A BOOK BY A QA ENGINEER WHO LOVES KIBANA

FANTASTIC ELASTIC

Level: Beginner

My Journey to Visualise Eurovision winners using Kibana dashboards

All done using the Elastic stack, fan sites, and a computer, free

Follow along to go from No data to Oh yeah, Kibana!

Fantastic Elastic

My Journey to Visualise Eurovision winners using Kibana dashboards

Anita Lipsky

This book is for sale at http://leanpub.com/FantasticElastic

This version was published on 2019-11-14

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Book Blurb

Book cover text

A book by a QA Engineer who loves Kibana

FANTASTIC ELASTIC

Level: Beginner

An interactive reading experience for data fans!

My Journey to Visualise Eurovision winners using Kibana dashboards

by Anita Lipsky

Teaser text

Imagine making simple and stunning visualisations of data, even if you are a total beginner!

This book follows one woman's journey into creating charts, graphs and dashboards of data from the Eurovision song context, from zero to hero!

All done using the Elastic stack, fan sites, and a computer, free

Follow along to go from No data to Oh yeah, Kibana!

About the book

This is a book by a QA (Quality Assurance) Engineer who loves Kibana.

It promises a fun, interactive reading experience for data fans, who can follow along just by reading, or trying out some or all of the steps for themselves.

If you are thinking of using Kibana, or about taking the Elastic Engineer Certification at some point, this could be a fun starting point.

About the author

Anita is a Quality Assurance (QA) and Software Test Automation Engineer who works on development teams using Agile and software best practices in Oslo, Norway. She is a speaker, blogger

and founder of www.purplebugs.com[1], and an active participant in both the Testing and Girls Can Do IT communities in Oslo.

Anita has a degree in Information Technology, and currently holds two ISTQB qualifications: the ISTQB Certified Tester Foundation Level and the ISTQB Agile Tester Foundation Extension.

You can connect with Anita via email[2] or on LinkedIn[3].

Not an official Elastic product

Please note this book is a hobby project and is in no way endorsed by Elastic.

[1] https://www.purplebugs.com
[2] mailto:anita@purplebugs.com
[3] https://www.linkedin.com/in/anita-lipsky-506360120

Dedication

This book is dedicated to
all women in I.T.
because you are awesome

Introduction

Purpose of this book

The purpose of this book is to walk myself and any others who want to join my journey through a fun idea of creating gorgeous and simple visualisations of data.

I want to use a charts and graph visualisation tool I love, Kibana, from elastic.co[4], around a topic I find interesting which has easy access to data online, the Eurovision Song Contest, because it is motivating to start with things that make me feel warm and fuzzy, without any pressure.

I hope you will be able to follow along, and want to learn using Kibana for making visualisations for any of your own data.

There may not be a huge amount of people who love the Eurovision Song Contest (ESC) and who love Kibana. If you are data lovers though, please do replace ESC with your own hobby project.

Also, it is just so fun to be able to understand, digest and create such visualisations in seconds… once everything is setup and being maintained nicely. Actually, nothing really takes just seconds, it just feels like it once the setup is all done!

Who this book is for

If you would like to follow along how to make fun and informative visualisations of data, even if you have never heard of Kibana before, and even if you have no data, then this is book is for you.

If you are comfortable with using computers and learning new things, and making some mistakes here and there, problem solving, then getting back into it, you will be fine.

If you have used source control tools before, and if you worked with data before, and if you have installed software of any kind before, you will be pretty comfortable.

If you don't like computers, or reading, or trying out new things, this is not for you.

How to use this book

Feel free to browse the chapter headings and skip to whatever interests you, and skip over what does not interest you.

Feel free to read only, or to try to follow along by installing the tools I mention, and use either the same data set I use, or one of your own.

[4] https://www.elastic.co

How I will do this project

I will blog, write book, and do at the same time

I will probably change things as I go, based on your feedback, my learnings, and anything else that pops up. However the overall goal will remain the same: make fab dashboards of Eurovision data, using Kibana, in a simple and timely manner, while having fun!

How to give feedback

You can contact me on LinkedIn, via my email address listed on the About page on my blog www.purplebugs.com[5], or by commenting on this book in LeanPub[6]. All feedback is welcome and appreciated!

When I first heard of Kibana

I am a QA (Quality Assurance) Engineer, otherwise known as a Software Test Automation Engineer in QA, which is in the software testing domain. That means I am part of developing software, getting it into the hands of clients and once there, ensuring it actually solves the problems each client is paying for it to solve. In reality it means I can break down imprecise, verbal conversations into small, precise sentences that can be easily understood and translated into code; generate data that simulates what the client will be doing with the software; automate that as much as possible and; provide feedback about this continually to both the developers and the business stakeholders.

What is really nice is to be able to share this kind of information in a simple and visual way as possible. Hence, Kibana came into my life.

I first was exposed to Kibana when our team switched from Waterfall (think two big, complex software releases per year) to Agile (think lots of small releases, constantly, in an automated way as possible).

This switch made us have to become more responsible for "seeing inside" the software: when might the system soon crash? Were the users generating many more errors in the hour following the release than usual, and if so why? And so on. At that time, the "ELK stack" was chosen, where the "K" was "Kibana" and boom… we were up and running, with the usual setup hiccups and learnings that a large scale software project entails when new software is incorporated into the mix.

What Kibana solved for me

At that time I was fortunate enough to only have to care about using the tools once set up, which meant that soon I realised I could answer questions such as: What percentage of our customers were

[5] www.purplebugs.com
[6] https://leanpub.com/FantasticElastic

using IE8, as opposed to Chrome? How many customers were from the same company, or timezone?

Knowing the answers to these questions, in real time, allowed me to see and assess risk more easily. For example, if I discovered a bug that was only in IE8, and not in Chrome, yet only 5% of our users were using IE8, and it was an area of the software not widely used, and there was a workaround, we did not have to hold of the release for that, but could fix it for the next release, usually withing a few days.

Kibana allowed me to quickly make visualisations of, for example, what percentage of users were using IE8 VS Chrome. I did this using a pie chart by the way, and a tip from a very helpful colleague. Then I added the chart to our already existing dashboard, so the product manager, ops (operations) and devs (software developers) could to get to know how our customers used our software in the real world, in real time. Teamwork, for the win!

TERMINOLOGY: What is the ELK stack?

"ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana.

Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch.

As of 2019 it is now the "Elastic Stack" instead of "ELK stack", because more products have been added to the suite. Though, for me it all started with Kibana, the "K" in the "ELK Stack".

Source: elk-stack[7]

Summary

Expect to learn steps from literally having an idea of some simple data in your head, such as your local sports team's scores, or your students' grade average, or prices of your favourite products at your local store, and know how to put those into a format that can them be imported into Elasticsearch, then visualised into separate charts and graphs, and a dashboard created from those visualisations with Kibana.

Australian spelling

So my native language is English, Australian spelling and all. Crikey! So be prepared for seeing a lot of "visualisation" with an "s" instead of the American "visualization" like is used throughout the Elasticstack. It just comes out of my fingers (not fingerz, ha) that way.

[7] https://www.elastic.co/elk-stack

Coming up...

Part One and Two of this book will take you on my journey from zero to hero, from having no data, to having created my own dashboard of visualisations from freely available data online.

Part Three of this book will also lead you into other topics, so you can get a taste of what you can do with these dashboards. It will not go into depth, just plant the seeds. It will also contains some more advanced tips for you to consider, or simply ignore.

Book Part One: From Zero...

When you have finished Part One and Part Two of this book, you will have followed me on my journey from having nothing but an idea in my head, to having created a Kibana board that visualises data that I found online.

These are the steps I will be taking in Part One

1. Create a data file to analyse
2. Install Elasticsearch and Kibana locally on my computer
3. Import this data file to Kibana

Then in "Book Part Two: To Hero" I will create some visualisations. More on that later.

Now, let's get started.

Create a data file to analyse

My initial thoughts

Create perhaps in a google drive spreadsheet that can be exported as a .csv file, for the past 10 years of Eurovision winners. Do some copy/paste manually from sites such as Wikipedia, if I am not able to find some database with this already. Don't try to make a data scraper or automate this at this stage, and store this data in source control, eg github.

What I ended up doing

I ended up creating a text file and of data of Eurovision winners from the past ten years, with data from Wikipedia, and storing this in github.

Install Elasticsearch and Kibana locally on my computer

My initial thoughts

Install Elasticsearch and Kibana locally on my computer at home, and on my laptop, depending on where I am working from. Do not try to create a Docker image or store this in source control at this point.

What I ended up doing

I ended up installing Elasticsearch and Kibana on my home computer, using the freely available Debian package manager.

Import this data file to Kibana

My initial thoughts

I didn't have any deep and meaningful initial thoughts here, beyond knowing that Elasticsearch was a pre-requisite to Kibana, and I need to install and have both up and running before looking for the most realistic way to simply import data into Kibana

What I ended up doing

I ended up modifying the structure of my data file to be .csv and importing that into Kibana. Initally when I created the data file, I was just gathering data into a plain text file, without the comma separated .csv parts, so I actually went back and updated the format.

Summary

Expect deciding upon and gathering some simple data to be pretty straightforward, as long as you are disciplined and do not try to be perfect or get too much data.

Expect the installation of Elasticsearch and Kibana to be slightly different than how I did it, unless you have the same setup as me.

Expect importing the data to be very straight forward, although the terminology such as "Index" and "Index Patterns" might be new.

Coming up…

Next I will jump into creating the data file. This file will be imported into Kibana in a later chapter, and will be the basis of the data used by the visualisations.

Create a data file to analyse

In this chapter you will follow along with how I create the data file for free, using my computer and some online tools.

In order to make data dashboards, you need data, and some software that will make the dashboard. You will learn how I do that in this chapter.

Create a raw data file

For this I went to Wikipedia's Eurovision winners page[8] and found a table with all the winners within seconds. Great!

It also has more data that I initially care about, such as how many points they won by, and who the runner up was for that year. I will ignore the tempation to add this data, to avoid digressing from my initial goal.

I then went to github, logged in with my existing purpblebugs account, and created a repo named fantasticelastic/[9]

In my case I selected the MIT licence, and created a Readme. After creating my repo, readme and selecting a license, I then added a file to my repo called eurovision_winners.csv and starting manually typing out the following in from Wikipedia:

[TODO] Update this file to remove spaces between the commas in the headings and data

```
year, country, song, performer, language
2019-05-18, Netherlands, Arcade, Duncan Laurence, English
2018-05-12, Israel, Toy, Netta, English
2017-05-13
```

I completed the list so the last ten years of winners were in it, saved the file and added a comment to my commit. This took me a few minutes. Great!

[8] https://en.m.wikipedia.org/wiki/Eurovision_winners
[9] https://github.com/purplebugs/fantasticelastic/

```
year, country, song, performer, language
2019-05-18, Netherlands, Arcade, Duncan Laurence, English
2018-05-12, Israel, Toy, Netta, English
2017-05-13, Portugal, Amar pelos dois, Salvador Sobral, Portugese
2016-05-14, Ukraine, 1944, Jamala, English
2015-05-23, Sweden, Heroes, Måns Zelmerlöw, English
2014-05-10, Austria, Rise Like a Phoenix, Conchita Wurst, English
2013-05-18, Denmark, Only Teardrops, Emmelie de Forest, English
2012-05-26, Sweden, Eurphoria, Loreen, English
2011-05-14, Azerbaijan, Running Scared, Ell & Nikki, English
2010-05-29, Germany, Satellite, Lena, English
```

I realise that the column with the date is called year, however contains the day and month also. I was just moving forward quickly and not giving it too much thought. More on that later.

