MONASH University

# Introduction to Visualisation and Storytelling

# What is visualisation?

Data visualisation:

- shows information found in (large sets of) data so a human can understand it
- underlying data: Database, big data, tabular data, most often numeric

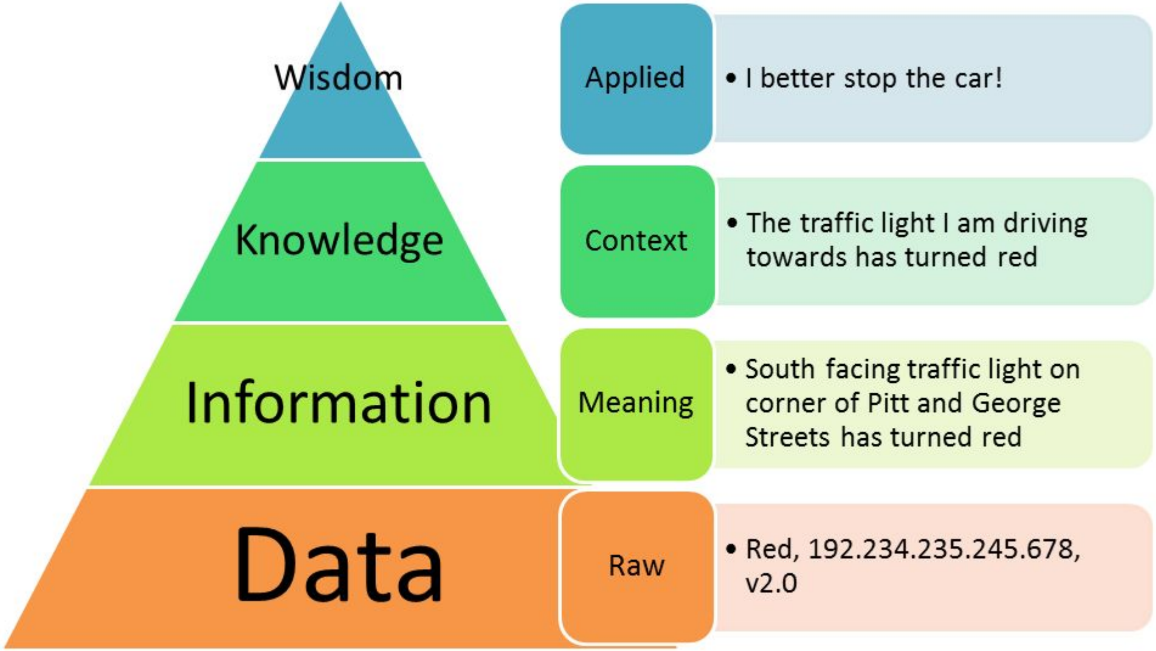
Scientific visualisation

- (3D) Modelling or simulation of scientific data
- Subset of computer graphics and data vis
- Goal is to help scientists gain insights/information about their data → Highly specialised

# Why visualisation

**Gain wisdom from data.**

**Enable Action!**



Wisdom — Applied
- I better stop the car!

Knowledge — Context
- The traffic light I am driving towards has turned red

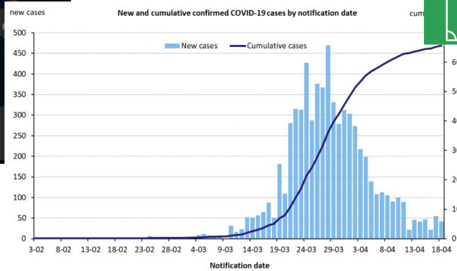
Information — Meaning
- South facing traffic light on corner of Pitt and George Streets has turned red

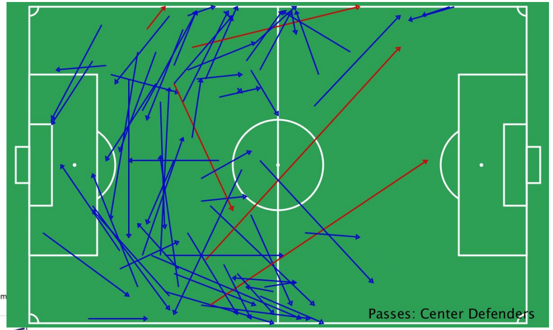
Data — Raw
- Red, 192.234.235.245.678, v2.0

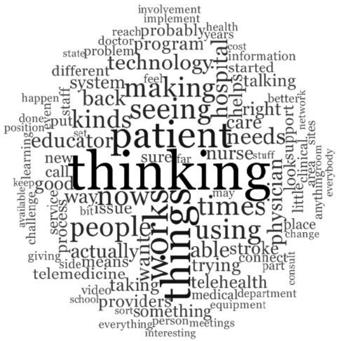
© 2011 Angus McDonald

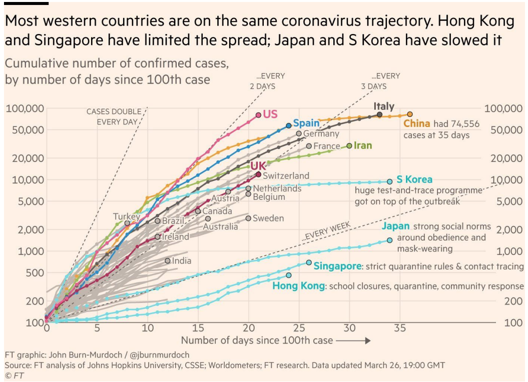
# Types of 2D visualisation

NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
AUST
ANTARCTICA


New and cumulative confirmed COVID-19 cases by notification date






Most western countries are on the same coronavirus trajectory. Hong Kong and Singapore have limited the spread; Japan and S Korea have slowed it

Cumulative number of confirmed cases, by number of days since 100th case

FT graphic: John Burn-Murdoch / @jburnmurdoch
Source: FT analysis of Johns Hopkins University, CSSE; Worldometers; FT research. Data updated March 26, 19:00 GMT
© FT

# Types of media

**Where to present the visualisation?**

- Beginners:
  - Offline presentations such as slide shows
  - Printed media such as flyers, posters, newspapers

- Intermediate/Advanced
  - **Browser-based**
  - standalone applications tailored for the vis
  - VR/AR

  → Interactivity

# Audience

- previous knowledge of topic?
- critical or supporting of your data?
- Do they trust you and your data?

  → Presentation can influence this

- How serious is the topic?

  → Artistic style should reflect all of this

- the more the vis will be exposed to public, the less complicated it should be

# Audience

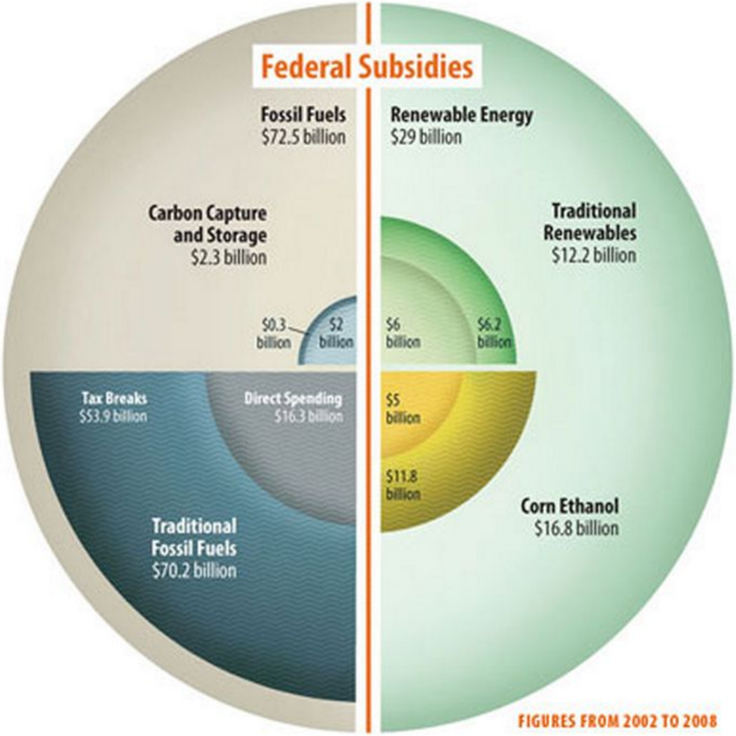
https://www.abc.net.au/news/2020-03-26/coronavirus-covid19-global-spread-data-explained/12089028

Scroll down for great vis for exponential growth, easily understandable for people with no mathematical background, as opposed to the graphs more commonly seen.

