# Data-scientists analysis code manual

This file contains the explanation and instructions of a code that can be used to check if 2 different scientists get significantly different results when a data-analysis is done of the exact same samples.

The code was tested for three different IDEs: visual studio code, visual studio and pycharm and it works fine in these environments, although coloured output cannot be printed to the python console in visual studio. The code is developed using python version 3.9. Please make sure that the necessary packages are installed. The packages are numerical python ‘numpy’, python analytical data system ‘pandas’, colorama and scientific python ‘scipy’. These packages can be installed by copying the following code into your terminal.

```terminal  
pip install <package name>  
```  
**Instructions**

1. Put the output of scientist 1 (all the files) in a folder with a name of your choosing
2. Put the output of scientist 2 (all the files) in a folder with a name of your choosing
3. Open folder of scientist 1, copy the path and paste it in the main function at the bottom of the script. Path1=r<insert your path here between “” as string>
4. Open folder of scientist 2, copy the path and paste it in the main function at the bottom of the script. Path2=r<insert your path here between “” as string>

The python file will loop over all the files in in both folders and read in the data correctly if the files are readable (csv and txt files). Blank cells are read in as Not a Number (NaN). A small amount of cells can not be read in properly and are also stored as NaN. The functions that take in NaN data will not throw an error but return NaN after the calculation.

1. Now, you can run the script. You will get information with colour codes that are not yet used at the moment. You will be asked if you want to print averages, stdevs, variances and medians for files with a large amount of columns (>30). Answer “y” or “n” in the python console and press enter.
2. Output is now printed to the python console, this is used for debugging purposes or to check if the data got read in correctly (black arrays at the beginning of the console output)
3. Once you see that the “Calculations are done =)”, you can check the filename in the python console and navigate to the place where your python script is located. Once you navigated over there, you will find a .txt file with all the results.

Note: If the files you are reading in are not correctly delimited, change the pd.read\_csv() function and make sure that the data is correctly read in by checking the black arrays at the beginning of the python console.

**Code explanation**

All of the code is executed and managed via the main() function at the end of the script in the following order.

1. colour\_coding\_info() print colour schemes to the console (currently unused colour schemes)
2. show\_results() ask the user if it wants all the averages etc. printed for larger datasets. This results in a bool stored as feedback.
3. The readable filenames are stored in Data1 and Data2 by the function Read\_in\_Data()
4. A dictionary is created in the function Generate\_Dictionaries() for each scientist and the datasets are stored in each dictionary respectively.
5. Compare\_Data() compares the amount of readable files in both folders. Since the starting data is the same, the amount of readable files should be the same.
6. Compare\_Datapoints() compares if the amount of datapoints is the same as should be.
7. Both Compare\_Data() and Compare\_Datapoints() return a bool that print a confirmation or an error to the screen respectively. In case the amount of readable files or datapoints is not the same, the program will sys.exit().
8. Calculate\_Results() calculates the average, stdev, variance and median for each column of each dataset.
9. Calculate\_Statistical\_Results() calculates the p-values of both a t-test and an f-test for each column of the corresponding files. This means that column 0, 1 and 2 etc. for both corresponding files (same filenames) will be t-tested and f-tested.
10. Interpret\_Statistical\_Results() is not called and thus does nothing. I will maybe implement in a next version to output coloured numbers based on the p-values.
11. Compare\_Results() divides the average of column 0 for a file for scientist 1 with the average of column 0 for the corresponding file for scientist 2. Value should thus be as close to 1 as possible.
12. A new .txt file is created in main()
13. Write\_Statistics() prints the p-values of both t-test and f-test for each column of the file.
14. Write\_Results() writes the Averages ratio and the averages, stdevs, variances and medians for each file.
15. Calculations are done =)

**Interpretation of the results**

T-test

The T-test performed is called Welch test. It does not assume that the variances are equal and it is a two-tailed test with mean1==mean2 as null hypothesis. If the p-value is 0,05 we can conclude that the null hypothesis should be rejected and that the alternative hypothesis mean1!=mean2 should be accepted with significance level of 95%.

F-test

The null hypothesis of the F-test assumes that var1==var2 and should be rejected once the p-value <0.05. At this p-value, we will accept the alternative hypothesis with var1!=var2 with a significance level of 95%.

Averages Ratio

Compares the average of each column for both files. Ratio should be close to one, preferred values differ less than 0,01.

Averages, stdevs, variances, median

The averages, standard deviations, medians and variances are printed for both files for each column. This will not be printed if “Feedback==False” and the file has over 30 columns.

It might be wise to look more at the statistical tests for files that contain many rows (look at the results file). It is common practice in big data to delete numbers that cannot be read in or to store them as NaN. The larger the datasets (rows), the smaller the impact on the p-values and the more accurate the results.

I recommend manually looking at the averages, stdevs etc. for smaller datasets, since these will be more accurate than the statistical tests and this is very doable for files with few columns.