ 5028.3074261 388.42343943 1073.11276605 ... 486.04679809
1134.89428891 39.95531261]
```

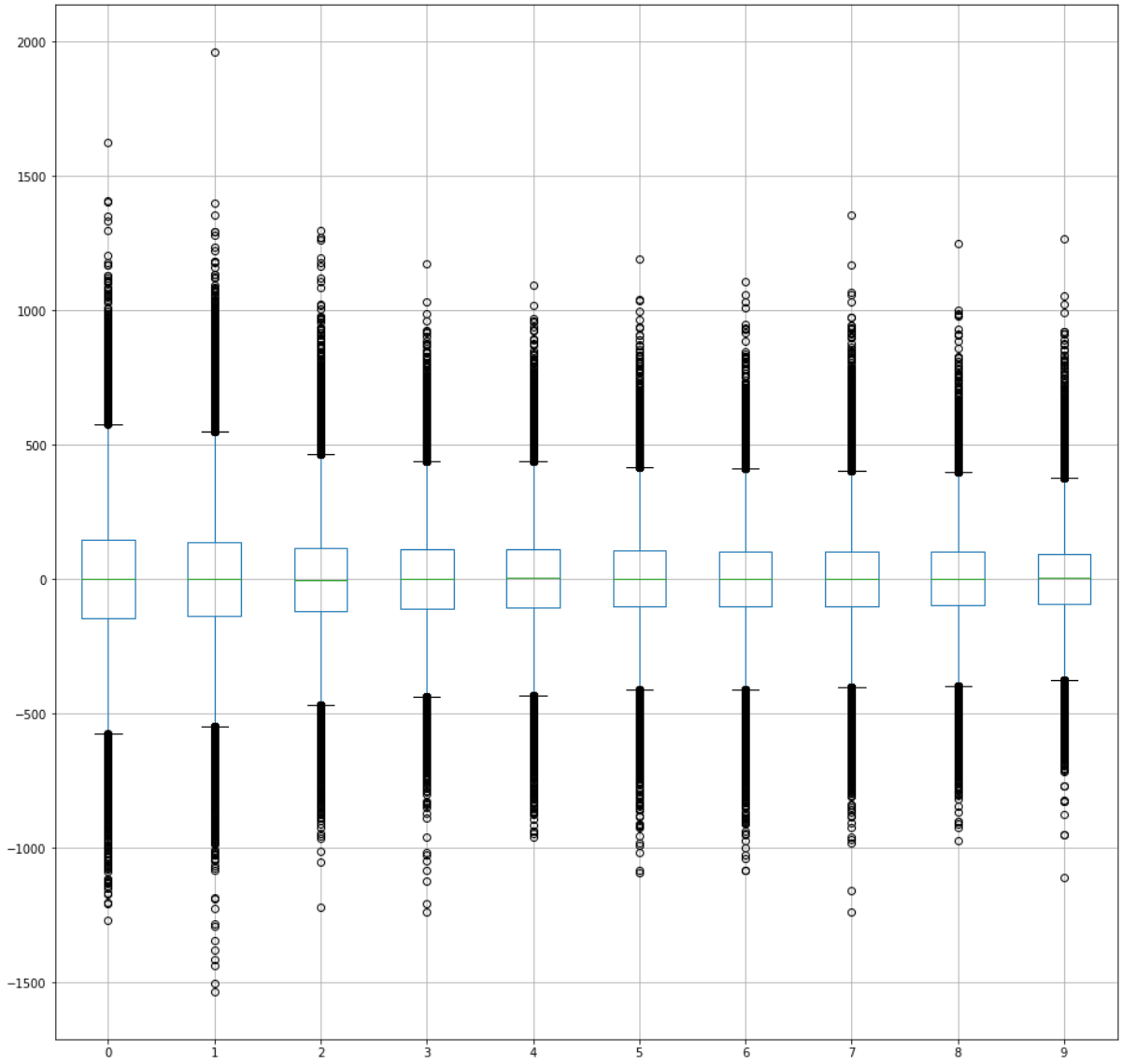
```
[-1015.38232173 3337.44691824 -4430.77078242 ... -129.44153824
151.47825469 153.98002354]]
```


15000
10000
5000
0
−5000
−10000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19



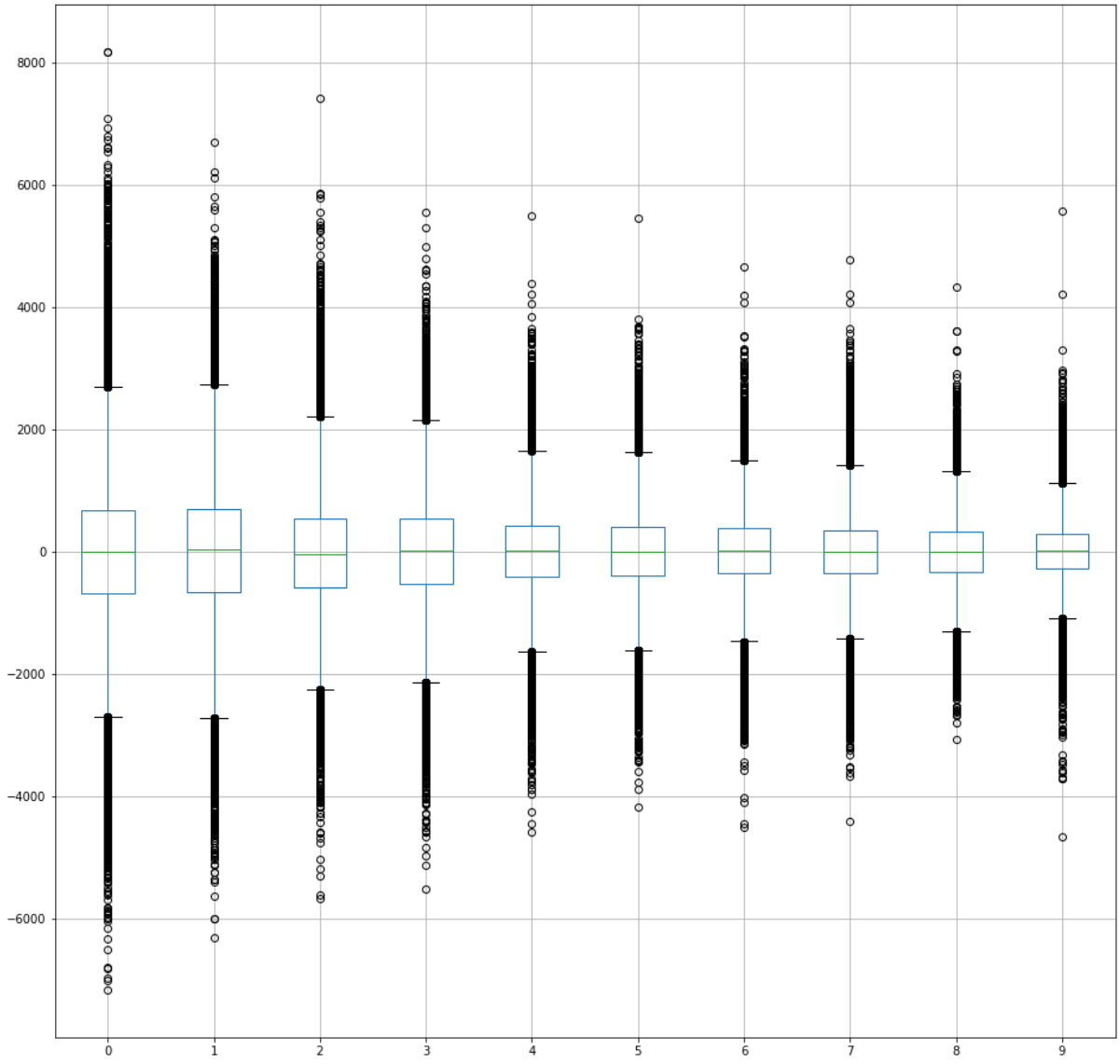
```
*** Transformed SELECTKBEST ***
(65856, 10)
Test 1
[[-422.205033 68.977914 -483.316293 ... -150.823587 -158.27646
-31.943726]
[-191.73689 259.108221 -35.791773 ... -64.802092 -35.126481
-89.511007]
[ 366.147176 -206.061556 -21.142321 ... 53.88305 18.238142
43.622182]
...
[ 214.769236 56.534752 -499.348094 ... 289.514517 340.60026
```

```
1.283359]
[ 285.649595 857.260039 -78.854101 ... -105.808715 -286.107867
22.838065]
[ -7.027664 428.999677 -197.878474 ... 126.456731 67.107772
-15.159047]]
```




```
*** Transformed data PCA + SELECTKBEST ***
(65856, 10)
Test 2
[[ 408.23130234 -2894.87840623 970.52219793 ... 333.2207937
-638.24261047 -103.89087966]
[ -158.33853374 209.2166804 -512.2507911 ... 222.28799044
41.69091798 49.34923789]
[ -470.20853717 -56.73446203 -370.47596797 ... 293.25099416
-89.32577709 -382.782457 ]
...