# Bad examples for Charts

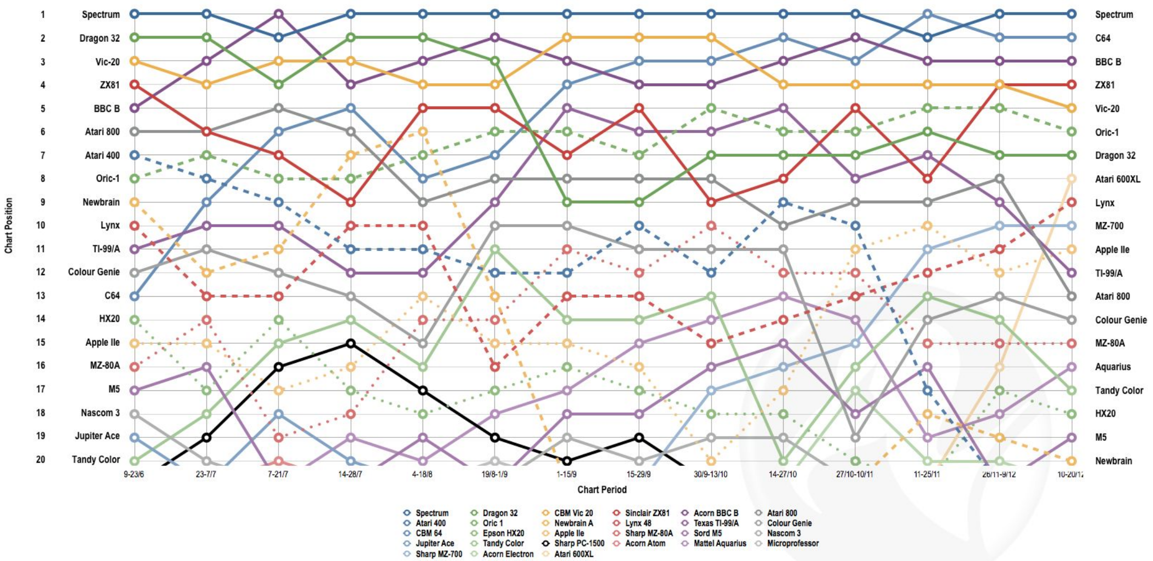
## Not understandable

- What do the several colour shades mean?
- the sums don’t add up to the distribution of parts of the pie chart
- several labels in the same colour



# Bad examples for Charts

## Too complicated/too much (charts of computer brands over a period of time)

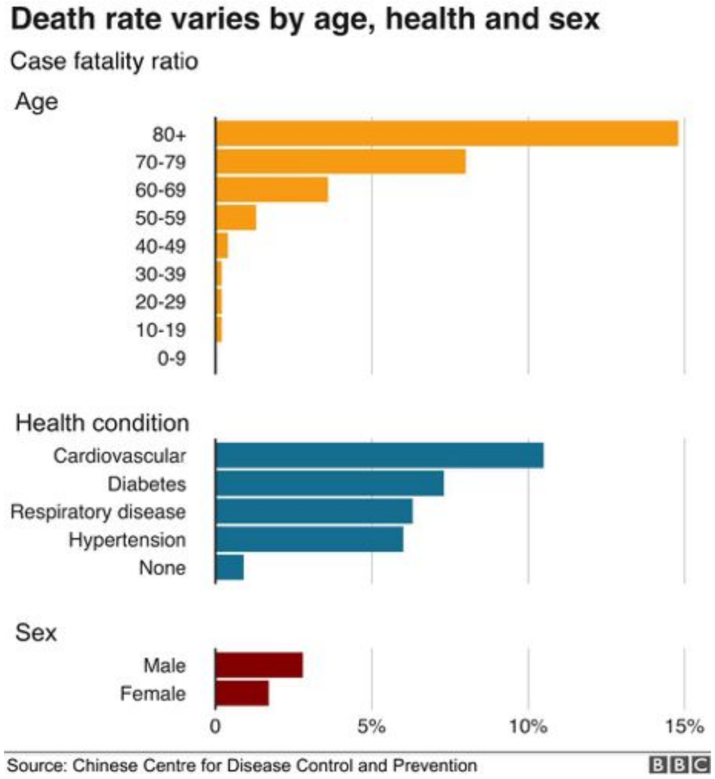


- literally TMI
- better to emphasize a few computer brands and greying out the rest
- could make brands selectable to highlight them → interactive

# Bad examples for Charts

## Misleading

- The x-axis ends at 15%. But the bar extending the full width is perceived as 100%



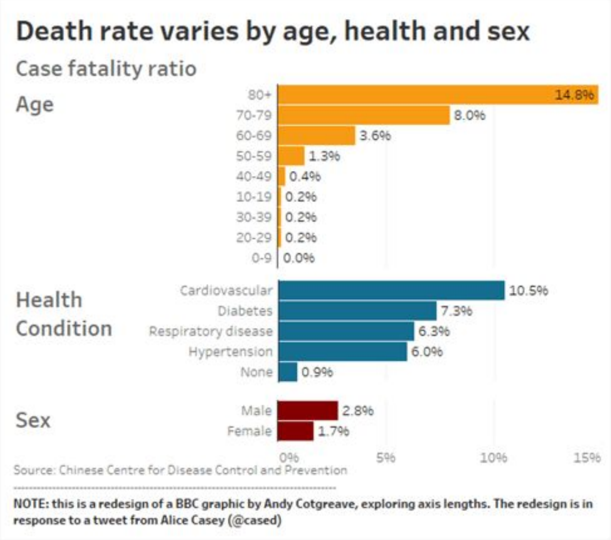
Death rate varies by age, health and sex

Case fatality ratio

Source: Chinese Centre for Disease Control and Prevention

# Bad examples for Charts

## Misleading: Better solution

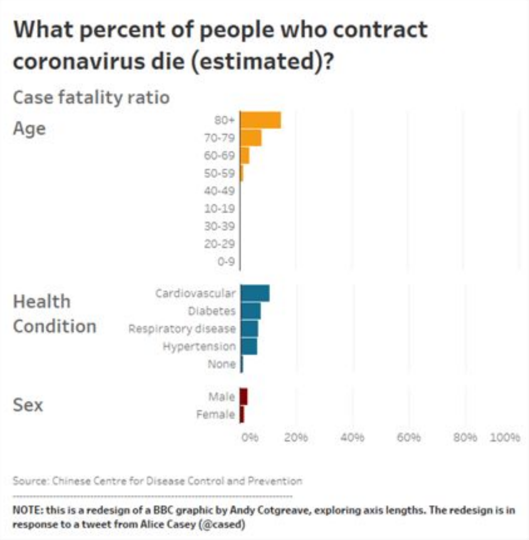


Death rate varies by age, health and sex

Case fatality ratio

Source: Chinese Centre for Disease Control and Prevention

NOTE: this is a redesign of a BBC graphic by Andy Cotgreave, exploring axis lengths. The redesign is in response to a tweet from Alice Casey (@cased)



What percent of people who contract coronavirus die (estimated)?

Case fatality ratio

Source: Chinese Centre for Disease Control and Prevention

NOTE: this is a redesign of a BBC graphic by Andy Cotgreave, exploring axis lengths. The redesign is in response to a tweet from Alice Casey (@cased)

# Storytelling

Historically, people have always told stories.

The best story tellers could create images in people’s head - they enabled them to visualise the story.

# Storytelling

Humans understand data a lot better through images.

Humans also remember a compelling story a lot better than a list of facts.

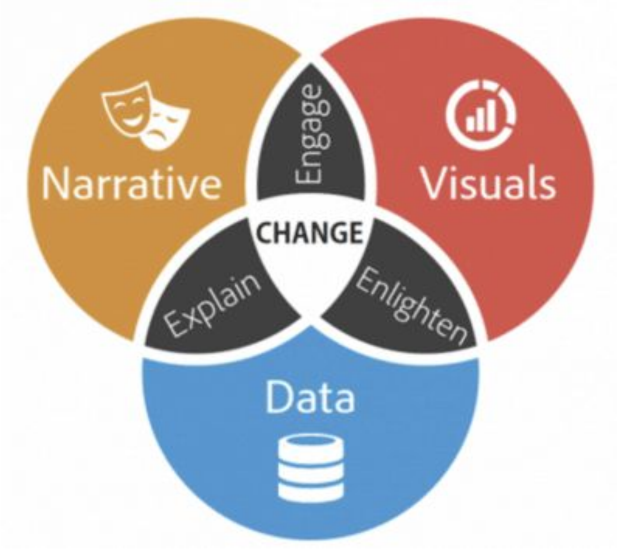
A story can evoke emotions, and show facts from different points of view, to help people gain understanding.

This can be very powerful and evoke

**Change**...

- ...of behaviour
- ...of thinking
- ...of opinion
- ...of laws
- ...

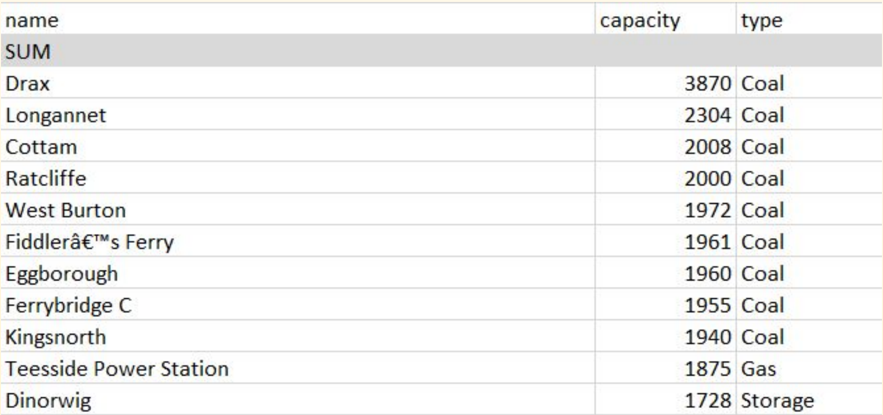
# Data visualisation and Storytelling



# From raw data to a 2D visualisation

# Example data - Excel

**Data: A few tables, one for each year, contain data about power plant capacities for plants in the UK, as well as their types (coal, solar, etc.)**



| name | capacity | type |
|---|---|---|
| SUM | | |
| Drax | 3870 | Coal |
| Longannet | 2304 | Coal |
| Cottam | 2008 | Coal |
| Ratcliffe | 2000 | Coal |
| West Burton | 1972 | Coal |
| Fiddlerâ€™s Ferry | 1961 | Coal |
| Eggborough | 1960 | Coal |
| Ferrybridge C | 1955 | Coal |
| Kingsnorth | 1940 | Coal |
| Teesside Power Station | 1875 | Gas |
| Dinorwig | 1728 | Storage |

(The first few rows of the table)

MONASH University

# Example data - how to visualise?

**Questions**

- What information is there? → Name of power plants with their capacities and types

- Is all the information useful? → Name of power plants not useful, disregard for now

- The interesting bit: In each year, how much capacity did each **type** of power plant have?

