DASU20 Reflection

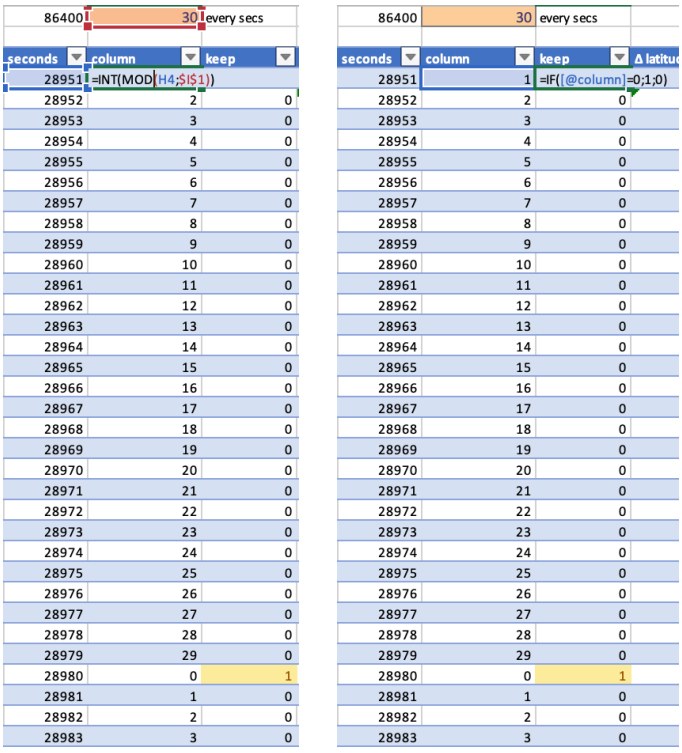
George Kontos - 1531409

In the first assignment the rest of the members of my group and I were assigned the same task which was to analyse the profile of a runner. This taught me how to extract data from visual information and use these to form hypotheses.

In the second assignment, I undertook the task of cleaning, analysing and visualising the data. Through this I improved my skills in excel. In the beginning it looked like it would be a simple operation where I would have to calculate the additional physical quantities and then make the graphs. When this was done the graphs showed a lot of variation between the data points and they needed to be simplified in order to be more easily understandable. To do that I learned how using formulas I could identify every Nth measurement and keep only those. That way of filtering the data was useful in our case as it allowed to experiment with how detailed we wanted the graphs to be, to compare and then to decide how many seconds would be ideal between the measurements we would keep.

In assignment three I again handled the data and created the visualisations. This time, in order for the process to be didactic I tried a different method of filtering out the data. This time I used python. My experience using python is not extensive so I had to learn how to calculate the extra quantities. Moreover, in the final assignment we became aware of various issues such as gaps in the measurements. These issues were the perfect opportunity to try out and learn different ways to fix them in python. Python functions I learned, for example, are interpolation for estimating unknown data points, resampling for keeping every Nth measurement and gaussian filtering. In addition to that, combining python and excel for using their different features was also an instructive experience to me. As much I would search I could not find an easy way to make linear regression lines in python. However, in excel is automatic. So, I learned how to export the modified datasets from python as excel and create the graphs with linear regressions in excel. This is an extremely easy and efficient way to use the best from both python and excel and it is surely going to be useful in the future. This work was of great importance in the project as most of the rest of the work was based on the visualisers produced by it. Even though I did not work on the part of drawing conclusions from the graphs I actively followed the process of making the motivational profiles and the final visualisation making myself aquatinted with the theoretical aspect of the project as well.

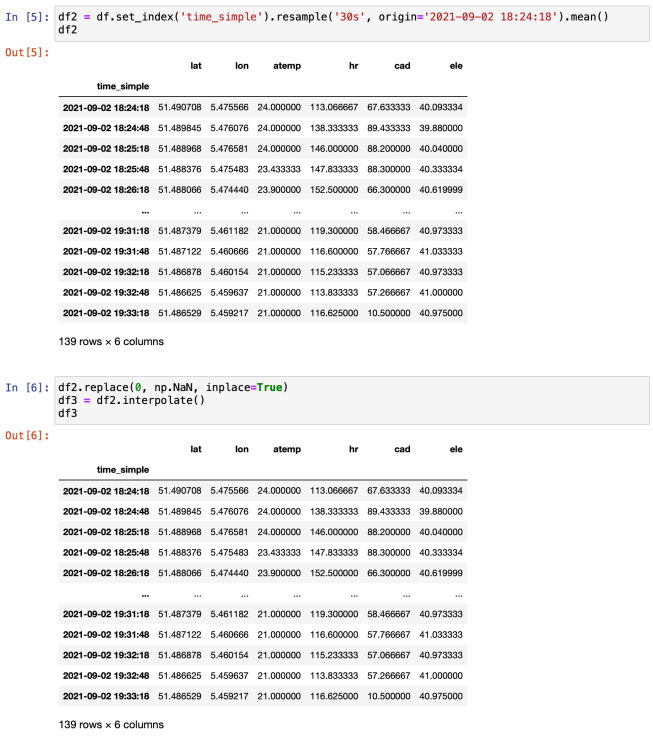
All in all, I believe this course was a great source of knowledge, useful for future projects. Most importantly, the realisation of the importance of the analysis and the research of the target user group, through motivational profiles, in designing an app was the greatest lesson. This showed me that making such an app for training and motivational purposes is not as easy as it seems and requires much more thought and research. This was something that I did not do in my past projects but its effectiveness in showing insights will be useful in the future.



