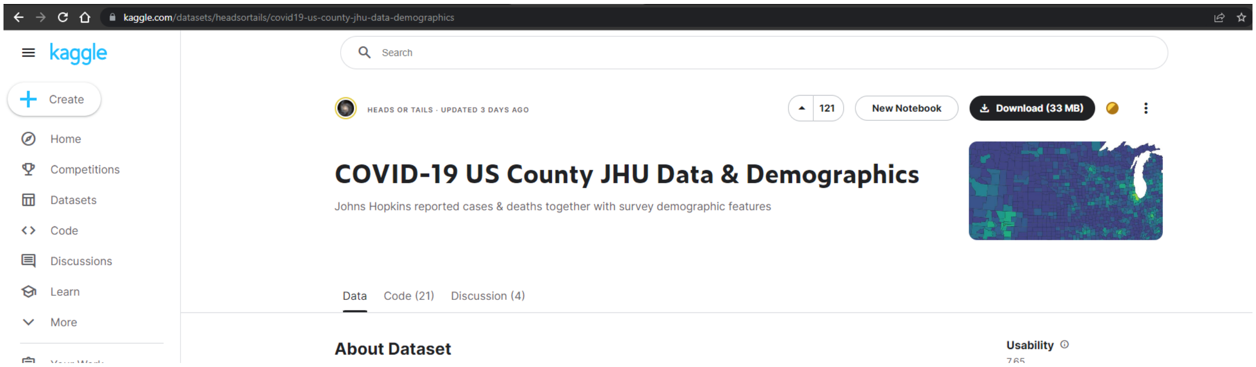
Group 2 Project Tutorial

William Gomez | Robin Huynh | Clever I Lemus | Luis Rodriguez | Jung Tae Cha

Importing CSV COVID-19 US County JHU & Demographics data into Kibana of Elasticsearch

Step 1 - Download the data from Kaggle

1. You need to launch and open Kibana
2. We are going to download the data files from the Kaggle site (https://www.kaggle.com/datasets/headsortails/covid19-us-county-jhu-data-demographics) and then import it into Elasticsearch.



kaggle.com/datasets/headsortails/covid19-us-county-jhu-data-demographics

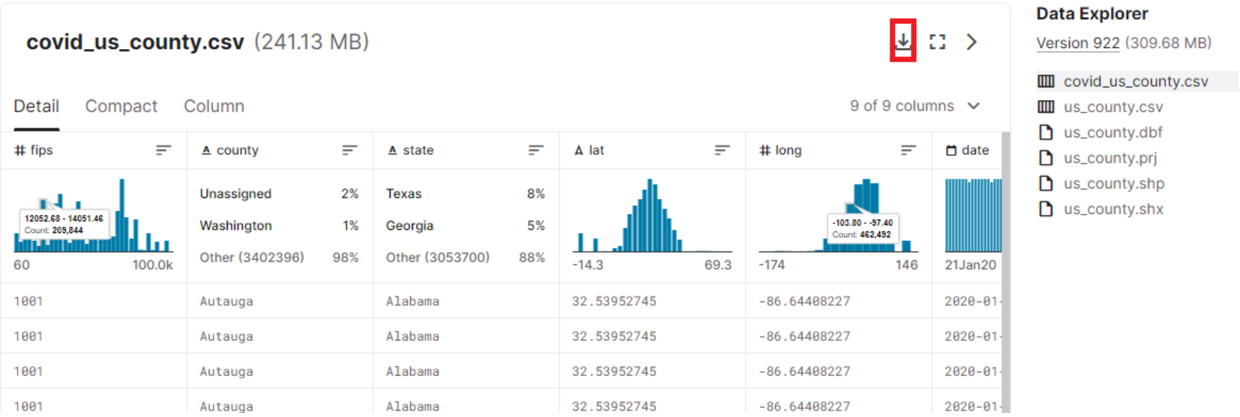
COVID-19 US County JHU Data & Demographics

Johns Hopkins reported cases & deaths together with survey demographic features

About Dataset

Usability

3. Once you get into the website, scroll down and click on the download button:



covid_us_county.csv (241.13 MB)

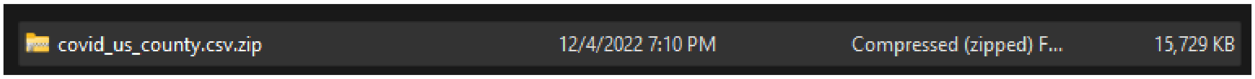
# fips	A county	A state	A lat	# long	date
60 100.0k	Unassigned 2% Washington 1% Other (3402396) 98%	Texas 8% Georgia 5% Other (3053700) 88%	-14.3 69.3	-174 146	21Jan20
1001	Autauga	Alabama	32.53952745	-86.64408227	2020-01
1001	Autauga	Alabama	32.53952745	-86.64408227	2020-01
1001	Autauga	Alabama	32.53952745	-86.64408227	2020-01
1001	Autauga	Alabama	32.53952745	-86.64408227	2020-01

Data Explorer

Version 922 (309.68 MB)

- covid_us_county.csv
- us_county.csv
- us_county.dbf
- us_county.prj
- us_county.shp
- us_county.shx

4. The file will be downloaded in a .ZIP format, let's extract it to our desired location.

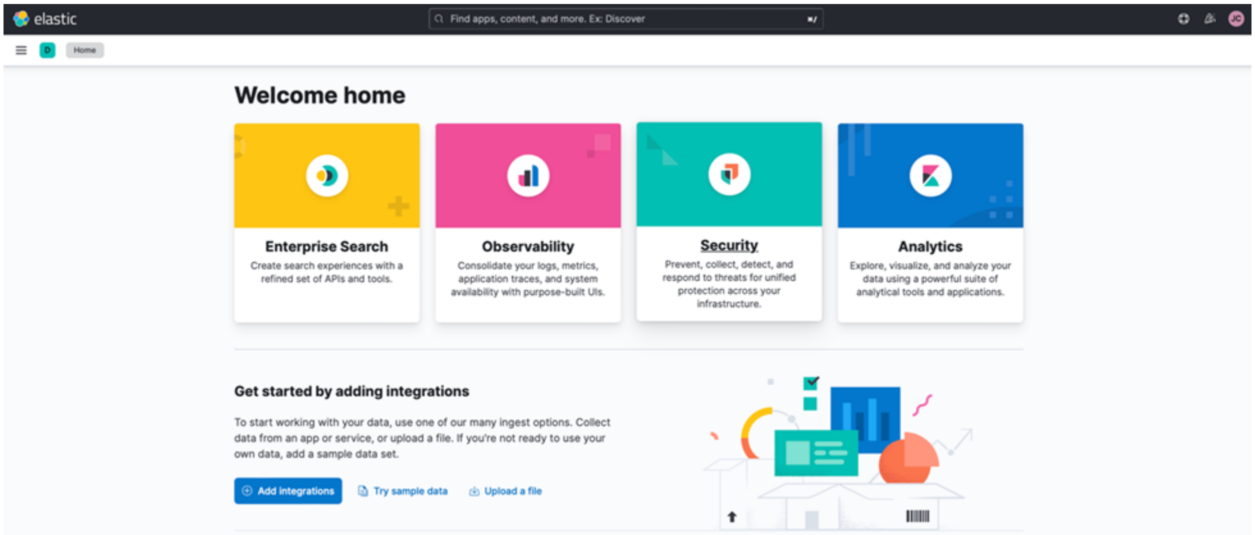


covid_us_county.csv.zip	12/4/2022 7:10 PM	Compressed (zipped) F...	15,729 KB

Step 2 - Import the data from CSV and Creating Data View

Now that we have the data to analyze, we need to import it and set up an index.

1. Within Kibana, click on **Upload a file**.



Welcome home

Enterprise Search
Create search experiences with a refined set of APIs and tools.

Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

Analytics
Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

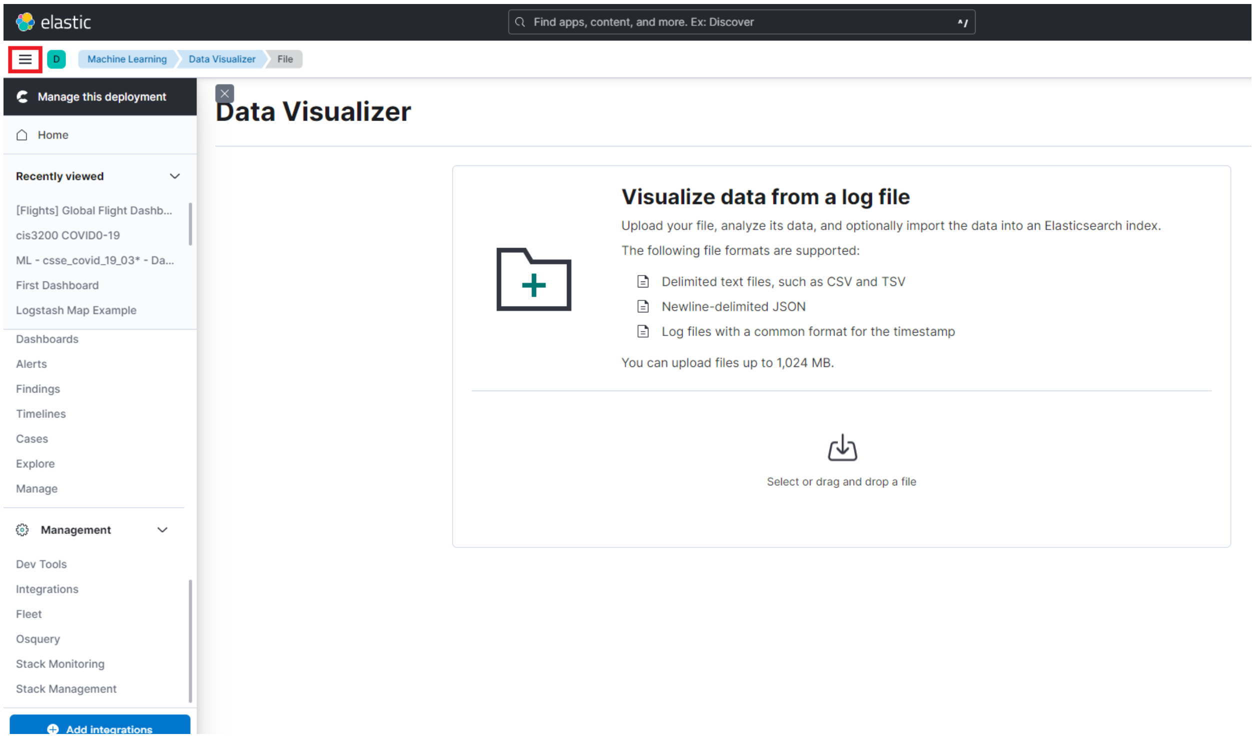
Get started by adding integrations

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, add a sample data set.

2. Let’s select our CSV data set by clicking on “**Select File**”.

*** If you get an error telling you that there is a limit to the upload size, we can fix that.

a. Let’s click on the menu button.



Data Visualizer

Visualize data from a log file

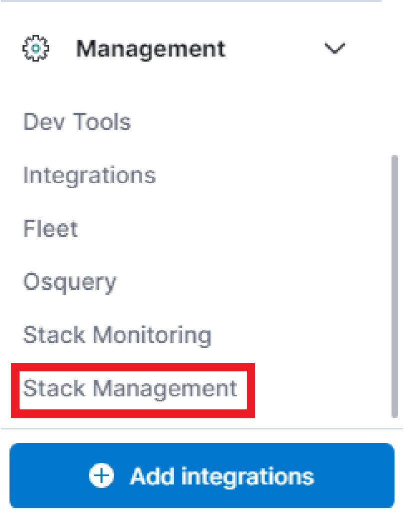
Upload your file, analyze its data, and optionally import the data into an Elasticsearch index.

The following file formats are supported:

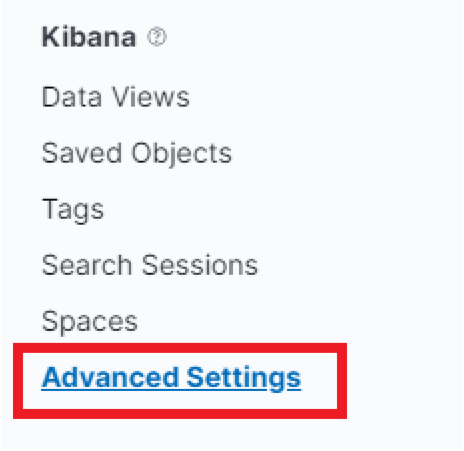
- Delimited text files, such as CSV and TSV
- Newline-delimited JSON
- Log files with a common format for the timestamp

You can upload files up to 1,024 MB.

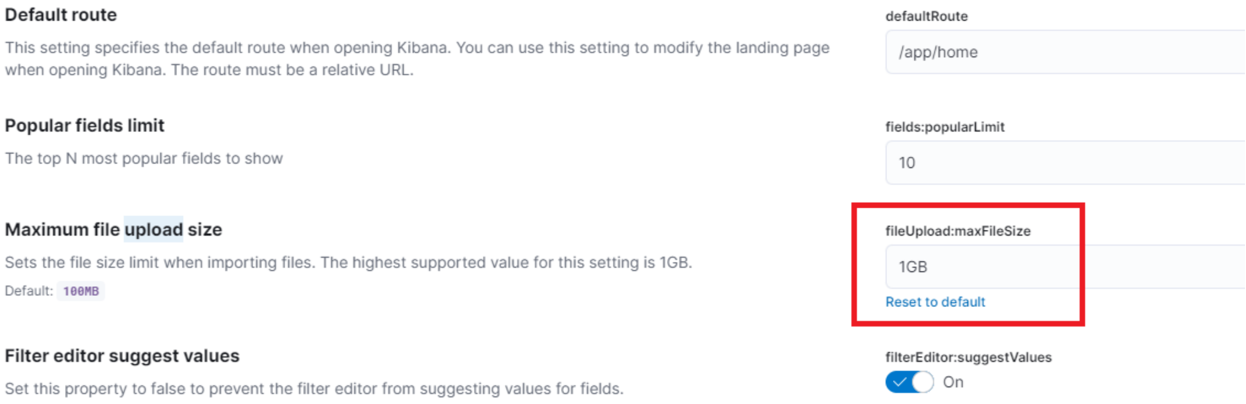
b. Let’s select “**Stack Management**” under the **“Management**” subnav.



c. Let’s click on “Advanced Settings”



d. Under “**General**” settings, we scroll down to **“Maximum file upload size**”. It seems like the default is 100MB, we can change it whatever size we desire, with the maximum upload size being **1GB**, so let’s change it to that.



Default route

This setting specifies the default route when opening Kibana. You can use this setting to modify the landing page when opening Kibana. The route must be a relative URL.

defaultRoute: /app/home

Popular fields limit

The top N most popular fields to show

fields:popularLimit: 10

Maximum file upload size

Sets the file size limit when importing files. The highest supported value for this setting is 1GB.

Default: `100MB`

fileUpload:maxFileSize: 1GB

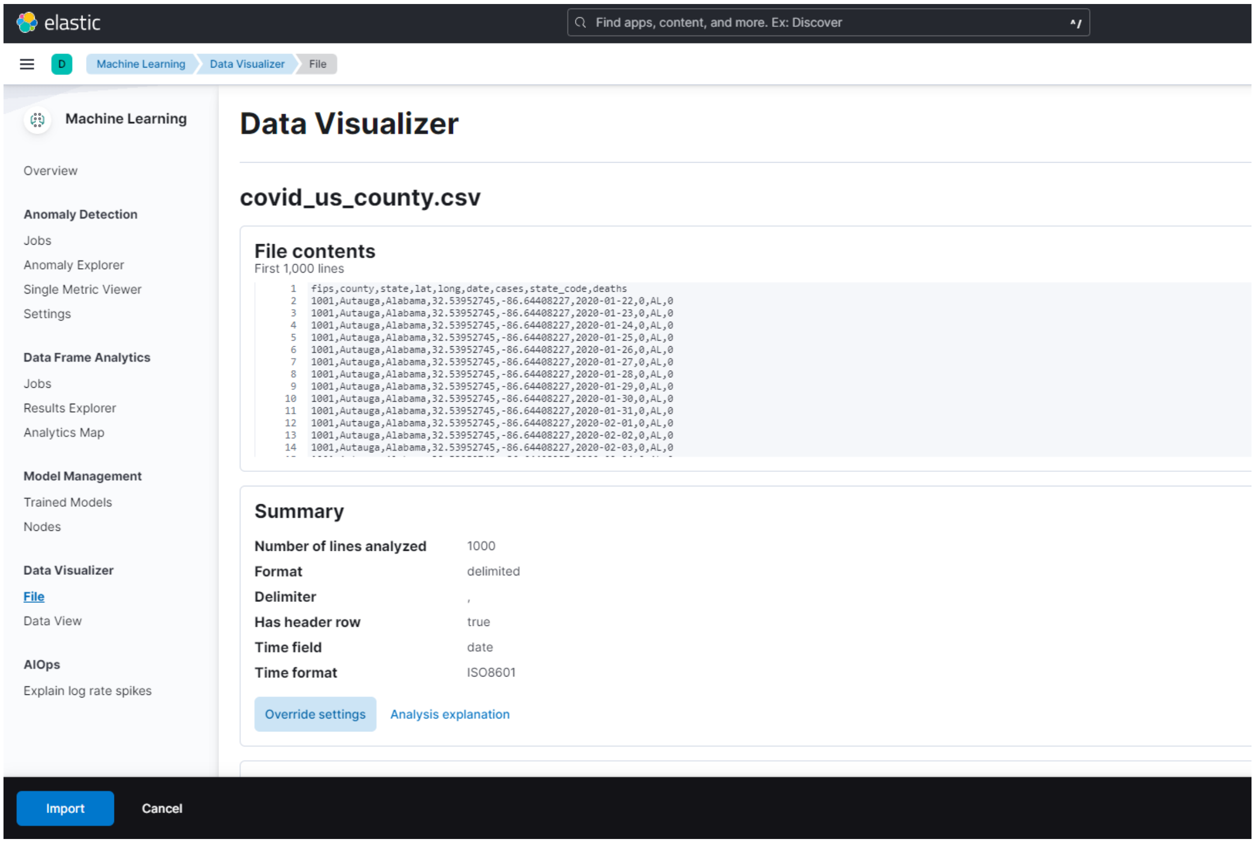
Filter editor suggest values

Set this property to false to prevent the filter editor from suggesting values for fields.

