**Hands on Visualize and Explore Data in Kibana**

We’ll be working with a dataset that we’ve become familiar with: Rennes Traffic Data. In, the Creating a Logstash pipeline recipe in [Chapter 5](https://subscription.packtpub.com/book/data/9781837634293/5), we created a **Time Series Data Stream** (**TSDS**) and used **Logstash** to periodically fetch, transform, and send real-time traffic data to Elasticsearch.

In this chapter, we will create a new data stream named metrics-rennes\_traffic-raw.

Here are the steps to follow:

1. In Kibana, go to **Management** | **Dev Tools**, and execute the commands provided here: [https://github.com/PacktPublishing/Elastic-Stack-8.x-Cookbook/blob/main/Chapter6/snippets.md#preparing-regular-data-stream-metrics-rennes\_traffic-raw](https://github.com/PacktPublishing/Elastic-Stack-8.x-Cookbook/blob/main/Chapter6/snippets.md" \l "preparing-regular-data-stream-metrics-rennes_traffic-raw).

These commands allow us to create the following components for our regular metrics-rennes\_traffic-raw data stream in the following order:

* 1. **Index lifecycle policy**: metrics-rennes\_traffic-raw-lifecycle-policy
  2. **Index mapping**: metrics-rennes\_traffic-mappings@raw
  3. **Index settings**: metrics-rennes\_traffic-raw-settings
  4. **Index template**: metrics-rennes\_traffic-raw-index-template

1. Once you have prepared the mapping, setting, and index template for the metrics-rennes\_traffic-raw data stream, go to your Logstash server and create the rennes-traffic-raw.conf Logstash configuration file in the Logstash configuration folder. For Debian, this folder is located at /etc/logstash/conf.d/. You can find the reference configuration file at this URL: <https://github.com/PacktPublishing/Elastic-Stack-8.x-Cookbook/blob/main/Chapter6/logstash-conf/rennes_traffic-raw.conf>. Should you deploy Logstash elsewhere, the directory layout for different Logstash installations is detailed here: <https://www.elastic.co/guide/en/logstash/8.12/dir-layout.html>.
2. Modify the output section of the configuration file with your own Elastic deployment connection details (Refer to the *Creating a Logstash pipeline* recipe in [*Chapter 5*](https://subscription.packtpub.com/book/data/9781837634293/5) for more details).
3. Restart your Logstash server to apply the configuration file with the following command:

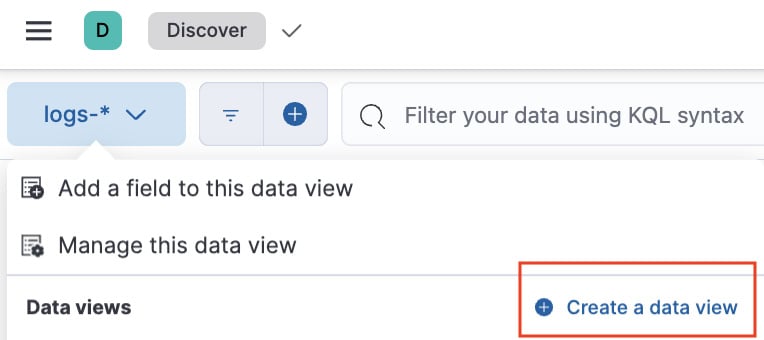
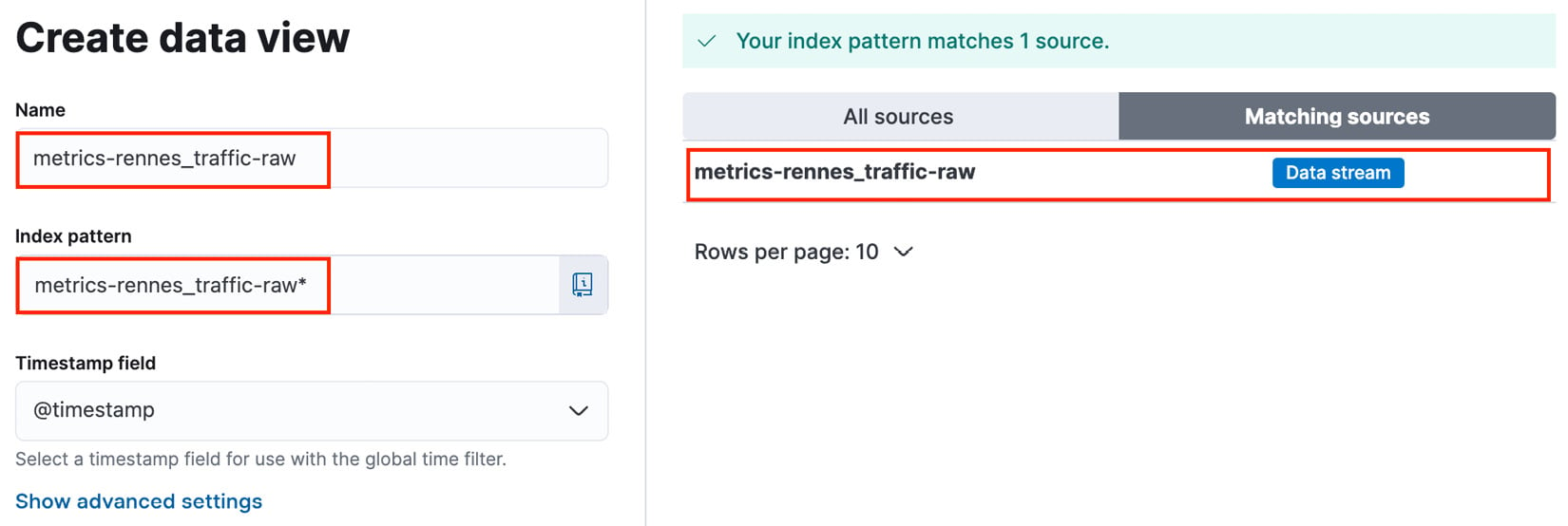
sudo systemctl restart logstash.service

At this stage, your Logstash instance has been configured to send the data to our new data stream, metrics-rennes\_traffic-raw. In the next steps, we will create a data view for the data stream to verify the data ingestion.

How to do it...