  → Data wrangling necessary to find out the sum of capacity

  for each type.

# Data Wrangling

### What does it mean?

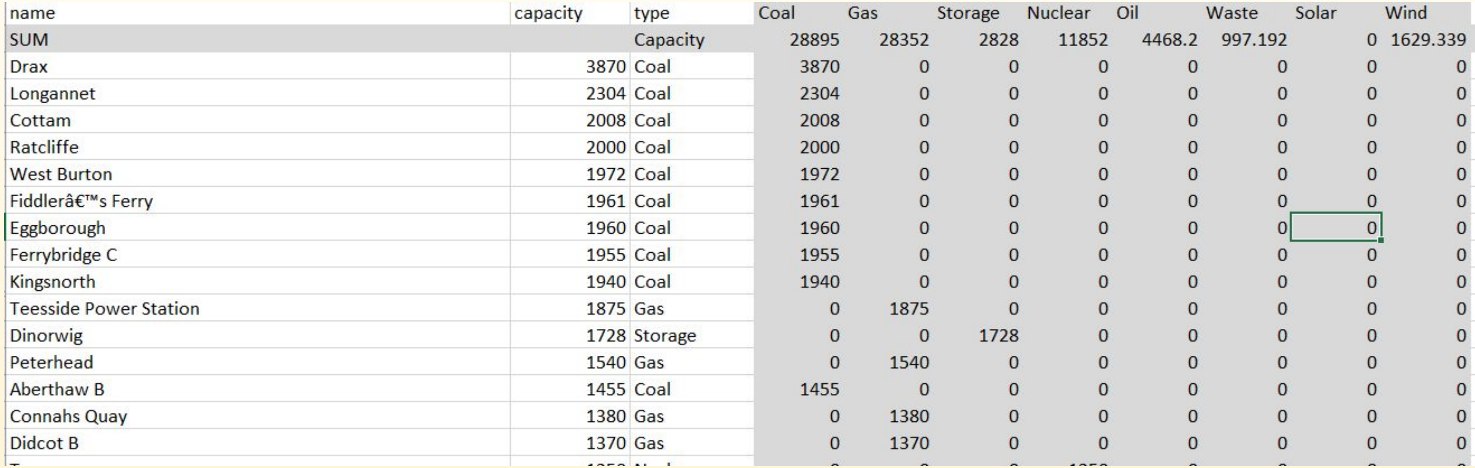
- unite all potentially heterogeneous data sources into a format usable by the means of visualisation
- remove unneeded bits of the data
- either remove incomplete data or predict the missing parts. This should be a very reliable prediction, and needs to be *made clear in the end product.*

### How to do it?

- Scripting or Programming
- Python very popular for such tasks
- SQL for databases
- manually if the dataset is not too large

# Example data - how to visualise?

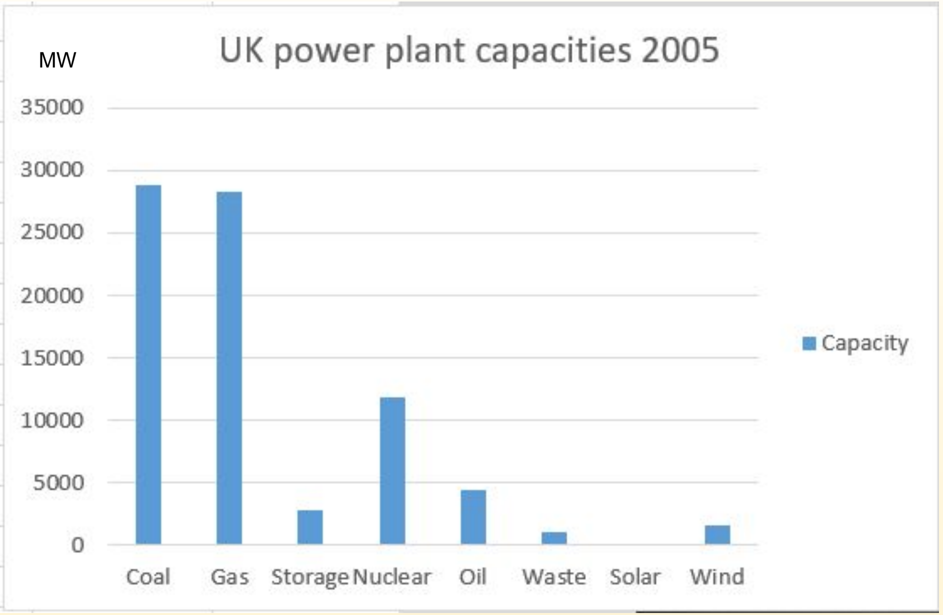
## Added columns after data wrangling



| name | capacity | type | Coal | Gas | Storage | Nuclear | Oil | Waste | Solar | Wind |
|---|---|---|---|---|---|---|---|---|---|---|
| SUM | | Capacity | 28895 | 28352 | 2828 | 11852 | 4468.2 | 997.192 | 0 | 1629.339 |
| Drax | 3870 | Coal | 3870 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Longannet | 2304 | Coal | 2304 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Cottam | 2008 | Coal | 2008 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Ratcliffe | 2000 | Coal | 2000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| West Burton | 1972 | Coal | 1972 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Fiddlerâ€™s Ferry | 1961 | Coal | 1961 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Eggborough | 1960 | Coal | 1960 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Ferrybridge C | 1955 | Coal | 1955 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Kingsnorth | 1940 | Coal | 1940 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Teesside Power Station | 1875 | Gas | 0 | 1875 | 0 | 0 | 0 | 0 | 0 | 0 |
| Dinorwig | 1728 | Storage | 0 | 0 | 1728 | 0 | 0 | 0 | 0 | 0 |
| Peterhead | 1540 | Gas | 0 | 1540 | 0 | 0 | 0 | 0 | 0 | 0 |
| Aberthaw B | 1455 | Coal | 1455 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Connahs Quay | 1380 | Gas | 0 | 1380 | 0 | 0 | 0 | 0 | 0 | 0 |
| Didcot B | 1370 | Gas | 0 | 1370 | 0 | 0 | 0 | 0 | 0 | 0 |

SUM row adding up all the types after offsetting the capacity value in new columns according to their type.

# Example data - how to visualise?

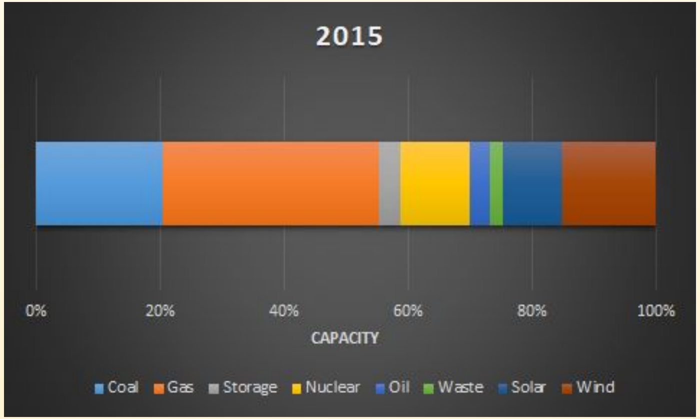


The values are comparable and relatable to each other. But with the raw numbers, most people won't be able to gain much information from this.

# Example data - how to visualise?

## Improvement

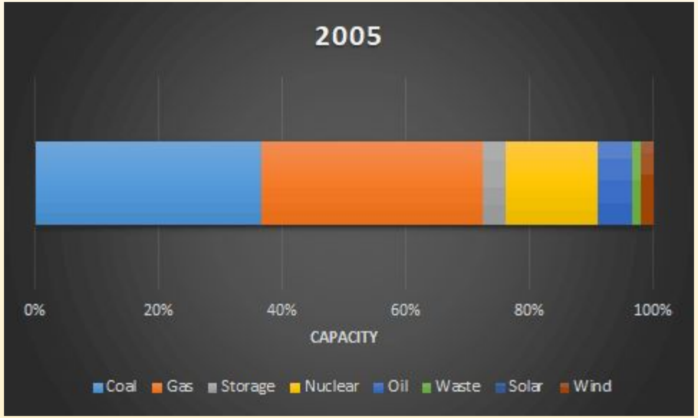
The capacity values are in direct relation to each other and they add up together to a whole. This can be displayed in a stacked bar to visualise how much of the “whole” each part takes.



