

Inputs:

Input 1 contains 4 columns: `campaign_name`, `spend`, `impressions`, and `clicks`. Campaign name consists of a variable number of arguments, separated with the delimiter “_”, but all campaign names contain an age, cluster and device. Furthermore, the age is always the first element, the cluster is always the second, and the device always starts at the third. Some, but not all, campaign names contain a date stamp following the device, but this is not a part of the device name. For example, if the campaign name is `31-40_notarget_htc_one_1GB_9114` then the age is “31-40”, the cluster is “notarget”, the device is “htc_one_1GB”, and the campaign data is from 9-1-2014. Furthermore, if there are two devices that have similar names (for example: `htc_one_1GB` and `htc_one`), they should be treated as unique devices.

Input 2 contains 3 columns: `campaign_name`, `actions`, and `object_type`. In this case, campaign name is always an age, cluster and device, separated by the delimiter “_”, but the order of the elements is variable. Actions contains a list of dictionaries, where each dictionary contains an `action_type` and a value. For example if a given campaign has the value `[{action_type:likes,value:12}]`, this means the campaign has 12 likes.

Note, input 1 can have multiple campaigns with same age, cluster and device (but different dates) while input2 cannot.

Output:

Using python, read in the two input files, map them on key, where key is (age,cluster,device) and create the following four tables, all with the same columns, and a totals row:

1. A table grouped by key
2. A table grouped by age
3. A table grouped by cluster
4. A table grouped by device

Columns:

1. Spend
2. Impressions
3. Storied Engagements = Like + Comment + Share
4. CPVV = Spend / Video Views if `object_type` is VIDEO, otherwise = 0
5. Count = Number of distinct campaign names from input1 per row

Please export your results to XLSX, with one table per tab.

Hints:

1. If you are using pandas, this can be achieved easily with `df.to_excel()`.
2. Each table in your final output should have parity with one another and also the two inputs. For example, total spend in each table should be equal as well as equal to the total spend in input1.csv

In addition to providing your final results please provide well-documented code, as well as a list of any assumptions made with regards to the data or process.