[ -734.39941316 955.19105045 -819.48424445 ... 483.03990499
-627.53178658 114.84507754]
```

```
[-1746.31879862 -1566.40363456 -538.68385777 ... 462.47712546
472.63302157 91.96324298]
[-1817.81182797 431.10840071 829.33293959 ... 426.35308882
-173.75324055 271.07035418]]
```



CODE :

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar 1 09:54:48 2019

@author: isabe
"""

"""
Il s'agit d'une deux fonctions de préprocessings, suivi de quelques
tests pour s'assurer que nos données ont bien étés modifiés.
Dans le premier cas, un preprocessing avec PCA.
Le second cas, une selection des features les plus pertinants.
```

```
Nous avons ensuite une fonction choixPrepro, qui nous sert a
selectionner le preprocessing voulu, chose que nous
utiliserons dans les tests, qui sont fait dans le main.
"""
from sklearn.pipeline import Pipeline
import warnings
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression
from sklearn.decomposition import PCA
from visualisation import duTP
import pandas as pd
from sys import path;


with warnings.catch_warnings():
    warnings.filterwarnings("ignore",category=DeprecationWarning)
    from sklearn.base import BaseEstimator
    from data_manager import DataManager # The class provided by
binome 1

    # Note: if zDataManager is not ready, use the mother class
DataManager


"""nombre de component pour PCA, et nombre de features pour
selectKbest"""
nbcomponent = 20;
nbkfeatures = 10;


"""Il s'agit ici du preprocessing PCA"""
class Preprocessor(BaseEstimator):

    def __init__(self):
        self.transformer =  PCA(n_components=nbcomponent)

    def fit(self, X, y=None):
        return self.transformer.fit(X, y)


    def fit_transform(self, X, y=None):
         return self.transformer.fit_transform(X)


    def transform(self, X, y=None):
        return self.transformer.transform(X)

    """Notre deuxieme methode de preprocessing, le selectKbest"""
class Preprocessor2(BaseEstimator):

    def __init__(self):
```

```
        self.transformer =  SelectKBest(f_regression, k=nbkfeatures)

    def fit(self, X, y=None):
        return self.transformer.fit(X, y)

    def fit_transform(self, X, y=None):
         return self.transformer.fit_transform(X, y)

    def transform(self, X, y=None):
        return self.transformer.transform(X)


"""Une fonction pour appeler plus facilement les méthodes lors du
test, ou nous combinons les methodes"""
#Permet de tester les différents préprocesseurs
def choixPrepro(option):
    if option == 0:
        print("\n\n*** Transformed data PCA ***")
        Prepro = Preprocessor()
    elif option == 1:
        print("\n\n*** Transformed SELECTKBEST ***")
        Prepro = Preprocessor2()
    elif option == 2:
        print("\n\n*** Transformed data PCA + SELECTKBEST ***")
        Prepro = Pipeline([('PCASelectKBest',
Preprocessor()),('SelectKBest', Preprocessor2())])
    elif option == 3:
        print("\n\n*** Transformed data SELECTKBEST + PCA ***")
        Prepro = Pipeline([('PCASelectKBest',
Preprocessor2()),('SelectKBest', Preprocessor())])
    return Prepro




"""C'est ici que nous testons le tout: nous checkons deja en sortie
sur console, si les données varient,
  et sur leur "shape" entre les données de base et les differentes
methode de preprocessing
  varient aussi comme on le souhaite. On double check en faisant
appel à une methode du fichier visualisation.py
  qui permet de checker de manière plus visuel.
  Enfin, on triple check en les mettant en format csv pour voir
s'ils sont bien formé en sortie."""



if __name__=="__main__":
# We can use this to run this file as a script and test the
Preprocessor
```

```
    input_dir = "C:\\Users\\isabe\\Downloads\\monet-
master\\starting_kit\\c1_input_data"
    output_dir = "./fichiers_preprocesses"
    basename = "perso"

    #Pour le test unitaire, on doit réduire le parametre k)

    D = DataManager(basename, input_dir) # Load data
    print("*** Original data ***")
    print(D.data['X_train'].shape)
    print(D.data['X_train'])
    Ddf=  pd.DataFrame(D.data['X_train'], D.data['Y_train'])
    duTP(Ddf,True)


    for i in range(3):
        Prepro = choixPrepro(i)
        test = Prepro.fit_transform(D.data['X_train'],
D.data['Y_train'])
        # Here show something that proves that the preprocessing
worked fine
        print(test.shape)
        print("Test ", i)
        print(test)
        df = pd.DataFrame(test)
        duTP(df,True)
       # nomfichier = 'test'+str(i)+'_train.data'
       # df.to_csv(nomfichier, index=False, header=False)
```