TIP: Make it work, then make it right

I want to keep the database simple for now, because my initial goal is to get to the point where I can create a Kibana dashboard with something that is good enough.

It is way to easy to get sidetracked and hung up on ideas that I know would be cool to have, although I know from experience it is important to get to the initial goal quickly, then refine. "Make it work, make it right" is a tip one of my mentors has said to me time and again, and it has saved me often enough from wasting time.

If you don't want to create this file yourself right now...

You can either 1. Copy the eurovision_winners.csv from my github account mentioned above, or 2. In Leanpub, where this book is published, go to Extras[10] and use the "Eurovision Winners - 10 entries only" file

Celebrate!

There is now a file containing data that will be imported into Kibana, then used to create the visualisations.

Coming up...

Next, I will install Elasticsearch and Kibana, which are the tools required for processing the data and creating the visualisations.

[10] https://leanpub.com/FantasticElastic/extras

Install Elasticsearch and Kibana on my computer

I know I keep talking about Kibana, Kibana, Kibana.

Actually, I will be installing two things:

1. Elasticsearch… to search and analyse the data
2. Kibana … to visualise and manage the data. This is where the magic happens!

Install Elasticsearch

I am installing everything locally on my computer, which is completely free. It is easy if you are familiar with installations.

If you are not comfortable with installations, then it is much simpler to go for the Hosted Elasticsearch and Kibana service, otherwise known as the Elasticsearch Service. Doing it this way will lock you into a trial period Elasticsearch Service[11] and you will then have to pay to use it.

In my case, I installed Elasticsearch myself using a Debian Package manager, which I found by clicking through and landing on this page Install Elasticsearch with Debian Package[12]

Install instructions for any computer are here: Elastic.co start page[13]

Because I did not install a hosted version, Elasticsearch is running on my computer.

Wait, Elastic VS Elasticsearch?

Elastic is the name of the company, and Elasticsearch is their core product that allows you to ingest and transform the data, among other things.

Start Elasticsearch

Since Elasticsearch is running locally on my computer, I need to start the service and ensure it is up and running. You will not need to start any service if you are using the hosted service. In my case, I followed the steps for starting elastic on my operating system

[11] https://www.elastic.co/cloud/elasticsearch-service
[12] https://www.elastic.co/guide/en/elasticsearch/reference/current/deb.html
[13] https://www.elastic.co/start

```
sudo systemctl start elasticsearch.service
```

Check Elasticsearch is up and running

I checked it was up and running in my browser by navigating to: http://localhost:9200/[14]