# Example data - how to visualise?

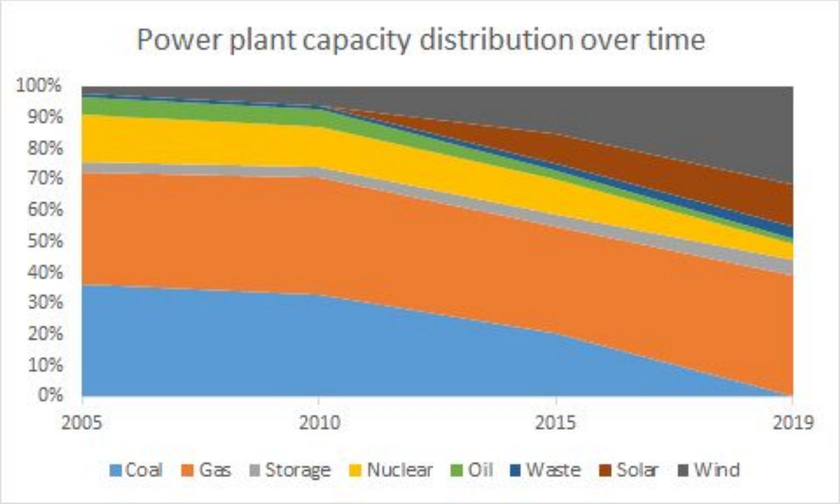
## Animation

- Charts can be combined in an animation when time can be added as a third dimension
- most photo apps can do this on any modern phone, or online tools
- very easy way to create animated vis with information gain!



# Example data - how to visualise?

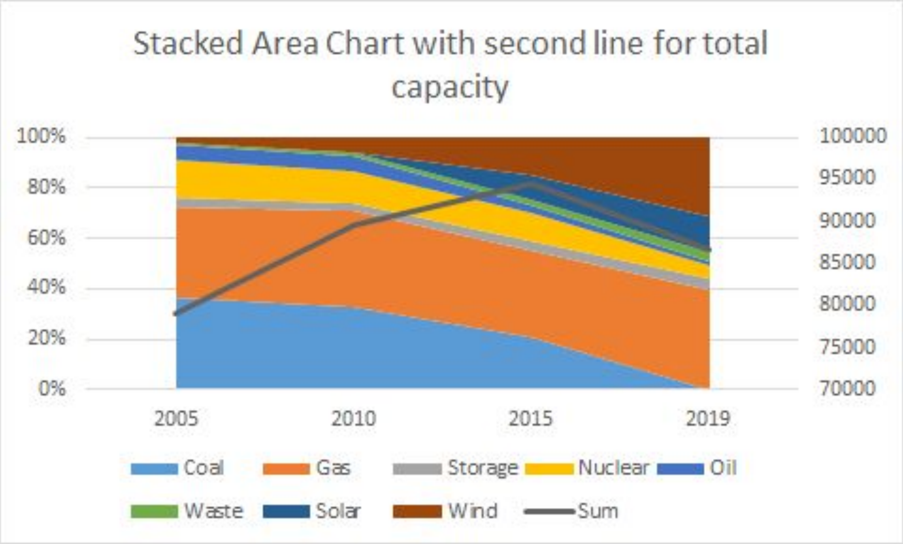
Alternative



Stacked area chart with times on the x-axis. To not distort the information, it is important the steps on the x-axis are equidistant

# Example data - how to visualise?

Alternative



“Sum” line shows that in 2019 the total capacity overall was lower than in 2018, which may be important information.

# Where to move on from here

Excel’s chart generation is nice for mostly simple tabular data. To visualise more complex data - or make more beautiful visualisations - other means are necessary.

- Python visualisation
- **Web visualisation**
- R
- Scientific visualisation
- VR/AR
- ...

# Web visualisation

# Why the web?

### Layout possibilities

The layout of the website is completely up to the developer. Menus, scrolling parts of the site, animations, natural support for user interaction are all great benefits of web development.

There is a lot of freedom and it gives possibilities for unusual approaches as well. Regarding **storytelling,** websites naturally offer very easy and efficient ways to weave a story with imagery.

### Accessibility

### Libraries

There are a lot of javascript libraries to do on-runtime visualisations, and most visualisations created with other means can be displayed on a website as well.

# Web Visualisation - Required Skills

## Beginner

- CSS/HTML5

  Create a basic one-page website including previously created images of vis, animations, and text to present visualisations along with a narrative

# Web Visualisation - Required Skills

## Intermediate

- Javascript, DOM Manipulation, Anchors, onclick event, onscroll event
  - can create a scrolling page where predefined static events happen as the user reaches certain points
  - JS Libraries:  jQuery, bootstrap, many more…

    → QoL, easy page layout, ...
- JSON
  - [https://www.w3schools.com/whatis/whatis_json.asp](https://www.w3schools.com/whatis/whatis_json.asp)
  - preferred data format for JS applications
  - converters from other types of data to JSON exist
  - many JS libraries expect JSON input

# Web Visualisation - Required Skills

## Advanced - 1

- Package Manager NPM
  - easy to install libraries to your project, add dependencies, define automatic workflows
- MVC: Model view controller
  - Separate what the user sees, the *view*, from the data, which is the *model*.
  - enables dynamic websites with easier code maintenance
  - change on runtime without directly interfering with the HTML
  - Available libs: **ReactJS**, KnockoutJS, Angular, ...

# Web Visualisation - Required Skills

## Advanced - 2

- mapboxGL (for map visualisations)
  - items drawn on the map are organised in layers
  - offers filter functionality for each layer, so can be changed on runtime based on user input or other events

- ES6 - EcmaScript 6
  - https://www.w3schools.com/js/js_es6.asp
  - “clean” Javascript
  - ReactJS uses this per default
- TypeScript

# Possibilities of the web

A **lot**.

A couple of visualisation libraries, not complete:

- D3js - most popular and extensive JS library
- ChartJS
- ThreeJS - 2D and 3D animations
- Raw graphs works with tabular data rather than Json
- There are **heaps** more!

Potentially useful other libraries

- jQuery, bootstrap, animeJS,
- For complex sites, or any site that will change on runtime, use a Model View Controller such as ReactJS

# Example - Scrolly teller with Map

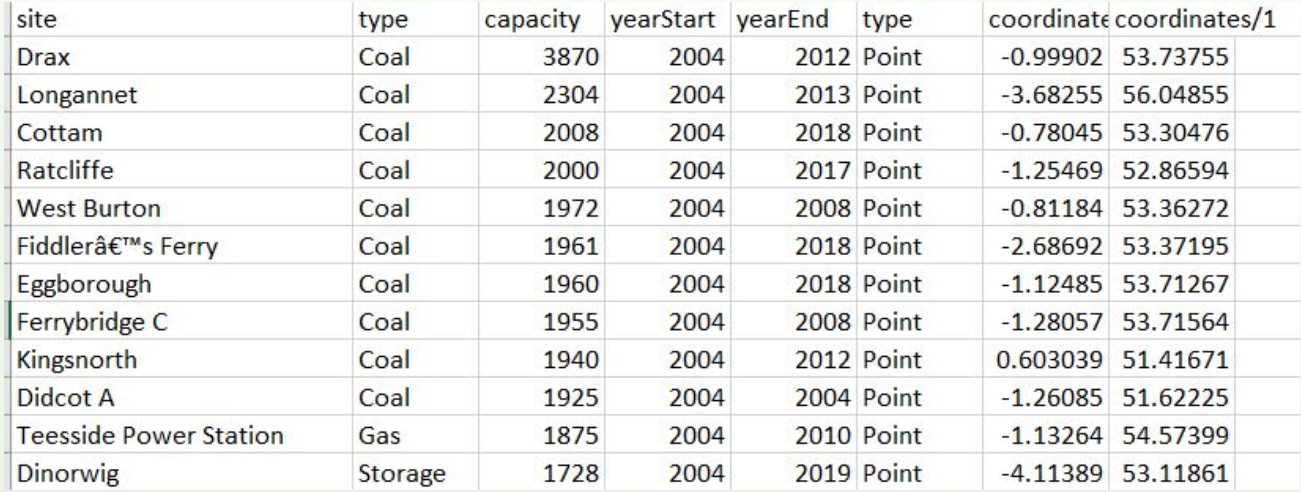
**What is a “Scrolly teller”?**

- Scroll through a website, reading a narrative, enhanced with imagery, videos, visualisations, etc..
- combines story telling with web interaction
- became popular in recent years, especially on News sites