filterEditor:suggestValues: On

With this, we fixed our upload issue and can continue uploading our data to Kibana.***

4. We select the covid_us_county.csv file, shortly it will show a preview of the data.



Data Visualizer

covid_us_county.csv

File contents

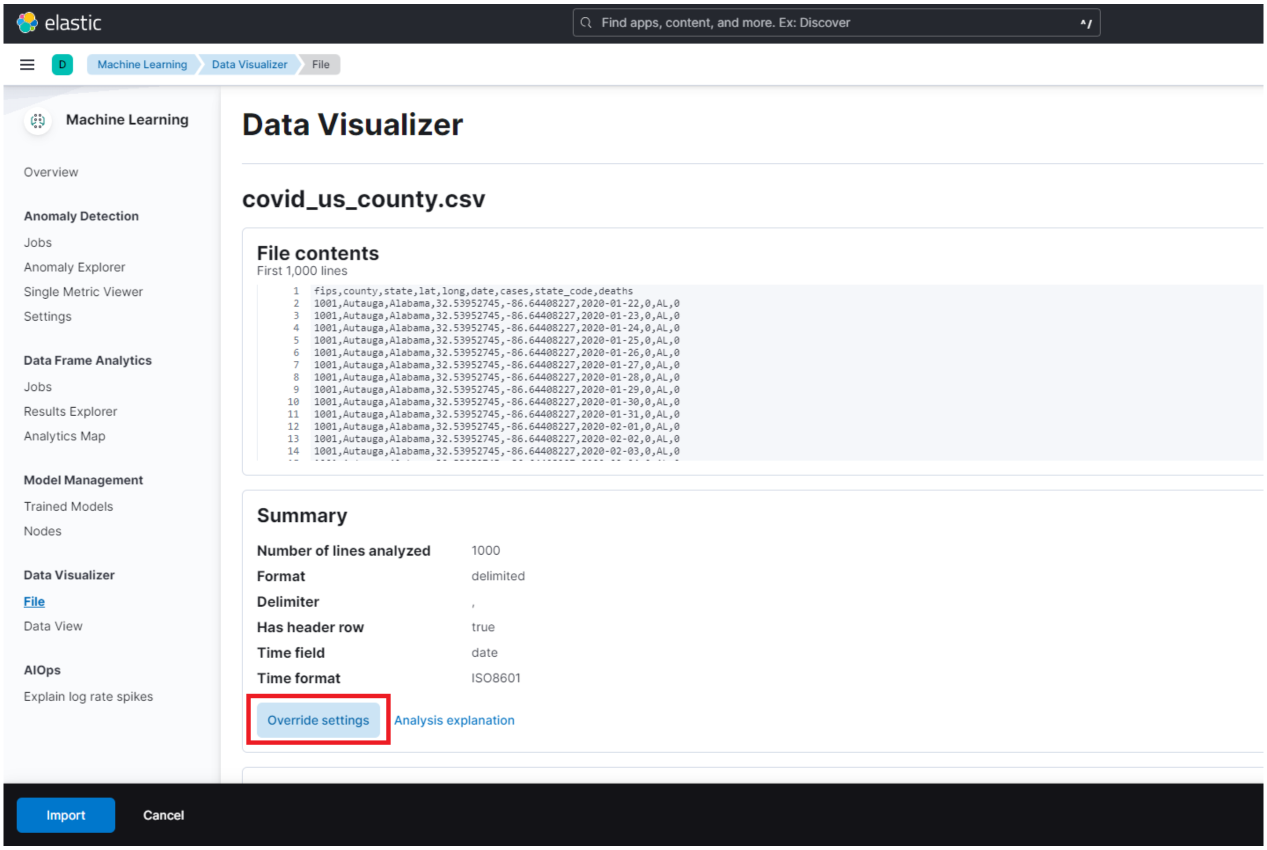
First 1,000 lines

```
fips,county,state,lat,long,date,cases,state_code,deaths
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-22,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-23,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-24,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-25,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-26,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-27,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-28,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-29,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-30,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-31,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-02-01,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-02-02,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-02-03,0,AL,0
```

Summary

Number of lines analyzed	1000
Format	delimited
Delimiter	,
Has header row	true
Time field	date
Time format	ISO8601

5. Now let’s select **Override settings** and change the column names as: “Lat” to “**Latitude**” and “Long_” to “**Longitude**”. Then, we click on **Apply**.



Data Visualizer

covid_us_county.csv

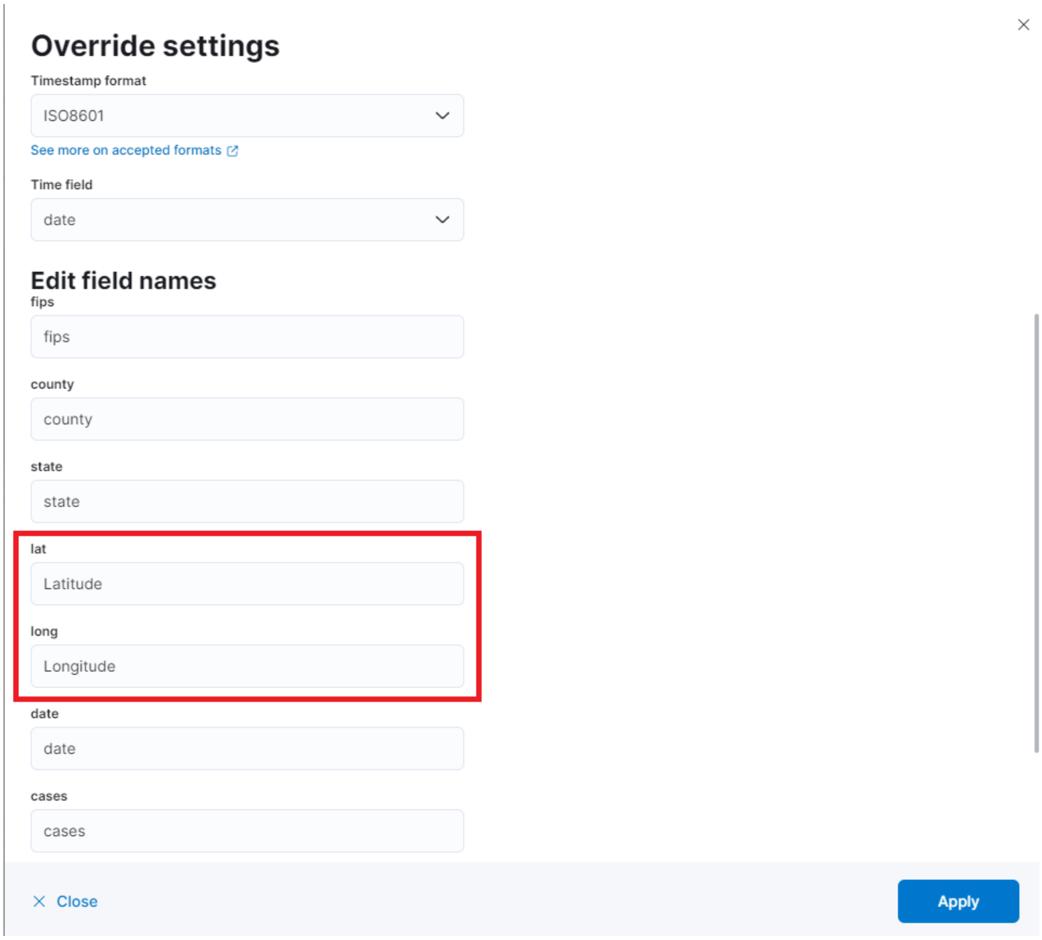
File contents

First 1,000 lines

```
fips,county,state,lat,long,date,cases,state_code,deaths
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-22,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-23,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-24,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-25,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-26,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-27,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-28,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-29,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-30,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-01-31,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-02-01,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-02-02,0,AL,0
1001,Autauga,Alabama,32.53952745,-86.64408227,2020-02-03,0,AL,0
```

Summary

Number of lines analyzed	1000
Format	delimited
Delimiter	,
Has header row	true
Time field	date
Time format	ISO8601



Override settings

Timestamp format: ISO8601

See more on accepted formats

Time field: date

Edit field names

fips: fips

county: county

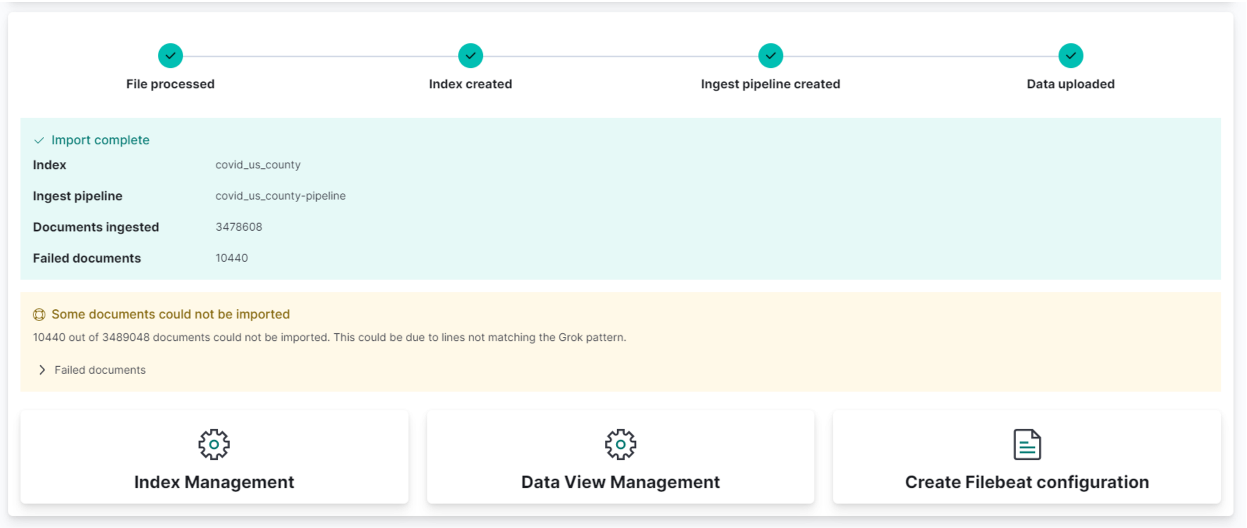
state: state

lat: Latitude

long: Longitude

date: date

cases: cases



Import complete

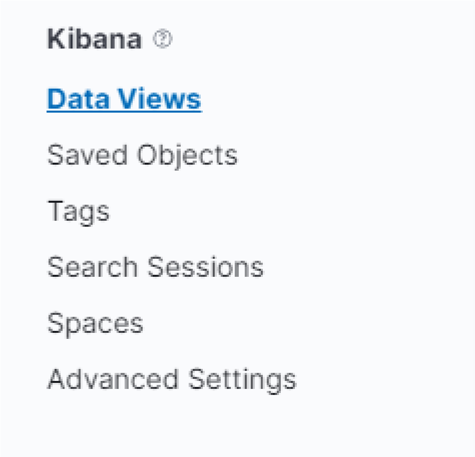
Index	covid_us_county
Ingest pipeline	covid_us_county-pipeline
Documents ingested	3478608
Failed documents	10440

Some documents could not be imported

10440 out of 3489048 documents could not be imported. This could be due to lines not matching the Grok pattern.

Failed documents

6. Now that we have our data uploaded, we can create our data view which will allow us to use that for many other things, such as visualizations and machine learning. We're going to click on Data Views under Kibana.



Kibana

Data Views

Saved Objects

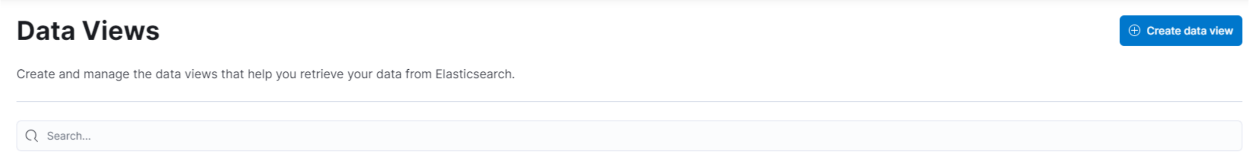
Tags

Search Sessions

Spaces

Advanced Settings

Select “Create Data View”.



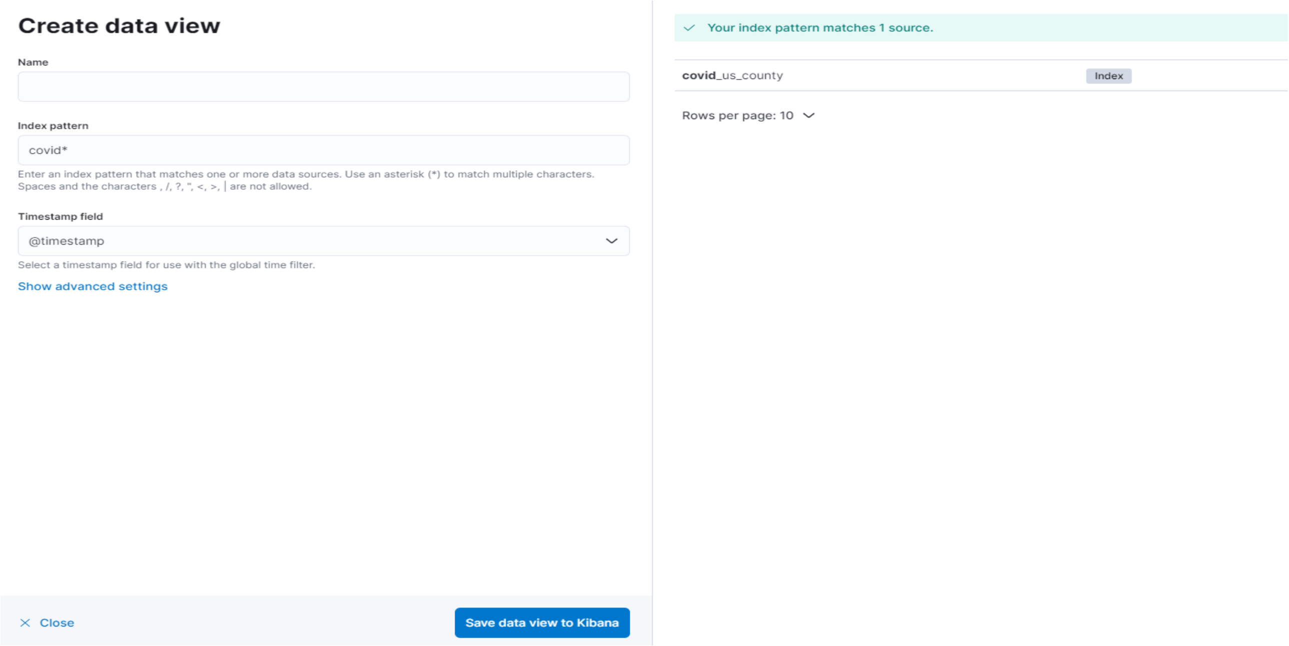
Data Views

Create data view

Create and manage the data views that help you retrieve your data from Elasticsearch.

Search...

Let’s choose an index pattern name and let’s make sure that it matches our uploaded data. Also, let’s make sure that timestamp is the selected timestamp pattern.



Create data view

Name

Index pattern

covid*

Enter an index pattern that matches one or more data sources. Use an asterisk (*) to match multiple characters. Spaces and the characters , /, ?, ", <, >, | are not allowed.

Timestamp field

@timestamp

Select a timestamp field for use with the global time filter.

Show advanced settings

× Close

Save data view to Kibana

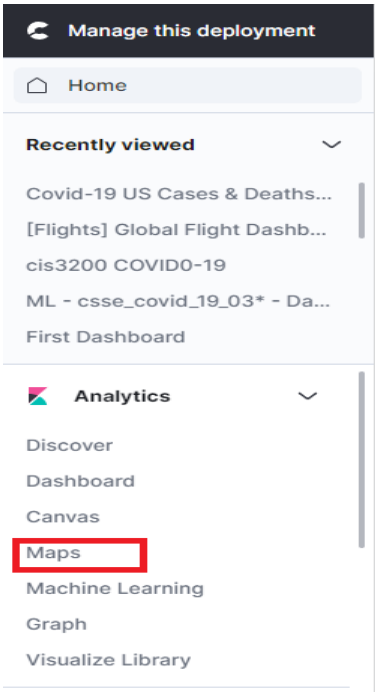
✓ Your index pattern matches 1 source.

covid_us_county	Index

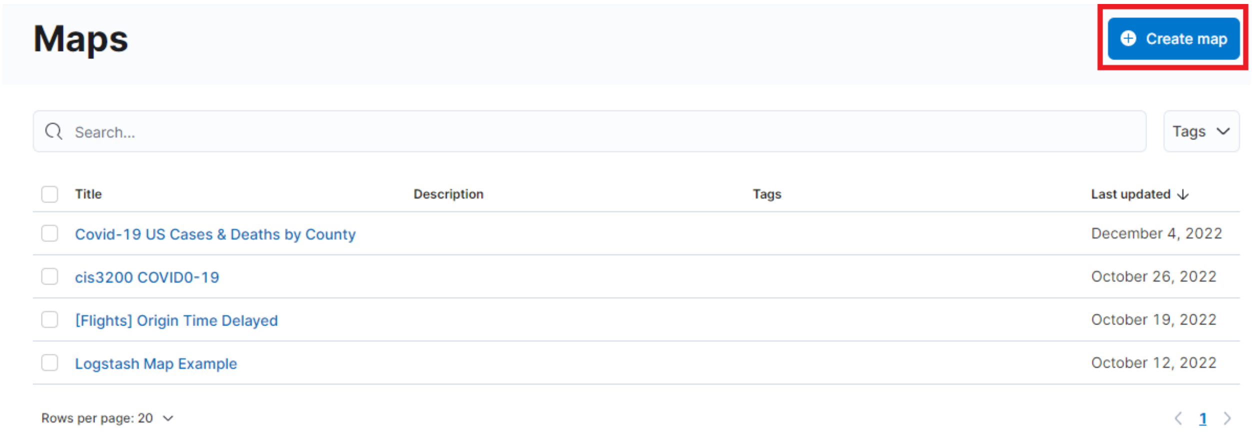
Rows per page: 10

Creating a Map on Elasticsearch

1. Now that we have our data view created. We can do many things with it on ElasticSearch. Let's start by creating a map using our geolocation that we created in the earlier steps. To do so we will head over to "**Maps**" under **analytics**.