You will first need to create a **data view**. Those familiar with Elastic Stack might be wondering what a data view is. Essentially, a data view is the rebranded term for what used to be called an **index pattern**. Both concepts function as a method for directing Kibana to the location of your Elasticsearch data. Essentially, a data view designates which indices, or groups of indices, Kibana should search through when retrieving data. Furthermore, data views offer the flexibility to modify data fields, including the creation of new fields using runtime fields.

Head to **Kibana**| **Analytics** | **Discover** and follow these steps:

1. In the upper left, navigate to the **Data view** menu and select **Create a data view**. This action will open the data view creation flyout, as shown in *Figure 6.1*:  
   
2. Enter the following information to configure your data view:
   1. Enter the name: metrics-rennes\_traffic-raw
   2. In the **Index pattern** field, start typing the name of the data stream. By typing metrics-rennes\_traffic-raw, you should see, on the right-hand side, the matching source.  
      
   3. Leave **Timestamp field** with the **@timestamp** value.
   4. Click on **Save data view to Kibana** at the top of the screen.

You should now see the Rennes data and the available fields in Discover, as illustrated in *Figure 6.3*:

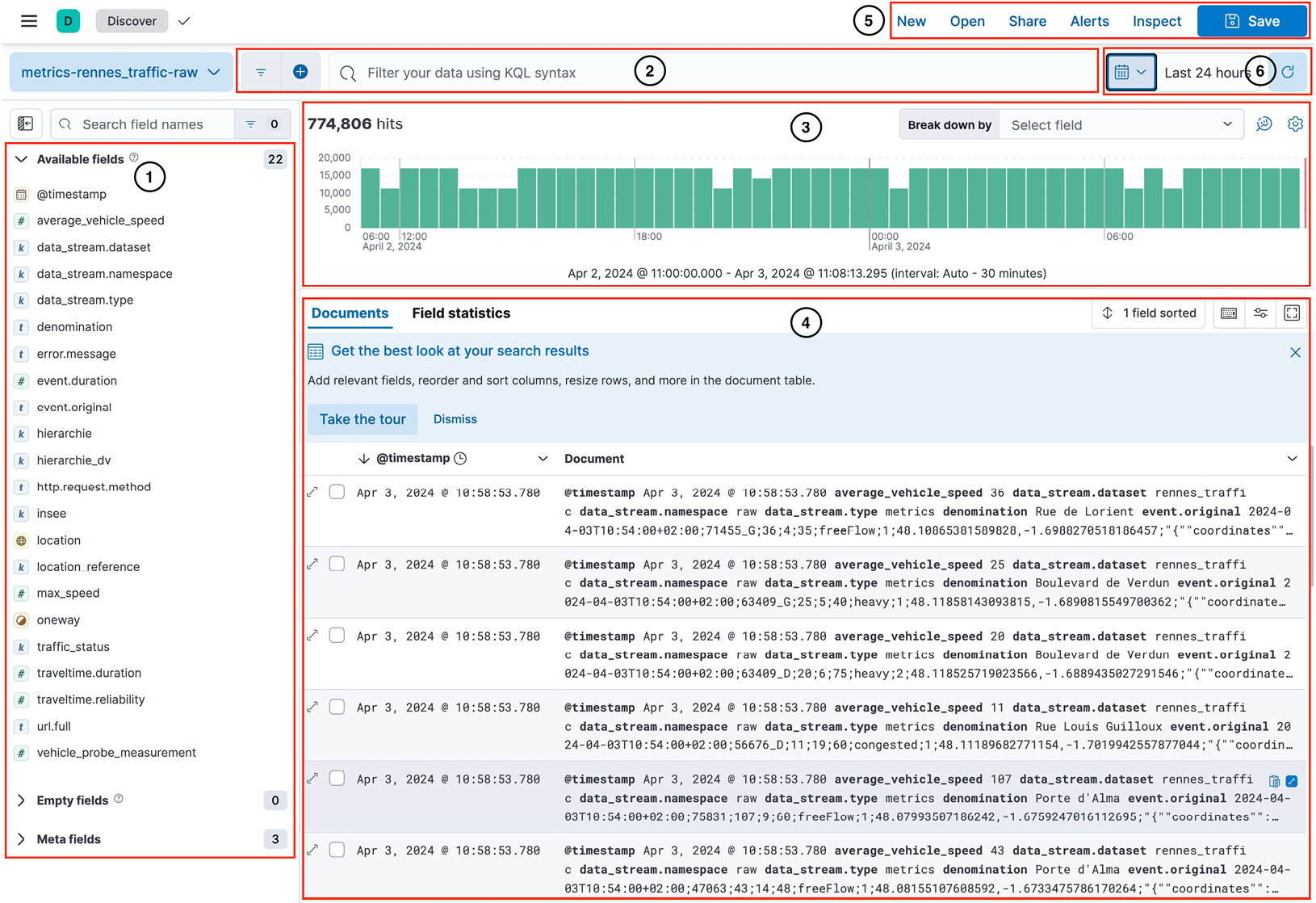


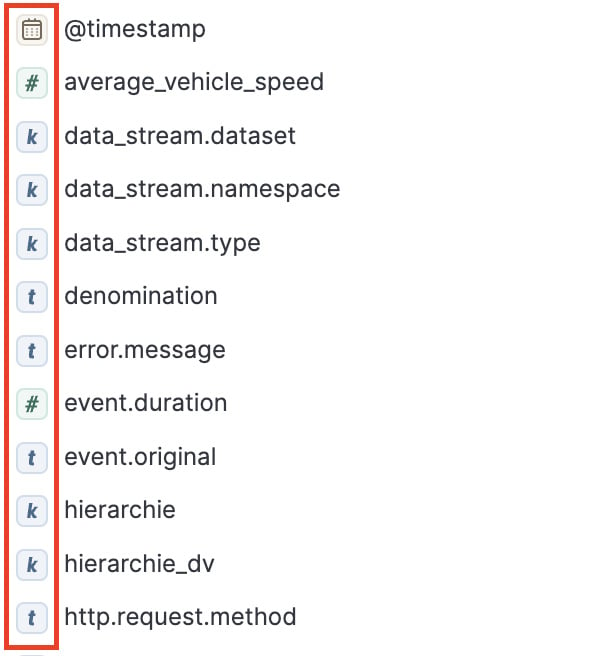
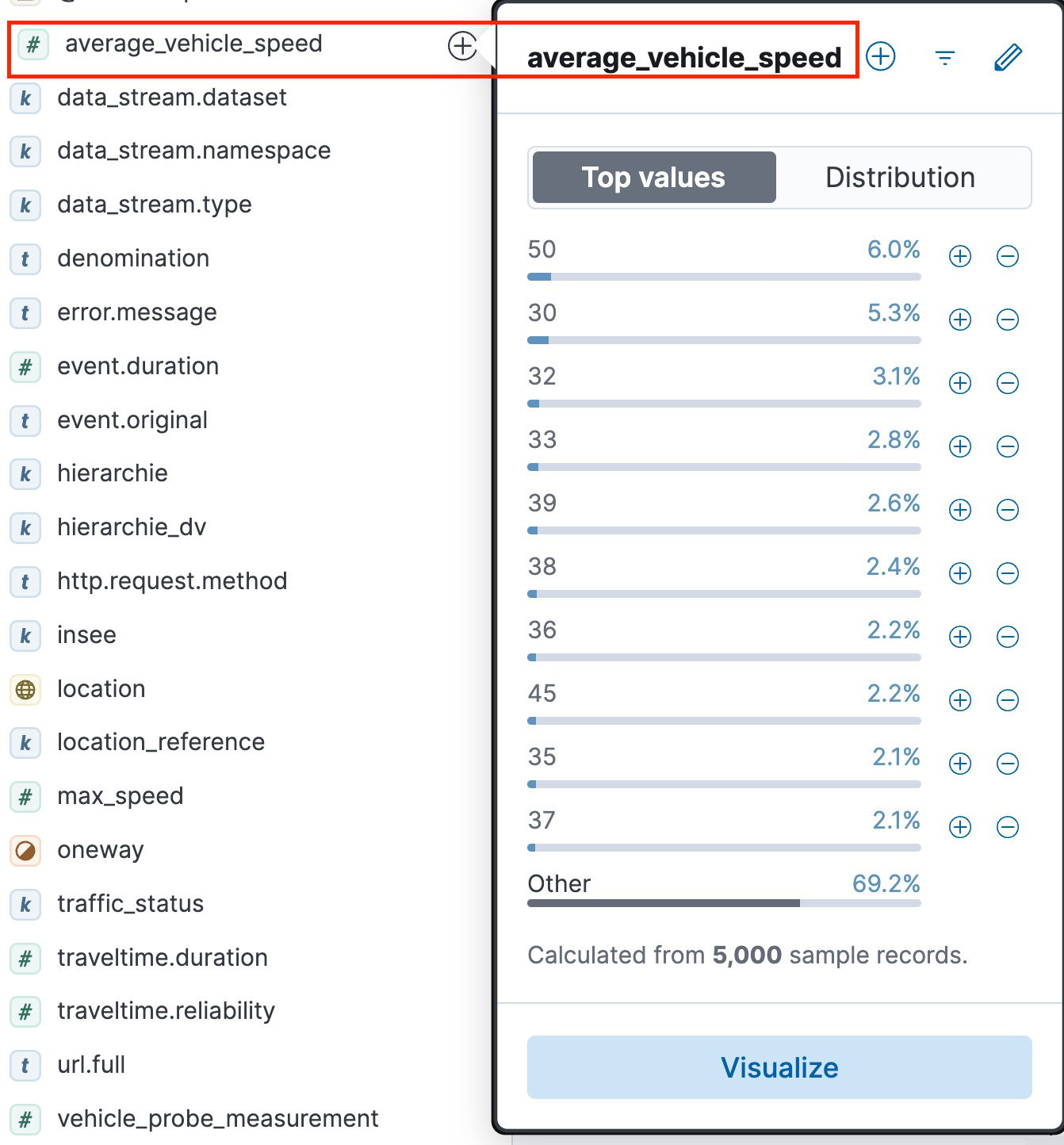
Figure 6.3 – Discover panel breakdown overview

Discover can be broken down into the following sections, as numbered in *Figure 6.3*:

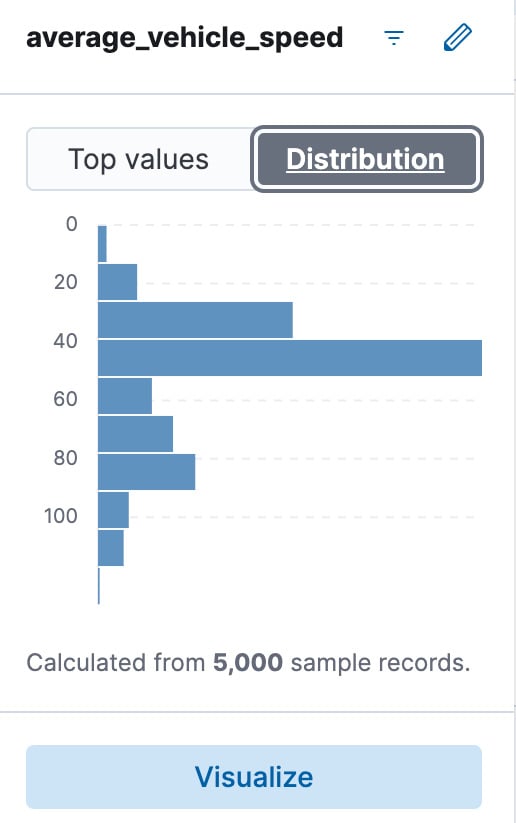
1. **Available fields**: This shows the available fields, empty fields, and meta fields from the documents.
2. **Search bar and filter**: Use these to apply **Kibana Query Language** (**KQL**) or **Lucene queries** for filtering and analyzing documents.
3. **Histogram**: By default, this displays the count of documents broken down by an automatically determined time interval.
4. **Documents table**: This shows the documents that match the criteria entered in the search bar or fall within the selected time range. It allows for customization to include additional columns.
5. **Menu bar**: This provides access to the main actions in Discover.
6. **Time picker**: This enables you to quickly select the time range you wish to apply to the documents.