# Example - Scrollyteller with Map

**Assume new data for our previous dataset**

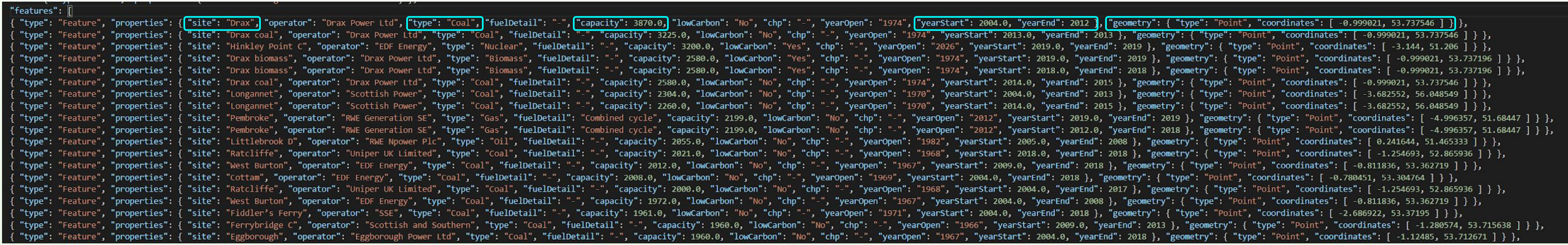
Now we also have location data for each power plant



| site | type | capacity | yearStart | yearEnd | type | coordinate | coordinates/1 |
|---|---|---|---|---|---|---|---|
| Drax | Coal | 3870 | 2004 | 2012 | Point | -0.99902 | 53.73755 |
| Longannet | Coal | 2304 | 2004 | 2013 | Point | -3.68255 | 56.04855 |
| Cottam | Coal | 2008 | 2004 | 2018 | Point | -0.78045 | 53.30476 |
| Ratcliffe | Coal | 2000 | 2004 | 2017 | Point | -1.25469 | 52.86594 |
| West Burton | Coal | 1972 | 2004 | 2008 | Point | -0.81184 | 53.36272 |
| Fiddlerâ€™s Ferry | Coal | 1961 | 2004 | 2018 | Point | -2.68692 | 53.37195 |
| Eggborough | Coal | 1960 | 2004 | 2018 | Point | -1.12485 | 53.71267 |
| Ferrybridge C | Coal | 1955 | 2004 | 2008 | Point | -1.28057 | 53.71564 |
| Kingsnorth | Coal | 1940 | 2004 | 2012 | Point | 0.603039 | 51.41671 |
| Didcot A | Coal | 1925 | 2004 | 2004 | Point | -1.26085 | 51.62225 |
| Teesside Power Station | Gas | 1875 | 2004 | 2010 | Point | -1.13264 | 54.57399 |
| Dinorwig | Storage | 1728 | 2004 | 2019 | Point | -4.11389 | 53.11861 |

Note: Instead of one table for each year, some data wrangling was done to add a “*yearStart*” and “*yearEnd*” column for each power plant and combine all data in one table.

# JSON representation of the data



```
"features": [
{ "type": "Feature", "properties": { "site": "Drax", "operator": "Drax Power Ltd", "type": "Coal", "fuelDetail": "-", "capacity": 3870.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1974", "yearStart": 2004.0, "yearEnd": 2012 }, "geometry": { "type": "Point", "coordinates": [ -0.999021, 53.737546 ] } },
{ "type": "Feature", "properties": { "site": "Drax coal", "operator": "Drax Power Ltd", "type": "Coal", "fuelDetail": "-", "capacity": 3225.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1974", "yearStart": 2013.0, "yearEnd": 2013 }, "geometry": { "type": "Point", "coordinates": [ -0.999021, 53.737546 ] } },
{ "type": "Feature", "properties": { "site": "Hinkley Point C", "operator": "EDF Energy", "type": "Nuclear", "fuelDetail": "-", "capacity": 3200.0, "lowCarbon": "Yes", "chp": "-", "yearOpen": "2026", "yearStart": 2019.0, "yearEnd": 2019 }, "geometry": { "type": "Point", "coordinates": [ -3.144, 51.206 ] } },
{ "type": "Feature", "properties": { "site": "Drax biomass", "operator": "Drax Power Ltd", "type": "Biomass", "fuelDetail": "-", "capacity": 2580.0, "lowCarbon": "Yes", "chp": "-", "yearOpen": "1974", "yearStart": 2019.0, "yearEnd": 2019 }, "geometry": { "type": "Point", "coordinates": [ -0.999021, 53.737196 ] } },
{ "type": "Feature", "properties": { "site": "Drax biomass", "operator": "Drax Power Ltd", "type": "Biomass", "fuelDetail": "-", "capacity": 2580.0, "lowCarbon": "Yes", "chp": "-", "yearOpen": "1974", "yearStart": 2018.0, "yearEnd": 2018 }, "geometry": { "type": "Point", "coordinates": [ -0.999021, 53.737196 ] } },
{ "type": "Feature", "properties": { "site": "Drax coal", "operator": "Drax Power Ltd", "type": "Coal", "fuelDetail": "-", "capacity": 2580.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1974", "yearStart": 2014.0, "yearEnd": 2015 }, "geometry": { "type": "Point", "coordinates": [ -0.999021, 53.737546 ] } },
{ "type": "Feature", "properties": { "site": "Longannet", "operator": "Scottish Power", "type": "Coal", "fuelDetail": "-", "capacity": 2304.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1970", "yearStart": 2004.0, "yearEnd": 2013 }, "geometry": { "type": "Point", "coordinates": [ -3.682552, 56.048549 ] } },
{ "type": "Feature", "properties": { "site": "Longannet", "operator": "Scottish Power", "type": "Coal", "fuelDetail": "-", "capacity": 2260.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1970", "yearStart": 2014.0, "yearEnd": 2015 }, "geometry": { "type": "Point", "coordinates": [ -3.682552, 56.048549 ] } },
{ "type": "Feature", "properties": { "site": "Pembroke", "operator": "RWE Generation SE", "type": "Gas", "fuelDetail": "Combined cycle", "capacity": 2199.0, "lowCarbon": "No", "chp": "-", "yearOpen": "2012", "yearStart": 2019.0, "yearEnd": 2019 }, "geometry": { "type": "Point", "coordinates": [ -4.996357, 51.68447 ] } },
{ "type": "Feature", "properties": { "site": "Pembroke", "operator": "RWE Generation SE", "type": "Gas", "fuelDetail": "Combined cycle", "capacity": 2199.0, "lowCarbon": "No", "chp": "-", "yearOpen": "2012", "yearStart": 2012.0, "yearEnd": 2018 }, "geometry": { "type": "Point", "coordinates": [ -4.996357, 51.68447 ] } },
{ "type": "Feature", "properties": { "site": "Littlebrook D", "operator": "RWE Npower Plc", "type": "Oil", "fuelDetail": "-", "capacity": 2055.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1982", "yearStart": 2005.0, "yearEnd": 2008 }, "geometry": { "type": "Point", "coordinates": [ 0.241644, 51.465333 ] } },
{ "type": "Feature", "properties": { "site": "Ratcliffe", "operator": "Uniper UK Limited", "type": "Coal", "fuelDetail": "-", "capacity": 2021.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1968", "yearStart": 2018.0, "yearEnd": 2018 }, "geometry": { "type": "Point", "coordinates": [ -1.254693, 52.865936 ] } },
{ "type": "Feature", "properties": { "site": "West Burton", "operator": "EDF Energy", "type": "Coal", "fuelDetail": "-", "capacity": 2012.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1967", "yearStart": 2009.0, "yearEnd": 2018 }, "geometry": { "type": "Point", "coordinates": [ -0.811836, 53.362719 ] } },
{ "type": "Feature", "properties": { "site": "Cottam", "operator": "EDF Energy", "type": "Coal", "fuelDetail": "-", "capacity": 2008.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1969", "yearStart": 2004.0, "yearEnd": 2018 }, "geometry": { "type": "Point", "coordinates": [ -0.780451, 53.304764 ] } },
{ "type": "Feature", "properties": { "site": "Ratcliffe", "operator": "Uniper UK Limited", "type": "Coal", "fuelDetail": "-", "capacity": 2000.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1968", "yearStart": 2004.0, "yearEnd": 2017 }, "geometry": { "type": "Point", "coordinates": [ -1.254693, 52.865936 ] } },
{ "type": "Feature", "properties": { "site": "West Burton", "operator": "EDF Energy", "type": "Coal", "fuelDetail": "-", "capacity": 1972.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1967", "yearStart": 2004.0, "yearEnd": 2008 }, "geometry": { "type": "Point", "coordinates": [ -0.811836, 53.362719 ] } },
{ "type": "Feature", "properties": { "site": "Fiddler’s Ferry", "operator": "SSE", "type": "Coal", "fuelDetail": "-", "capacity": 1961.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1971", "yearStart": 2004.0, "yearEnd": 2018 }, "geometry": { "type": "Point", "coordinates": [ -2.686922, 53.37195 ] } },
{ "type": "Feature", "properties": { "site": "Ferrybridge C", "operator": "Scottish and Southern", "type": "Coal", "fuelDetail": "-", "capacity": 1960.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1966", "yearStart": 2009.0, "yearEnd": 2013 }, "geometry": { "type": "Point", "coordinates": [ -1.280574, 53.715638 ] } },
{ "type": "Feature", "properties": { "site": "Eggborough", "operator": "Eggborough Power Ltd", "type": "Coal", "fuelDetail": "-", "capacity": 1960.0, "lowCarbon": "No", "chp": "-", "yearOpen": "1967", "yearStart": 2004.0, "yearEnd": 2018 }, "geometry": { "type": "Point", "coordinates": [ -1.12485, 53.712671 ] } },
```

In JavaScript this format can be directly converted into objects, which is very convenient as  opposed to parsing a table-like format.

