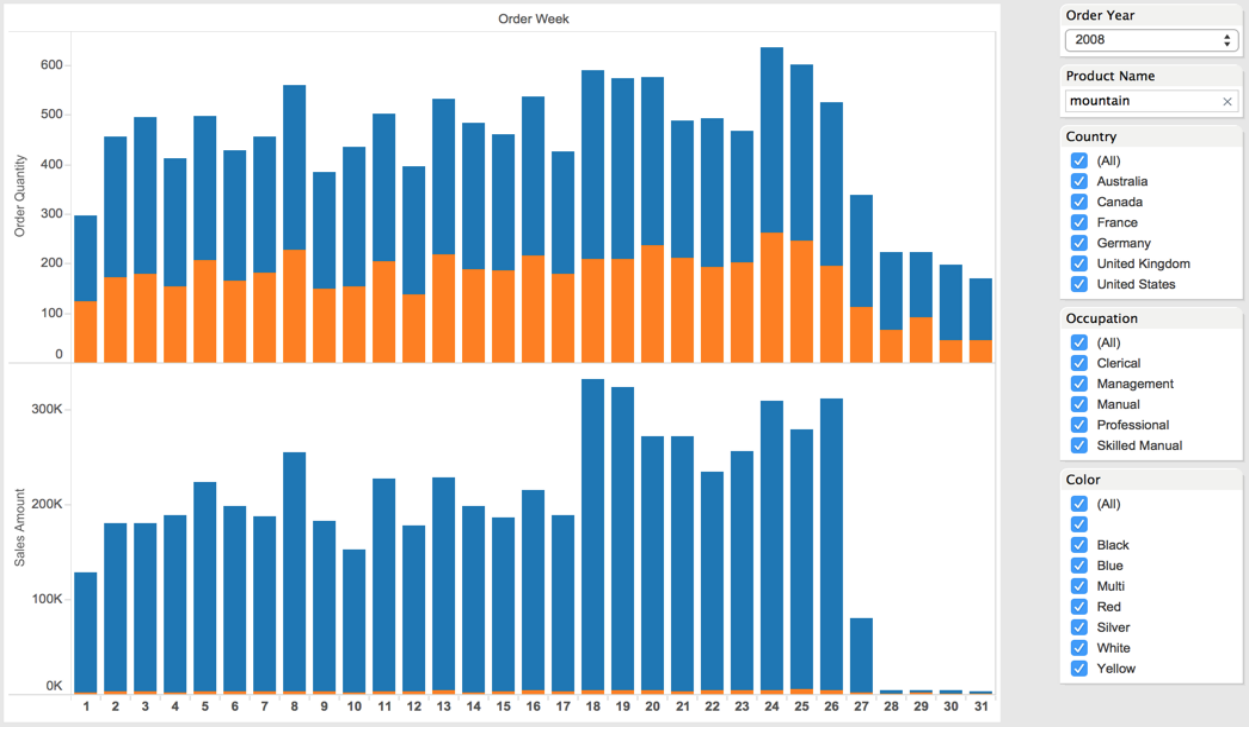
Filters and Dynamic Aggregates:

How AtScale aggregates help get the most out of Tableau filters

TIPS FROM ATSCALE: One of the great capabilities of Tableau is the ability to create visible, data-driven filters as part of an interactive visualization. We'll share a tip here on how to capitalize and get the most out of Tableau filters whether you are querying small or large datasets in Hadoop.

Key Components of Tableau Filters

Tableau's ability to create visible, data-driven filters as part of an interactive visualization allows users to dynamically adjust the conditions used to generate a visualization and analyze the resulting data. For example, the visualization below allows users to look at weekly sales trends for select production (in this case any Product Name that contains the string "mountain"), for a specific year (in this case "2008") and for a selection of buyer Occupations and product Colors.



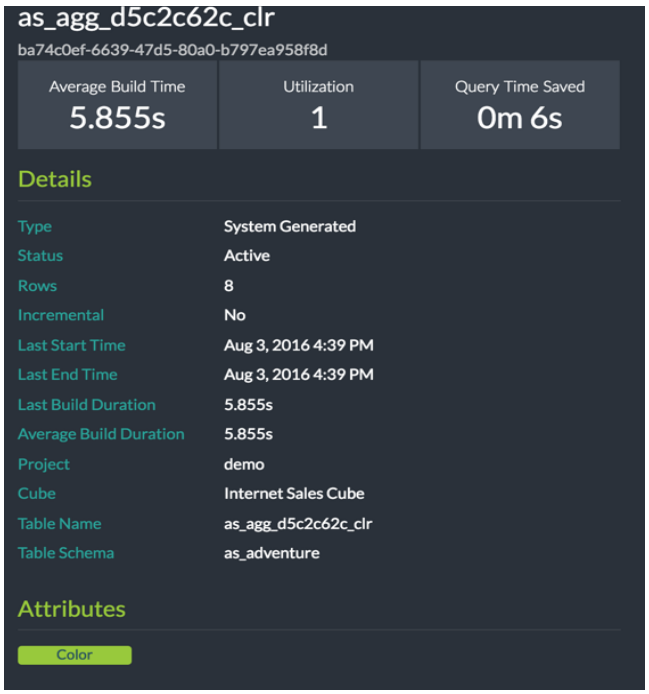
For smaller data sets, the contents of these filters - the list of distinct values for Occupation, Color, Year, and so on - can be obtained by doing a simple "SELECT DISTINCT" from the underlying data source. However, as datasets become larger, the population of filter contents can be cost-