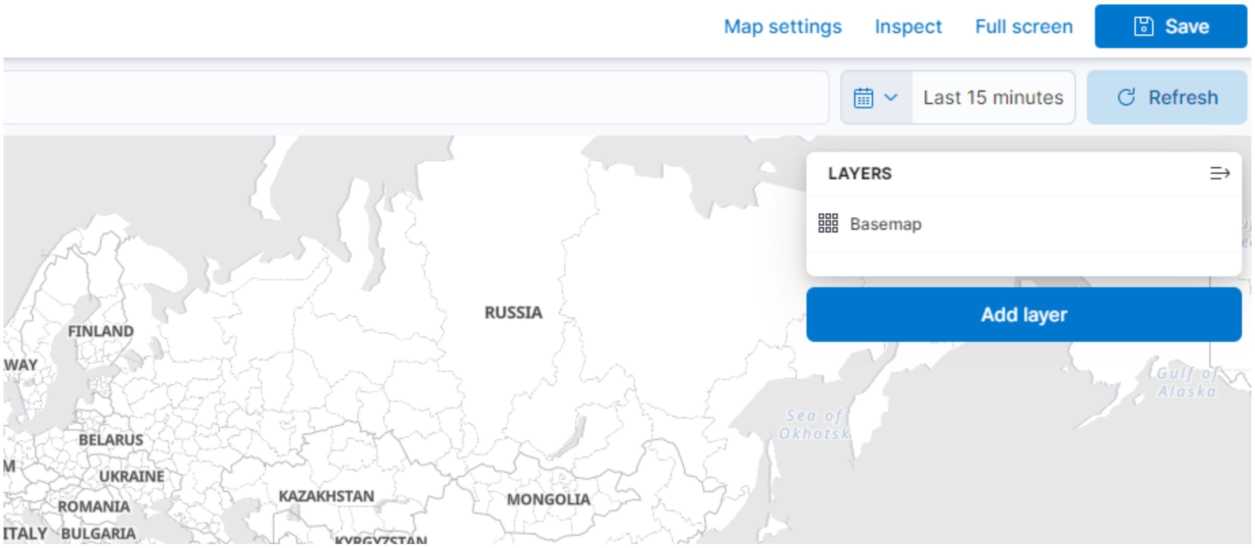


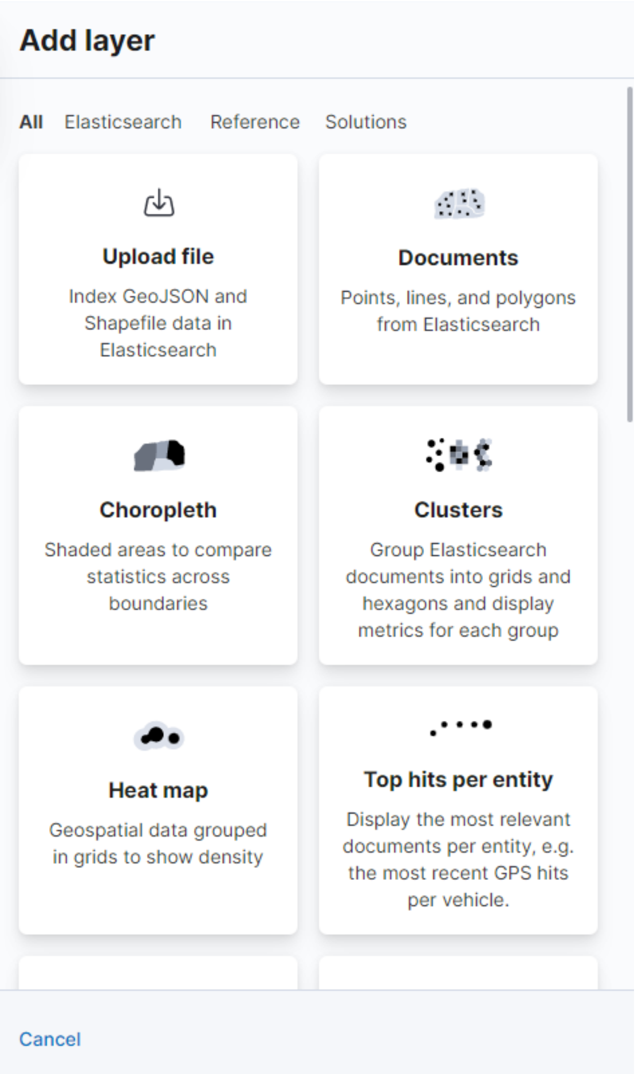
2. We are going to click on "**Create Map**".



Title	Description	Tags	Last updated ↓
Covid-19 US Cases & Deaths by County			December 4, 2022
cis3200 COVID0-19			October 26, 2022
[Flights] Origin Time Delayed			October 19, 2022
Logstash Map Example			October 12, 2022

3. Let’s click on “**Add Layer**” and let’s select “**Clusters**”




Add layer

All Elasticsearch Reference Solutions

Upload file
Index GeoJSON and Shapefile data in Elasticsearch

Documents
Points, lines, and polygons from Elasticsearch

Choropleth
Shaded areas to compare statistics across boundaries

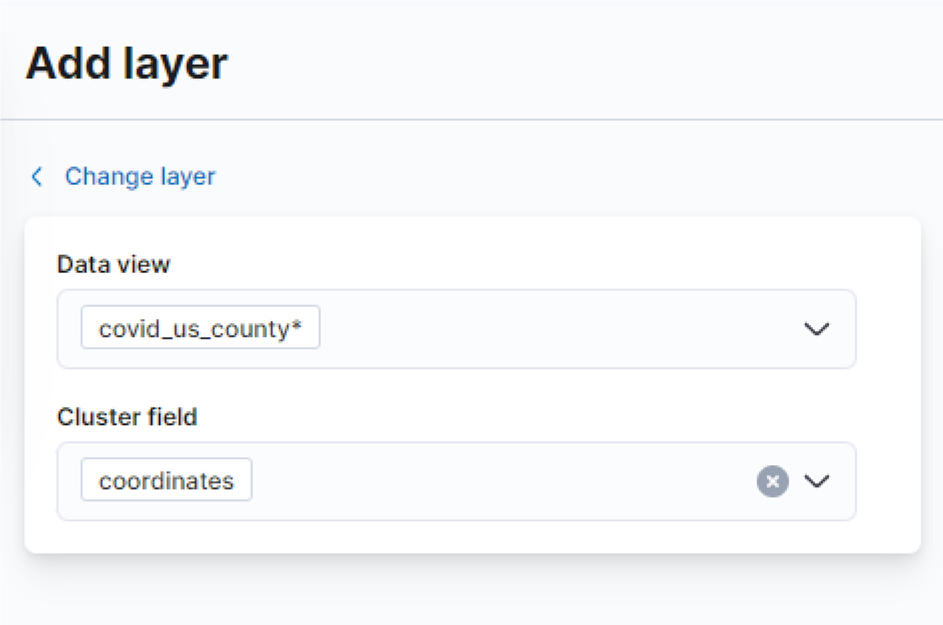
Clusters
Group Elasticsearch documents into grids and hexagons and display metrics for each group

Heat map
Geospatial data grouped in grids to show density

Top hits per entity
Display the most relevant documents per entity, e.g. the most recent GPS hits per vehicle.

Cancel

4. Let’s select the data set we uploaded. Let’s also make sure we select the desired time range. For our dataset it will be **Jan 22,2021** to **Dec 3, 2022**. Time shouldn’t matter.



Add layer

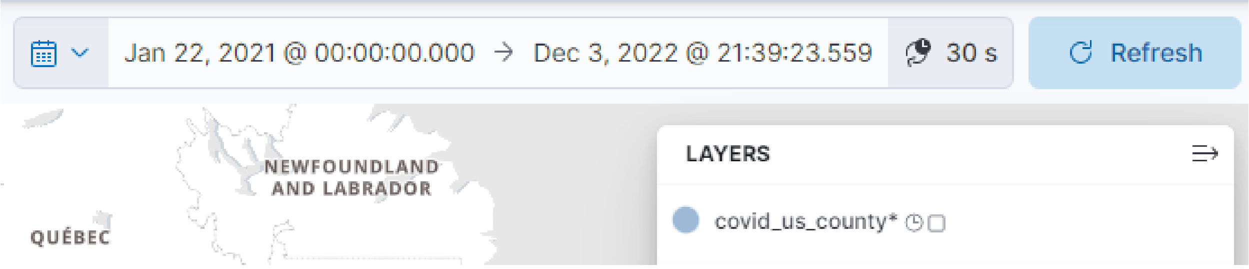
‹ Change layer

Data view

covid_us_county*

Cluster field

coordinates



Jan 22, 2021 @ 00:00:00.000 → Dec 3, 2022 @ 21:39:23.559

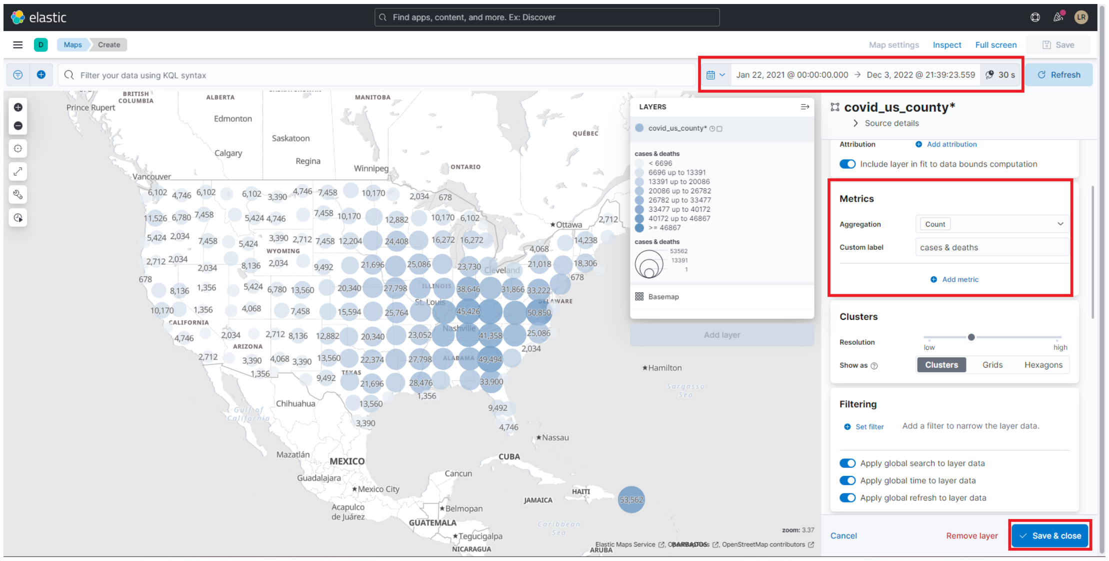
30 s

Refresh

LAYERS

● covid_us_county*

5. Once we have selected the time range and selected the coordinates field with a custom label name, we are greeted with our map. We can click save so that we can use it in the future for any possible need.



Jan 22, 2021 @ 00:00:00.000 → Dec 3, 2022 @ 21:39:23.559 30 s

Include layer in fit to data bounds computation

Aggregation: Count

Custom label: cases & deaths

Show as: Clusters

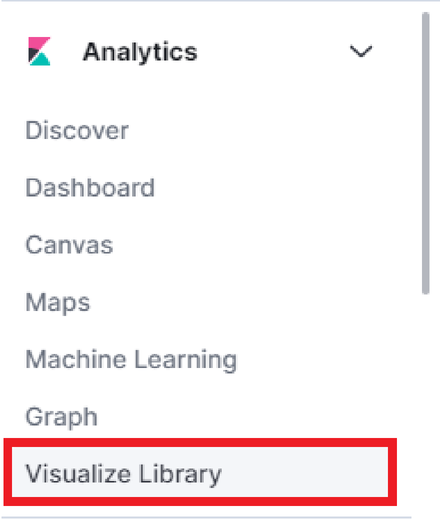
Apply global search to layer data

Apply global time to layer data

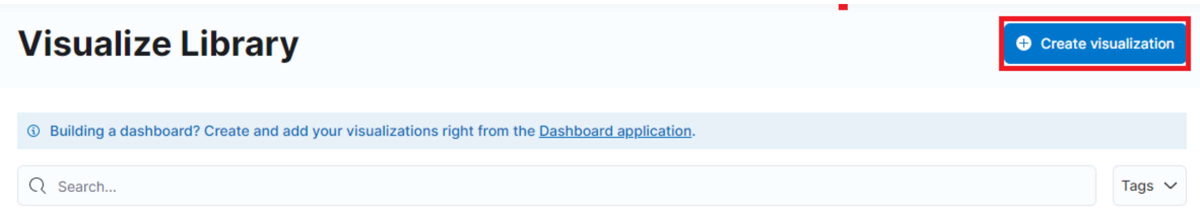
Apply global refresh to layer data

Creating a Pie Chart

1. At this step, we already have our data set uploaded and we already created our data set. If not, please refer to the beginning of this document.
2. On the left hand side, let's open up the sliding elastic menu and let's click on "Visualize Library" under "Analytics".

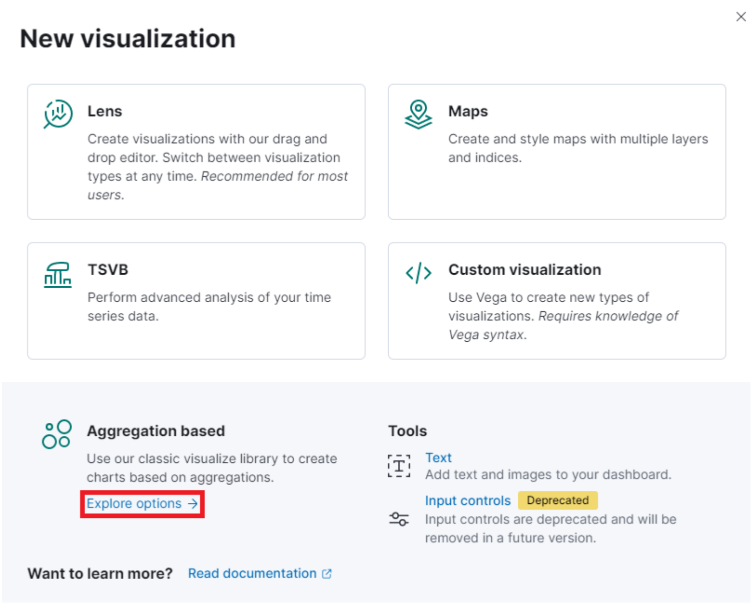


3. You will see the following screen, it should be blank if this is your first visualization. We'll simply click on the blue button that says "Create Visualization".



Building a dashboard? Create and add your visualizations right from the Dashboard application.

4. The page will prompt us with several options that elastic provides for our data, let's click on "Explore Options" under the "Aggregation based" section.



New visualization

Lens
Create visualizations with our drag and drop editor. Switch between visualization types at any time. *Recommended for most users.*

Maps
Create and style maps with multiple layers and indices.

TSVB
Perform advanced analysis of your time series data.

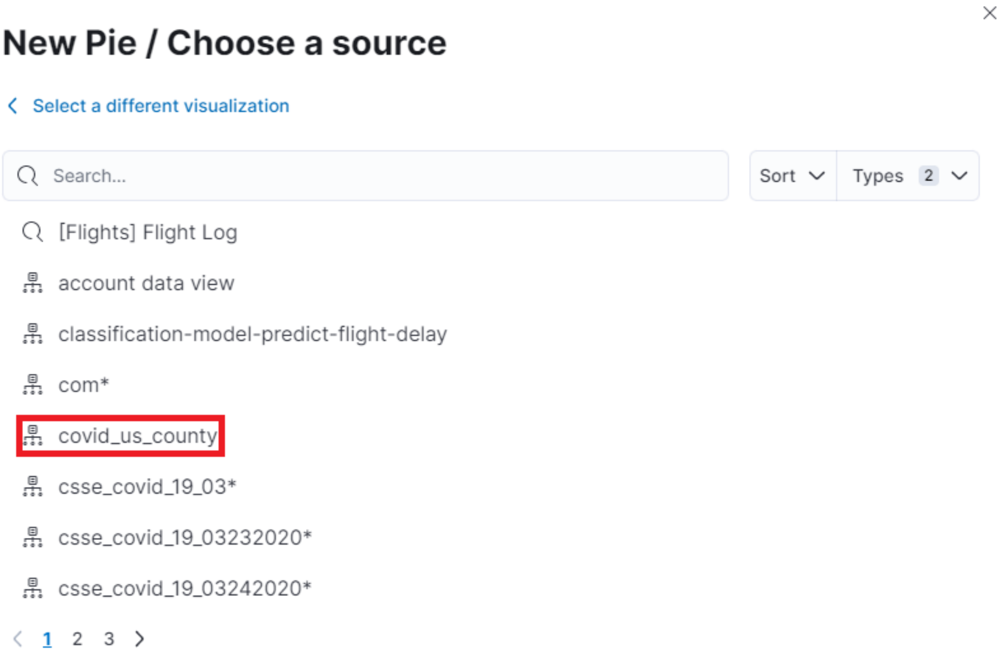
Custom visualization
Use Vega to create new types of visualizations. *Requires knowledge of Vega syntax.*

Aggregation based
Use our classic visualize library to create charts based on aggregations.
Explore options →

Tools
Text
Add text and images to your dashboard.
Input controls Deprecated
Input controls are deprecated and will be removed in a future version.