I saw the following in my browser, so that means it is up and running.

```
{
  "name" : "home",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "-W61xodqQRWAfikfx1PJHA",
  "version" : {
    "number" : "7.1.0",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "606a173",
    "build_date" : "2019-05-16T00:43:15.323135Z",
    "build_snapshot" : false,
    "lucene_version" : "8.0.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Note - this took maybe 10 seconds or more to be up and running, so I had to be a little patient.

Install Kibana

If you are using the Hosted Elasticsearch and Kibana service you will not need to install Kibana in addition - it will already be available to you.

In my case, I do need to install Kibana, so I went back to the Elastic.co start page[15]

I followed the links again to find the Kibana Debian package installer on this page Install Kibana with Debian Package[16]

To install Kibana I ran the command `sudo apt-get update && sudo apt-get install kibana`

[14] http://localhost:9200/
[15] https://www.elastic.co/start
[16] https://www.elastic.co/guide/en/kibana/current/deb.html

Start Kibana

To start Kibana I ran the command `sudo systemctl start kibana.service`

Check Kibana is up and running

If you are using the Hosted Elasticsearch and Kibana service you will simply navigate to the URL provided to you when you register.

Because I installed Kibana locally on my computer, I checked it was up and running in my browser by navigating to: http://localhost:5601[17]

This took me to a "Welcome to Kibana" page

Note - this took maybe 20 seconds or more to be up and running, so I had to be a little patient.

Optional: Add Kibana's sample data

The page gave me an option to Try Sample Data which I selected, then selected the sample eCommerce orders data.

I plan to add my Eurovision data next, though for now it is nice to know I have something else for playing around with too.

Celebrate!

Kibana and Elasticsearch are available and up and running.

Coming up...

Next, I will import data that will be used to create visualisations in Kibana.

[17] http://localhost:5601

Import this data file to Kibana

Ok, now that I have data to analyze and Elasticsearch that can do the analyzing, I need to import the data.

I'm keeping it simple for the purpose of this book.

Create an Index

All documents in Elasticsearch are stored inside of one index or another. Source: elasticsearch/reference[18]

Therefore we need to import the data and save it as an index.

1. In Kibana, click on Machine Learning in the left hand panel
2. In the subnav, click on Data Visualizer
3. Under Import Data, click Upload File, and drag and drop the .csv file there
4. Click "Import"
5. Enter "eurovision_winners" in the Index Name - do not select Index Pattern
6. Click on Advanced
7. Look through the contents and notice that the year field is importing as "text", or possibly the number format "long", which will be updated to "date" in the next step
8. Under Mappings, manually change the year "keyword" type to "date"
9. Click Import.

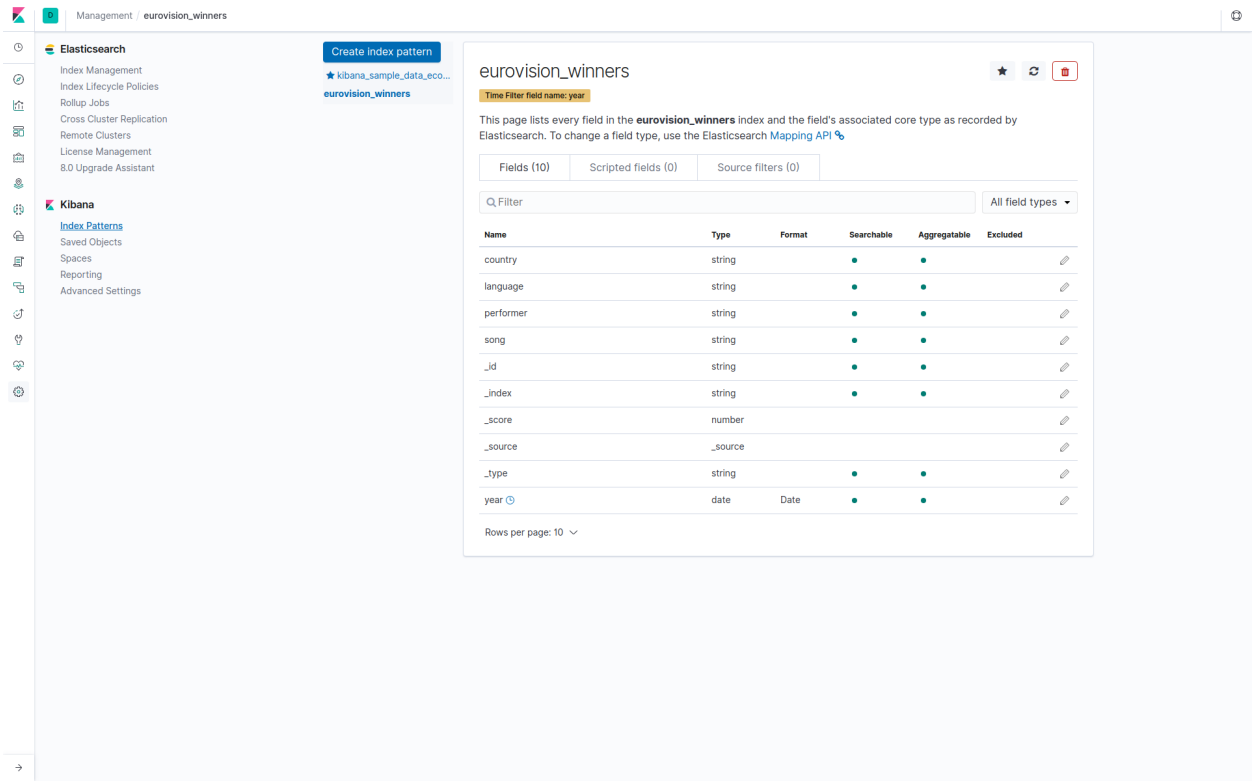
Create an Index Pattern

An Index Pattern lets Kibana know which index or group of indexes I want to work with.

In this simple case, the Index Pattern will only match the one index I just created. However, in typical real world scenarios, visualisations are based on large amounts of data that come from many indices.

Nevertheless, creating an Index Pattern is a pre-requisite for working with the data in Kibana, so Kibana knows where to find and aggregate the data.

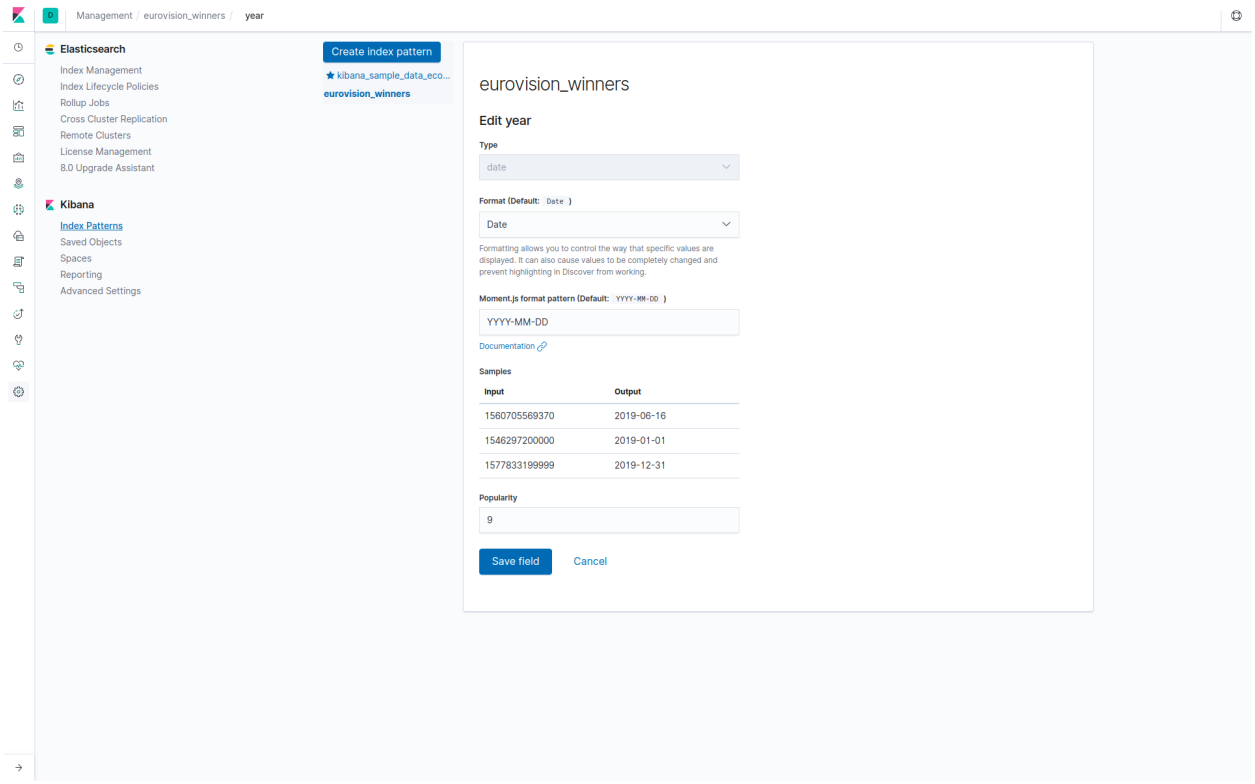
[18] https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-create-index.html



Index pattern - default

1. In Kibana, go to Management > Index Patterns
2. Enter “eurovision_winners” index pattern, and wait for the message “Success! Your index pattern matches 1 index.” and the button “Next step” to show
3. Click “Next step”
4. In “Configure Settings” select “year” for the Time Filter field name and click “Create Index Pattern”
5. Wait for the success message
6. Click the edit pencil next to the “year” field
7. Select “Date” from the Format dropdown list
8. Enter YYYY-MM-DD as the format pattern
9. Save field

It is now possible to find this index pattern under Management > Index Patterns then clicking on “eurovision_winners” in the list of index patterns.



Index pattern - date format

Tip: More details on importing .csv files

Here is a great blog post by a Product Manager at elastic.co who has written an entry level description of how to import a .csv file:

aftershock-therapy-with-elasticsearch-and-csv-data-import[19]

Note location data should be converted to correct format

I am already thinking that my location data probably needs to converted to latitude and longitude. Though, that seems kind of complicated to figure out right now, so I'll make a TO DO for this for later.

TO DO list so far

1- Update location data so it will show on a Kibana map

[19] https://www.elastic.co/blog/aftershock-therapy-with-elasticsearch-and-csv-data-import

Tip: Note down TO DOs because “You Ain’t Gonna Need It, Yet”

… otherwise known as “YAGoNIY”

When I notice things that should be addressed, though will detract from the current goal, I note a TO DO in a list.

Then when I have achieved my goal, I go back through the TO DOs and pick them up.

In this case, I will note a TO DO for figuring out how to convert the location data to something that will be mappable to an actual latitude and longitude on a map.

Celebrate!

The Eurovision data is alive and well inside the Elastic Stack, ready for me to create some visualisations with it.

Coming up…

I am now ready to go to Part Two of the book, which is where I will Discover, Visualize then create a Dashboard with the data. Note, those three terms in capital letters are actual terminology and menu items found in Kibana.

Book Part Two: ...To Hero

Now that Elasticsearch and Kibana are up and running, and we have some indexed data, it is time to make some visualisations.

These are the steps I will be taking in Part Two

1. In Discover, take a peek at the data to get some ideas for how to visualise it in different ways
2. In Visualize, create some visualisations, each which is a type of chart, graph, map, word cloud, and so on.. there are many possibilities
3. In Dashboard, select the best visualisations and put them together on a dashboard

This is the typical flow when working with data and visualisations.

Clean the data

I will probably want to do some data enhancements at this point, such as correcting any typos or errors, and adding more data. I will come back to that as needed.

First, I will just create something simple that works.

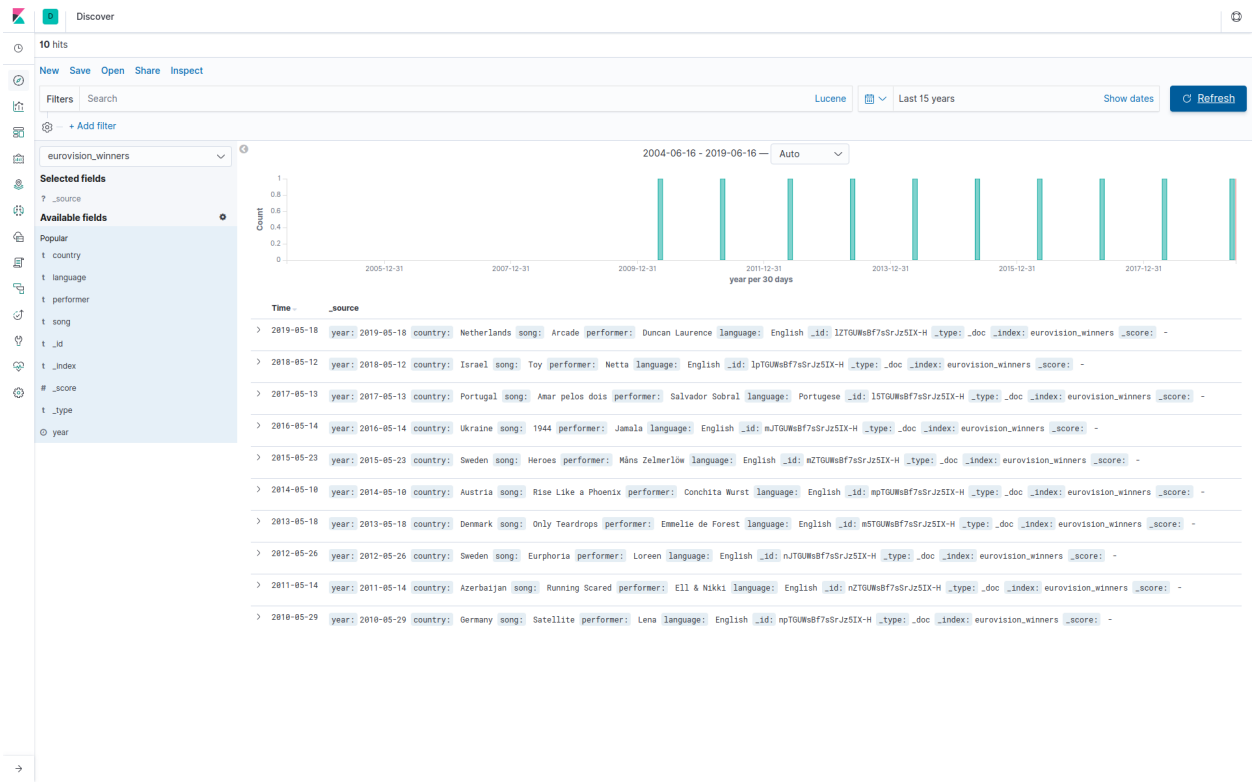
Discover

Here is where I want to look at the data.

I want to know if it actually got imported at all, and if it is searchable by date.

Did the data import at all?

1. Navigate to “Discover” in the left hand menu
2. Look for “eurovision_winners” right above the “Selected fields” heading, and select it if it is not automatically there
3. If there are other Elasticsearch indices available, I need to click on the dropdown list and look for “eurovision_winners” and select it, if it is not the first item in the list.
4. Because by default Kibana only shows data from the last 15 minutes, there will be no data shown. I will come to that in the next section



Discover - default

Of course, something might have gone wrong. If “eurovision_winners” is nowhere to be found, then troubleshoot Discover my data

TIP: Troubleshooting Discover my data

Go to Management > Index Management and look for "eurovision_winners" there. If it is not present, go back to the Importing data chapter and follow the steps, because the data was not uploaded at all.

If it was present in Index Management, go to Management > Index Patterns and look for "eurovision_winners" there. If it is not present, go back to the Importing data chapter and follow the steps for creating an index pattern.

If it is present, look for the options to delete the Index Pattern and create it again.

The simplest is probably to delete "eurovision_winners" and start again, because if there are there and you do not see the data as expected, probably a step or setting was missed during indexing.

Since the data is such a small file, it is really fast to upload and index, so don't worry about deleting and starting again. It is a great way to learn.

Is the data searchable by date?

Here is where I want to ensure the data is searchable by date.

By default Kibana shows only the last 15 minutes of data, and the last date entry is from May 2019 which is more than 15 minutes ago. Therefore I want to update the time filter to show more than the last 15 minutes of data.

1. Make sure I am still in Discover
