A: Implement the DoAggregations method as performant as possible for 3 supported aggregations: SUM, AVERAGE and COUNT DISTINCT.

Below is the link containing the solution for the above problem.

<https://dotnetfiddle.net/vIAgBn>

Time complexity is – O(m\*n)

Space complexity is also O(m\*n)

B: Are there any trade-offs if we allow a generic aggregations method to be used? E.g. other developers use your library and provide their own aggregation methods (such as Func<>)

* Yes, the aggregation method determines the time and space complexity.
* Additional functions will also need further code changes to allow generic methods and unit tests need to be written to handle that.