Want to learn more? Read documentation

5. Let’s select our desired data set.

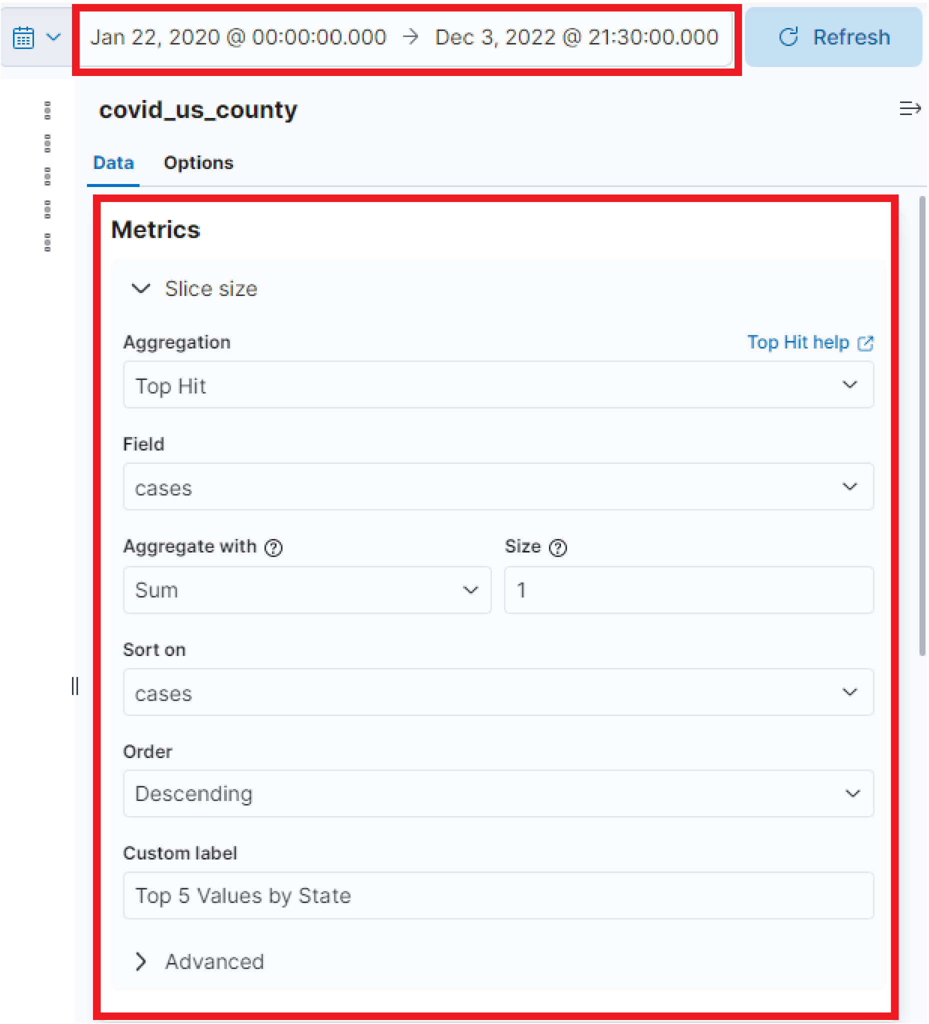


New Pie / Choose a source

‹ Select a different visualization

[Flights] Flight Log
account data view
classification-model-predict-flight-delay
com*
covid_us_county
csse_covid_19_03*
csse_covid_19_03232020*
csse_covid_19_03242020*

6. The following step requires us to select the date range from our data set. We must also use the following options. For aggregation we want “**Top Hit**”. For the field we will be using “**cases**” since that’s what we’re looking for in this situation. Under “**Aggregate with**” we went with **Sum**, but Elastic provides other options such as “Average”, “Max”, or “Min”, it depends on what it is you are trying to present with your visualization.



Jan 22, 2020 @ 00:00:00.000 → Dec 3, 2022 @ 21:30:00.000

covid_us_county

Metrics

Slice size

Aggregation: Top Hit

Field: cases

Aggregate with: Sum

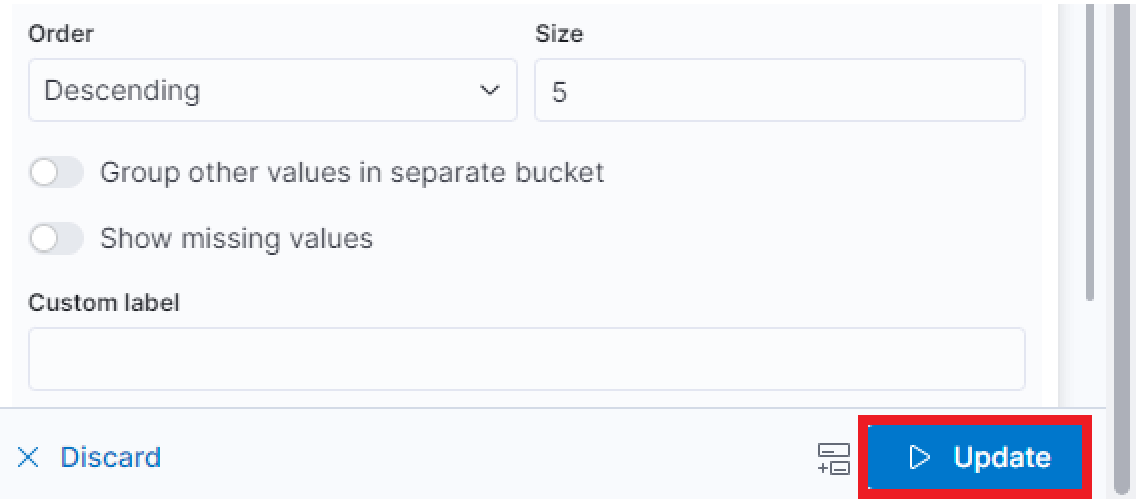
Size: 1

Sort on: cases

Order: Descending

Custom label: Top 5 Values by State

7. Now that we have customized all of our desired fields, let’s click on update so that Elastic can apply our changes.



Order: Descending

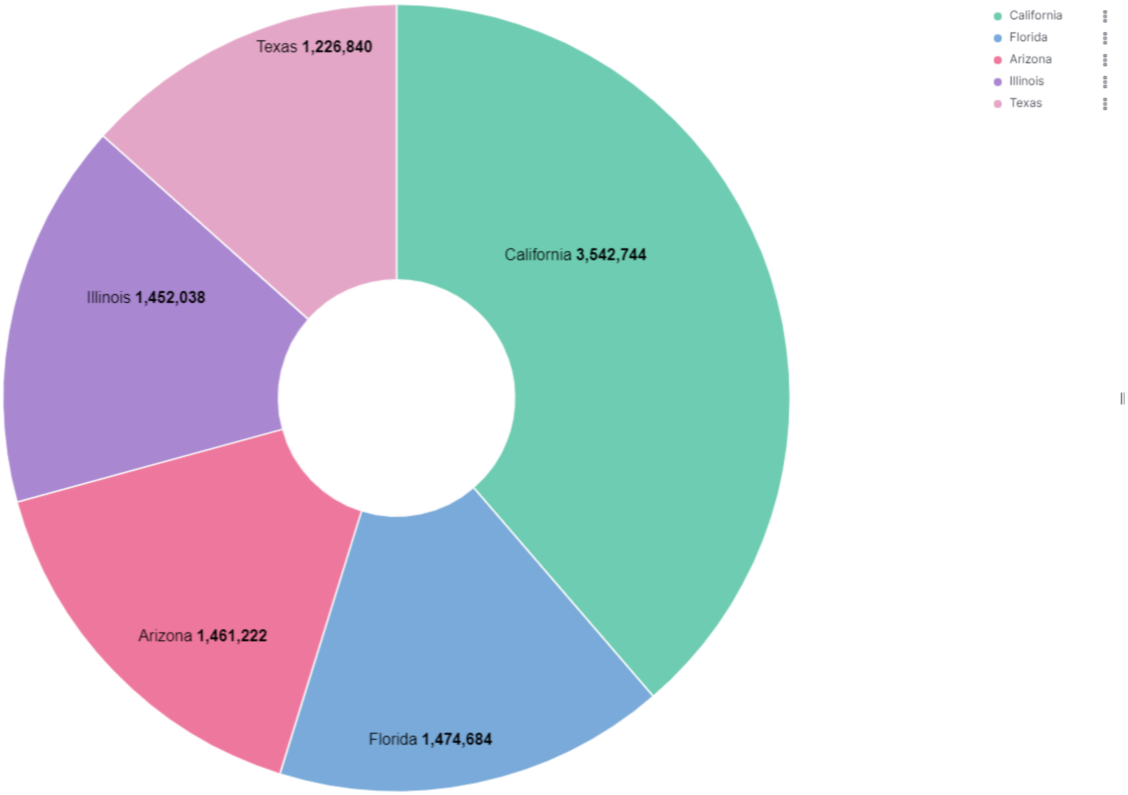
Size: 5

○ Group other values in separate bucket

○ Show missing values

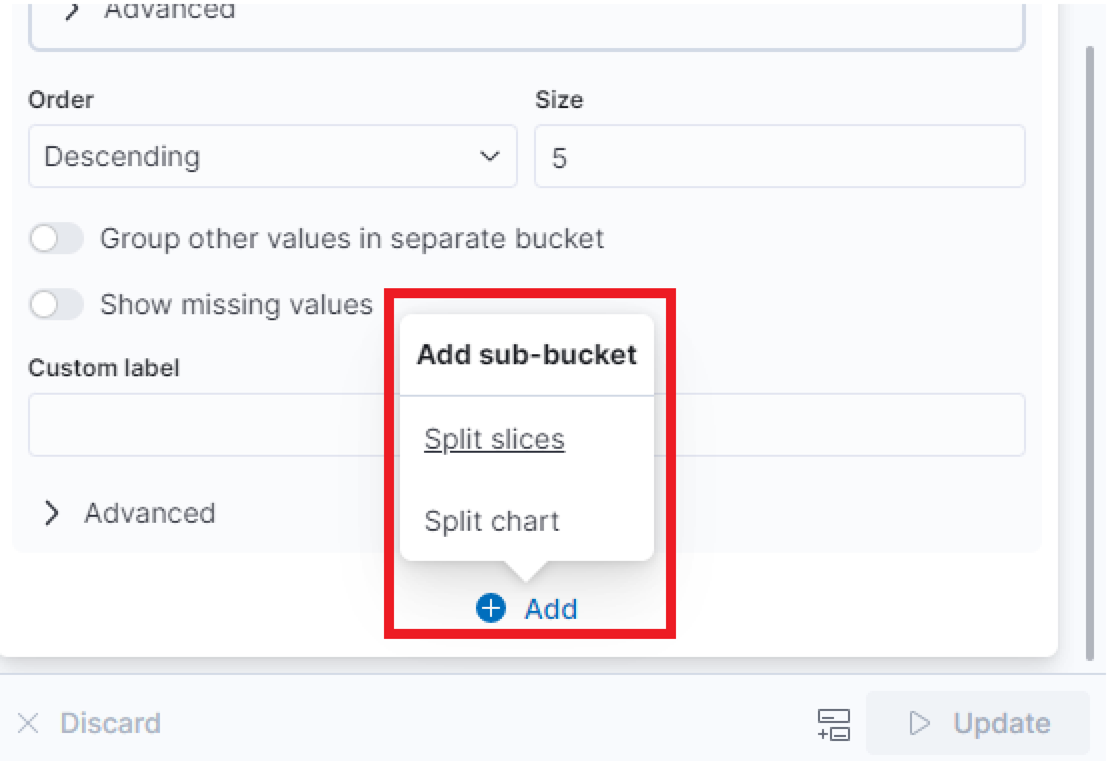
Custom label

8. As you can see, we should now be able to see the top 5 cases sorted by state.



9. In instances, you may be asked to include the timestamp that was created back when we were creating our data view. We can include this in our pie chart, with a couple of extra steps. Let’s start by adding an additional bucket.

10. Select “**Split slices**”



Order: Descending

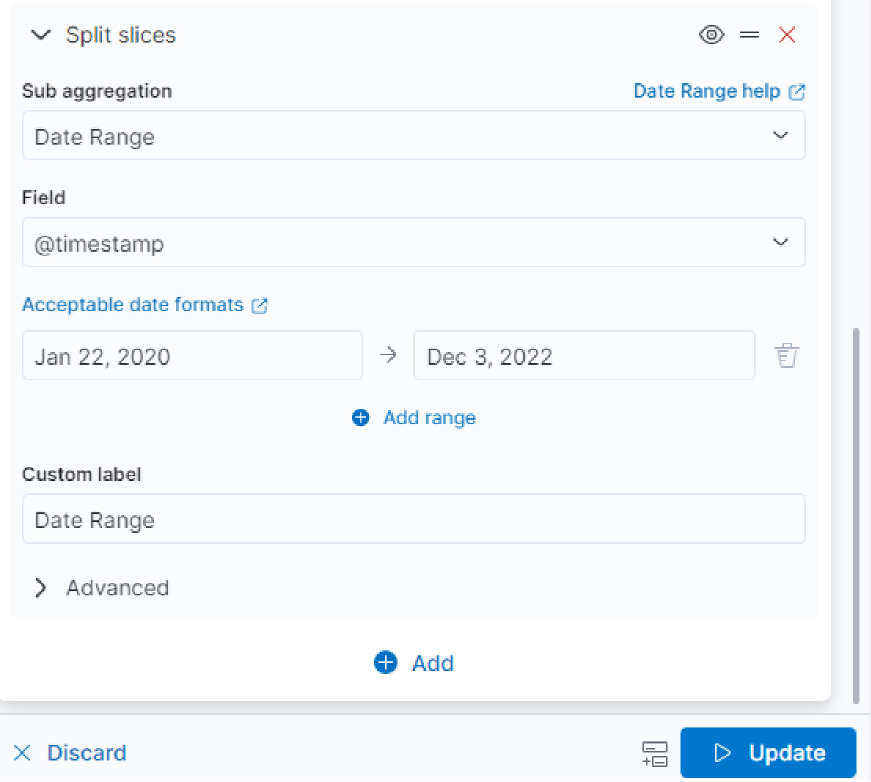
Size: 5

Group other values in separate bucket

Show missing values

Custom label

11. Let’s select “**Data Range**” for our Sub aggregation and choose “**timestamp**” for the field. Of course, we must also select the actual dates from our data set.