prohibitive. For example, in our demo above, the values for the Color attribute are stored as key:value pairs in the core fact table (the sales_log table) in our Hadoop data set. In order to calculate the distinct values for color, the following query against the fact table is required:

```
SELECT
REGEXP_EXTRACT(factinternetsales_t2.product_info,'(^|,)color:([^,]+)',2)
FROM
   As_adventure.factinternetsales

GROUP BY 1
```

If this query needs to be executed against a fact table that may have million or billions of rows every time this dashboard is viewed, this will result in excessive load on the Hadoop cluster, as well as a very poor end user experience.



In order to eliminate these redundant queries for dimension values, the AtScale Adaptive Cache will dynamically generate aggregate tables to satisfy future Tableau queries for filter values. For example, in the case of "Color", AtScale will generate an aggregate table immediately after the initial query for the filter values. As a result, all subsequent queries for filter conditions will execute against this aggregate table instead of the fact table. The resulting query looks like.

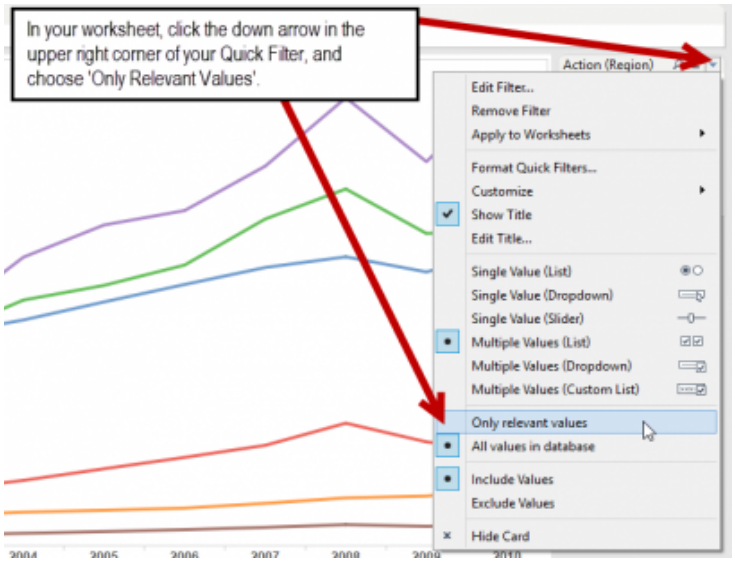
```
SELECT
   as_agg_d5c2c62c_clr_t2.color_c1
FROM
   as_adventure.as_agg_d5c2c62c_clr
GROUP BY
   1
```

This approach allows Tableau users to benefit from the rich filtering capabilities of Tableau while enabling an interactive visualization experience on Hadoop-scale data. However, in order to realize these benefits there are several "rules of thumb" that are recommended to ensure that filter population is as fast as possible.

The first is to eliminate the use of "Show Relevant Value" settings on ALL quick filters. This is suggested for two reasons:

This is suggested for two reasons:

1. Best Practice:
   This is a generally accepted best practice for improving Tableau workbook performance for large data sets.



2. Performance:
   As described above, the AtScale Engine has been designed to populate quick filters from aggregate tables. As a result "Show Relevant" values queries from Tableau can create filter conditions that are challenging for the AtScale Engine to resolve. For simple cases, the AtScale Query Engine will, in fact, try to ignore additional "WHERE" clauses on these queries. In general, the 'show relevant' values can generate very expensive queries that make aggregate table usage impractical or impossible. For example, in the sample visualization above, in order to show relevant values for a particular filter condition it is first required to find all fact table values that satisfy all other filter conditions in the view. So to get the relevant values for Color the engine would first need to find all fact records for the selected Year, Product Names, Countries, and Occupations. Then the SELECT DISTINCT COLOR values query would need to run on the fact table to populate the table. The sequential execution of these queries for all filter conditions would lead to significant performance degradation, especially with a large number of filters on the page. As a result, the AtScale Engine has not been optimized to support these "Relevant Value" queries - when possible it will ignore extra conditions on SELECT

Summary

For these reasons, we strongly recommend NOT using the "Show Relevant Values" option for any filters in Tableau.

We also recommend turning off the "Auto Update Filters" and "Auto Update Worksheet" options in the Worksheet > Auto Update menu option in Tableau. There are a number of different resources that recommend this approach when working with larger data sets. This allows the end-user to make a number of changes (e.g. filter selections) in the Worksheet without needing to wait for the data to refresh in between each selection.

By following these approaches Tableau users can get the best interactive experience using Quick Filters while still benefitting from the AtScale approach to supporting BI on Hadoop.