We will cover each of the sections in detail in the next steps of the recipe.

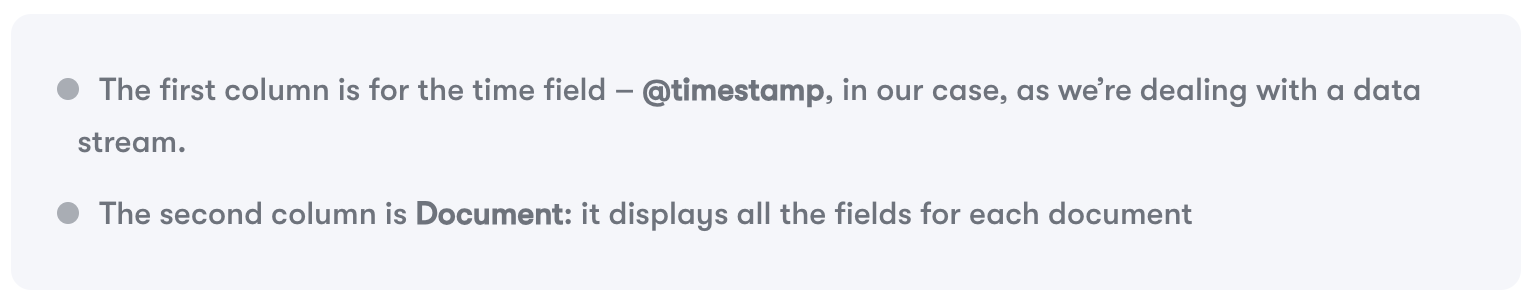
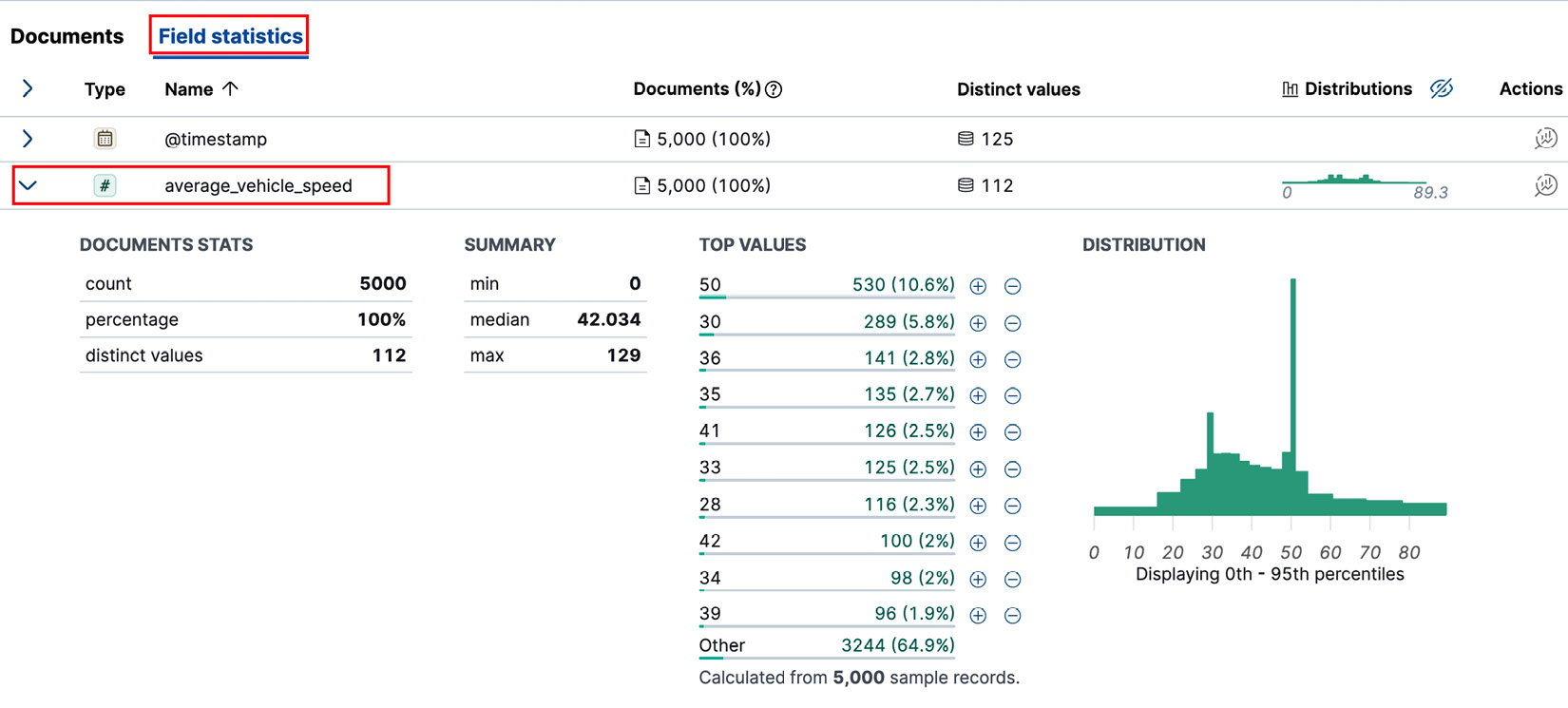
Now that you have some data in Discover, we can start exploring the indexed documents:

1. On the left of the screen, you can see a list called **Available fields**. Those are the non-empty fields in our dataset. There is a small icon at the left of each field name; this serves as an indication of the field type (keywords, numeric, geo, date, etc.):  
   
2. From the available fields, click on the **average\_vehicle\_speed** field and you’ll see a contextual window showing two columns – **Top values** and **Distribution**:  
     