Split slices

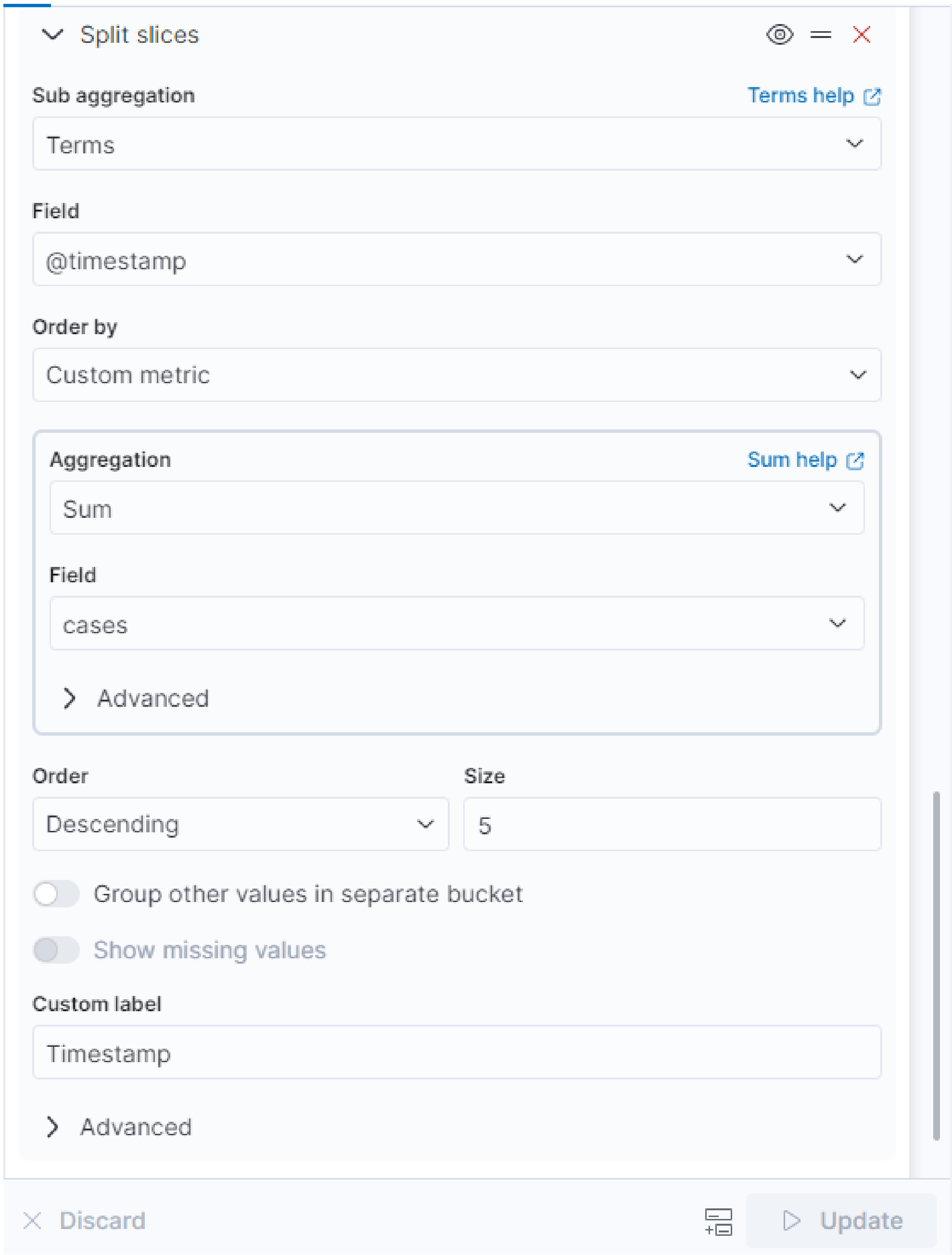
Sub aggregation: Date Range

Field: @timestamp

Jan 22, 2020 → Dec 3, 2022

Custom label: Date Range

12. We will click through advance and let it know we want 5 cases and for it to be in Descending order.



Sub aggregation: Terms

Field: @timestamp

Order by: Custom metric

Aggregation: Sum

Field: cases

Order: Descending

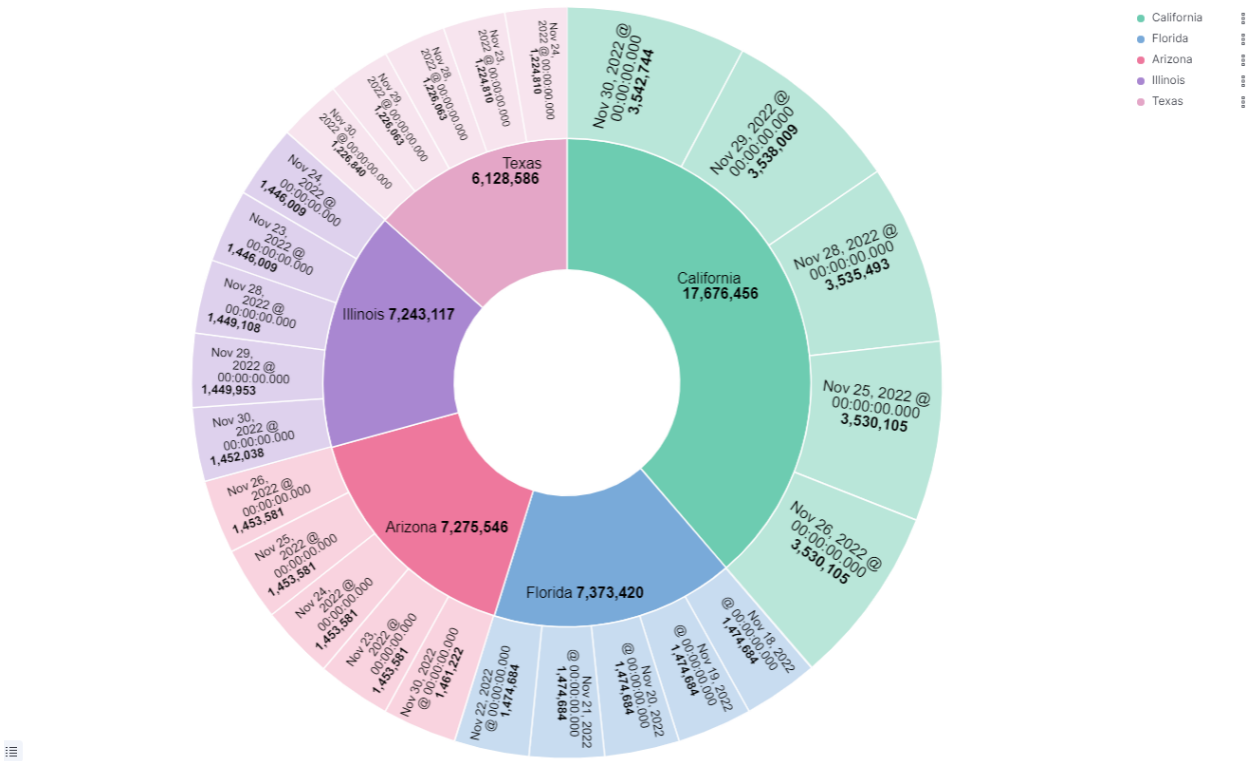
Size: 5

○ Group other values in separate bucket

○ Show missing values

Custom label: Timestamp

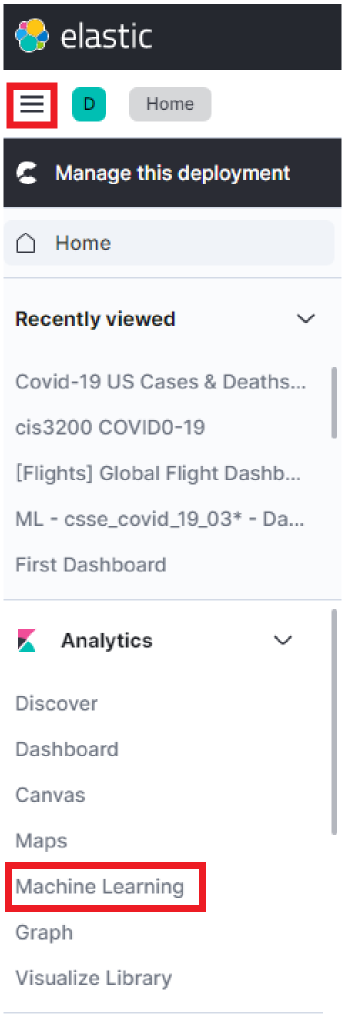
13. Here are the results, a new pie surrounds our previous pie, this time, it provides the date and shows the number of cases updated on that date.



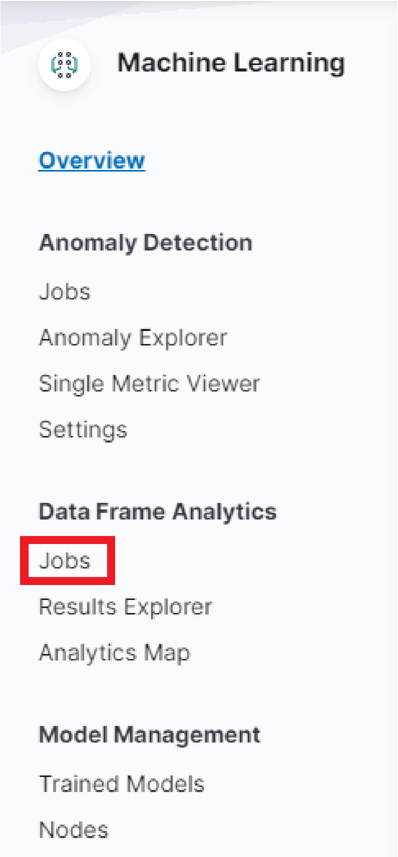
Creating a Regression Job in Elastic's Machine Learning

Another feature that Elastic provides is several analytics that use machine learning. We can identify anomalies on data. We can also make predictions based on the data we feed it. That's what we will be doing in this situation. Our data is based on COVID-19 cases based on county, so let's use Elastic's Machine Learning to predict future cases, per county.

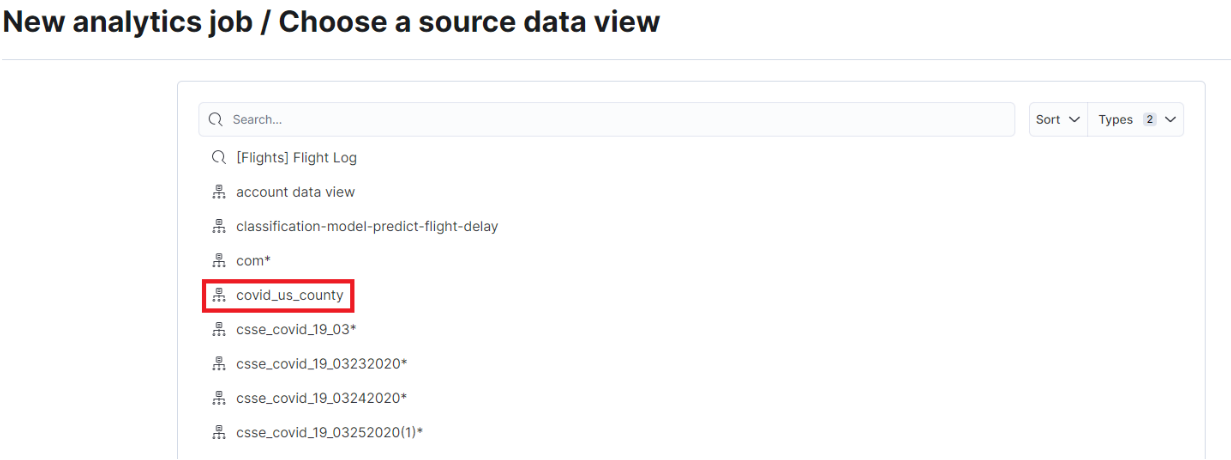
1. First, let's click open the sliding menu on Elastic, scroll down to "**Analytics**" and select "**Machine Learning**".



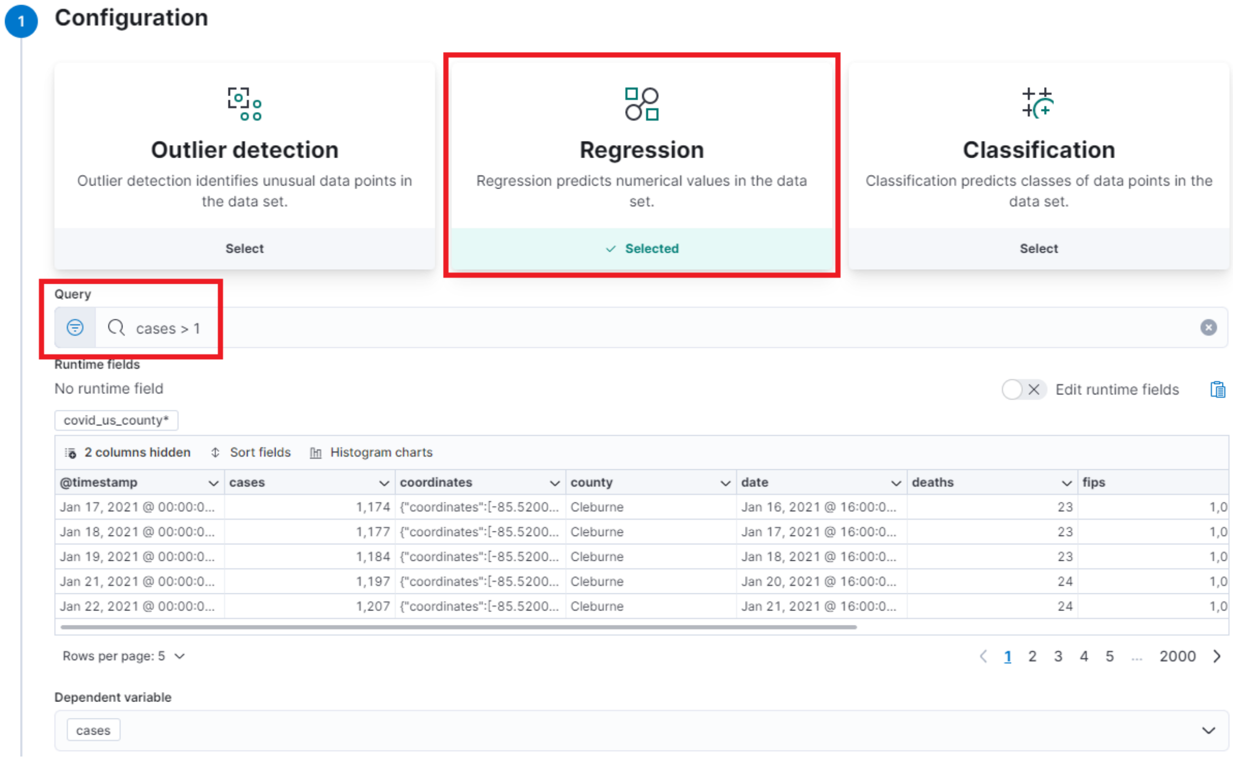
2. We are first greeted with the Overview page. This is where we will be able to see all the jobs we create in the future. For now, on the left hand side, under **"Data Frame Analytics**", lets click on "**Jobs**".



3. The page shows us all of our Data Frame Analytics jobs, we're going to click on the blue button that says "**Create Job**".
4. On this page, we can see all of our data views. Let's choose the data view that we created for this project. We titled it "**covid_us_county**".


New analytics job / Choose a source data view

5. After selecting the data view, Elastic wants to know what we would like to do with the data. We're gonna be predicting numerical values in the data set, which is exactly what “Regression” does, so let’s choose that option. Since we’re predicting cases, let’s make sure that for the query section we choose **“cases > 1**”. We also want to set our “**Dependent Variable**” to be “**cases**”.



1 Configuration

Outlier detection

Outlier detection identifies unusual data points in the data set.

Select

Regression

Regression predicts numerical values in the data set.

✓ Selected

Classification

Classification predicts classes of data points in the data set.

Select

Query

cases > 1

Runtime fields

No runtime field

Edit runtime fields

covid_us_county*

2 columns hidden · Sort fields · Histogram charts

@timestamp	cases	coordinates	county	date	deaths	fips
Jan 17, 2021 @ 00:00:0...	1,174	{"coordinates":[-85.5200...	Cleburne	Jan 16, 2021 @ 16:00:0...	23	1,0
Jan 18, 2021 @ 00:00:0...	1,177	{"coordinates":[-85.5200...	Cleburne	Jan 17, 2021 @ 16:00:0...	23	1,0
Jan 19, 2021 @ 00:00:0...	1,184	{"coordinates":[-85.5200...	Cleburne	Jan 18, 2021 @ 16:00:0...	23	1,0
Jan 21, 2021 @ 00:00:0...	1,197	{"coordinates":[-85.5200...	Cleburne	Jan 20, 2021 @ 16:00:0...	24	1,0
Jan 22, 2021 @ 00:00:0...	1,207	{"coordinates":[-85.5200...	Cleburne	Jan 21, 2021 @ 16:00:0...	24	1,0

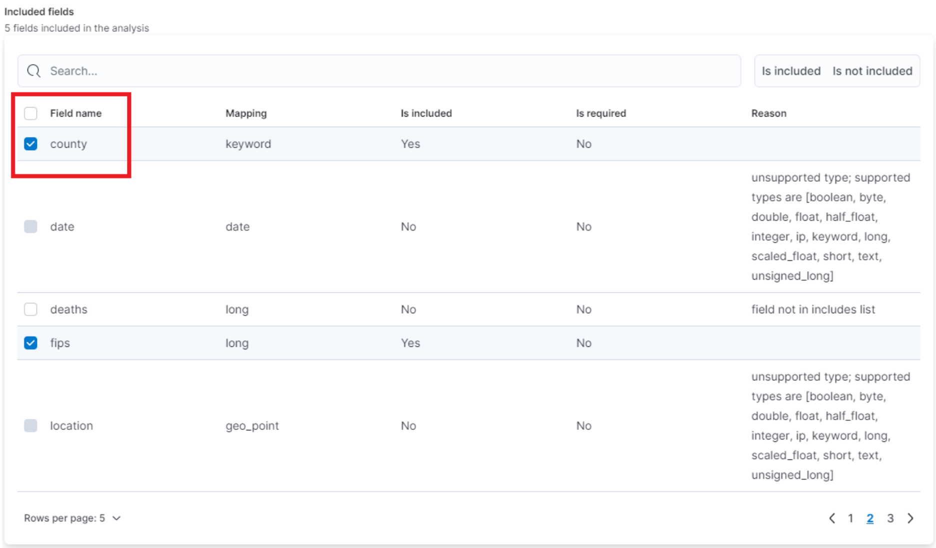
Rows per page: 5

< 1 2 3 4 5 ... 2000 >

Dependent variable

cases

6. To start the process of machine learning, we’re gonna go through 4 more steps to set up the job. Please refer to the following images so that we can set up our job correctly.
    a. For our fields, since we are trying to predict per county, let’s make sure that the field is selected.

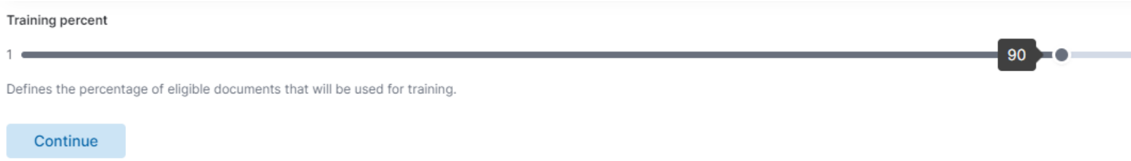


Included fields

5 fields included in the analysis

☐ Field name	Mapping	Is included	Is required	Reason
☑ county	keyword	Yes	No	
☐ date	date	No	No	unsupported type; supported types are [boolean, byte, double, float, half_float, integer, ip, keyword, long, scaled_float, short, text, unsigned_long]
☐ deaths	long	No	No	field not in includes list
☑ fips	long	Yes	No	
☐ location	geo_point	No	No	unsupported type; supported types are [boolean, byte, double, float, half_float, integer, ip, keyword, long, scaled_float, short, text, unsigned_long]

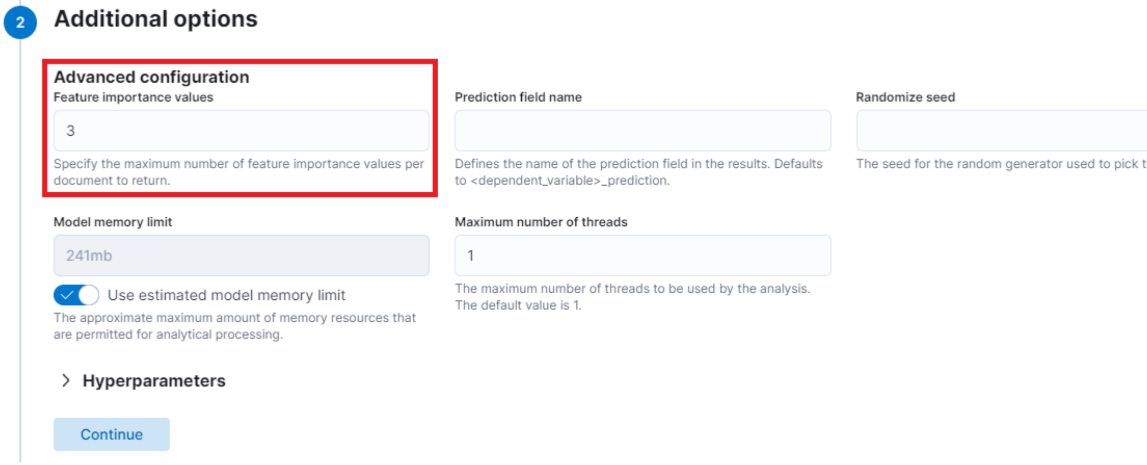
b. For our training percent, we want our prediction to be as accurate as possible, so let's choose 90 for the Training percent. Keep in mind that the higher this percent is, the LONGER it will take to show the results.



Training percent

Defines the percentage of eligible documents that will be used for training.

c. Let's set 3 for our feature importance value and let's keep the recommended memory limit.



2 Additional options

Advanced configuration

Feature importance values: 3

Specify the maximum number of feature importance values per document to return.

Prediction field name

Defines the name of the prediction field in the results. Defaults to <dependent_variable>_prediction.