# Understanding the data

## And think about what supports the narrative

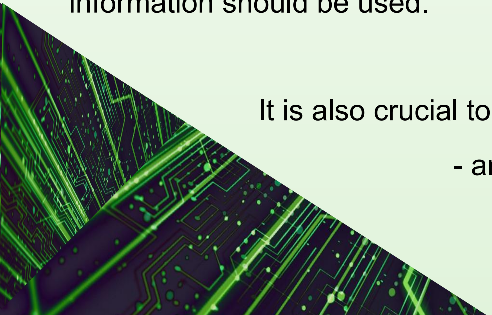
It is important the person visualising the data **understands** it. Else, crucial information could be missed or displayed wrong.

This dataset shows the *name* and *location* of power plants in the UK, which *type* of power generation they are, along with the *capacity* and the *start* and *end year of relevance* for this visualisation.

Note that some of these plants were opened a lot earlier than the *yearStart*. However the **narrative** of the story of this visualisation focuses on how and why the power generation in the UK transformed in the last 15 years, starting from 2004.

So, for this vis, it is only important if a given plant was active in the time frame 2004-2015, other data can be discarded.

If a *different* narrative is chosen it is absolutely valid to go back further in time, and it is **up to the narrator** to choose which information should be used.



It is also crucial to pinpoint the **events in the story** that need to be enhanced by visualisation or other imagery

- and how.

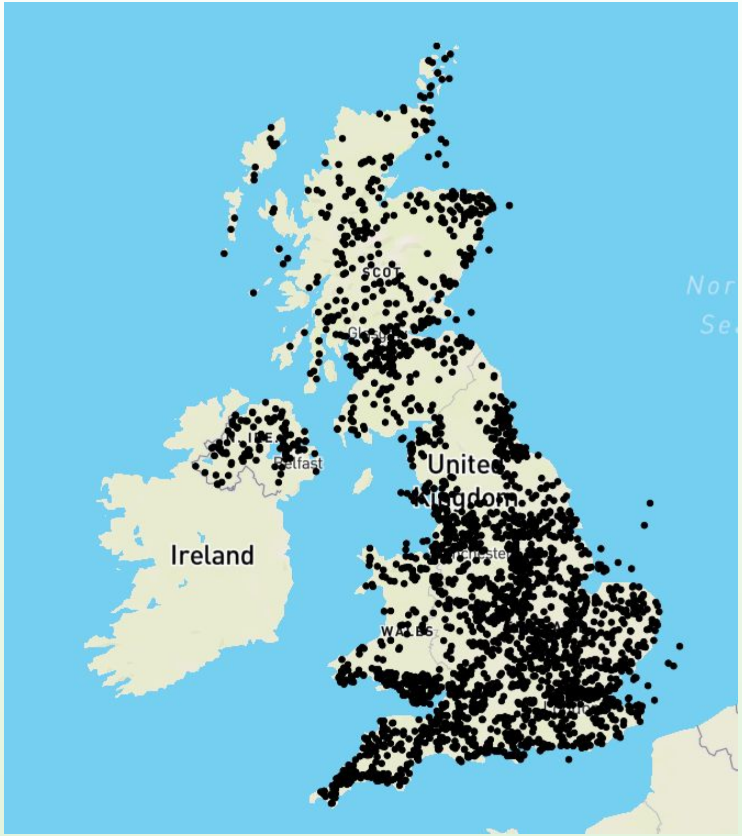
# Example - Scrollyteller with Map

Not very useful, not pretty.

How can this be improved?

💡

Drawing dots on a map is essentially a scatter plot in a 2D grid.



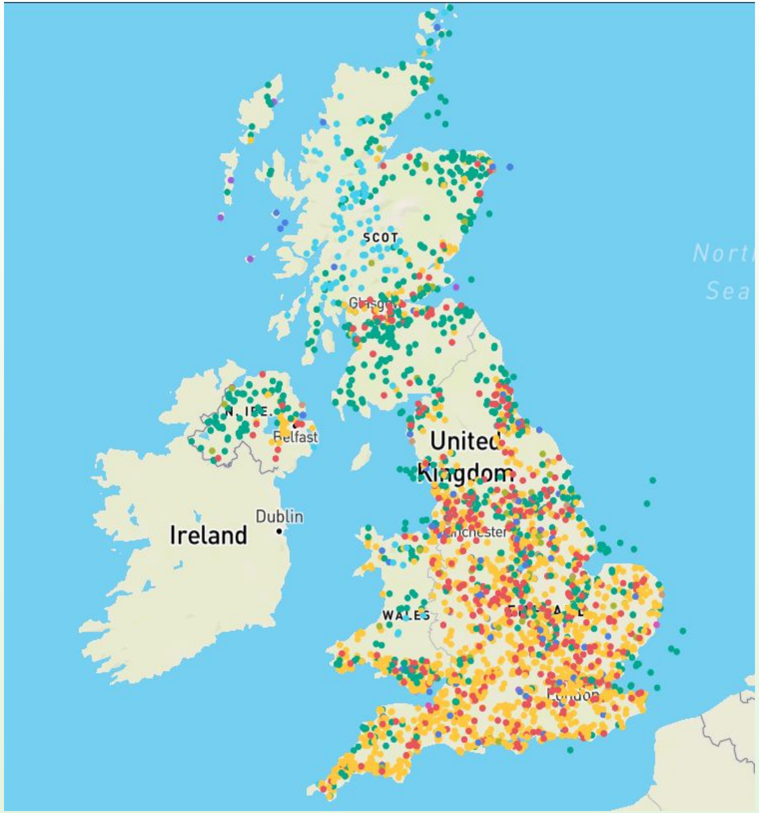
# Example - Scrollyteller with Map

Each type of plant has a different colour now.

Solar (yellow) in the South, while there is quite a bit of wind (green) offshore and in the North.

It is still very cluttered.

How can this be improved?



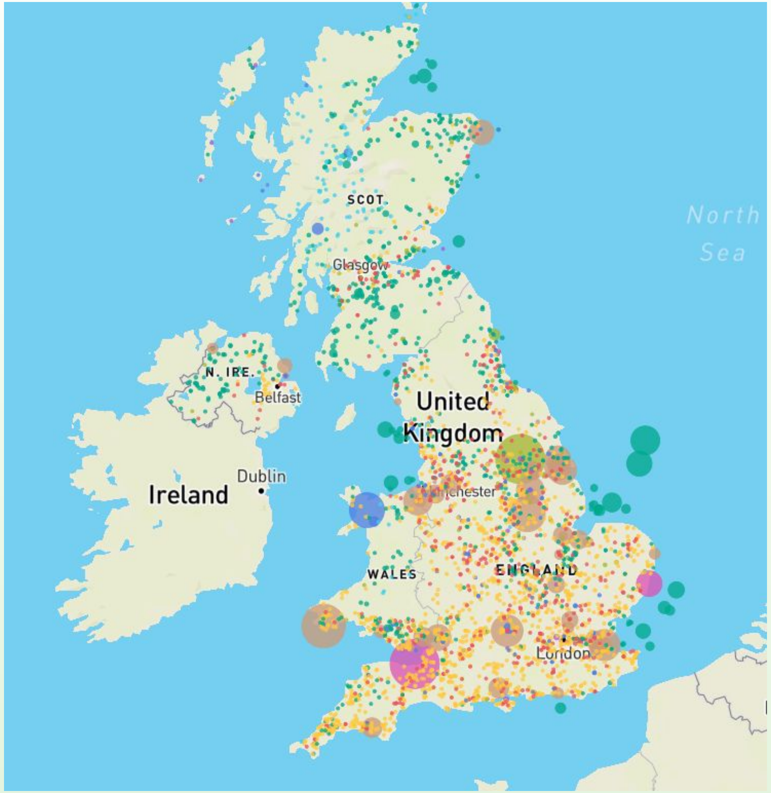
MONASH University

# Example - Scrollyteller with Map

Changed the size of the dots according to the *capacity* column.

The dots are also now a bit transparent to prevent hiding dots behind others.

A couple of large gas plants (brown) appear, the offshore wind parks seem huge and the solar panels are tiny, but numerous.



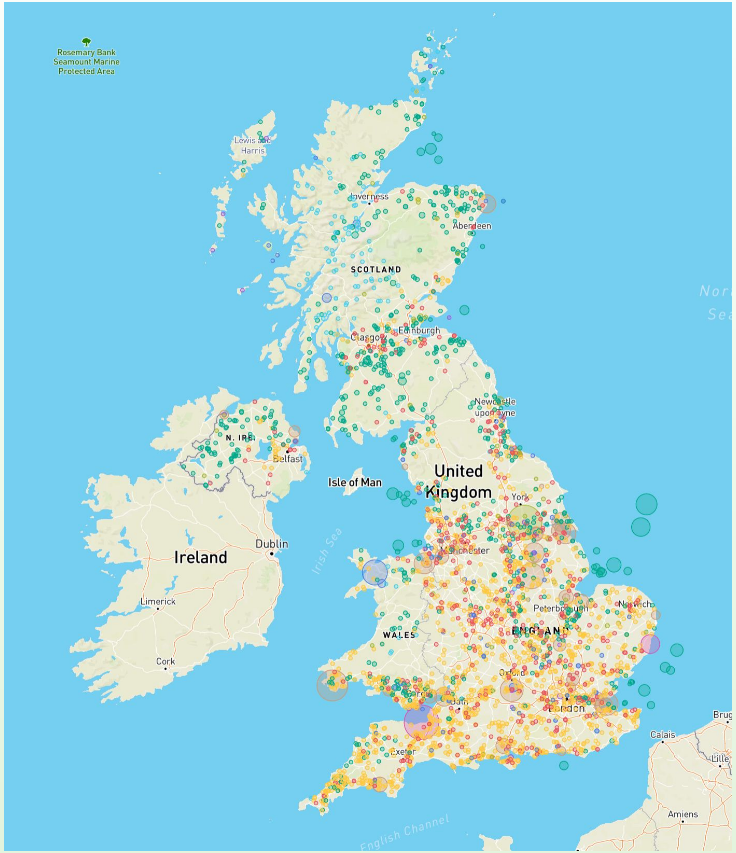
# Example - Scrollyteller with Map

Different circle style

It is worth it to try different styles to find what feels right for the audience.

