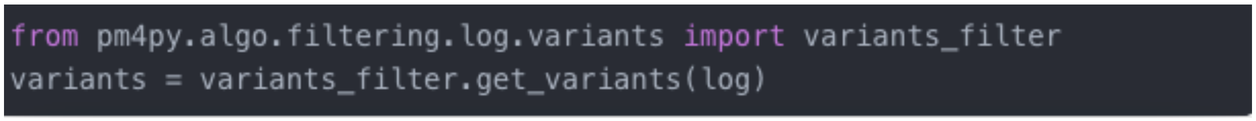
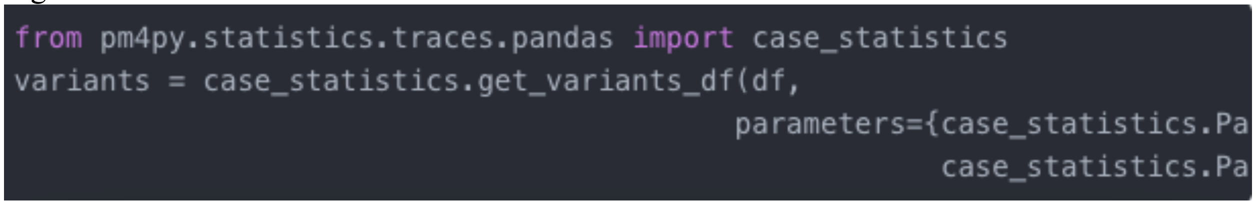
Filtering based on variants.

A variant is a set of cases that share the same control-flow perspective, so a set of cases that share the same classified events (activities) in the same order. In this section, we will focus for all methods first on log objects, then we will continue with the dataframe.



```
from pm4py.algo.filtering.log.variants import variants_filter
variants = variants_filter.get_variants(log)
```

Figure - 1



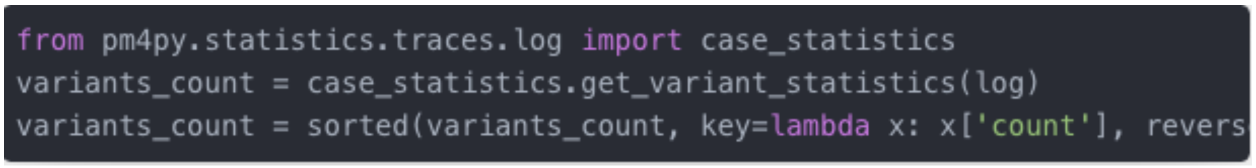
```
from pm4py.statistics.traces.pandas import case_statistics
variants = case_statistics.get_variants_df(df,
                                           parameters={case_statistics.Pa
                                                       case_statistics.Pa
```

Figure -2

To get the list of variants contained in a given log, the following code could be used. The first code is for an log object, the second for a dataframe. The result is expressed as a dictionary having as key the variant and as value the list of cases that share the variant.

If the number of occurrences of the variants is of interest, the following code retrieves a list of variants along with their count (so, a dictionary which key is the variant and the value is the number of occurrences).

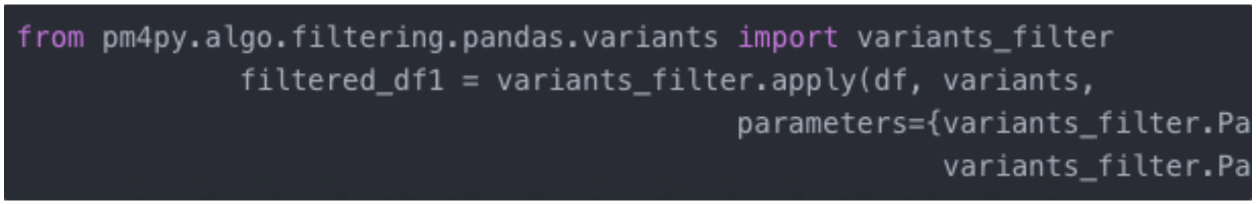
To filter based on variants, assume that variants is a list, whereby each element is a variant (expressed in an equal way as in the variants retrieval method). The first method can be applied on log objects, the second can be applied on dataframe objects. Note that the variants given in variants are kept.



```
from pm4py.statistics.traces.log import case_statistics
variants_count = case_statistics.get_variant_statistics(log)
variants_count = sorted(variants_count, key=lambda x: x['count'], revers
```