Randomize seed

The seed for the random generator used to pick t

Model memory limit: 241mb

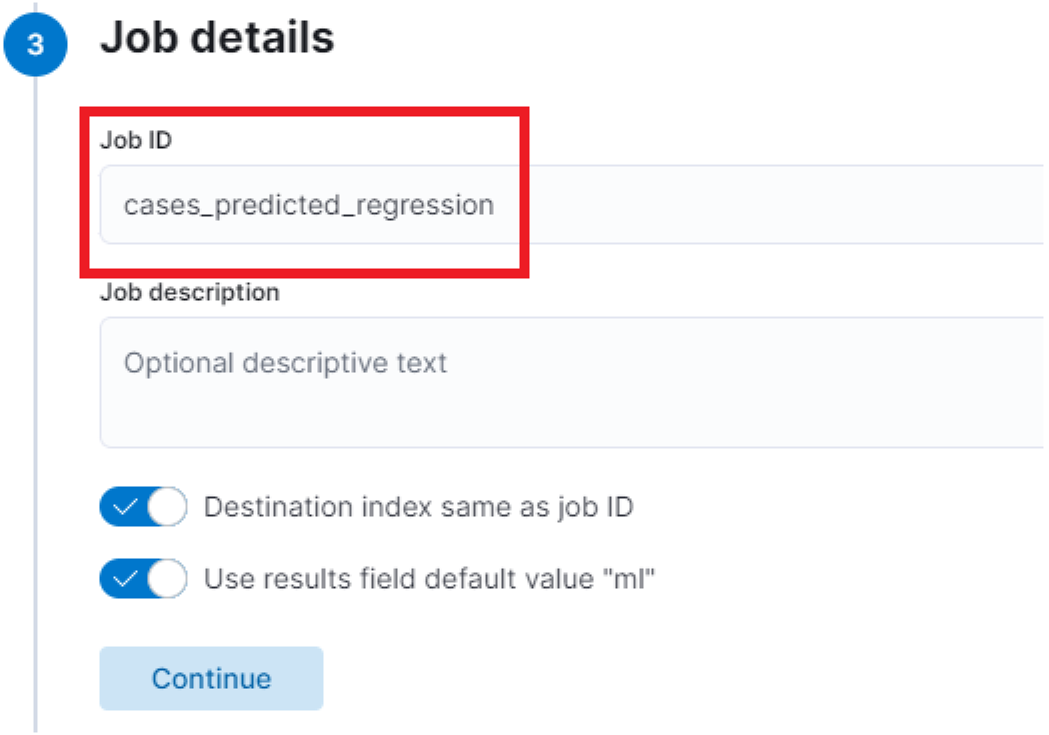
☑ Use estimated model memory limit

The approximate maximum amount of memory resources that are permitted for analytical processing.

Maximum number of threads: 1

The maximum number of threads to be used by the analysis. The default value is 1.

d. Let's name our job.



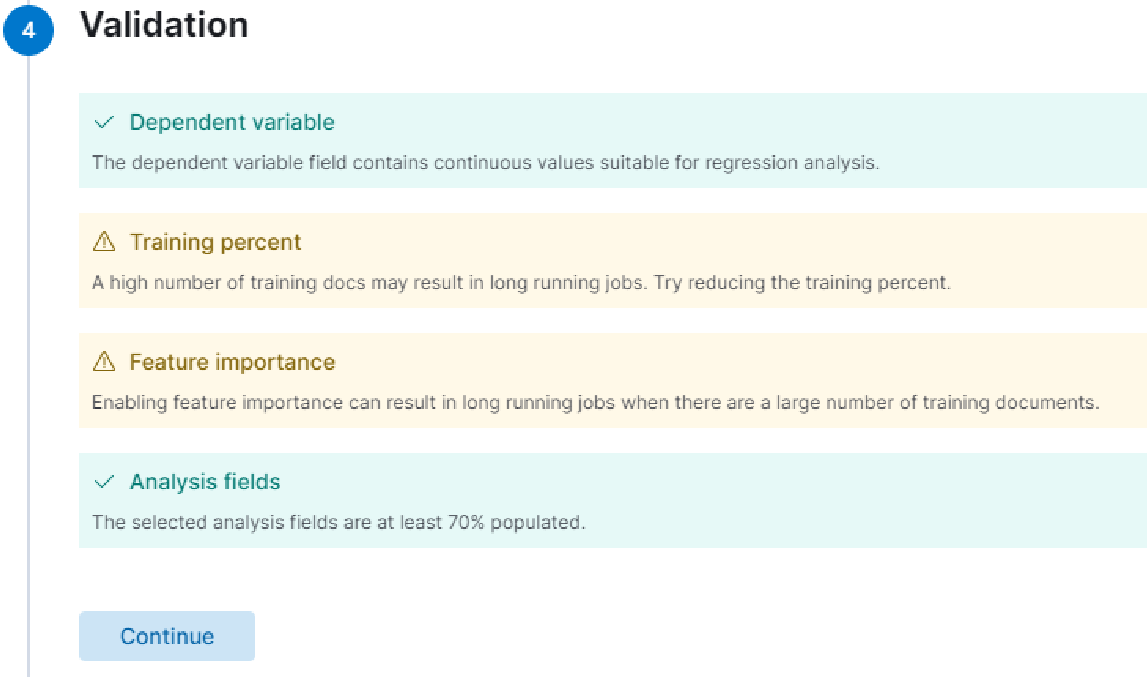
Job ID: cases_predicted_regression

Job description: Optional descriptive text

☑ Destination index same as job ID

☑ Use results field default value "ml"

e. The validation step is just telling us that we have our dependent variable and it's letting us know that by choosing to have a high training percent and a feature importance, the process may take longer. It's informational, we can click on continue.



✓ Dependent variable

The dependent variable field contains continuous values suitable for regression analysis.

⚠ Training percent

A high number of training docs may result in long running jobs. Try reducing the training percent.

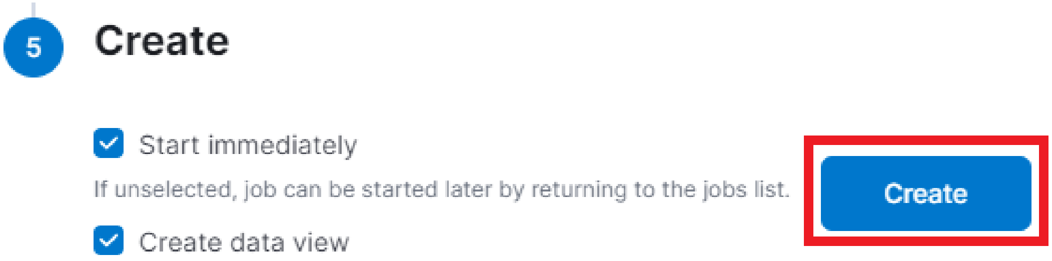
⚠ Feature importance

Enabling feature importance can result in long running jobs when there are a large number of training documents.

✓ Analysis fields

The selected analysis fields are at least 70% populated.

f. We're done setting our specifications. Let's click on the "**Create**" button and now the fun part begins: Waiting!



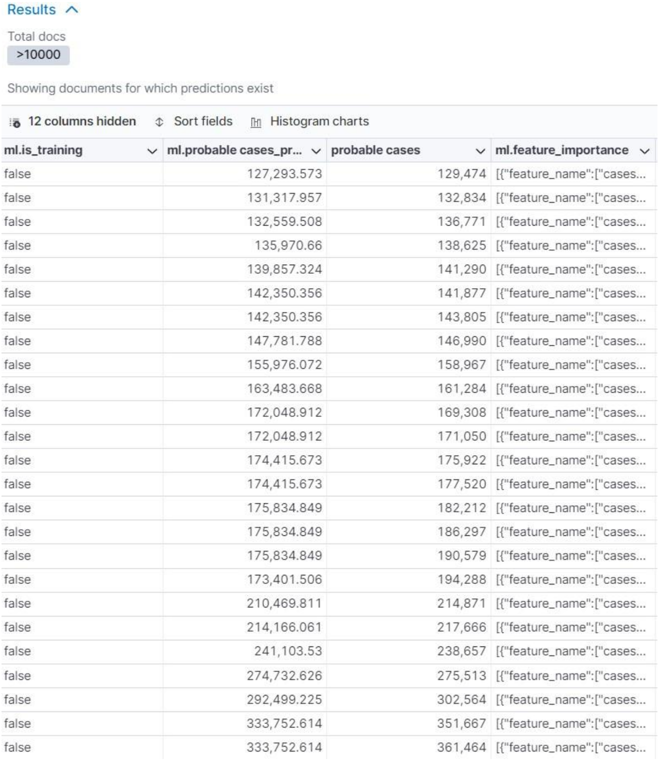
Create

☑ Start immediately

If unselected, job can be started later by returning to the jobs list.

☑ Create data view

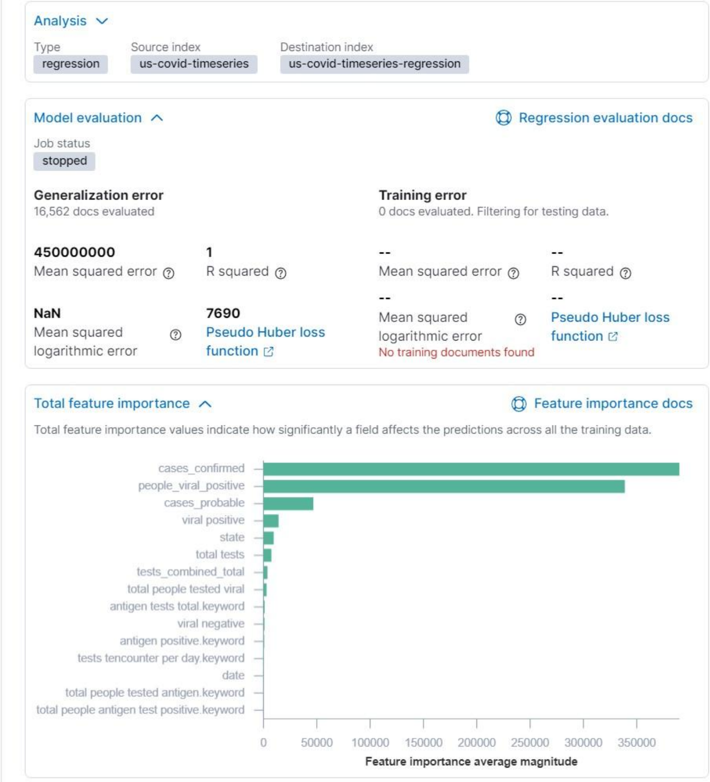
7. Now that progress is complete, let's take a look at our results. Your results should be different than ours when it comes to your Mean Square error (MSE). When you get a 1, it means that the prediction has perfect accuracy, which is highly unlikely. Due to time constraints, we were unable to rerun the machine learning process to get more accurate results.



Total docs
>10000

Showing documents for which predictions exist

ml.is_training	ml.probable cases_pr...	probable cases	ml.feature_importance
false	127,293.573	129,474	[{"feature_name":["cases...
false	131,317.957	132,834	[{"feature_name":["cases...
false	132,559.508	136,771	[{"feature_name":["cases...
false	135,970.66	138,625	[{"feature_name":["cases...
false	139,857.324	141,290	[{"feature_name":["cases...
false	142,350.356	141,877	[{"feature_name":["cases...
false	142,350.356	143,805	[{"feature_name":["cases...
false	147,781.788	146,990	[{"feature_name":["cases...
false	155,976.072	158,967	[{"feature_name":["cases...
false	163,483.668	161,284	[{"feature_name":["cases...
false	172,048.912	169,308	[{"feature_name":["cases...
false	172,048.912	171,050	[{"feature_name":["cases...
false	174,415.673	175,922	[{"feature_name":["cases...
false	174,415.673	177,520	[{"feature_name":["cases...
false	175,834.849	182,212	[{"feature_name":["cases...
false	175,834.849	186,297	[{"feature_name":["cases...
false	175,834.849	190,579	[{"feature_name":["cases...
false	173,401.506	194,288	[{"feature_name":["cases...
false	210,469.811	214,871	[{"feature_name":["cases...
false	214,166.061	217,666	[{"feature_name":["cases...
false	241,103.53	238,657	[{"feature_name":["cases...
false	274,732.626	275,513	[{"feature_name":["cases...
false	292,499.225	302,564	[{"feature_name":["cases...
false	333,752.614	351,667	[{"feature_name":["cases...
false	333,752.614	361,464	[{"feature_name":["cases...



Analysis

Type	Source index	Destination index
regression	us-covid-timeseries	us-covid-timeseries-regression

Model evaluation

Regression evaluation docs

Job status: stopped

Generalization error
16,562 docs evaluated

450000000	Mean squared error
1	R squared
NaN	Mean squared logarithmic error
7690	Pseudo Huber loss function

Training error
0 docs evaluated. Filtering for testing data.

--	Mean squared error
--	R squared
--	Mean squared logarithmic error
--	Pseudo Huber loss function

No training documents found

Total feature importance

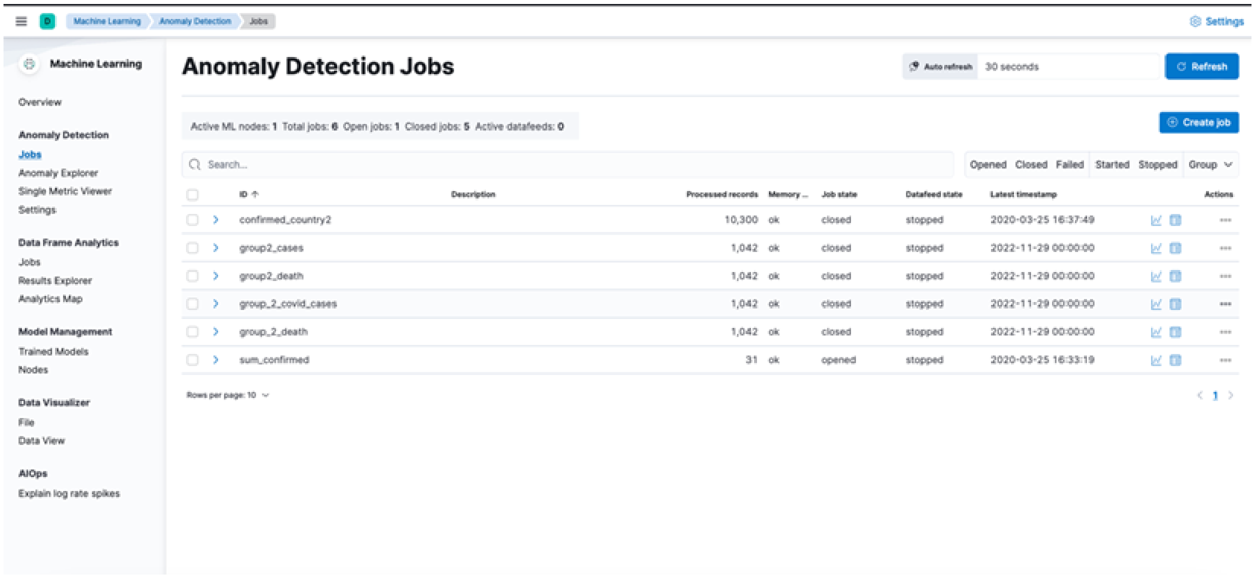
Feature importance docs

Total feature importance values indicate how significantly a field affects the predictions across all the training data.

cases_confirmed
people_viral_positive
cases_probable
viral positive
state
total tests
tests_combined_total
total people tested viral
antigen tests total.keyword
viral negative
antigen positive.keyword
tests tencounter per day.keyword
date
total people tested antigen.keyword
total people antigen test positive.keyword
0
50000
100000
150000
200000
250000
300000
350000
Feature importance average magnitude

Creating Forecast

1. Once the data has been uploaded, go to Analytics > Machine Learning > Anomaly Detection > Jobs



Anomaly Detection Jobs

Auto refresh: 30 seconds

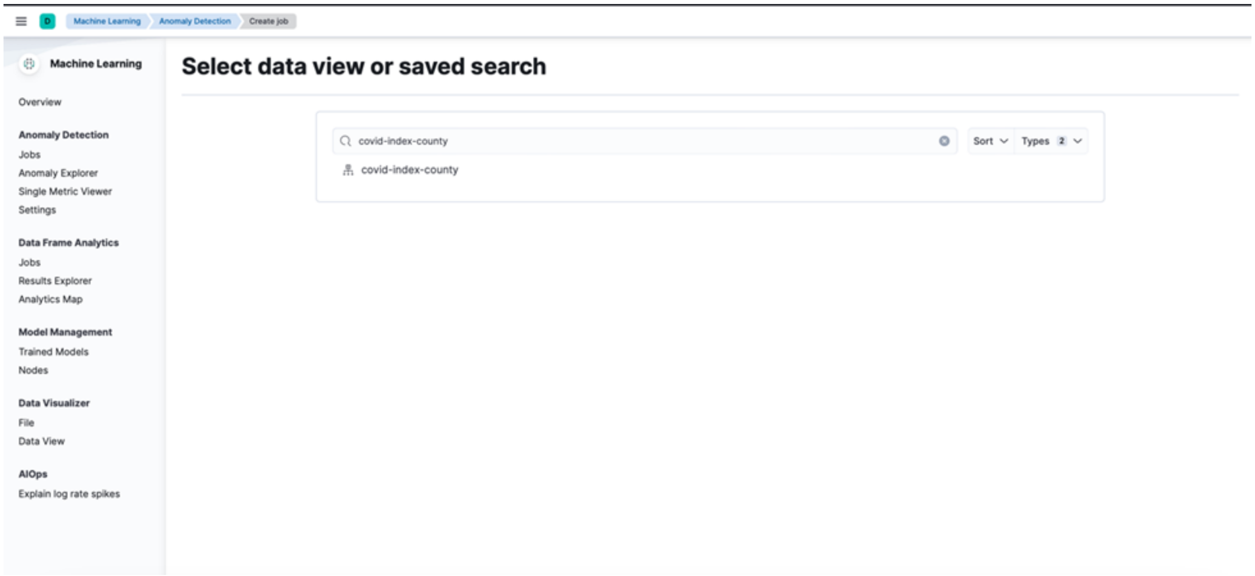
Active ML nodes: 1 Total jobs: 6 Open jobs: 1 Closed jobs: 5 Active datafeeds: 0

ID ↑	Description	Processed records	Memory ...	Job state	Datafeed state	Latest timestamp	Actions
confirmed_country2		10,300	ok	closed	stopped	2020-03-25 16:37:49	
group2_cases		1,042	ok	closed	stopped	2022-11-29 00:00:00	
group2_death		1,042	ok	closed	stopped	2022-11-29 00:00:00	
group_2_covid_cases		1,042	ok	closed	stopped	2022-11-29 00:00:00	
group_2_death		1,042	ok	closed	stopped	2022-11-29 00:00:00	
sum_confirmed		31	ok	opened	stopped	2020-03-25 16:33:19	

Rows per page: 10

2. Now, press the **create job** button

3. Select the data that we have uploaded on Kibana (covid-index-county)



Select data view or saved search

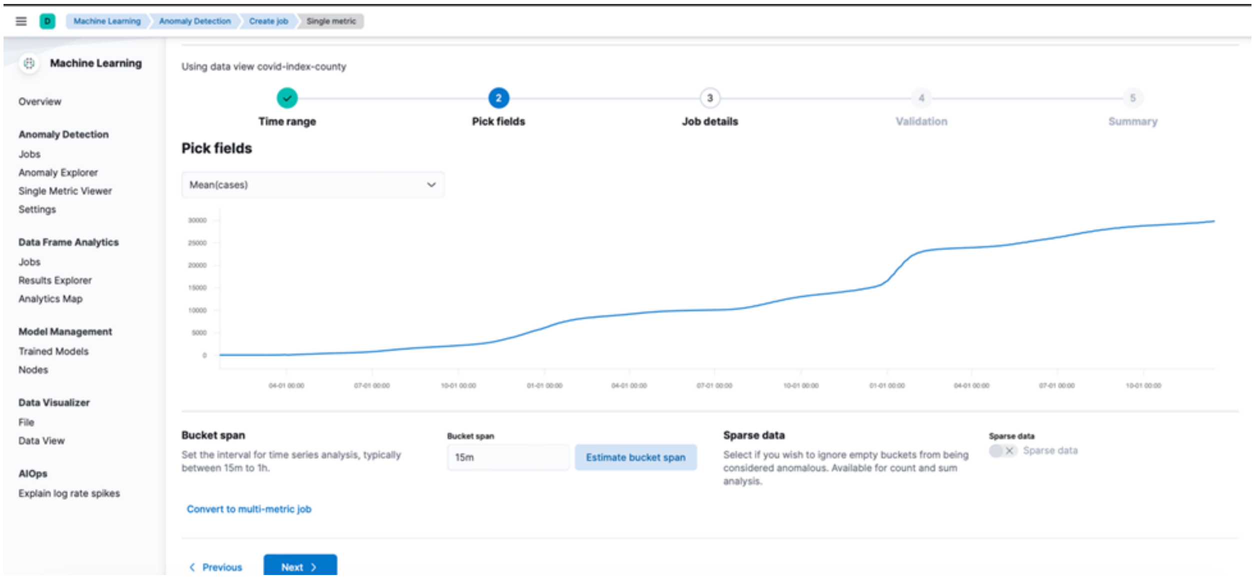
covid-index-county

covid-index-county

4. Once Selected, press on **single metrics**

5. On Create Job, Single Metrics, for the time range, press the '**Use full range data**' and press next

6. On the field, select **mean(case)** and press next



Using data view covid-index-county

Pick fields

Mean(cases)

Bucket span

Set the interval for time series analysis, typically between 15m to 1h.

