Manual: Pluto vs. Rataplan python script

1. **Installing the correct libraries**

```
#Script that compares differences between Pluto and Rataplan
from matplotlib import pyplot as plt
import pandas as pd
import os
import sys
import numpy as np
import scipy.stats as sp
import re
from colorama import Fore
```

All above python libraries can (and have) to be installed. An easy way to do this is by typing pip install <insert library name here> in the terminal, e.g. pip install numpy.
Matplotlib is used for making plots, pandas for reading in the data from the csv files, OS is used for walking over paths and sys is used to crash the system with a self-written error message. Numpy is used for fast array calculations, scipy for it's statistical functions, re is short for regular expressions (regex) and colorama is added to print coloured output to the python terminal (makes debugging easier).

2. **Changing the paths in the main() function**

```
if __name__ == "__main__":
    Script_Path=r'C:\Users\Vincent2\PycharmProjects\Carbyonscript'
    Rataplan_Path= r'C:\Users\Vincent2\PycharmProjects\Carbyonscript\vincent_share_0568\output'
    Pluto_Path= r'C:\Users\Vincent2\PycharmProjects\Carbyonscript\vincent_share_0568\pluto'
    Feedback = False  #if true then also print all averages, stdevs, medians and variances for eve
```

Before you are going to use the script, you should change the paths in the main() function. An example can be seen in the piece of code above. Assign the r-string of the folder that contains your python script to Script_Path. Then you can assign Rataplan_Path and Pluto_Path to the path of the folders where the Rataplan and Pluto outputs are stored respectively. But make sure that the directory that contains subdirectories to both the Rataplan and Pluto output are stored in the same directory as your python script (look at the paths above).

3. **Changing the boundaries in the main function**

```
#################################################################'BOUNDARIES'#####################################################################
RSquaredBoundary = 0.995          #R^2                        # Adjust boundaries here
AverageRatioUpper = 1.01          #average1/average2 if avg1>avg2
AverageRatioLower = 0.99          #average1/average2 if avg2>avg1
TTestBoundary = 0.90              #Ttest lower boundary
FTestBoundary = 0.40              #Ftest lower boundary
##################################################################################################################################################
```

The boundaries can be changed depending on the requirements the data should fulfill. It will loop over user-selected files, and check for each column if the column fulfills all the boundaries. When a boundary is not fulfilled it will write its value and the value of the boundary both to the 'txt' file. RSquaredBoundary allows you to check if the columns reach a R^2-value. Changing AverageRatioUpper and AverageRatioLower allows you to check the relative error on both averages. When this is 1.01 and 0.99, then this means that the average of the column of the Rataplan file divided by the average of the column of the Pluto file cannot be higher than 1.01 and cannot be lower than 0.99 (relative error = max 1%) before writing out an error to "Warnings.txt". The TTestBoundary and FTestboundary contain the minimal p-values of the T-test and F-test respectively. Remember that you only have a significant difference (90% lowest significance level) when the p-values are equal or below 0.1.

4. **Changing the important files to check the boundaries for and to make R^2-plots for.**

```
def Crucial_Data_Interpretation(Dictionaries, RSquaredBoundary, AverageRatioUpper, AverageRatioLower, TTestBoundary, FTestBoundary):
    with open("Warnings.txt", "x") as file2:
        for index in range(len(Dictionaries[0])):
            if "integral" in Dictionaries[0][index]['name']:
                file2.write("////////////////////////////////////////////////" + "Filename: " + str(Dictionaries[0][index]['name'] + "////
                for column in range(Dictionaries[0][index]["Dataset"].shape[1]):
                    BoundariesFulfilled = True
```

```
def plot_Average_Ratio(Dictionaries, Script_Path):

    directory = "r_squared_graphs"              # Make directory to store all subdirectories (one for each important csv file)
    path = Script_Path
    dirpath = os.path.join(Script_Path, directory)
    os.makedirs(dirpath, exist_ok=True)

    for index in range(len(Dictionaries[0])):
        if "integral" in Dictionaries[0][index]['name']:
            subdirpath = os.path.join(dirpath, Dictionaries[0][index]['name'])        # make one subdir for each important csv file
            os.makedirs(subdirpath, exist_ok=True)
```

The Boundaries are checked in Crucial_Data_Interpretation() and the R^2-plots are made in plot_Average_Ratio(). The yellow if-loop specifies on which csv files you want to check for boundaries or for which files you want to make R^2-plots for. You can select the files to loop over by simply changing "integral" to whatever file you want to read in. Make sure that you check that you are not using "raw" for instance because then every file with 'raw' in it will go into the loop.