Figure - 3

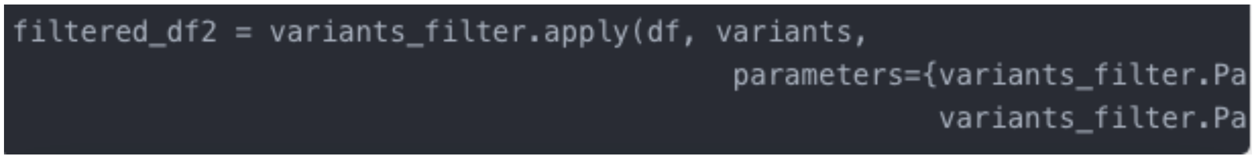
Contrary to the previous example, suppose you want to filter the given variants out. Again, let variants be a list, whereby each element is a variant.



```
from pm4py.algo.filtering.pandas.variants import variants_filter
            filtered_df1 = variants_filter.apply(df, variants,
                                                 parameters={variants_filter.Pa
                                                             variants_filter.Pa
```

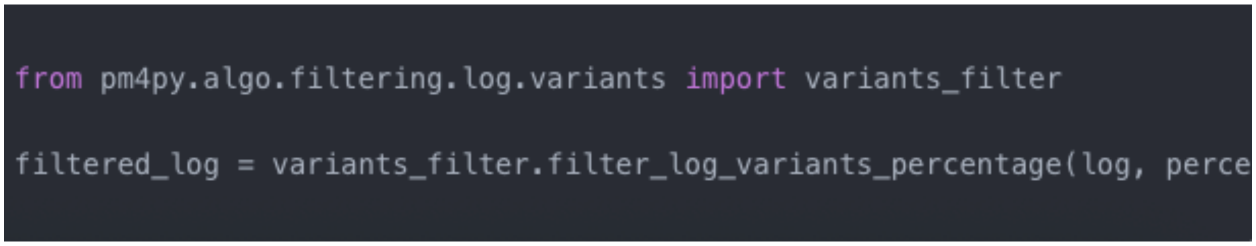
Figure - 4

A filter to keep automatically the most common variants could be applied through the
 apply_auto_filter method. This method accepts a parameter
called DECREASING_FACTOR (default value is 0.6; further details are provided in the start activities filter).



```
filtered_df2 = variants_filter.apply(df, variants,
                                     parameters={variants_filter.Pa
                                                 variants_filter.Pa
```

Figure – 5



```
from pm4py.algo.filtering.log.variants import variants_filter

filtered_log = variants_filter.filter_log_variants_percentage(log, perce
```

Figure - 6

On the event log objects, a filter on the variants percentage can be applied as shown in the following example. The percentage of variants to keep must be specified in
the **percentage** parameter as a number between **0** (only the most frequent variant is kept) and **1** (all the variants are kept).

To be more explicit, the variants filter on percentage works as follow, given the percentage **P**:

- The variants of the log are found along with their number of occurrences
- A number N is chosen such that if we take all the variants with at least N occurrences, we include a percentage of cases that is at least P, while if we choose N+1 we would include a percentage of cases that is below P.

If the log contains the following variants:

- ABC (2 occurrences)
- A,B,C,AB,AC,BC,CB,BA,CA,ABCD,ABCE,ABCF,ABCG,ABCH,ABCI,ABCL,ABCM ,ABCN (1 occurrence each)

Then:

- with percentage=1, all the 20 cases would have been kept.
- If we choose percentage=0.1, and N=1, then we include all the cases, while choosing N=2 we include only the cases of the first variant (that are the 5% of the log, hence N=2 is not valid according to the above principle).
- If we choose percentage=0.05, and N=2, then we include exactly 5% of the cases of the log, that is the minimum requirement.