Bucket span: 15m

Sparse data

Select if you wish to ignore empty buckets from being considered anomalous. Available for count and sum analysis.

Sparse data: Sparse data

7. Enter the job id as 'covid-19-cases-forecast'

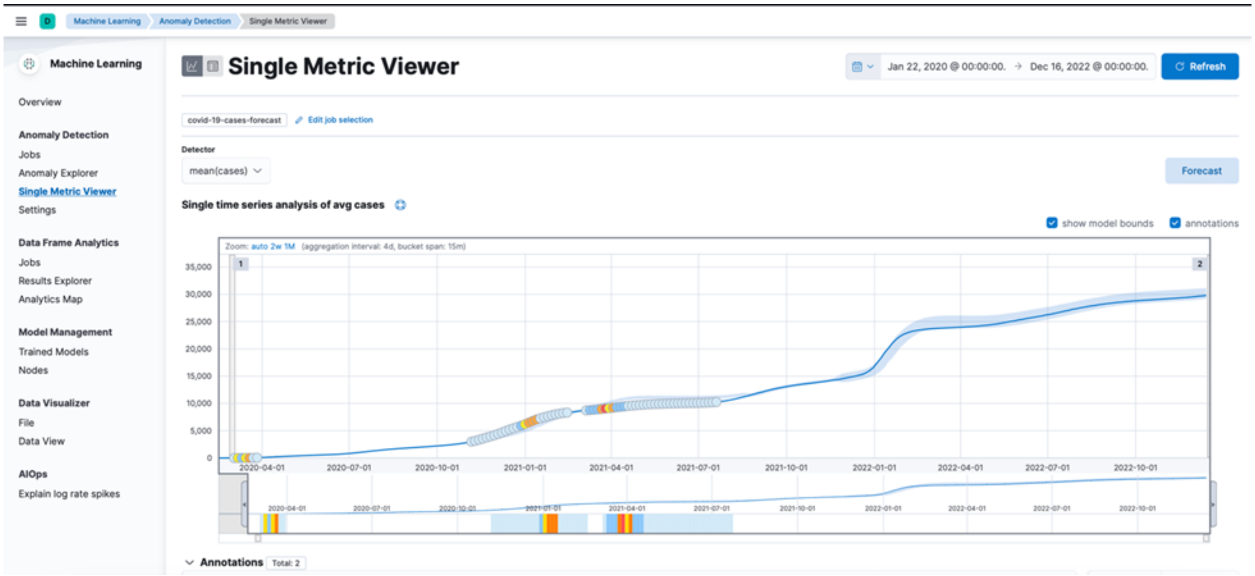
8. Press next repeatedly until you arrive on the summary page.

9. On the summary page, press the ‘create job’ button

10. Once the job has been successfully created, press the ‘view results button’

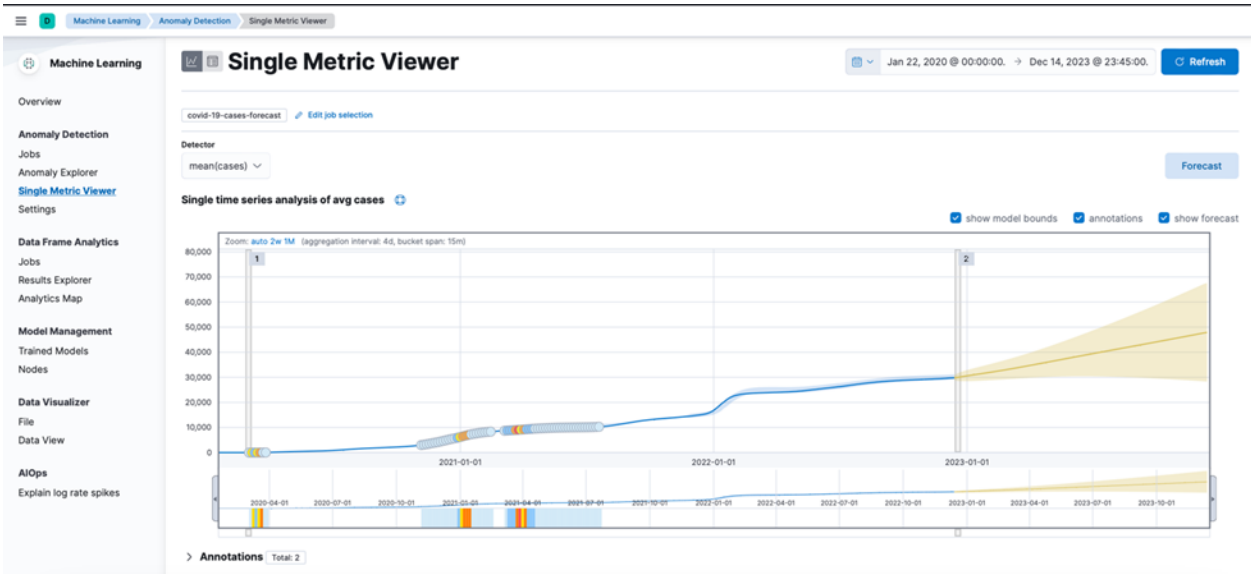
    a. On the results button, the data might not appear until the end because it is still loading. In this case, please wait a little and refresh the page until all the data appears.

11. Once the data is fully loaded press on the ‘forecast button’



12. To create a forecast for a year, enter **365d** and **press forecast**

13. One completed, **drag** the **Single time series** to the end to the entire forecast for **next year**



Jan 22, 2020 @ 00:00:00. → Dec 14, 2023 @ 23:45:00.

covid-19-cases-forecast

Detector
mean(cases)

Single time series analysis of avg cases

Zoom: auto 2w 1M (aggregation interval: 4d, bucket span: 15m)

14. Now you have a forecast of the average cases for a year.

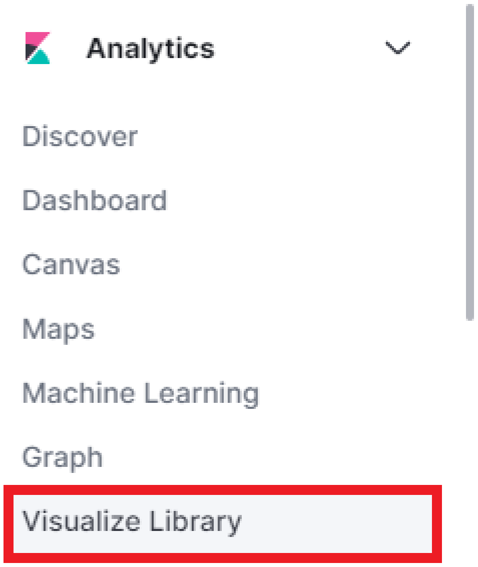
Future work, possibly, for your term project

You may increase the range of the forecast up to **3650 days** which is 10 years from now. Also you may change the average number of cases to average number of death when creating the job by changing the field from **mean(cases) to mean (death).**

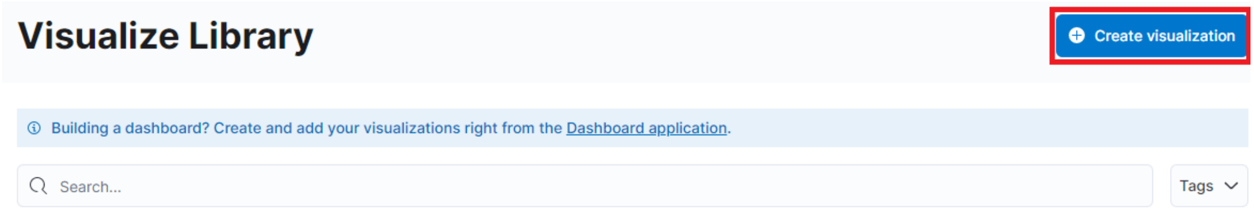
Creating a Horizontal Bar & Vertical Stacked Bar Graph

Elasticsearch provides tools to create quick graphs. We will be using the tool “Lens” to create these 2 bar graphs.

1. First we will be heading to the Visualize Library by clicking on it under the Analytics

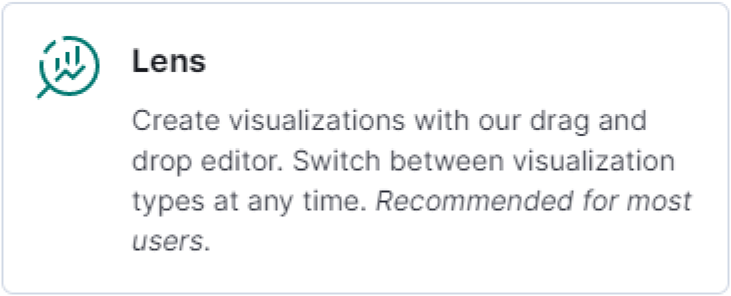


2. Let’s click on “**Create visualization**”.



Building a dashboard? Create and add your visualizations right from the Dashboard application.

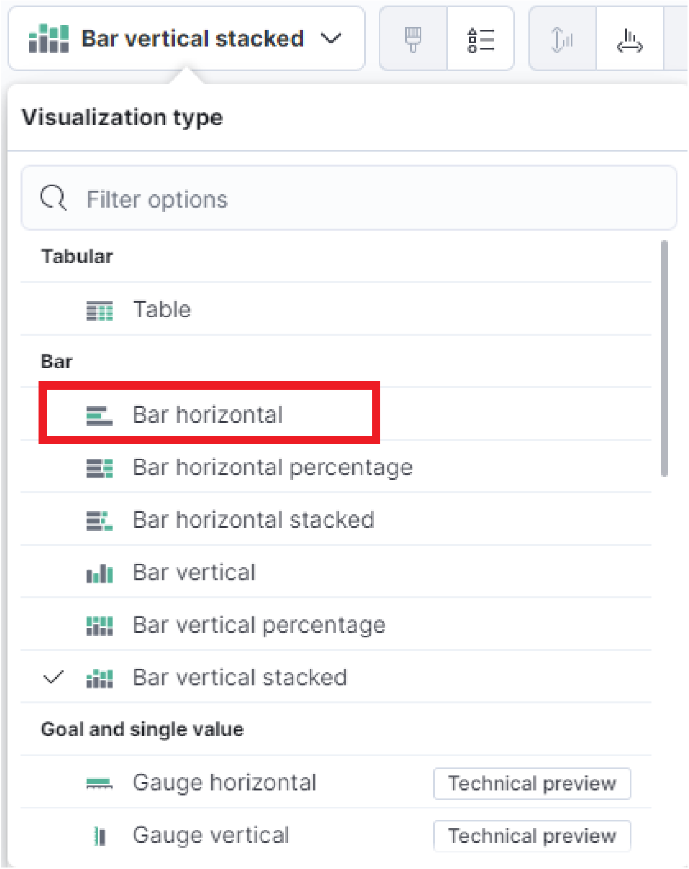
3. We will be prompted by several options to create a visualization, but we will go with Elastic’s easy, drag and drop, option: Lens.



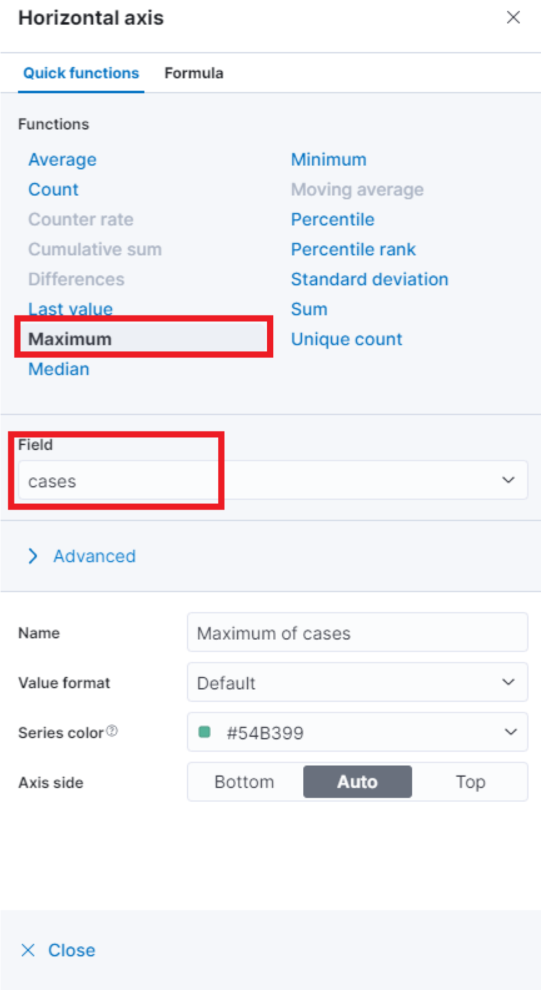
Lens

Create visualizations with our drag and drop editor. Switch between visualization types at any time. *Recommended for most users.*

4. Let’s make sure that we select “**Bar Horizontal**”, since that’s the one we are creating for now.



5. These are the options we are going with to see the Top 5 Counties with Confirmed cases in the U.S. Under functions, we are choosing “**Maximum**”. Under field, we will be choosing “**cases**”.



Field: cases

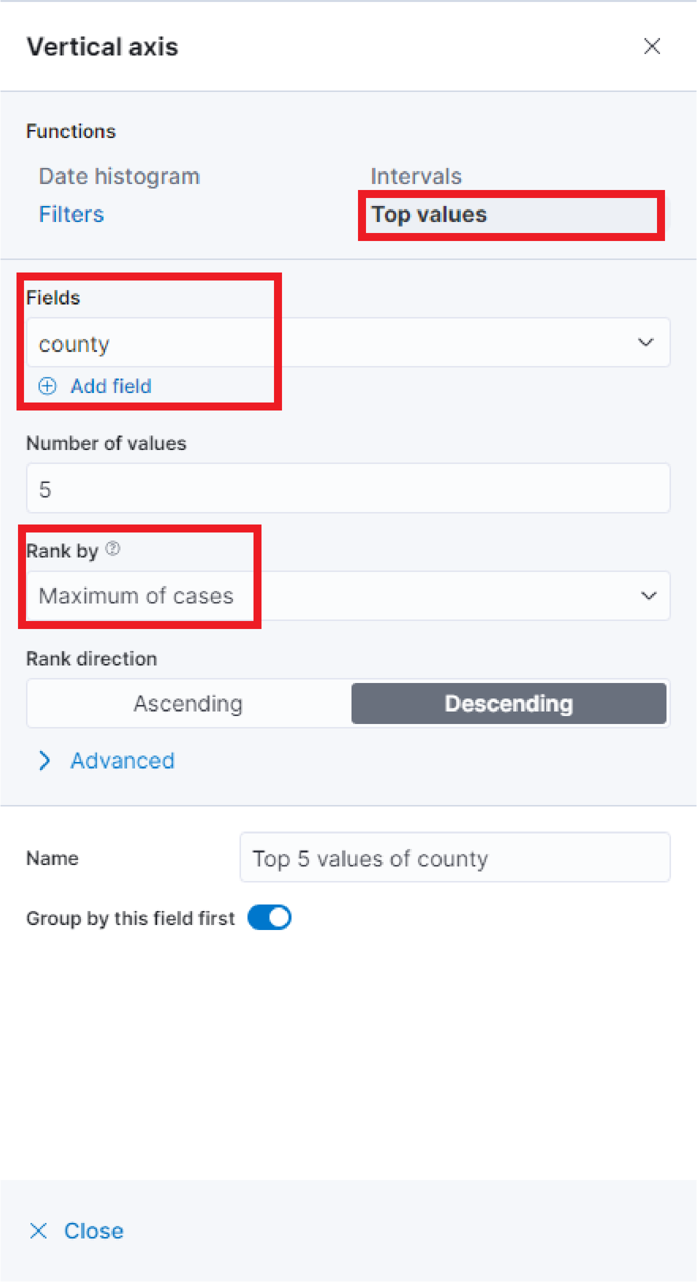
Name: Maximum of cases

Value format: Default

Series color: #54B399

Axis side: Auto

6. For the Vertical axis, we will be selecting "**Top Values**" for the Functions, "**County**" for fields, and Ranking by "**Maximum of cases**".



