Events CSV (events-csv) TODO

* **TimeQuery and Early Parsing**. A time query should be pushed into the parser’s API, because we can optimize by not parsing complex events if they do not fall within the time window.
* **Workflow**

csv headers --last 'jhq0[1,4].\*mp-grams.\*onsumerCount' ./metrics.csv

csv --from <timestamp handled in a central location> --to <> -o <id-corresponding-to-mpgrams-before>, <id-corresponding-to-mpgrams-after>, …. <file>

With this I get the differences between active and standby and back during a failure event, and I can decide whether HA Franklin works.

* Refactor SimpleDateFormat everywhere, not thread safe.
* **Time Queries and Header Handling** – should be keeping track of the last header, so we have more info about the timed CSV events when applying time queries, or the self-contained information is sufficient? We need to consider this because the headers are non-timed, and if we’re using a time query, they will be thrown away. We probably don’t want that because they have a special meaning in CSV. Build a few test cases and model this.
* CSV vs Output  
  + CVSFormat, CVSFormatter –how those match with what’s now in Output.
* **CSV output in “processing” and “CSV”.** There is code that deals with CSV output both in events-processing/output and in the events-csv. Unify, one set of tests.
* **Reconcile ‘processing’ OutputFormat and ‘CSV’ CSVFormat,** when they are used together.
* **Databot has AsynchronousCsvLineWriter**, which also uses CSVFormatter. Reconcile.
* in-line graph