2. In the top right area select last 15 years (not minutes!) from the date selector, and Apply
3. All the data will appear below, along with all available fields such as location and language shown in the left hand panel

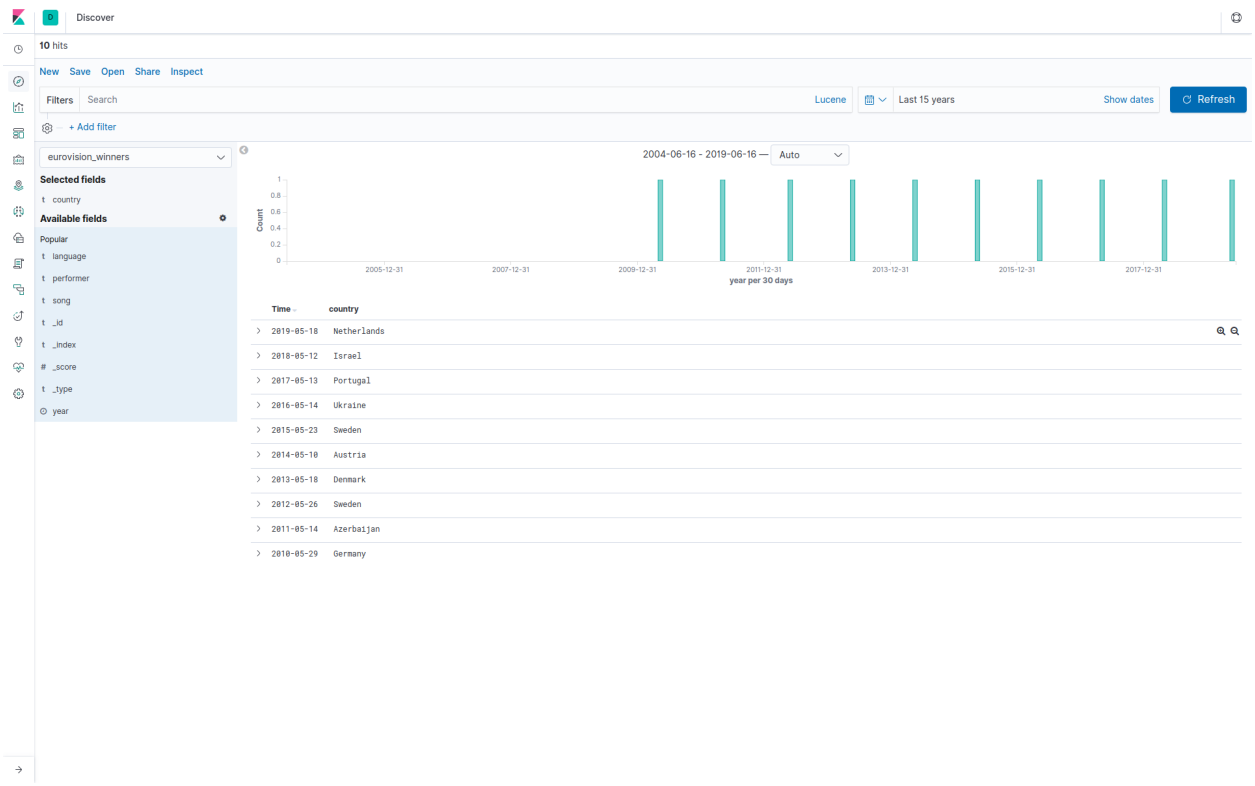
Play around with the data selector to get a feel. Selecting the last 5 years will filter on only data from the last five years, for example.

If I do not find the date selector and there is data shown, it means the data is not searchable by date. In other words, the indexing did not pick up that "year" was a date field. If this is the case, then go back to Part One importing data, and ensure that the "year" field was updated to be in "date" format.

Quickly view the data in Discover in a readable way

1. To organise the results easily, look for the Country field in the left panel and click "Add"
2. Note the data returned now shows just Time and Country, giving a clean and simple overview
3. Click "Remove" next to the Country field and note that this toggles the option off

4. Another way to do this is to to expand a row of data, move the mouse to find the toggle column option next to some data and click.
5. Play around with adding and removing columns.



Discover - country field

TIP: To start again in Discover

To start again when playing around in this space, click New then select “eurovision_winners” again

Filtering

...using Fields in the left hand panel

1. In Language field in the left hand panel, select the + magnifying glass next to English
2. There will now only be nine entries shown, with Portugal not showing since the song was in Portugese, not English
3. Note at the top Filters shows language:English
4. Click X to remove the filter

...using expanded document in the right hand panel

1. In the right hand panel, expand one entry item
2. All fields with their corresponding values should be shown
3. Move the mouse to the language row and select the + magnifying glass next to English
4. Note at the top Filters shows language:English

Save a search

1. Click Save, enter “English songs only” and Confirm Save
2. The search will now be saved.

Open a saved Search

1. To open a saved search, click Open, and select the “English songs only” search
2. The search will be shown, and the name will be shown in the Discover breadcrumbs at the top

This saved search will be used when creating a visualisation.

Celebrate!

I have now viewed the data in Kibana, and ensured the data can be filtered by date. I have tried other filters, and cleared the filters again. I have also saved a search, and opend a saved search.

Coming up...

Next, I will create some visualisations such as a vertical bar chart, a pie chart, and a word cloud. I will start with a simple “quick cheat” way to make a visualisation.

Visualize

It's time to create graphs and charts, or visualisations as they are called in Kibana.

The idea is to showcase relationships between the data so that humans can quickly interpret and better understand the data.

All of these visualisations will be shown on one dashboard later on.

Create a "quick cheat" Vertical Bar chart

1. In Discover, select the last 15 years, and then click on the word "Country" in the left hand field panel.
2. Click Visualize when it expands
3. I am now in the Visualize section, and have created a Vertical Bar chart. It clearly shows Sweden as the winner: in other words, they won the most times out of the available data I uploaded - two times - where all other countries won once only.

An overview of the Visualize area and toolkit

Using the "quick cheat" visualization to get a quick overview of how the Vertical Bar chart was created, I now take a closer look at the chart.

What is displayed on the X and Y Axes

1. In Vizualise, "Data" should be selected by default under "eurovison_winners" left hand panel. If not, click it
2. Under Metrics, click the arrow next to "Y-Axis" to expand it, and note that "Count" is the Aggregation selected, with no custom label
3. Under Buckets, expand "X-Axis" and note "Terms" is the Aggregation selected, by field Country, ordered by metric Count, descending, size 20, with no custom label

Change some defaults

1. Still under Buckets, "X-Axis", change the size to three and click the green Play button at the top of the panel to Apply changes
2. Note that only three countries are shown: Sweden, Austria and Azerbaijan
3. Note the label shows as "country: Descending" on the chart
4. Change the "X-Axis" custom label to "Country" and Apply changes
5. Note the label shows as "Country" on the chart

Filtering

1. Under Buckets, “X-Axis”, change the X-Axis size to 10 and Apply changes. This will still use all the data, since there are only 10 entries. Only 9 colums show, because Sweden has two entries that won.
2. Using the data selector in the top right area, select last five years and Refresh. Five entries are shown, each that won one time.
3. Select the last 10 years and click Refresh, to view all the data again.
4. Click the bar labelled “Sweden”. Note that only Sweden shows, and Filters shows country:Sweden set.
5. Click X next to filters to remove the filter. Note all the data is shown again.

Create a Vertical Bar chart from scratch

1. Click Visualize in the very left panel, to get to the Create Visualization page
2. Click + to create new
3. Select Vertical Bar
4. Select Source: Index pattern “eurovision_winners”
5. Now I will be in the Visualize section
6. Under Buckets, X-Axis, select Aggregation:Terms, Field:Country, Order By Metric:Count, Order: Descending 20, then Apply changes
7. I should see the same the same Vertical Bar chart created earlier with all the data. If this is not the case, check the date filter and adjust to see the last 10 years, then refresh.
8. Save the visualisation as “Winners vertical bar”

TIP: Troubleshooting

Change the date filter to the last 15 minutes and Refresh. No data will be shown.

When creating new visualisations, if no data is shown, ensure the date filter is set correctly.

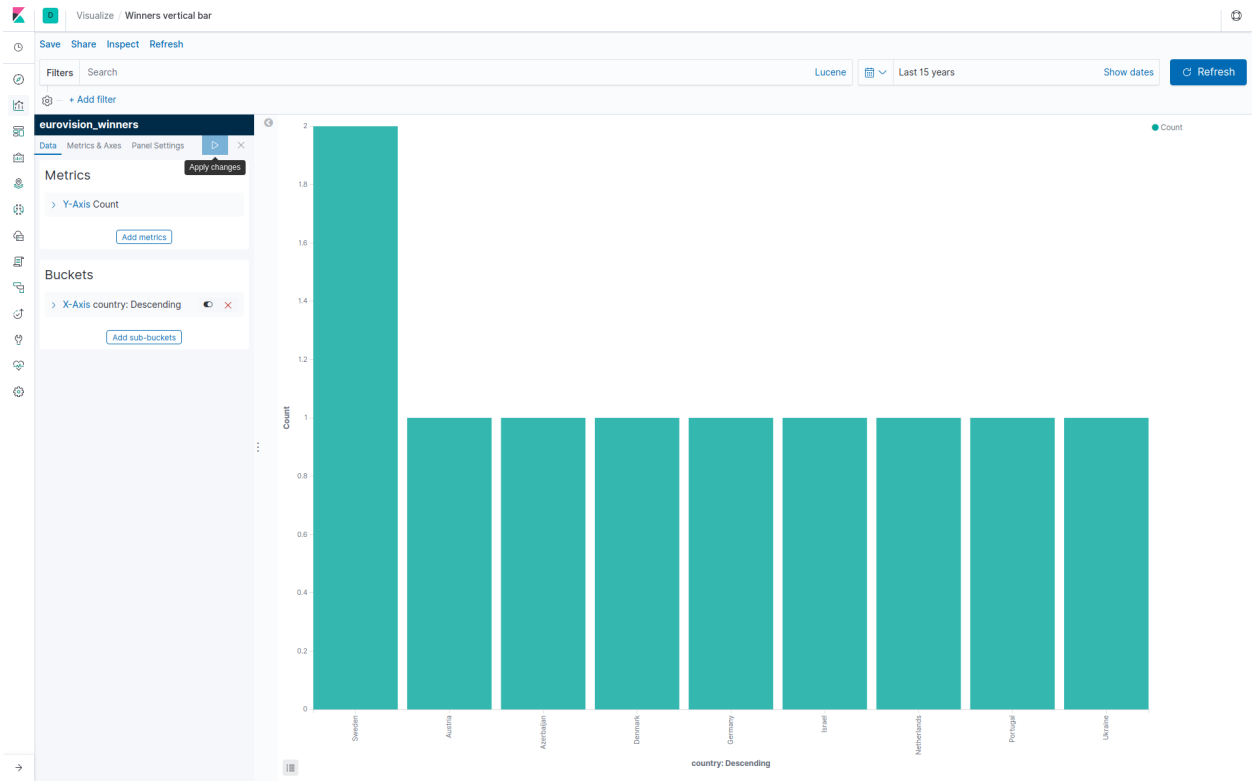
Since the default is to show the last 15 minutes, only the most 15 minutes of data will show for for any new vizualisations created with this data set.

Create a Vertical Bar chart from a saved query

This is the same as creating one from scratch, except instead of selecting Source: Index pattern “eurovision_winners”, select Saved Search: English songs only.

1. Click Visualize in the very left panel, to get to the Create Visualization page
2. Click + to create new

3. Select Vertical Bar
4. Select select Saved Search: English songs only
5. Now I will be in the Visualize section
6. Under Buckets, X-Axis, select Aggregation:Terms, Field:Country, Order By Metric:Count, Order: Descending 20, then Apply changes
7. I should see the same the same Vertical Bar chart created earlier with all the data. If this is not the case, check the date filter and adjust to see the last 10 years, then refresh.
8. Note there are only eight vertical bars in the chart, since Portugal has been filtered out by the saved search



Winners Vertical Bar

Create a Tag Cloud

1. Click Visualize in the very left panel, to get to the Create Visualization page
2. Click + to create new
3. Select Tag Cloud
4. Select Source: Index pattern “eurovision_winners”
5. Now I will be in the Visualize section
6. Make sure at least the last ten years are showing, as per the date selector in the top right section
7. I will see a giant word “all” as the Tag Cloud visualization.

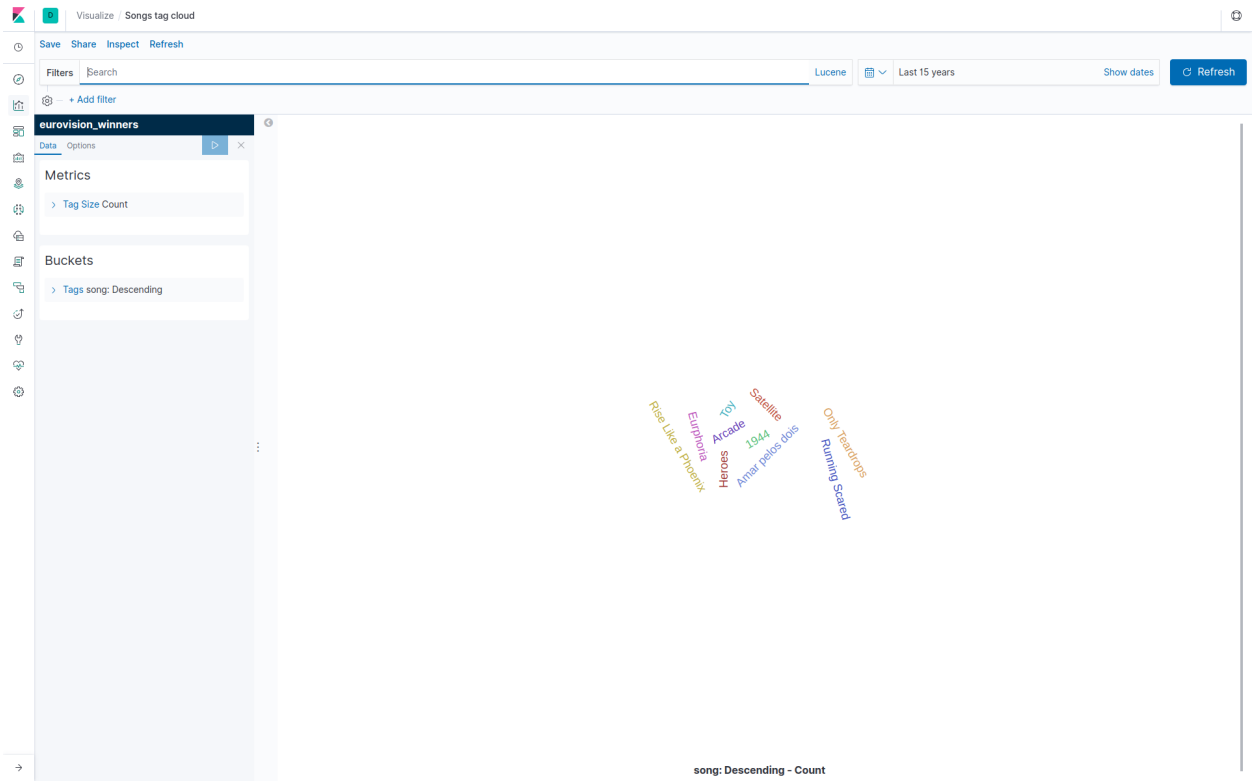
8. Under Buckets, X-Axis, select Aggregation:Terms, Field:Song, Order By Metric:Count, Order: Descending 20, then Apply changes
9. All songs will be shown in the Tag Cloud, in various colours

Options for the Tag Cloud visualization

Each visualization type has various option. I want to explore these to see how to give the visualisation a colourful Eurovision feel.

1. Instead of being in Data, select Options
2. Drag the slider to play with increasing the font size, and apply changes
3. The text should now be larger
4. Under Orientations, select Right angled, and apply changes
5. The song titles are now showing as a combination of verical and horizontal text, which is quite playful!
6. Under Orientations, select Multiple, and apply changes
7. The song titles are now showing at various angles, still in a large font, in various colours. This really gives it a nice Eurovision vibe!
8. Save the visualisation as “Songs tag cloud”

While playing with this I notice there is a spelling error in the song title Euphoria. Never mind, I’ll come back to that later and add it to my to do list.



Songs tag cloud

TO DO list

1. Update location data so it will show on a Kibana map
2. Update spelling of Euphoria

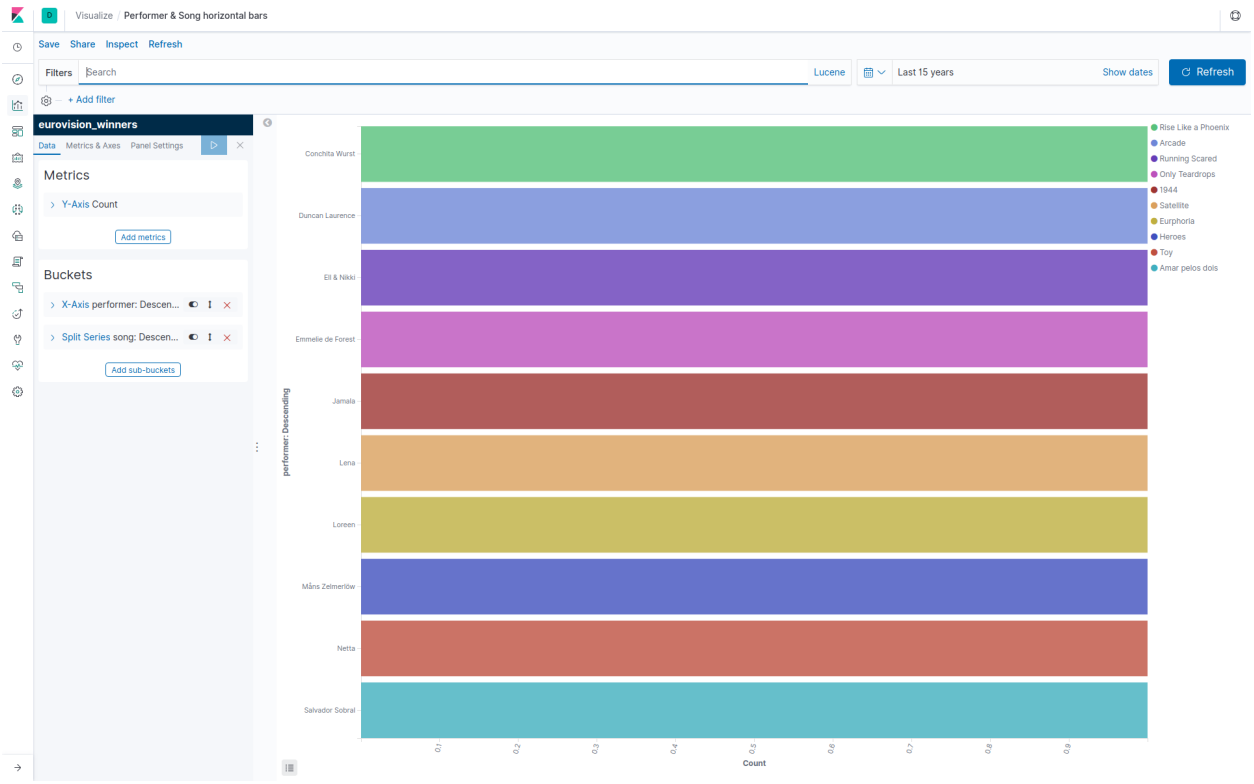
Creating a Horizontal Bar chart

1. Click Visualize in the very left panel, to get to the Create Visualization page