     
   Figure 6.5 – Top values and Distribution

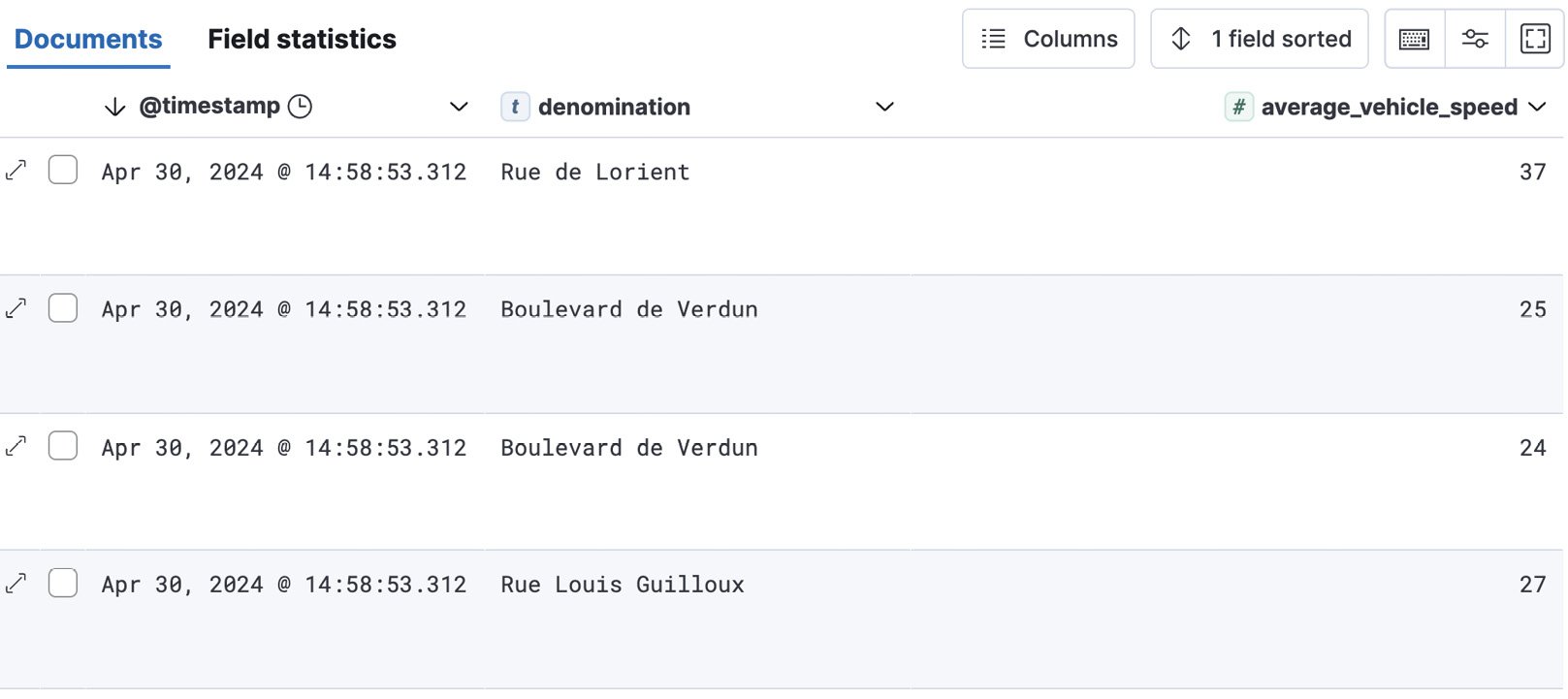
The **Top values** column shows the most common values that appear within a specific field across all documents that match the current search or filter criteria. It provides a quick overview of the predominant values for that field, which is useful for identifying patterns, trends, or anomalies in your data

1. Click on the **Distribution** column to get an idea of the value distribution across the documents. This feature allows you to visualize the frequency of values or ranges of values within a field across the dataset that matches the search or filter criteria. This can be particularly useful for numeric fields or date fields where you’re interested in understanding the spread or distribution of values:  
     