5. Now you can run the program (pycharm, terminal, VSC, visual studio).

```
C:\Users\Vincent2\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Vincent2/PycharmProjects/Carbyonscript/PlutoversusRataplan.py
Rataplan files:  ['s0171_e0568_Ad_integral_overview.csv', 's0171_e0568_Ad_outcome.csv', 's0171_e0568_Ad_pocahontas_curves.csv', 's0171_e0568_Ad_pocahontas_total_curves.csv', 's01
Pluto files  ['s0171_e0568_Ad_integral_overview.csv', 's0171_e0568_Ad_pocahontas_curves.csv', 's0171_e0568_Ad_pocahontas_total_curves.csv', 's0171_e0568_C02-44_Ad_corrected_curve
Amount of Rataplan files:  24
Amount of Pluto files:  21
From which file do you want to remove? Answer Files_Pluto or Files_Rataplan: Files_Rataplan
Which letters do you want to remove?
Amount of corresponding files:  21
Corresponding files:  ['s0171_e0568_Ad_integral_overview.csv', 's0171_e0568_Ad_pocahontas_curves.csv', 's0171_e0568_Ad_pocahontas_total_curves.csv', 's0171_e0568_C02-44_Ad_correc
Reading in Pluto files...
```

On the image above you can see the Rataplan files and Pluto files that the script recognizes and you can also see the amount of Rataplan and Pluto files. Now you will asked from which file to remove.

If Rataplan files have the same name as the Pluto files, then you can give either Files_Rataplan or Files_Pluto as answer and on which letters you want to remove, you can answer nothing. Otherwise specify if you want to remove from Rataplan and Pluto and select which letters you want to remove, so that the filenames are equal (sometimes there is a date before the filename e.g. 20230323). Once these 2 questions have been answered the output will be generated as can be seen below.

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Program executed :)
You can find the results of the calculations as Statistics.txt and the graphs in the r_squared_graphs folder of this script.