Other things to consider with maps:

- aggregating points based on zoom level
- visibility - cluttering



# Example - Scrollyteller with Map

**A few notes**

- For all the images, the *same dataset* was used. mapboxGL was only told some conditions about how to draw the circles.
- The images are for the year 2019. mapboxGL accepts filters - the filter here is:
  - ○ *yearEnd* value must be larger or equal to 2019,
  - ○ *yearStart* value must be smaller or equal to 2019.

  All other rows are filtered out.

- Combining this with a scrolly teller enables filtering on the map on runtime as the user scrolls and reads through a narrative
- Other events can be filtered, e.g:

  in the year solar power took off, you could highlight only solar on the map to emphasize that, or highlight a major individual coal plant in the year it shut down.

# Example - Scrollyteller with Map

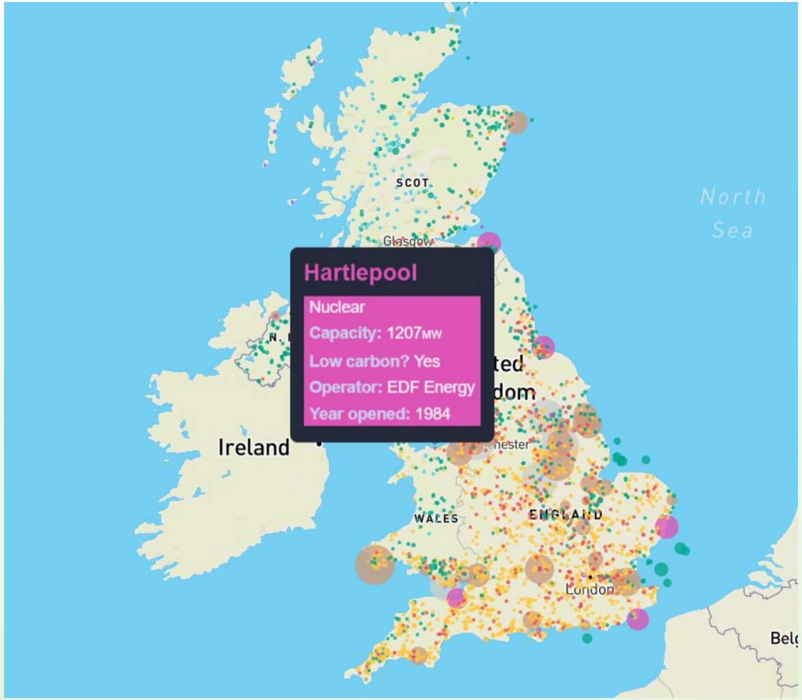
More functionality:

Optional for users to mouse over each plant and display more information.

The map can also be zoomable to get a clearer view for smaller areas in the UK.

Explore the functionality at

[https://interactive.carbonbrief.org/how-uk-transformed-electricity-supply-decade/](https://interactive.carbonbrief.org/how-uk-transformed-electricity-supply-decade/)



Hartlepool

Nuclear

Capacity: 1207MW

Low carbon? Yes

Operator: EDF Energy

Year opened: 1984

# Example - Scrollyteller with Map

On the website, in year 2008, other filters are applied to highlight certain types of power plant.

There is a **narrative** and the visualisation reacts to it. The reader/viewer is taken through the historical events in the UK and it is visualised on the map.

The stacked bar on the left shows the distribution of power generation type in the active year. It also updates as the user scrolls.

# Questions when creating this vis

**What has to be displayed and understood at first glance?**

- which year is currently “active”
- location of power plant
- size/capacity of power plant
- Percentage of power generation for each power type. This is in a different chart than the map but also updates when the user scrolls as it changes over the years.

**What can optionally be displayed in an interactive manner?**

- additional information for individual power plants (mouse hover)
- more spaced out plants upon zoom in, or a general overview upon zoom out
- there can be plenty of images or clickable links in the narrative

# Narrative and Vis - which first? (1)

## Narrative first

Sometimes there is a story to tell, and to get it across to the viewer, visualisations help a lot.
In this case, it needs to be identified what **can** be visualised and in which way.
This is comparable with the job a **movie director** or a game designer has - have a story, chose how to present it. Often, this is a very creative job.

Fact-based stories, like the current COVID-19 evolving situation, are also in this category and there exist many visualisations about many different aspects of this topic.

# Narrative and Vis - which first? (2)

## Narrative first

Having a **clear goal,** a clear message to convey via the narrative is crucial. Stick to that goal and consider it when creating the visualisations.

A **clear storyline** is important to create an engaging story. Reasons why a story arc changes can be emphasised by visualisations. It is good to have the ability to answer potential upcoming questions → Interactivity
Otherwise, the imagery and vis should support everything necessary, but be as simple as possible.

# Narrative and Vis - which first? (3)

## Data first

The other case is that there is a lot of data available, which gives insight to certain topics.

Researchers often collect large amounts of data, to find out *if there is a story to it.* Data mining and “Big Data” in general also falls in this category:

The challenge is to gain information from that data and sometimes you find out things you didn’t know yet - one of the goals of research.

# Covid-19, the story

### Collective of Covid-19 data visualised

https://www.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf6

Johns Hopkins Map vis and graphs

### Animated visualisation

https://www.technologyreview.com/2020/03/27/950263/the-covid-19-pandemic-in-two-animated-charts/

Two animated charts using the data from Johns Hopkins

→ Shows the same thing differently,  people will find one or the other more appealing.

# 3D Visualisation

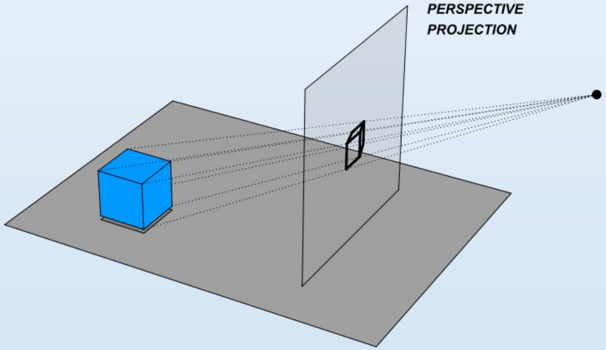
# What is 3D Visualisation

- 3D Visualisation is a set of techniques that allows us to produce 3D renderings that can be used on both desktops, handhelds and virtual reality headsets.

- Developed to represent 3D phenomena, eg: from instruments or simulations, with an emphasis on realistic rendering, perhaps with the addition of a time component (animation) (https://en.wikipedia.org/wiki/Scientific_visualization)

# 3D Vis - Techniques

- We are still working with 2D images as the final output, all our displays are flat 2D planes:
- 3D Vis requires translation to 2D Image from particular viewpoint from an infinite number of possible views and is impossible to encapsulate in a single image.



# 3D Vis - Techniques

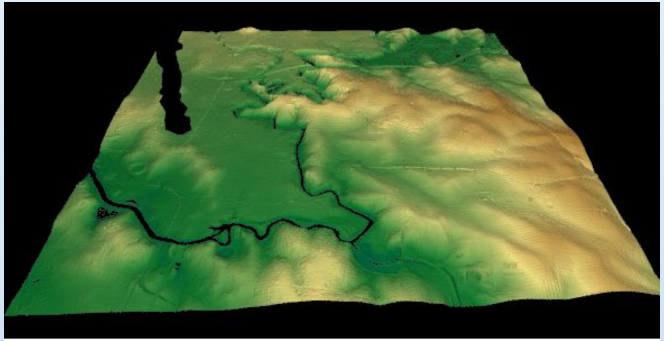
- Using modern graphics processing hardware we can create an illusion of a 3D world or view to examine and analyse a model.
- Visualising large data sets requires a high end graphic capability, which might not be available on the average laptop or mobile device.
- Cloud platforms increasingly provide availability (AWS, NeCTAR, Google Colab)

# Realistic vs abstract

A 2D visualisation of map provides a perspective of a realistic landscape, usually from an unusual point of view (such as a very much zoomed out birds-eye view, omitting details.

In a 3D visualisation of a map things like rivers, buildings, mountains or other landscape elements can be realistically displayed.

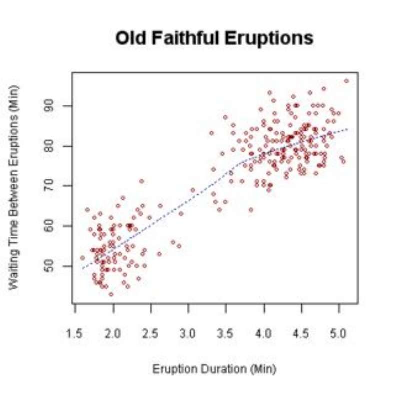
This opens possibilities to *explore places* from several points of views and *simulate* scenarios by adding/removing components.