   Figure 6.6 – Distribution histogram for average\_vehicle\_speed

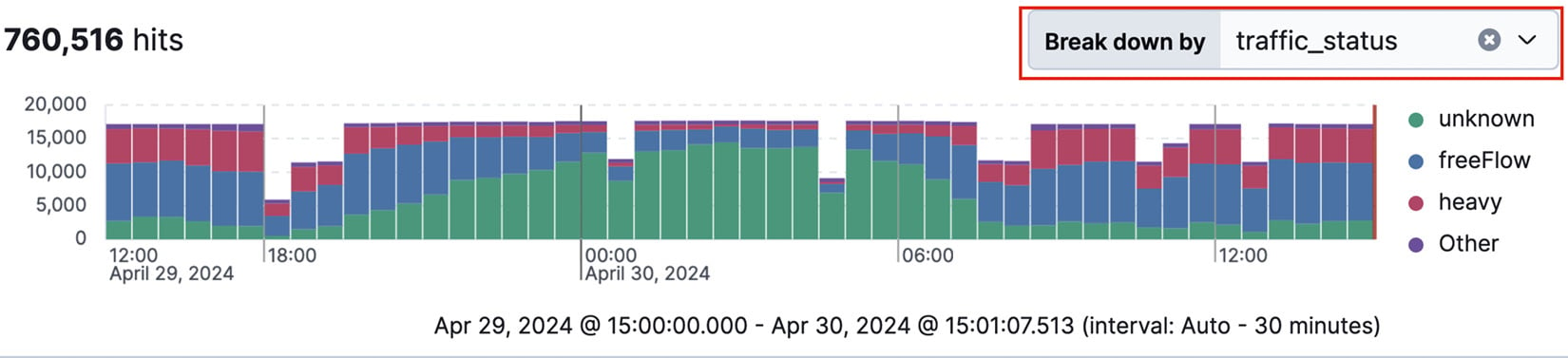
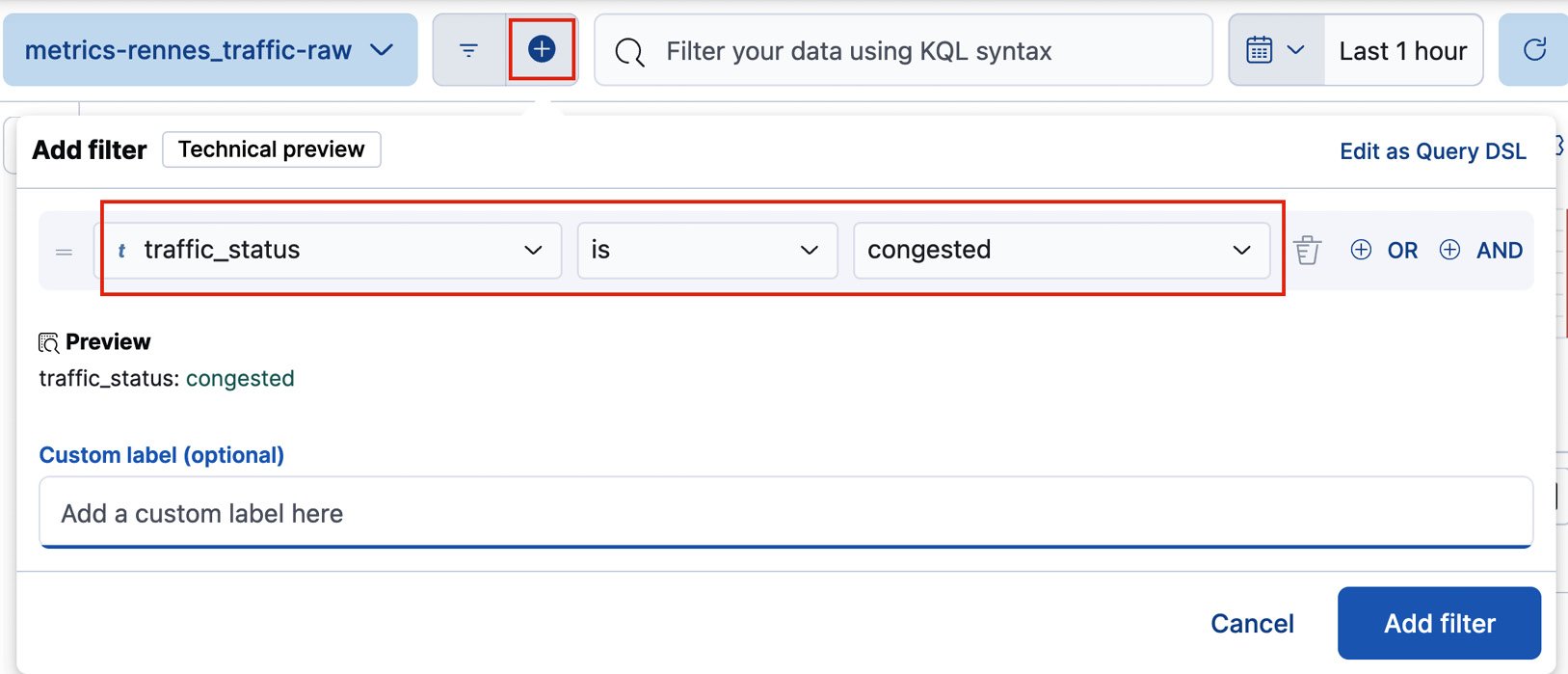
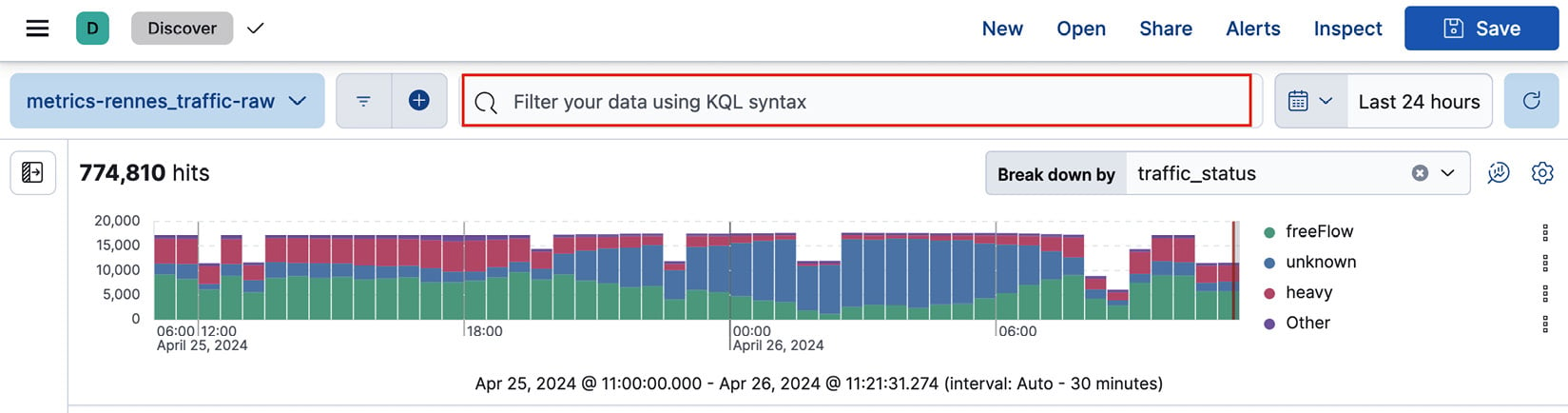
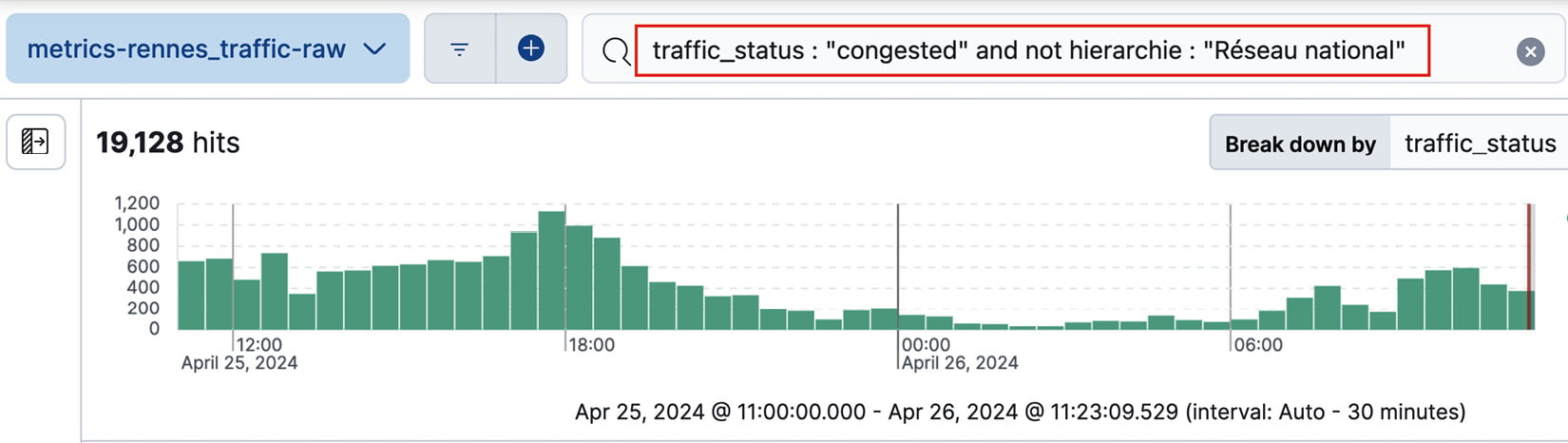
In our case, we see that we have most traffic documents with an average speed of 40 km per hour.