2. Click + to create new
3. Select Horizontal Bar
4. Select Source: Index pattern “eurovision_winners”
5. Under Buckets, X-Axis, select Aggregation:Terms, Field:Performer, Order By Metric:Count, Order: Descending 20, then Apply changes
6. Each bar will represent a performer. The bars are quite thin.
7. Under Buckets, select Split Series, Aggregation:Terms, Field:Song, Order By Metric:Count, Order: Descending 20, then Apply changes
8. Each bar now has a different colour, which represents a split for each unique paired performer and song entry
9. The Song names appear in the key of the visualization

Options for the Horizontal Bar visualization

To give the chart more visual impact I will expore the options

1. Instead of being in Data, select Metrics and Axes
2. Under Metrics: Count select Mode: Stacked and apply changes
3. Now the bars are wider, making the colours pop, which gives the visualization a real Eurovision feel!
4. Move my mouse over the bars and note how the performer and song information pops up as a mouse tooltip
5. Save the visualization as “Performer & Song horizontal bars”



Performers and Songs horizontal bar

TIP: Selecting a vizualisation

It can be tricky deciding how best to vizualise data that shows the many interesting details and relationships.

Here is a guide that can help.

Visual Vocabulary chart[20]

[20] https://raw.githubusercontent.com/ft-interactive/chart-doctor/master/visual-vocabulary/poster.png

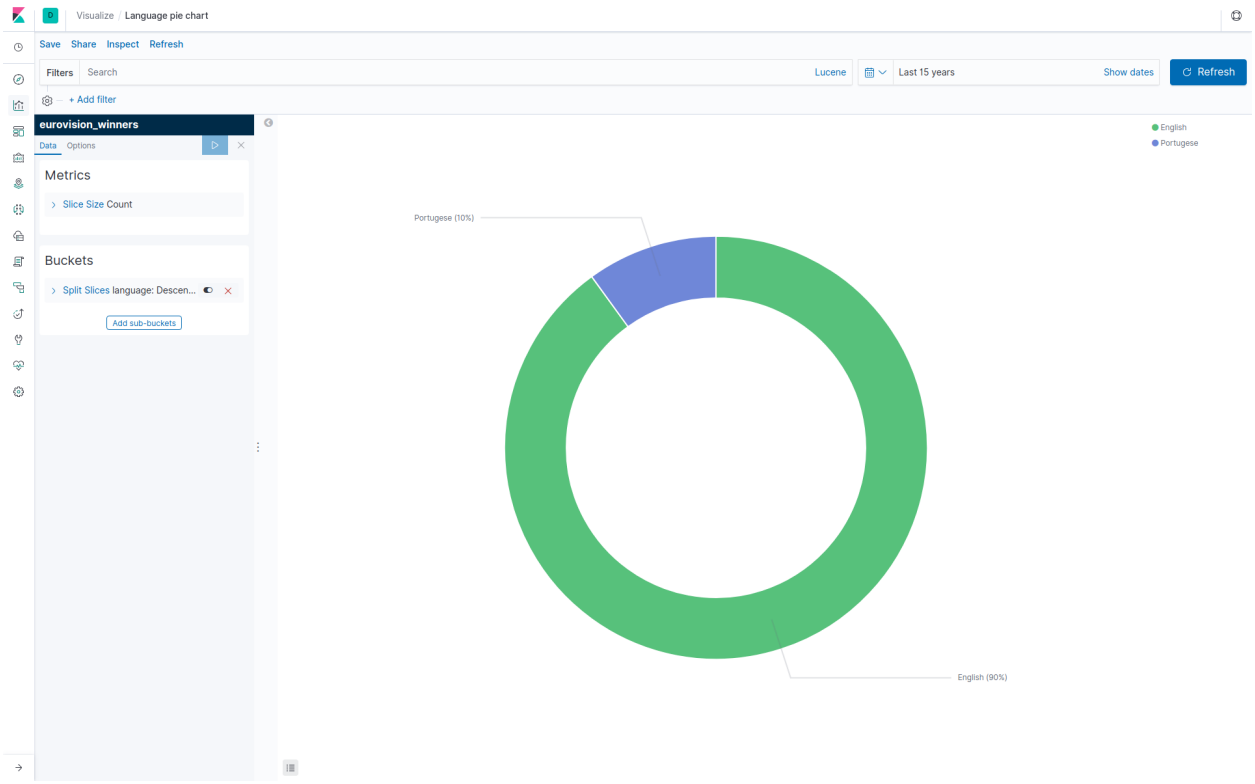
Creating a Pie chart

1. Click Visualize in the very left panel, to get to the Create Visualization page
2. Click + to create new
3. Select Pie
4. Select Source: Index pattern “eurovision_winners”
5. Under Buckets, selet Split Slices, Aggregation:Terms, Field:Language, Order By Metric:Count, Order: Descending 20, then Apply changes
6. A donut shape appears with two colours, one for each language. The larger piece of donut represents the songs most performed in one language, English.
7. The Languages appear in the key of the visualization

Options for the Pie visualisation

To make the data more clear, I will expore the options

1. Instead of being in Data, select Options
2. Under Label Settings select Show Labels and apply changes
3. Now a line with the name of the language points to each langauge
4. Move my mouse over the bars and note how the language, number and percentage of songs performed in that language, pops up as a mouse tooltip
5. Save the visualization as “Language pie chart”



Language pie chart

Celebrate!

I have... 1. created a bar chart quickly from within Discover 2. create a visualization based on a saved query 3. become familiar with exploring different options for different visualizations 4. created four different visualizations, each with a fun Eurovision vibe!

Coming up...

In less than a minute, create a dashboard with all of these saved visualisations.

Dashboards

Elastic.co is a search company with a simple goal: to solve the world's data problems with products that delight and inspire.

When I created my first dashboard, I was so delighted and inspired, I wanted to solve all my data problems with it.

Within a few minutes, I think you will feel the same way.

Create a dashboard using the saved visualizations

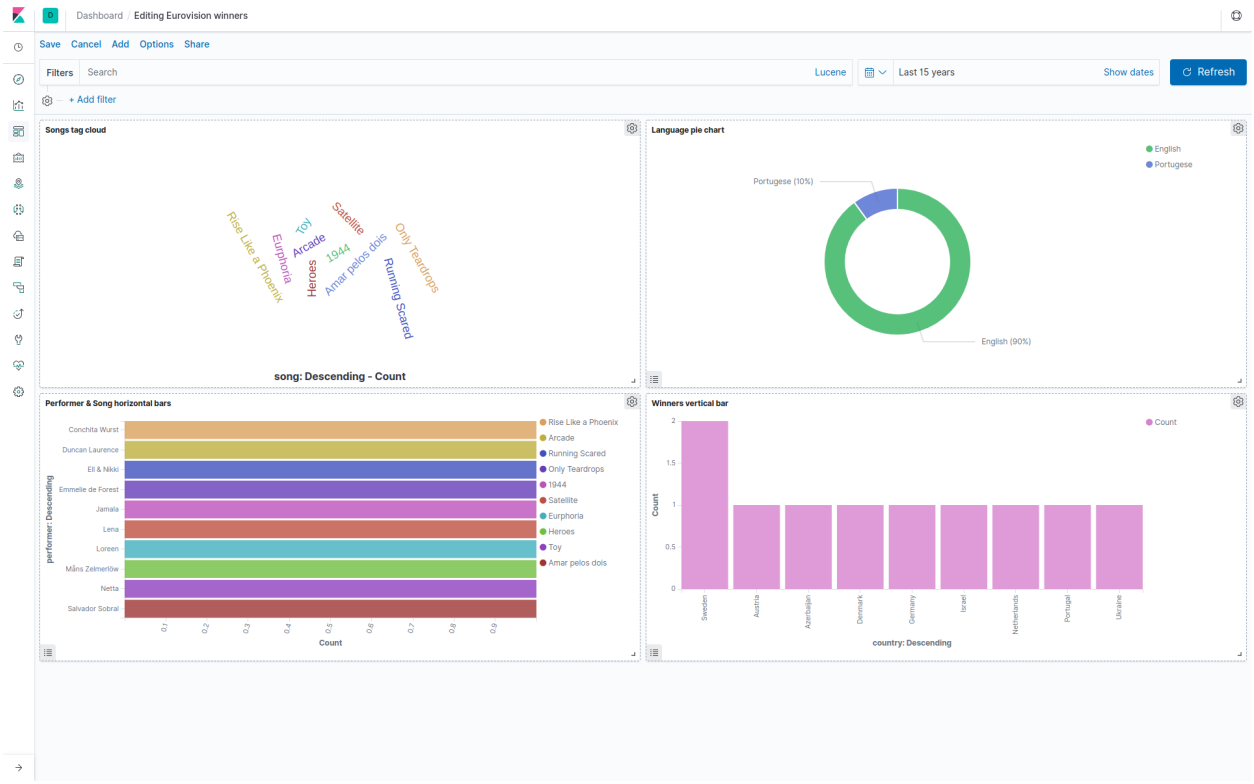
This is so fast and easy, this can be done in less than a minute

1. Navigate to "Dashboard" in the left hand menu
2. Click Create new Dashboard
3. Ensure the date selector is still showing at last the last ten years
4. Click Add and select "Winners vertical bar" from the list
5. Note that it shows up on the main page
6. Click the other three visualizations from the list: "Songs tag cloud", "Performer & Song horizontal bars", and "Language pie chart", one after another
7. When done, click back into the main page to hide the panel listing the visualizations
8. Note that you have created a gorgeous and colourful dashboard!
9. Save the dashboard as "Eurovision Winners" and Confirm Save

Adjusting the dashboard

There are a few quick wins that can be done, to balance out the look and feel

1. Click Edit
2. Hover over the visualizations and drag them around so they are positioned as follows:
3. "Performer & Song horizontal bars" on the top left
4. "Songs tag cloud" on the top right
5. "Language pie chart" on the bottom right
6. "Winners vertical bar" on the botton right
7. Note each visualization snaps to align into place when moved around
8. Save the changes to the dashboard



Dashboard

Filtering on the dashboard

When I was writing this part in the book, I felt delighted and inspired all over again, the same feeling I had when I had just created my first dashboard.

The beauty of a Kibana dashboard is how easy and intiutive it is to get to know the data. Also, it looks so nice!

Filter on the country that won the most

1. Click on Sweden in the winners vertical bar
2. Note the Count shows as 2, indicating Sweden has won twice
3. Filters in the top right will show country:Sweden selected
4. Note the song tag cloud shows the two song names that Sweden won with. (Ugh, I can see the typo “Eurphoria” really clearly now, which should be “Eurphoria”, though it’s on my to do list to correct later)
5. Note the Language pie chart shows 100% English, so we know both songs were performed in English

6. Note the Performer Song and vertical bars shows the performer labelled on the left, the song name marked by the colour key, and hovering over each bar shows the performer and song listed together
7. Click the X next to the filter to clear the filter and see all songs again

Filter on the only language that is not English

1. Click Portugese in the Language Pie chart
2. Note the Songs tag cloud is now empty. I don't know why, though I see a yellow warning sign in the bottom corner of empty tag cloud and hover over it
3. It says the container is too small and some tags might be omitted. I'll add that to my TO DO list for later, and move on. For now, it's no big deal on.
4. Note the Performer Song and vertical bar shows the Performer as Salvador Sobral, and the song as Amar pelos dois. A beautiful song, and brings back happy memories of watching his live performance on TV!
5. Click the X next to the filter to clear the filter and see all songs again

Filter on favourite artists

… using the filter at the top play around with selecting one or more of my favourite artists.

Editing a visualization on the dashboard

I would like to change the colour of the winners vertical bar simply because there is too much green for my taste

I believe this can be done in about twenty seconds, even if it is my first time.

If not, it takes about three seconds, or as fast as I can click a few times and save.

1. On the Winners vertical bar chart, select the cog icon and Edit Visualization
2. This will take me to the Winners vertical bar Visualization for editing
3. Click the "Count" label on the chart on the top right side
4. A whole of coloured dot options will appear
5. Select another colour. In my case, I selected a nice rich pink, and all the bars on the chart changes to pink
6. Save, and Confirm Save
7. Go back to Dashboards, and note the Winners vertical bar chart now shows in pink
8. Save the dashboard, and Confirm Save

TO DO list

1. Update location data so it will show on a Kibana map
2. Update spelling of Euphoria
3. Investigate if the Tag cloud can show Portugese, if language:Portugese filter is selected on the dashboards

Be delighed and inspired

I am certainly delighted and inspired by the data now.

1. Sit back, and take a long, happy look at the dashboard
2. Click around some more
3. Smile, in a delighted and inspired way :)

Optional: Export the dashboard

This step is important if you would like to have a portable project, as discussed in the next part of the book.

Reference: Managing Saved Objects[21]

Export a dashboard using the UI

1. Go to Management > Kibana > Saved Objects
2. Select the dashboard to export
3. Select Export > Include related objects
4. Save the export.ndjson someplace for future reference

This export is a backup, and can be re-imported at a later time if required.

Celebrate!

I have create a dashboard with all of these saved visualisations, and exported it as a backup.

Coming up...

I will go through my TO DO list in order to enhance and clean the data. I will also add more data to the initial set of data.

[21] https://www.elastic.co/guide/en/kibana/current/managing-saved-objects.html

Data enhancements

[TODO] Write intro

[TODO] Write out summary

[TODO] Write out that I will pick the easiest win first, which is updating the spelling error

[TODO] Format the following steps, starting with step one which is to use Dev Tools

Get all data from the index

GET eurovision_winners/_search

Scroll down and look for _id of document with song spelt as “Eurphoria”

A snip of this document will be like:

```
{
  "_index" : "eurovision_winners",
  "_type" : "_doc",
  "_id" : "DTKnWm0BDynNCGR1kkk_",
  "_score" : 1.0,
  "_source" : {
    "year" : "2012-05-26",
    " country" : " Sweden",
    " song" : " Eurphoria",
    " performer" : " Loreen",
    " language" : " English"
  }
},
```

Copy out the value of _id

Then update just that document _id as follows, ensuring the value of _id is what was just copied:

POST eurovision_winners/*doc/DTKnWm0BDynNCGR1kkk*/_update { “doc”: { “ song”: “Euphoria” } }

Get this document to ensure the typo was corrected

GET eurovision_winners/*doc/DTKnWm0BDynNCGR1kkk*

Celebrate!

[TODO] write a summary of what was achieved

Coming up...

[TODO]

Book Part Three: Getting Fancy

These will be the topics covered. Don't worry if the topics don't make sense now. You can learn more when we get there.

Now I want to make the project portable

...using inspiration from elastic-gcdit[22] which was a great initiative and probably the most fun workshop I have attended. Thank you Girls Can Do IT, Oslo!

...in order to have a package that can be stored in the cloud, and thus used and run by anyone.

Let's take a look at this in the next chapter

Now I want the world to see this

Now that these gorgeous dashboards have been created, what should I do with them?

Did you know that the dashboard can be incorporated into any software project?

For example, a website or an app would be a great way to share this info with Eurovision fans

Summary

[TODO]

[22] https://github.com/htmevik/elastic-gcdit

Make the project portable

Make the whole project portable, because ... [TODO]

To do this I will be using Docker[23]

What is Docker?

Docker is a tool used to create containerised development environments.

In this way, the project can be easily installed and uninstalled on various computers, thus being portable.

The steps to make the project portable

1. Install Docker
2. Install Elasticsearch using Docker
3. Install Kibana using Docker
4. Run Elasticsearch using Docker
5. Run Kibana using Docker
6. Check Elasticsearch and Kibana are running
7. Put these commands in a bash script
8. Including the data as part of this portable project

Install Docker

Install docker installed on your operating system. Info can be found in docs for Docker[24] for your operating system.

This is a pre-requisite for the next steps.

Install Elasticsearch using Docker

Run `docker pull docker.elastic.co/elasticsearch/elasticsearch:7.2.0`

This will grab the officially supported docker image for elasticsearch version 7.2.0

There will be some logs output while it downloads and installs.

[23] https://www.docker.com
[24] https://docs.docker.com

TIP: Offically supported elastic.co Docker images

Here is the list of offically supported elastic.co docker images[25] in case you want to grab other versions.

Install Kibana using Docker

Run `docker pull docker.elastic.co/kibana/kibana:7.2.0`

There will be some logs output while it downloads and installs.

Run Elasticsearch using Docker

Run `docker run -d --rm --name elasticsearch_1 -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:7.2.0`

Run Kibana using Docker

docker `run -d --rm --name kibana_1 --link elasticsearch_1:elasticsearch -p 5601:5601 docker.elastic.co/kibana/kibana:7.2.0`

Check Elasticsearch and Kibana are running

Run `docker ps` to see the list of running containers.

There should be only two, one named `elasticsearch_1` and one named `kibana_1`

In addition, as per the earlier chapter in my book about installing Elasticsearch and Kibana, you can havigate to http://localhost:9200/[26] and http://localhost:5601[27] to verify they are running.

They took maybe 10-20 seconds to start running, so be patient.

Put these commands in a bash script

I created a script setup-elastic-kibana-data.sh that does this.

Remember to make the script executable, eg `chmod +x setup-elastic-kibana.sh`

I also added some TODOs and some wait times, and some robustness such as removing any previously installed containers with the same name, as follows:

[25] https://www.docker.elastic.co/
[26] http://localhost:9200/
[27] http://localhost:5601