86400	30	every secs	
seconds	column	keep	
28951	=INT(MOD(H4;I1))		
28952	2	0	
28953	3	0	
28954	4	0	
28955	5	0	
28956	6	0	
28957	7	0	
28958	8	0	
28959	9	0	
28960	10	0	
28961	11	0	
28962	12	0	
28963	13	0	
28964	14	0	
28965	15	0	
28966	16	0	
28967	17	0	
28968	18	0	
28969	19	0	
28970	20	0	
28971	21	0	
28972	22	0	
28973	23	0	
28974	24	0	
28975	25	0	
28976	26	0	
28977	27	0	
28978	28	0	
28979	29	0	
28980	0	1	
28981	1	0	
28982	2	0	
28983	3	0	

86400	30	every secs	
seconds	column	keep	Δ latitu
28951	1	=IF([@column]=0;1;0)	
28952	2	0	
28953	3	0	
28954	4	0	
28955	5	0	
28956	6	0	
28957	7	0	
28958	8	0	
28959	9	0	
28960	10	0	
28961	11	0	
28962	12	0	
28963	13	0	
28964	14	0	
28965	15	0	
28966	16	0	
28967	17	0	
28968	18	0	
28969	19	0	
28970	20	0	
28971	21	0	
28972	22	0	
28973	23	0	
28974	24	0	
28975	25	0	
28976	26	0	
28977	27	0	
28978	28	0	
28979	29	0	
28980	0	1	
28981	1	0	
28982	2	0	
28983	3	0	

Filtering with excel



```
In [5]: df2 = df.set_index('time_simple').resample('30s', origin='2021-09-02 18:24:18').mean()
        df2
```

Out[5]:

time_simple	lat	lon	atemp	hr	cad	ele
2021-09-02 18:24:18	51.490708	5.475566	24.000000	113.066667	67.633333	40.093334
2021-09-02 18:24:48	51.489845	5.476076	24.000000	138.333333	89.433333	39.880000
2021-09-02 18:25:18	51.488968	5.476581	24.000000	146.000000	88.200000	40.040000
2021-09-02 18:25:48	51.488376	5.475483	23.433333	147.833333	88.300000	40.333334
2021-09-02 18:26:18	51.488066	5.474440	23.900000	152.500000	66.300000	40.619999
...	...	...	...	...	...	...
2021-09-02 19:31:18	51.487379	5.461182	21.000000	119.300000	58.466667	40.973333
2021-09-02 19:31:48	51.487122	5.460666	21.000000	116.600000	57.766667	41.033333
2021-09-02 19:32:18	51.486878	5.460154	21.000000	115.233333	57.066667	40.973333
2021-09-02 19:32:48	51.486625	5.459637	21.000000	113.833333	57.266667	41.000000
2021-09-02 19:33:18	51.486529	5.459217	21.000000	116.625000	10.500000	40.975000

139 rows × 6 columns

```
In [6]: df2.replace(0, np.NaN, inplace=True)
        df3 = df2.interpolate()
        df3
```

Out[6]:

time_simple	lat	lon	atemp	hr	cad	ele
2021-09-02 18:24:18	51.490708	5.475566	24.000000	113.066667	67.633333	40.093334
2021-09-02 18:24:48	51.489845	5.476076	24.000000	138.333333	89.433333	39.880000
2021-09-02 18:25:18	51.488968	5.476581	24.000000	146.000000	88.200000	40.040000
2021-09-02 18:25:48	51.488376	5.475483	23.433333	147.833333	88.300000	40.333334
2021-09-02 18:26:18	51.488066	5.474440	23.900000	152.500000	66.300000	40.619999
...	...	...	...	...	...	...
2021-09-02 19:31:18	51.487379	5.461182	21.000000	119.300000	58.466667	40.973333
2021-09-02 19:31:48	51.487122	5.460666	21.000000	116.600000	57.766667	41.033333
2021-09-02 19:32:18	51.486878	5.460154	21.000000	115.233333	57.066667	40.973333
2021-09-02 19:32:48	51.486625	5.459637	21.000000	113.833333	57.266667	41.000000
2021-09-02 19:33:18	51.486529	5.459217	21.000000	116.625000	10.500000	40.975000

139 rows × 6 columns

Resampling with python (top) - Interpolation with python (bottom)