Process finished with exit code 0
```

5. The output

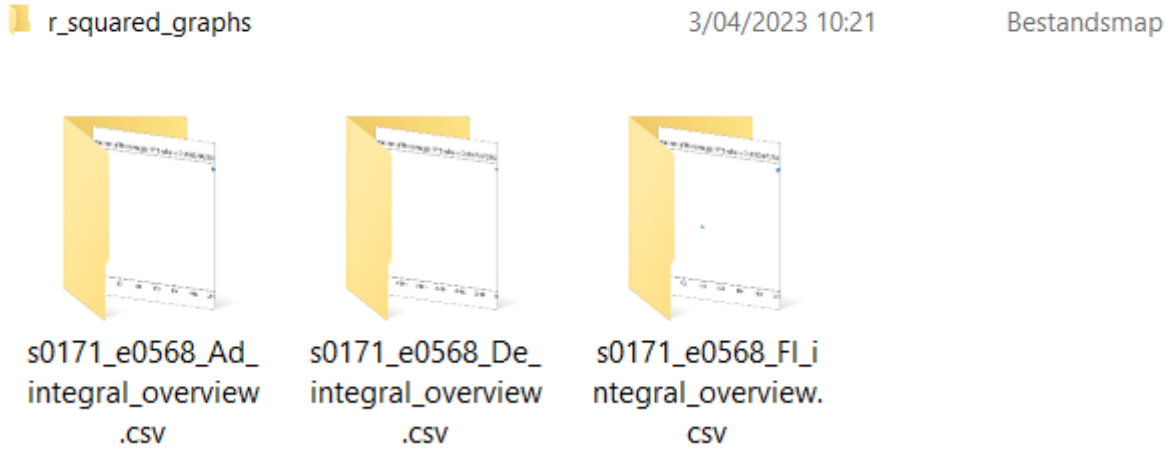
The program outputs 2 'txt' files which can all be found in the same folder as your python script. The first one is called 'Statistics.txt' and contains the average ratios of each column, the averages, stdev, variance and medians if the file has less than 30 columns and the p-values of the F-tests and the T-Tests. The headers of the columns that were read in can be found next to the average ratio columns. This is very <u>important</u> since Rataplan and Pluto output does not contain the same data. Therefore I tried to compare the same data of each file as you can see below. Sometimes, there will be no data to be compared and then I chose to compare the columns that were very similar but were for instance measured under different flow. In such cases you cannot properly make any statistical conclusions of these files, but it might help to check if the values look okay on first sight.

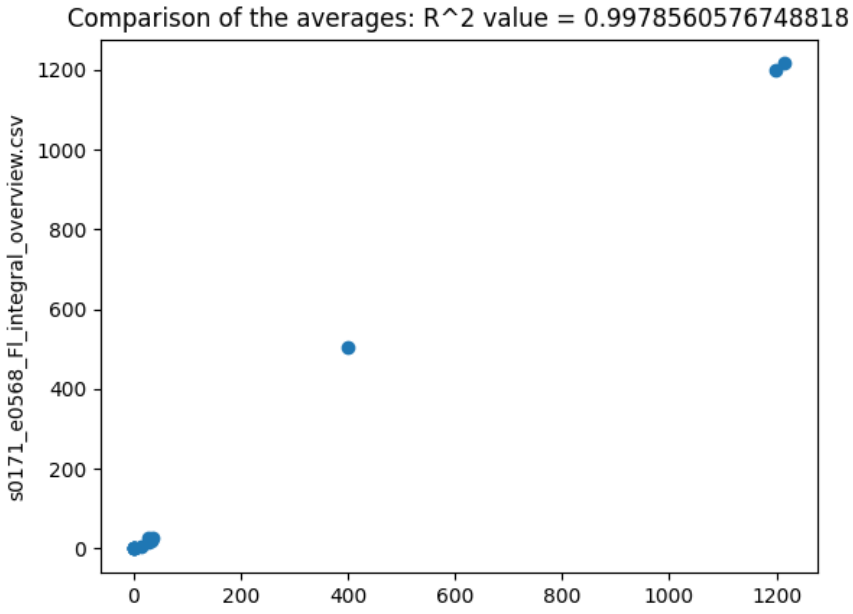
```
////////////////////////////////////////////////////////////////////////FILENAME: s0171_e0568_Ad_integral_overview.csv////////////////////

Path 1: C:\Users\Vincent2\PycharmProjects\Carbyonscript\vincent_share_0568\plutos0171_e0568_Ad_integral_overview.csv
Path 2: C:\Users\Vincent2\PycharmProjects\Carbyonscript\vincent_share_0568\outputs0171_e0568_Ad_integral_overview.csv

Rows: 2  Columns: 13

Averages ratios:
Averages ratio column 0: 1.0                      Q (l/min)      vs.        Q (l/min)
Averages ratio column 1: 1.0                      Adsorptiontime (s)      vs.        Adsorptiontime (s)
Averages ratio column 2: 0.9827727645611156                      Temp Ad      vs.        Temp Ad
Averages ratio column 3: 1.010915071216643                     SchT H2O (s)      vs.        SchT Water calibrated (s)
Averages ratio column 4: 0.17130519568373143                     H2O_SchTs-de      vs.        Water calibrated_SchTs-de
Averages ratio column 5: 0.6627631508564643                      H2O_100s-de      vs.        Water calibrated_100s-de
Averages ratio column 6: 0.5050490984526848                      H2O_250s-de      vs.        Water calibrated_250s-de
Averages ratio column 7: 0.17130519568373143                     H2O_tots-de      vs.        Water calibrated_tots-de
Averages ratio column 8: 1.0099226423952261                      SchT CO2-44 (s)      vs.        SchT Carbon dioxide calibrated (s)
Averages ratio column 9: 1.0029777552560604                      CO2-44_SchTs-de      vs.        Carbon dioxide calibrated_SchTs-de
Averages ratio column 10: 0.9851092834788702                     CO2-44_100s-de      vs.        Carbon dioxide calibrated_100s-de
Averages ratio column 11: 0.9947905634656651                     CO2-44_250s-de      vs.        Carbon dioxide calibrated_250s-de
Averages ratio column 12: 1.0026304182335488                     CO2-44_tots-de      vs.        Carbon dioxide calibrated_tots-de