```
#!/bin/bash

#Usage - To install Elasticsearch and Kibana using docker on your local machine, run:
# ./setup-elastic-kibana.sh

#TODO Add robustness eg: to fail fast if Docker not installed

# download Elasticsearch & Kibana images
docker pull docker.elastic.co/elasticsearch/elasticsearch:7.2.0
docker pull docker.elastic.co/kibana/kibana:7.2.0

echo "FETCHED docker.elastic.co/elasticsearch/elasticsearch:7.2.0"
echo "FETCHED docker.elastic.co/kibana/kibana:7.2.0"

# Remove previously installed containers
docker stop elasticsearch_1
docker stop kibana_1

# Run elasticsearch
echo "Starting elasticsearch_1"

docker run -d --rm --name elasticsearch_1 -p 9200:9200 -p 9300:9300 -e "discovery.ty\
pe=single-node" docker.elastic.co/elasticsearch/elasticsearch:7.2.0 && sleep 20s

# Takes about 20 secs
# Or perform GET http://localhost:9200/ and assert on response?

# Run Kibana
echo "Starting kibana_1"
docker run -d --rm --name kibana_1 --link elasticsearch_1:elasticsearch -p 5601:5601\
 docker.elastic.co/kibana/kibana:7.2.0 && sleep 20s

echo '--------------- list of running containers ---------------------'
docker ps
echo '----------------------------------------------------------------'
```

The latest version of this script is available on github at ./setup-elastic-kibana.sh[28]

[28] https://github.com/purplebugs/fantasticelastic/blob/master/setup-elastic-kibana.sh

A note on docker compose

Note that originally I used Docker Compose[29], a more high level way of creating docker containers. This is described in the Appendix for future reference.

Including the data as part of this portable project

This will be covered in the following chapter.

Celebrate!

We now have the possibility to run one script that will install the tools to work on data, on different machines, making the project partly portable.

Coming up...

I will go through some options for how to include the data to be part of this portable project.

Without this we currently only have a script where the tools are portable, and not the data for creating visualisations.

[29] https://docs.docker.com/compose/

Including the data as part of this portable project

Add the data in an automated way to the script created in the previous chapter, in order to have a portable way to install both the tools and the data for creating visualisations.

The steps we will take

There are two parts to this.

The first part, figuring out the commands to run, has been moved to the Appendix.

The second part, adding the commands to the bash script, is covered in this chapter.

The reason is so that I can run one command that will result in installing elasticsearch, kibana, and data being imported

1. Create a .json file with the JSON documents
2. Add the bulk create data command to the script
3. Run the script
4. Check running the script now automatically creates Elasticsearch, Kibana, with the data available
5. Alternatively, get the installer script from github

Create a .json file with the JSON documents

This file is also available in github[30]

Create this file and save it in the same directory the script resides in.

[30] https://github.com/purplebugs/fantasticelastic/blob/master/eurovision_winners.json

```
{"index":{"_id":"1"}}
{"year": "2019-05-18","country": "Netherlands", "song": "Arcade", "performer": "Dunc\
an Laurence", "language": "English" }
{"index":{"_id":"2"}}
{"year":"2018-05-12","country":"Israel","song":"Toy","performer":"Netta","language":\
"English"}
{ "index" : { "_id" : "3" } }
{"year":"2017-05-13","country":"Portugal","song":"Amar pelos dois","performer":"Salv\
ador Sobral","language":"Portugese"}
{ "index" : { "_id" : "4" } }
{"year":"2016-05-14","country":"Ukraine","song":1944,"performer":"Jamala","language"\
:"English"}
{ "index" : { "_id" : "5" } }
{"year":"2015-05-23","country":"Sweden","song":"Heroes","performer":"Måns Zelmerlöw"\
,"language":"English"}
{ "index" : { "_id" : "6" } }
{"year":"2014-05-10","country":"Austria","song":"Rise Like a Phoenix","performer":"C\
onchita Wurst","language":"English"}
{ "index" : { "_id" : "7" } }
{"year":"2013-05-18","country":"Denmark","song":"Only Teardrops","performer":"Emmeli\
e de Forest","language":"English"}
{ "index" : { "_id" : "8" } }
{"year":"2012-05-26","country":"Sweden","song":"Eurphoria","performer":"Loreen","lan\
guage":"English"}
{ "index" : { "_id" : "9" } }
{"year":"2011-05-14","country":"Azerbaijan","song":"Running Scared","performer":"Ell\
 & Nikki","language":"English"}
{ "index" : { "_id" : "10" } }
{"year":"2010-05-29","country":"Germany","song":"Satellite","performer":"Lena","lang\
uage":"English"}
```

Add the bulk create data command to the script

Here is the extra command to add the data. Paste this command at the end of the previously created script for installing and running Elasticsearch and Kibana.

```
curl -s -H "Content-Type: application/json" -XPOST http://localhost:9200/eurovision_\
winners/docs/_bulk --data-binary "@eurovision_winners.json"'
```

Optional

If you like, you can try running the command in the console first. Be sure to have Elasticsearch up and running before doing so.

Run the script

Now run the full script. My version is called `./setup-elastic-kibana-data-bulk.sh`

Alternatively, get the installer script from github

Grab the latest installer from here github[31]

The contents of it look like:

```
#!/bin/bash

#Usage - To install Elasticsearch and Kibana using docker on your local machine, run:
# ./setup-elastic-kibana.sh

#TODO Add robustness eg: to fail fast if Docker not installed

# download Elasticsearch & Kibana images
docker pull docker.elastic.co/elasticsearch/elasticsearch:7.2.0
docker pull docker.elastic.co/kibana/kibana:7.2.0

echo "FETCHED docker.elastic.co/elasticsearch/elasticsearch:7.2.0"
echo "FETCHED docker.elastic.co/kibana/kibana:7.2.0"

# Remove previously installed containers
docker stop elasticsearch_1
docker stop kibana_1

# Run elasticsearch
echo "Starting elasticsearch_1"

docker run -d --rm --name elasticsearch_1 -p 9200:9200 -p 9300:9300 -e "discovery.ty\
pe=single-node" docker.elastic.co/elasticsearch/elasticsearch:7.2.0 && sleep 20s

# Takes about 20 secs
```

[31] https://github.com/purplebugs/fantasticelastic/blob/master/setup-elastic-kibana-data-bulk.sh

```
# Or perform GET http://localhost:9200/ and assert on response?

# Run Kibana
echo "Starting kibana_1"
docker run -d --rm --name kibana_1 --link elasticsearch_1:elasticsearch -p 5601:5601\
 docker.elastic.co/kibana/kibana:7.2.0 && sleep 20s

echo '--------------- list of running containers ----------------------'
docker ps
echo '-----------------------------------------------------------------'


#docker exec -it elasticsearch_1 bash <--- Do not need to be in interactive mode for\
 this! Can simply run the bash commands as follows:

echo "About to create the index: eurovision_winners..."
echo "... add bulk add the data from an external file: eurovision_winners.json ..."
echo "... where each row is known as a document in elasticsearch, and setting the do\
cument indexes as the numbers in the path"

curl -s -H "Content-Type: application/json" -XPOST http://localhost:9200/eurovision_\
winners/docs/_bulk --data-binary "@eurovision_winners.json"
```

Check the script automatically created Elasticsearch, Kibana, with the data available

1. Navigate to `http://localhost:5601`
2. In Devtools run `GET eurovision_winners/_search`
3. The data should be shown in the response

Alternative options

Alternatively, importing a .json file in one cURL commands as per the Appendix: Using devtools to get commands for importing data

Celebrate!

[TODO] write a summary of what was achieved

Coming up…

[TODO]

Including the dashboard as part of this portable project

[TODO]

Import the dashboard using the UI

A precondition is that the dashboard created earlier was exported.

You will need the exported file `export.ndjson` to import in the following step.

The steps to import are:

1. Go to Management > Kibana > Saved Objects
2. Select the dashboard to export
3. Select Import > Automatically overwrite all saved objects
4. Management > Kibana > Index Pattern > Define Index Pattern > enter “eurovision*” so it matches “eurovision_winners” Index
5. Select Year as Time filter > Create Index pattern
6. Verify it imported by going to the dashboards and opening it up
7. Tip: Remember to filter on at least a the last few years to see any data

[TODO - automate creating index pattern]

Reference: Managing Saved Objects[32]

Experimental: Import the dashboard using an api

At time of writing, there is an experimental api that allows exporting and importing of dashboards

Source: dashboard-import-api-export[33]

Example dashboard url, snip: `http://localhost:5601/app/kibana#/dashboard/813b7920-9051-11e9-9fea-3dbdb6`
`g=(filters:!....`

[32] https://www.elastic.co/guide/en/kibana/current/managing-saved-objects.html
[33] https://www.elastic.co/guide/en/kibana/current/dashboard-import-api-export.html

1. Get the dashboard id by navigating to the dashbaord and copying the id from url which are all characters from after dashboard/ up until the question mark. In this example it is `813b7920-9051-11e9-9fea-3dbdb697437e`
2. Run the command to exports all saved objects associated with and including the dashboard, substituting the id to be the actual id `curl -X GET "http://localhost:5601/api/kibana/dashboards/export?d -H 'kbn-xsrf: true' | tee dashboard.log`
3. The output will be both printed on the screen and saved to the dashboard.log file
4. Import the dashboard by running `POST api/kibana/dashboards/import?exclude=index-pattern`
5. Use the complete response body from the construct the import cURL and paste it when prompted

Celebrate!

[TODO] write a summary of what was achieved

Coming up...

[TODO]

What's Next?

I want the world to see these gorgeous dashboards and charts.

Product managers all over the world have development teams that use elastic to incorporate charts, graphs and dashboards into their software and applications.

So now it's my turn to pop on my product manager hat, have a little brainstorming session with myself and decide how to share this info.

What kind of product should I create

[TO DO]

Creating a demo with the dashboard

A great way to enhance a sales pitch for products that the company envisions however has not quite completed (and we are pretty much there all the time in software development, right?) is to wow them with data that the potential customer can easily digest.

In the same way, I am going to try to wow you, dear reader, with a demo of this awesome Eurovision data nerd fan dashboard, which is not yet a software product, just some cool charts and bells and whistles.

[TO DO]

… will go into more detail on this in my follow up book: "Fantastic Elastic 2: Demo-tastic Deal Sealer"

Creating a website with the dashboard

[TO DO]

Creating an app with the dashboard

[TO DO]

… will go into more detail on this in my follow up book "Fantastic Elastic 3: App-tastic ways to show off data visualisations"

Conclusion

[TODO]

Canvas VS Kibana

Since I started my journey, I have learned that Elasticsearch's Canvas is the tool of choice - just a menu click away really - for formatting the visualisations created in Kibana for business presentations, whereas Kibana is more a tool for digging, analysing and creating the standalone charts and graphs. Canvas has options such as the ability to drag in a logo and export the visualisations to be used in slide presentations.

Credits

[WIP]

1. Girls Can Do IT workshop: contact them…
2. N who talked about her project, contact her…
3. Z who gave tips on technical writing, contact her…
4. K - mentor tips…. ask him
5. J - docker inspiration… ask him
6. Who else??

Rough notes

How to store Kibana dashboard in docker container[34]

You can export your dashboard into a JSON file and create a script that sends a request to Kibana's "import dashboard" API endpoint with this file as a payload. Here's an example of how docker-compose can run a script like this: https://github.com/elastic/stack-docker/blob/master/docker-compose.yml#L116 180. Here is some information on the "import dashboard" API: https://github.com/elastic/kibana/issues/14872 81.

Have another 'service' in docker compose that depends_on the elasticsearch image, and this 'service' is actually just a script that uses curl to insert the data[35]

Have a short lived container outside Kibana[36]

Example in github: elastic-stack-6.3[37]

[34] https://discuss.elastic.co/t/how-to-store-kibana-dashboard-in-docker-container/137459
[35] https://discuss.elastic.co/t/how-to-create-elasticsearch-docker-container-create-indexes-and-seed-data-in-one-step/167561
[36] https://discuss.elastic.co/t/not-able-to-run-the-script-to-create-dashboard-in-docker-kibana/138999
[37] https://github.com/swarmee/projects/tree/master/elastic-stack-6.3

Appendix: Docker Compose

While learning and writing, I first tried to use Docker Compose to create the Docker setup.

However, I wanted to have a better understanding of how docker worked, so I decided to skip this and use docker in its plain format instead. That is what I described earlier in the book.

In any case, I am keeping this docker compose installation info here in the appendix for future reference.

What is Docker Compose

Docker Compose[38] is a tool for defining and running multi-container Docker applications

It is a more high level way of creating docker containers.

The steps I will take

1. Install Docker compose to create the docker image - https://docs.docker.com/compose/install/
2. Populate docker-compose.yml with with elasticsearch and kibana
3. Run the image
4. Check Elasticsearch and Kibana are up and running

Install Docker compose to create the docker image

Source: https://docs.docker.com/compose/install/

I did this on linux by doing the following

1. `sudo curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
2. Apply executable permissions to the binary: `sudo chmod +x /usr/local/bin/docker-compose`
3. Test the installation: `docker-compose --version`
4. Result: `docker-compose version 1.24.0, build 1110ad01`

... or install in one command: `sudo apt install docker-compose`

To uninstall for any reason: `sudo rm /usr/local/bin/docker-compose`

[38] https://docs.docker.com/compose/

Populate docker-compose.yml with elasticsearch and kibana

Based on: elasticsearch-kibana-with-docker-compose[39]

[TODO contact to give credits]

Populate docker-compose.yml with the following configuration:

```
# ./docker-compose.yml

version: '3'

services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:6.8.0
    environment:
      - cluster.name=docker-cluster
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
    ulimits:
      memlock:
        soft: -1
        hard: -1
    ports:
      - "9200:9200"
  kibana:
    image: docker.elastic.co/kibana/kibana:6.8.0
    ports:
      - "5601:5601"
```

Run the Docker image

```
sudo docker-compose up -d
```

Note the first time this can take up to a few minutes to download the images

Check Kibana and elasticsearch are up and running

You might need to wait a few moments

[39] https://alysivji.github.io/elasticsearch-kibana-with-docker-compose.html

1. Elasticsearch is available at http://localhost:9200/
2. Kibana UI is up at http://localhost:5601/

Troubleshooting: check the docker images are running using

```
docker ps
```

I should see something like

```
CONTAINER ID        IMAGE                                                 COMMAND   \
               CREATED             STATUS              PORTS                         \
      NAMES
6202695f3e8c        docker.elastic.co/elasticsearch/elasticsearch:6.3.2   "/usr/loca\
l/bin/dock…"   5 seconds ago       Up 3 seconds        0.0.0.0:9200->9200/tcp, 9300/\
tcp   dev_elasticsearch_1
cf2c02756264        docker.elastic.co/kibana/kibana:6.3.2                 "/usr/loca\
l/bin/kiba…"   5 seconds ago       Up 3 seconds        0.0.0.0:5601->5601/tcp        \
      dev_kibana_1
```

Note - to quickly destroy this environment for any reason, use:

```
docker-compose down
```

and use the following again to run the image again when needed. It will go faster after the first time as it will not need to download large files from scratch again

```
docker-compose up -d
```

Appendix: Using devtools to get commands for importing data

Figure out the commands to run

For several steps I use Kibana Dev Tools > Console to test out the commands on a new index with the same data

For each command that requires saving the cURL command, hover over the tool icon and "copy as cURL" to save the command to be used in the docker image later on.

1. Create the index via command line, and save the "cURL" command
2. Set mappings for the fields, and save the "cURL" command
3. Convert the .csv file to json
4. Load the data set via command line, and save the "cURL" command
5. Alternatively, import the data using explicit curl commands, without using a .json file
6. Check the data is available

Create the index via command line, and save the "cURL" command

1. In Dev Tools, enter the following and execute: `PUT eurovision_winners`
2. Click on the tool icon and copy as cURL
3. Paste the text into a notepad for later use. It will be: `curl -XPUT "http://elasticsearch:9200/eurovision_winners"`

Response will be

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "eurovision_winners"
}
```

Set mappings for the fields, and save the “cURL” comannd

There are a couple of steps

1. Get the existing mappings
2. Use the result to create a command to set the mappings

Get existing mappings for the fields

Before I load the eurovision_winners and logs data sets, I must set up mappings for the fields. Mappings divide the documents in the index into logical groups and specify the characteristics of the fields.

Source: tutorial-load-dataset[40]

1. In console dev tools, get mappings first: `GET eurovision_winners/_mapping`
2. Click on the tool icon and copy as cURL
3. Paste the text into a notepad for later use. It will be: `curl -XGET "http://localhost:9200/eurovision_-winners/_mapping`
4. Copy the Json result into notepad someplace for later use. It will look like the following:

Json result:

```
{
  "eurovision_winners" : {
    "mappings" : {
      "_meta" : {
        "created_by" : "ml-file-data-visualizer"
      },
      "properties" : {
        " country" : {
          "type" : "keyword"
        },
        " language" : {
          "type" : "keyword"
        },
        " performer" : {
          "type" : "keyword"
        },
```

[40] https://www.elastic.co/guide/en/kibana/current/tutorial-load-dataset.html

```
        " song" : {
          "type" : "keyword"
        },
        "year" : {
          "type" : "date"
        }
      }
    }
  }
}
```

Set mappings for the fields

1. Using Dev Tools type the command and do NOT run it: `PUT eurovision_winners/_mapping`
2. Modify the mappings reponse to only keep the "properties" key value pairs, and paste it below the command in devtools
3. Run the command - it should look like the command below
4. Copy the cURL command and store it in a notepad for later use

Dev Tools command

```
PUT eurovision_winners/_mapping
{
      "properties" : {
        " country" : {
          "type" : "keyword"
        },
        " language" : {
          "type" : "keyword"
        },
        " performer" : {
          "type" : "keyword"
        },
        " song" : {
          "type" : "keyword"
        },
        "year" : {
          "type" : "date"
        }
      }
}
```

cUrl command

```
curl -XPUT "http://localhost:9200/eurovision_winners/_mapping" -H 'Content-Type: app\
lication/json' -d'
{
      "properties" : {
        " country" : {
          "type" : "keyword"
        },
        " language" : {
          "type" : "keyword"
        },
        " performer" : {
          "type" : "keyword"
        },
        " song" : {
          "type" : "keyword"
        },
        "year" : {
          "type" : "date"
        }
      }
}'
```

Convert the .csv file to json

I did this by

1. using an online csv to json tool[41], which resulted in the JSON below
2. saving the JSON to a file named `eurovision_winners.json`
3. Then modifying it to have the following structure, as per https://www.elastic.co/guide/en/elasticsearch/referen bulk.html[42]
4. Note: ensure the final line is a new line, and the file is formatted as below

This file is also available in github[43] “‘ {“index”:{“_id”:”1”}} {“year”: “2019-05-18”,”country”: “Nether-lands”, “song”: “Arcade”, “performer”: “Duncan Laurence”, “language”: “English” } {“index”:{“_-id”:”2”}} {“year”:”2018-05-12”,”country”:”Israel”,”song”:”Toy”,”performer”:”Netta”,”language”:”English”} {“index” : { “_id” : “3” } } {“year”:”2017-05-13”,”country”:”Portugal”,”song”:”Amar pelos dois”,”performer”:”Salvador Sobral”,”language”:”Portugese”} { “index” : { “_id” : “4” } } {“year”:”2016-05-14”,”country”:”Ukraine”,”song”:1944,”perfor { “index” : { “_id” : “5” } } {“year”:”2015-05-23”,”country”:”Sweden”,”song”:”Heroes”,”performer”:”Måns

[41] https://www.csvjson.com/csv2json
[42] https://www.elastic.co/guide/en/elasticsearch/reference/7.1/docs-bulk.html
[43] https://github.com/purplebugs/fantasticelastic/blob/master/eurovision_winners.json

Zelmerlöw”,”language”:”English”}{“index”:{“_id”:“6”}}{“year”:”2014-05-10”,”country”:”Austria”,”song”:”Rise Like a Phoenix”,”performer”:”Conchita Wurst”,”language”:”English”} { “index” : { “_id” : “7” } } {“year”:”2013-05-18”,”country”:”Denmark”,”song”:”Only Teardrops”,”performer”:”Emmelie de Forest”,”language”:”English”}{“index”:{“_id”:“8”}}{“year”:”2012-05-26”,”country”:”Sweden”,”song”:”Eurphoria”,”perfo{“index”:{“_id”:“9”}}{“year”:”2011-05-14”,”country”:”Azerbaijan”,”song”:”Running Scared”,”performer”:”Ell & Nikki”,”language”:”English”}{“index”:{“_id”:“10”}}{“year”:”2010-05-29”,”country”:”Germany”,”song”:”Satellite”,

```
## Load the data set via command line, and save the "cURL" command

Ensure I am in the correct directory, then on the bash command line run: `curl -s -H\
 "Content-Type: application/json" -XPOST http://localhost:9200/eurovision_winners/do\
cs/_bulk --data-binary "@eurovision_winners.json"`


## Another way to import in bulk, without have to create a separate .json file

[TODO : create index with correct mapping for date field first]

This bulk command is the equivalent of doing the following, without having an extern\
al file to the your data in:
```

Create the index curl -XPUT “http://elasticsearch:9200/eurovision_winners”

Add the data, where each row is known as a “document” in elasticsearch, setting the document indexes as the numbers in the path

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/1” -H ‘Content-Type: application/j-son’ -d’{ “year”: “2019-05-18”,”country”: “Netherlands”, “song”: “Arcade”, “performer”: “Duncan Laurence”, “language”: “English” }

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/2” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2018-05-12”,”country”:”Israel”,”song”:”Toy”,”performer”:”Netta”,”language”:”English”}

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/3” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2017-05-13”,”country”:”Portugal”,”song”:”Amar pelos dois”,”performer”:”Salvador Sobral”,”language”:”Portugese”}

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/4” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2016-05-14”,”country”:”Ukraine”,”song”:1944,”performer”:”Jamala”,”language”:”English”}

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/5” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2015-05-23”,”country”:”Sweden”,”song”:”Heroes”,”performer”:”Måns Zelmerlöw”,”language”:”Englis

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/6” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2014-05-10”,”country”:”Austria”,”song”:”Rise Like a Phoenix”,”performer”:”Conchita Wurst”,”language”:”English”}

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/7” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2013-05-18”,”country”:”Denmark”,”song”:”Only Teardrops”,”performer”:”Emmelie de Forest”,”language”:”English”}

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/8” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2012-05-26”,”country”:”Sweden”,”song”:”Eurphoria”,”performer”:”Loreen”,”language”:”English”}

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/9” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2011-05-14”,”country”:”Azerbaijan”,”song”:”Running Scared”,”performer”:”Ell & Nikki”,”language”:”I

curl -XPUT “http://elasticsearch:9200/eurovision_winners/_doc/10” -H ‘Content-Type: application/j-son’ -d’ {“year”:”2010-05-29”,”country”:”Germany”,”song”:”Satellite”,”performer”:”Lena”,”language”:”English”}
“‘

Check the data is available

1. Navigate to `http://localhost:5601`
2. In Dev Tools run `GET eurovision_winners/_search`
3. The data should be shown in the response