Vertical axis

Functions

Top values

Fields

county

Number of values

5

Rank by

Maximum of cases

Rank direction

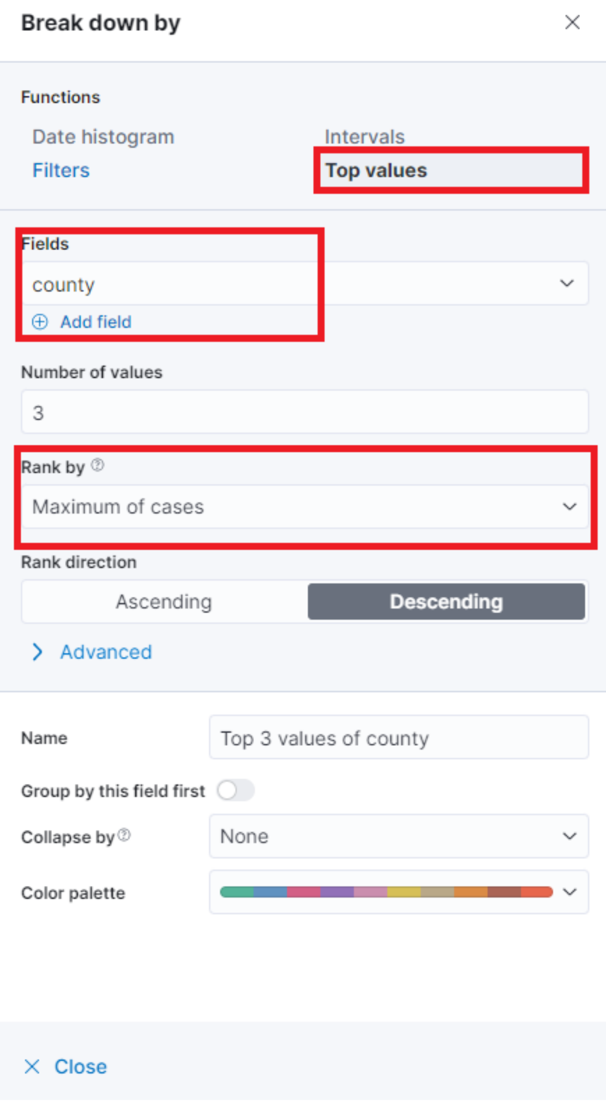
Descending

Name

Top 5 values of county

Group by this field first

7. We want to break everything down by “**Top Values**” with “**county**” under Fields and Rank by “**Maximum Cases**”.



Break down by

Functions

Date histogram

Intervals

Filters

Top values

Fields

county

⊕ Add field

Number of values

3

Rank by

Maximum of cases

Rank direction

Ascending | Descending

Advanced

Name: Top 3 values of county

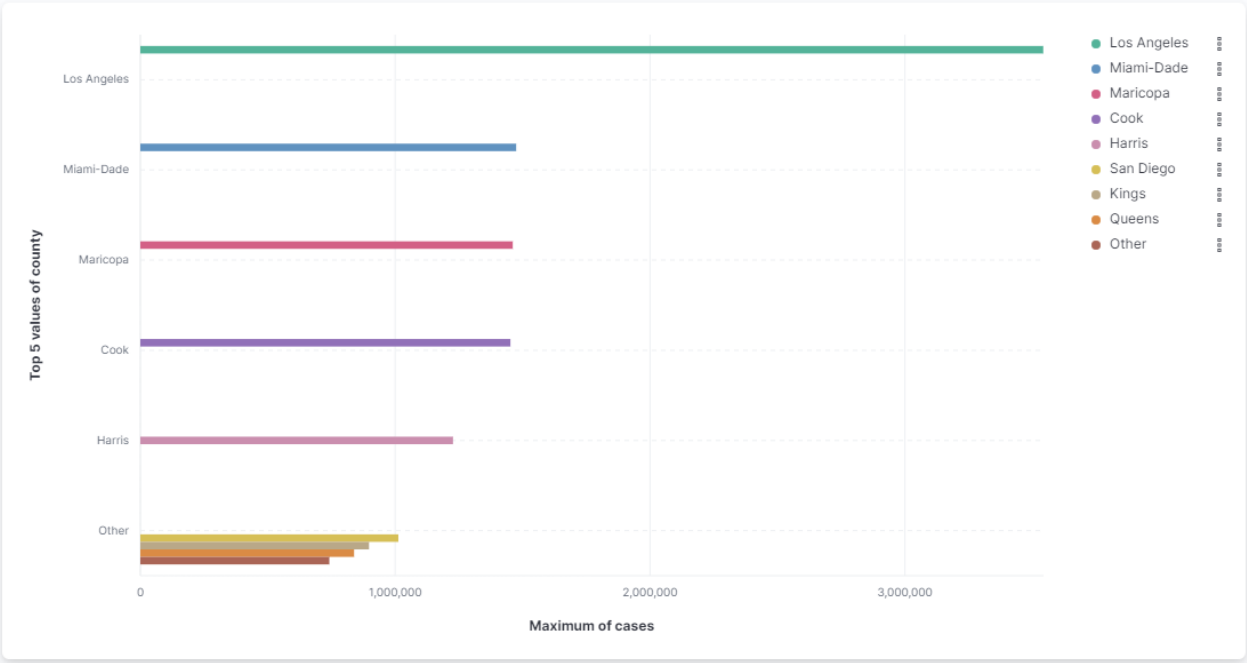
Group by this field first ○

Collapse by: None

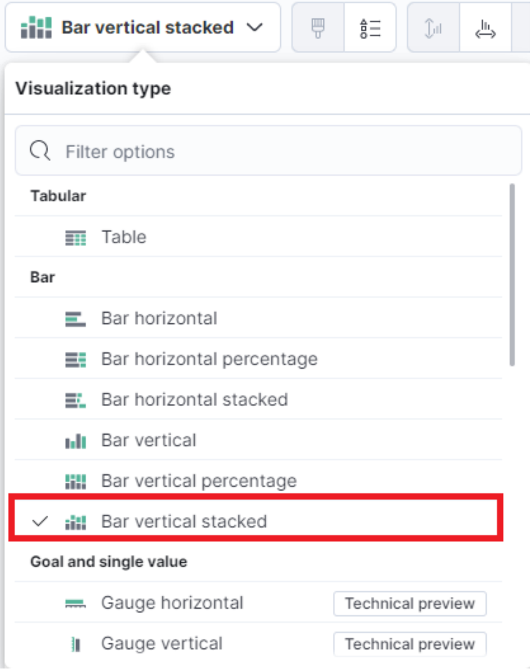
Color palette

Close

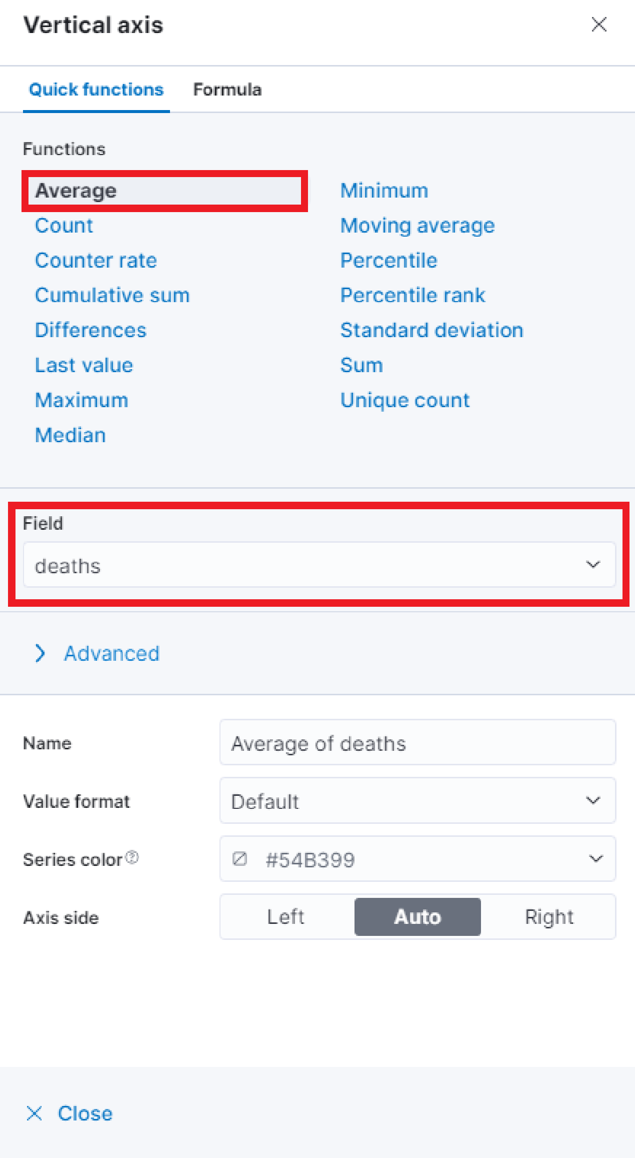
8. Below we can see our results. Los Angeles county seems to be the county with the highest number of cases.



9. To create the Vertical Stacked Bar Graph, we're going to repeat the first initial steps. Up to the point where we let Elastic know what bar graph we want to create. Now we will choose "**Ber vertical stacked**".



10. For our Vertical axis, "**Average**” will be our function and we will select **“deaths**” as our field.



Vertical axis

Quick functions Formula

Functions

Average
Count
Counter rate
Cumulative sum
Differences
Last value
Maximum
Median
Minimum
Moving average
Percentile
Percentile rank
Standard deviation
Sum
Unique count

Field

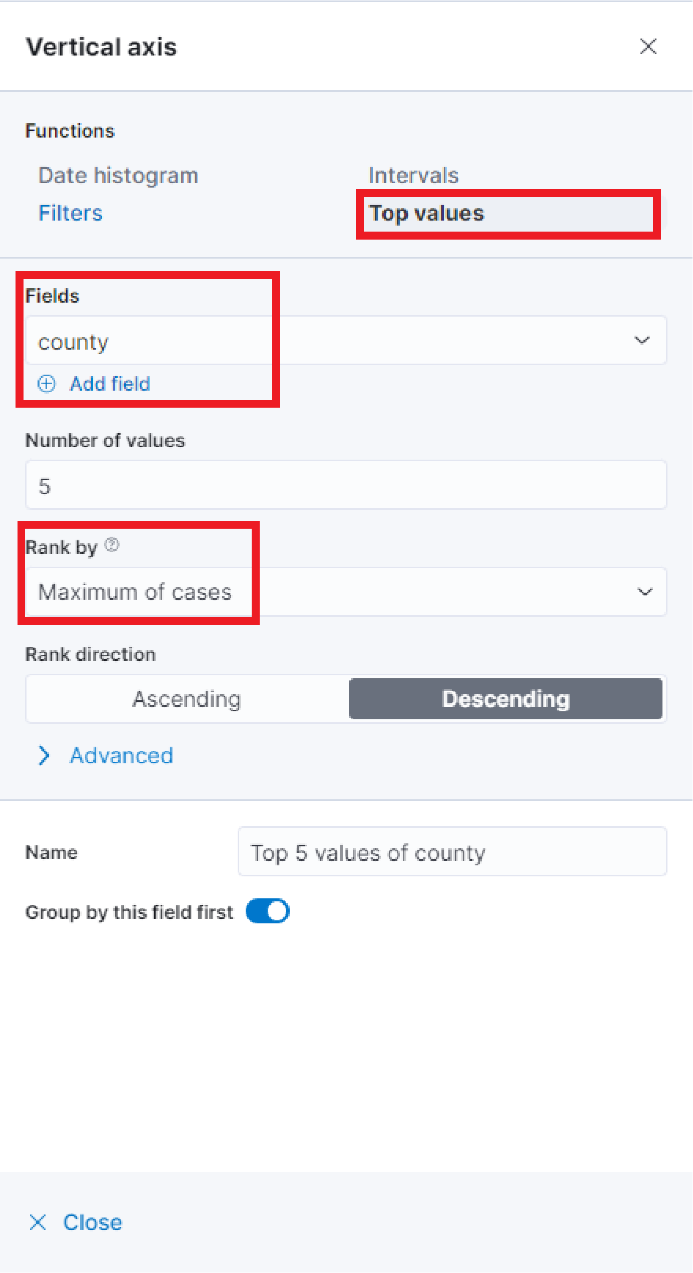
deaths

Advanced

Name	Average of deaths
Value format	Default
Series color	#54B399
Axis side	Left [✓ Auto] Right

Close

11. Our Vertical axes will have the “**Top Values**” as the Functions, with “**county**” selected as our Fields.



Vertical axis

Functions

Date histogram

Intervals

Filters

Top values

Fields

county

Add field

Number of values

5

Rank by

Maximum of cases

Rank direction

Ascending

Descending

Advanced

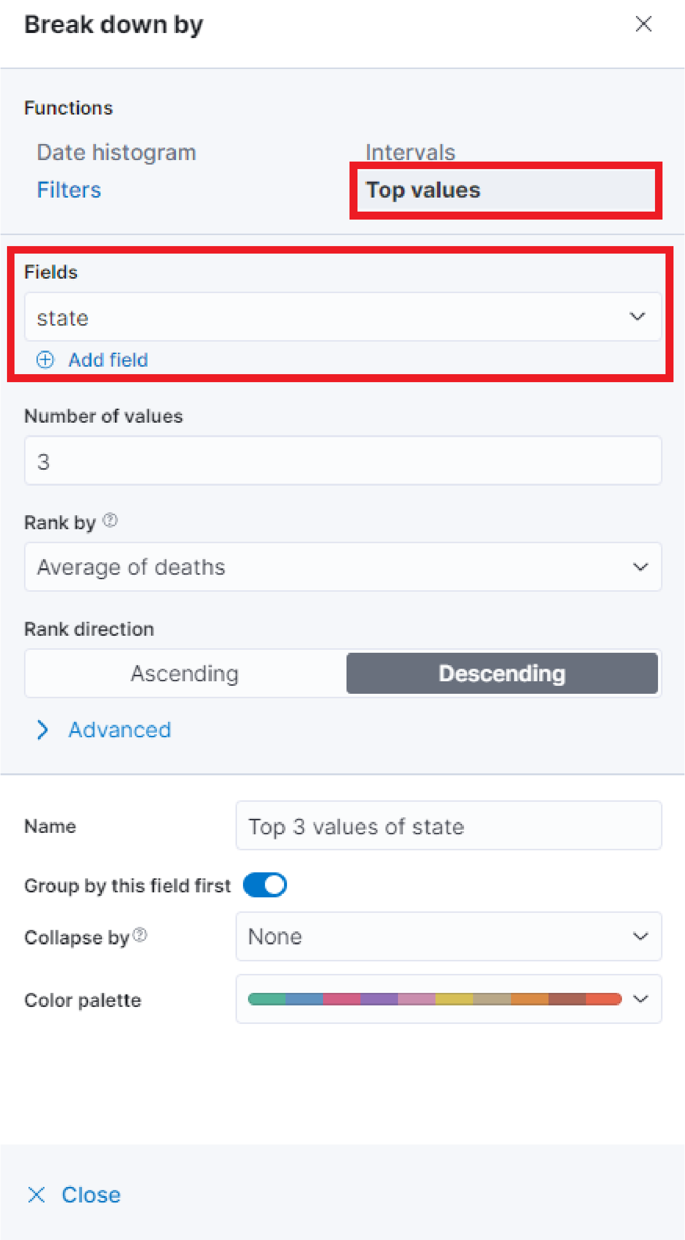
Name

Top 5 values of county

Group by this field first

Close

12. Let’s break down everything by “**Top Values**” under Functions, with “**state**” as our fields.



Break down by

Functions

Top values

Fields

state

Number of values

3

Rank by

Average of deaths

Rank direction

Descending

Name

Top 3 values of state

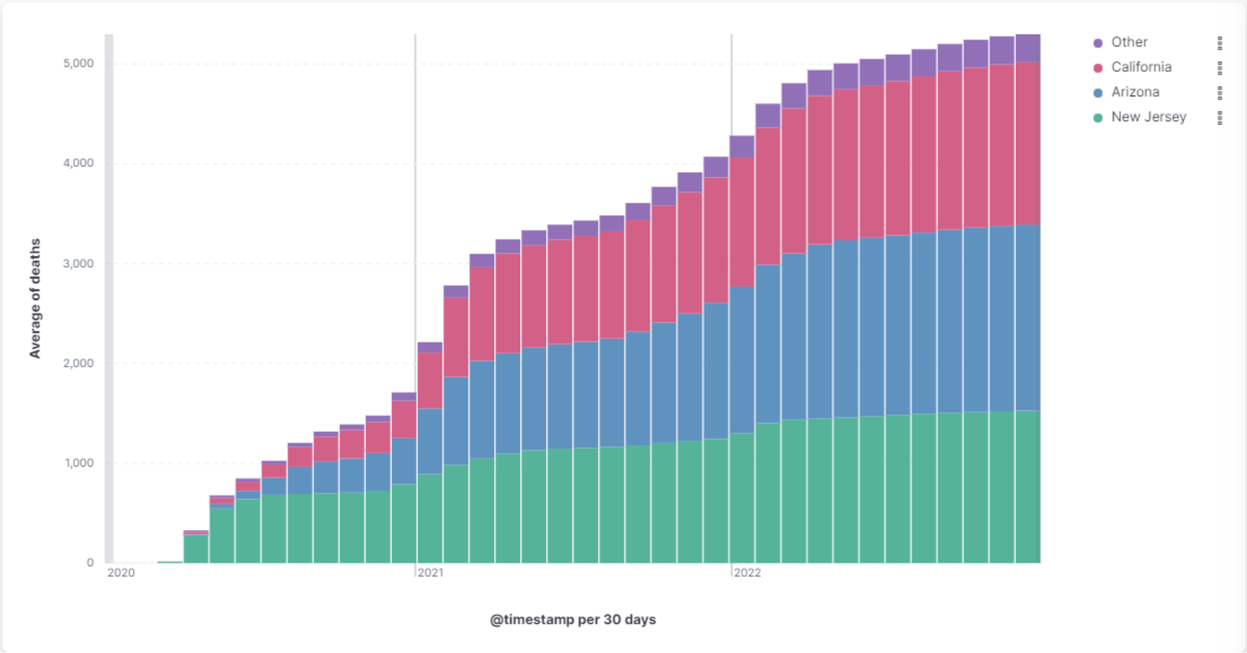
Group by this field first

Collapse by

None

Color palette

13. Once we have finished entering our settings, we can see our final results, the top 3 average amount of deaths per state.



This is the end of the tutorial