1. Now, let’s look at the **Documents** table displayed in the center of the **Discover** application. By default, it has two columns:  
     
   You can expand a single document and view all the fields either in table or JSON format.
2. Starting with Kibana version 8.1, a valuable machine learning feature, **Field statistics**, was integrated into Discover. Click on **Field statistics** at the top of the table and expand the average\_vehicle\_speed field to quickly view its top values, cardinality, the percentage of documents in which it is present, and its distribution:  
     
   Figure 6.7 – Field statistics

Let’s update the table with two new columns.

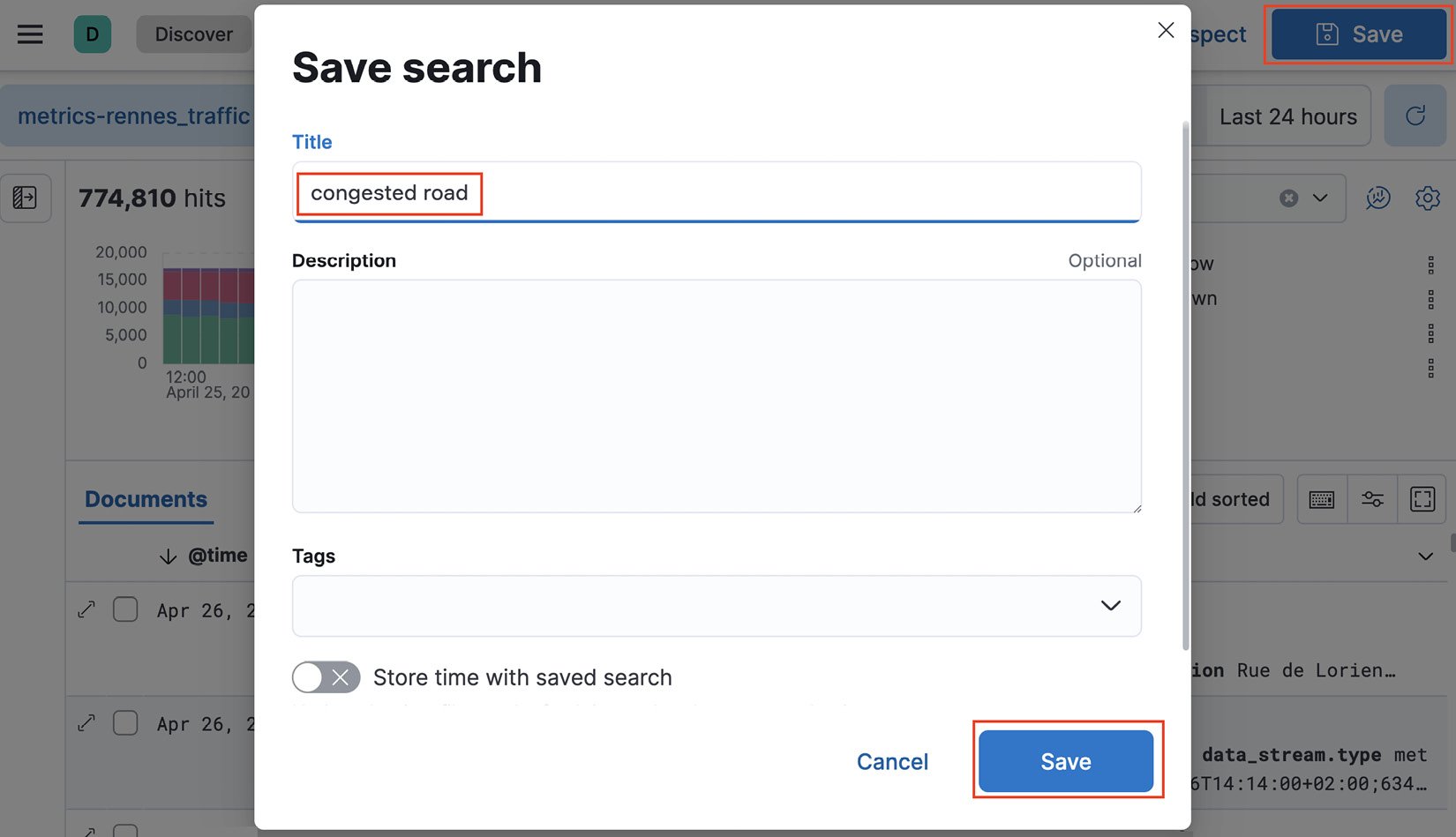
1. Start by selecting the **denomination** field from the options on your left and drag it onto the **Documents** table to add it as a new column. Then, similarly, add the **average\_vehicle\_speed** field by dragging it into the table. These changes will enrich your table with data displaying the average speed for each road, alongside the timestamps of these measurements:  
     