Calculation results file 1.
Average:
[ 9.00000000e-01  1.21600000e+03  2.99500000e+01  1.22846450e+03
 -2.85550129e+01 -1.64826189e+01 -2.58093147e+01 -2.85550129e+01
  1.22725850e+03 -2.28710461e+00 -8.51106981e-01 -1.38901894e+00
 -2.28631257e+00]
Stdev:
[ 0.6        16.          1.05       15.9795      7.0602311   9.27966688
  8.07589056  7.0602311  14.7735      0.37795396  0.50486062  0.56808052
  0.378746  ]
Variance:
[3.60000000e-01 2.56000000e+02 1.10250000e+00 2.55344420e+02
 4.98468632e+01 8.61122174e+01 6.52200084e+01 4.98468632e+01
 2.18256302e+02 1.42849196e-01 2.54884241e-01 3.22715480e-01
 1.43448531e-01]
Median:
[ 9.00000000e-01  1.21600000e+03  2.99500000e+01  1.22846450e+03
 -2.85550129e+01 -1.64826189e+01 -2.58093147e+01 -2.85550129e+01
  1.22725850e+03 -2.28710461e+00 -8.51106981e-01 -1.38901894e+00
 -2.28631257e+00]
```

On the image below you can see that 'Warnings.txt' checks the boundaries of the Ad_integral_overview.csv file for every column. When the column fulfills the boundaries that were set then it writes 'All boundaries fulfilled for column: X' otherwise it will write which value is too low, what the value is and what the value of the boundary is.

```
//////////////////////////////////////////////////Filename: s0171_e0568_Ad_integral_overview.csv//////////////////////////////////////////////////////////////////////////////////////

All boundaries fulfilled for column: 0

All boundaries fulfilled for column: 1

Column: 2 => average_ratio differs too hard!
Average ratio value is 0.9827727645611156 and lower boundary is set at 0.99 and upper boundary is set at 1.01

Column: 2 => T test value too low!
T test value is 0.7816887120679747 and boundary is set at 0.9

Column: 3 => average_ratio differs too hard!
Average ratio value is 1.010915071216643 and lower boundary is set at 0.99 and upper boundary is set at 1.01

Column: 3 => T test value too low!
T test value is 0.6166716333923293 and boundary is set at 0.9

Column: 4 => average_ratio differs too hard!
Average ratio value is 0.17130519568373143 and lower boundary is set at 0.99 and upper boundary is set at 1.01

Column: 4 => T test value too low!
T test value is 0.4200772486580087 and boundary is set at 0.9

Column: 4 => F test value too low!
F test value is 0.04178196868234685 and boundary is set at 0.4

Column: 5 => average_ratio differs too hard!
Average ratio value is 0.6627631508564643 and lower boundary is set at 0.99 and upper boundary is set at 1.01

Column: 5 => T test value too low!
T test value is 0.7244202336546217 and boundary is set at 0.9

Column: 5 => F test value too low!
F test value is 0.310073561328417 and boundary is set at 0.4

Column: 6 => average_ratio differs too hard!
Average ratio value is 0.5050490984526848 and lower boundary is set at 0.99 and upper boundary is set at 1.01

Column: 6 => T test value too low!
T test value is 0.5787698608493203 and boundary is set at 0.9
```

As you can see below, a directory called 'r_squared_graphs' has been created and contains 3 subdirectories with the corresponding R^2-plots. The R^2 value can be seen on the title of the graph.




Comparison of the averages: R^2 value = 0.9978560576748818
s0171_e0568_FI_integral_overview.csv
0
200
400
600
800
1000
1200