   Figure 6.8 – Documents table with a denomination column added

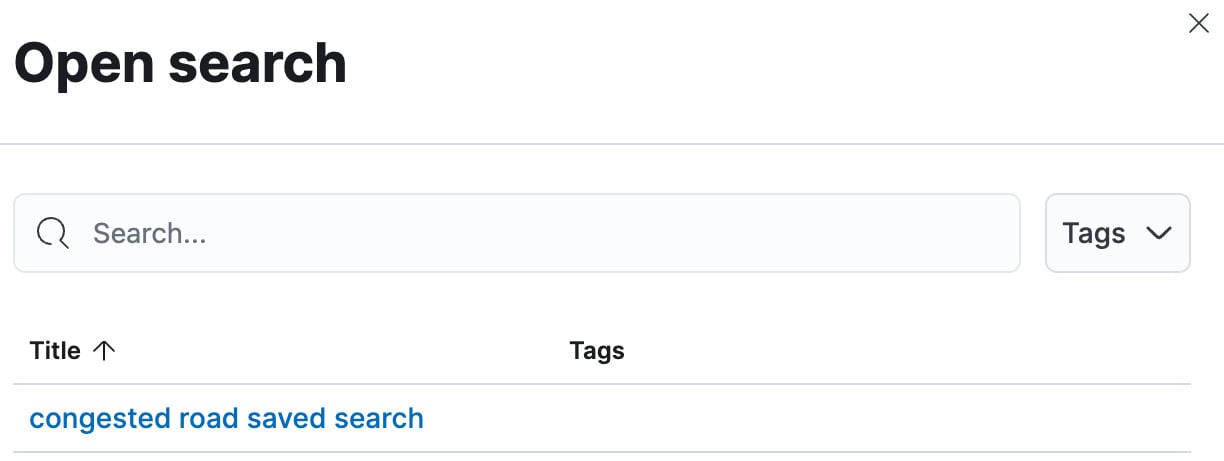
You can sort the table by any of the columns simply by clicking on the column header at the top of the table.

1. At this stage, you have likely noticed the histogram visualization displayed above the **Documents** table. The histogram is one of Discover’s hallmark features, and it can be customized to suit your requirements.
2. Above the histogram visualization, look for the **Break down by** field. Click on the drop-down menu next to this text and select **traffic\_status**. The histogram will now segment the data by traffic status, providing a clear visualization of the status distribution across the incoming documents, as illustrated in *Figure 6.9*:  
   
3. One of Discover’s standout features is its ability to combine text search and filtering. Suppose we want to filter our documents by a specific traffic status, such as the **congested** one. Click on the **+** sign to the right of the data view name to open the **Add filter** menu. Configure the following and click on **Add filter**:
   1. **Field**: **traffic\_status**
   2. **Operator**: **is**
   3. **Value**: **congested**  
        
      **Once the filter is applied, notice that the Discover page updates to display only the documents with the congested traffic status.**
4. Next, we will utilize the KQL to achieve the same outcome as the filter we just set. Remove the existing filter and click inside the query bar at the top of the histogram:  
   
5. Enter the following query and hit *Enter*:  
     
     
   Notice how the autocomplete feature suggests field names as you type. We can refine our query by adding another filter for the road type with the following condition:  
     
     
     
     
   Let’s save our customized query and **Documents** table as a saved search. Saved searches allow users to quickly access frequently used queries and filters, enhancing workflow efficiency. They are also useful when building dashboards.

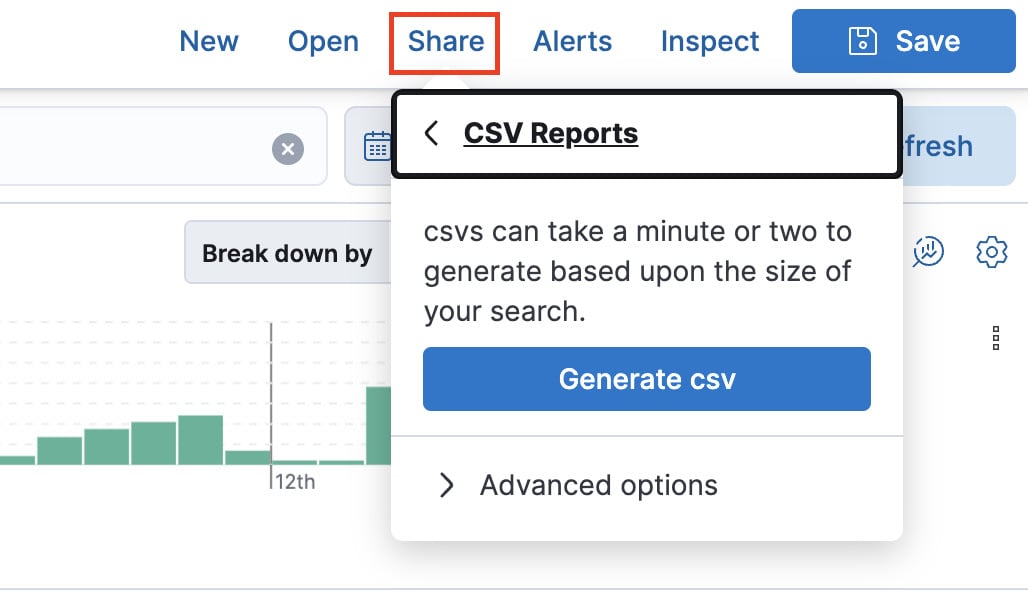
traffic\_status : "congested" and not hierarchie : "Réseau national"

traffic\_status: " congested"

1. Click **Save** in the top-right corner of the screen and name it—let’s call it congested road in the popup:  
   
2. To access your saved search, click on the **Open** option in the top-right corner of the screen, and select your saved search, as shown in *Figure 6.14*:



1. Finally, let’s export our results as an easily shareable CSV report. Click **Share** in the top-right corner, choose **CSV Reports**, and then select **Generate CSV**.  
     
   Note

The CSV export feature in Discover within Kibana does have limits. By default, it is configured to export up to a maximum number of rows, which can be adjusted in the Kibana settings (kibana.yml). The limit is in place to manage performance and resource utilization on both the Elasticsearch cluster and the Kibana server. If the number of documents matching your search criteria exceeds this limit, only the top documents, up to the specified limit, will be exported. To export larger datasets, adjustments to the xpack.reporting.csv.maxSizeBytes settings in kibana.yml may be necessary. However, always consider the performance implications of exporting large datasets.  


1. The CSV is generated in the background; once it is completed, a popup at the bottom right of the screen will appear. You can now click on the **Download report** button to retrieve the CSV file:  
   