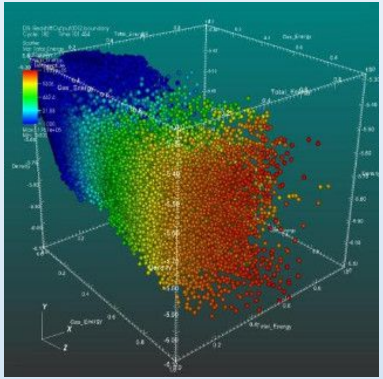


# Realistic vs abstract

Graphs and other axes plots are abstract. They don’t try to display the underlying data in a realistic way, but provide a mental model of the data. This is mostly used in 2D, but can be used in 3D -

but 2D techniques don’t always translate well into 3D models!



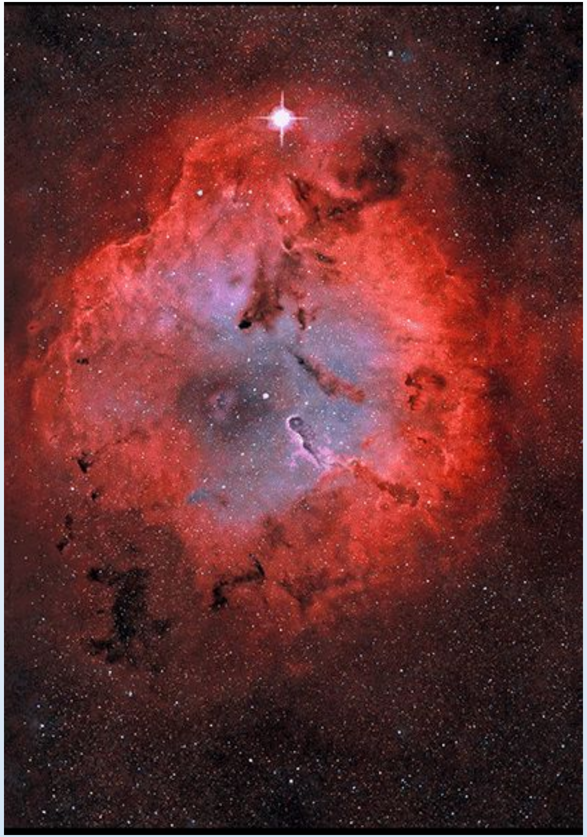


# Why 3D Visualisation?

*Mu Cephei IC 1396 Nebula - from hubble telescope*

# Why 3D Visualisation?

*3D animation of same area of space : J-P Metsävainio*



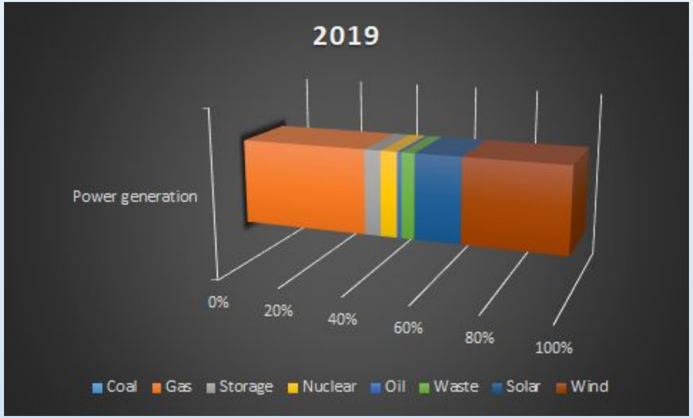
# Why 3D Visualisation

- The data is inherently 3D (eg: 3D models, scans, geospatial data that includes the height axis)
- The data is abstract and difficult to condense to 2D or comprehend with a 2D plot
- The data can be mapped to an inherently 3D dataset, eg: can be plotted on a geospatial 3d dataset to map to locations
- For outreach / increased impact, fun : we often simply find 3D visualisations more compelling than flat 2D images, but keeping that in mind…

# When not to use 3D Visualisation

- If you can communicate your data clearly in a 2D plot
- If adding a 3rd dimension just provides too much visual information and complexity without adding anything
- [https://serialmentor.com/dataviz/no-3d.html](https://serialmentor.com/dataviz/no-3d.html)

Bad



# 3D Visualisation - Data types

The first step in visualising data in 3D is to translate it into a data type that can be easily rendered into a 3D scene.

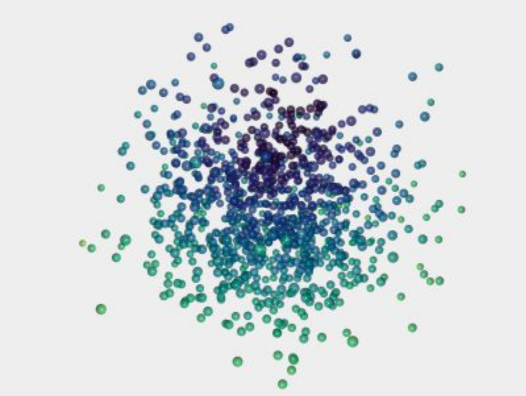
Some data already matches nicely with one of these types, but other sources will need you to come up with a way to turn more abstract data into 3D spatial representations.

The three most widely used data types for this task are **Points, Meshes** and **Volumes**. Each have different properties and uses and techniques exist to translate data between these types.

All these types can be combined to create more complex visualisations.

# 3D Visualisation - Data types

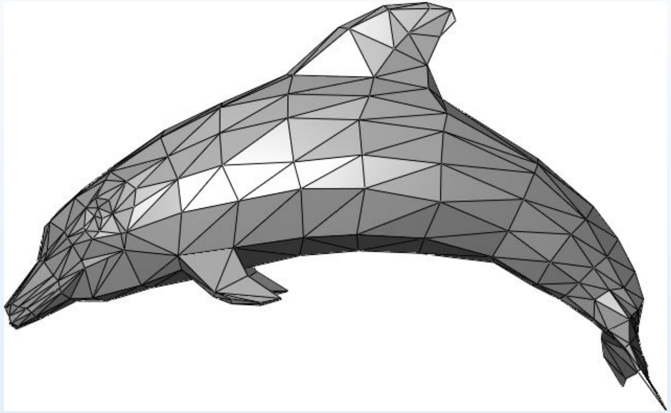
## Point



- most simple, requires X,Y,Z coordinates plus rendering parameters such as colour, size, etc. which may or may not be tied to further data values
- LiDAR / Photogrammetry : Quick recreation of a real world scene or object from photographic footage or laser scans (AgiSoft PhotoScan, Phone based 3D scanning)
- Points as data points : points can be used to show elements of more abstract data types or even represent moving objects
- Common file formats : LAS, LAZ, OBJ, PLY etc

# 3D Visualisation - Data types

## Mesh

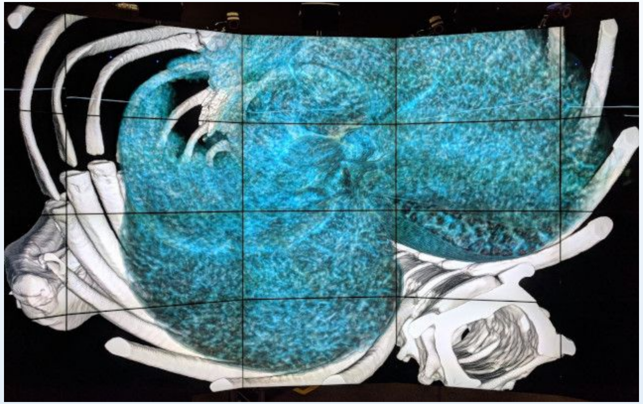


The building block of most 3D graphics : used to create the illusion of objects by drawing the surface only (via a connected “Mesh” of triangles)

- Created in 3D modelling applications (Blender, Maya)
- Can be created from the real world via 3D scanning (to point cloud, then further processing to a mesh)
- Can be extracted from Volume data (isosurface) or generated from a dense point cloud.
- Common file formats : OBJ, PLY, STL, FBX etc

# 3D Visualisation - Data types

## Volume



Think of stacks of 2D images : resulting in cubes of data points on a regular grid in 3D

- Direct result of 3D image scans : microscopy, CT scans, MRI scans
- 3D Simulations and numerical models
- Can plot directly via Volume Rendering - slow but often very fine quality rendering or extract surface or point data for simpler plots (slices, isosurfaces)
- Common file formats : DICOM, TIFF, PNG, RAW

# 3D visualisation - Available Tools

**Methods not requiring Programming Skills**

- Desktop tool such as scientific visualisation or 3D modelling application
  - All-in-one environments: ParaView, Vislt, Drishti
  - 3D modeling applications: Blender, Maya → Hand editing of 3D meshes
- previs web portal to preview visualisations (online, but discontinued)

**Methods requiring Programming Skills**

- Visualisation libraries/tools to create 3D plots and models
  - Tied to the programming environment of choice: Matlab, Python, R, Web ( → Matplotlib, D3, Plotly, GGPlot)
- Game development engine: Unity3D, UnrealEngine, etc

# Summary

# Visualisation - Do’s and Don’ts

**Do**

- Be as simple as possible, but include everything that’s necessary
- choose the correct chart for your type of data
- Make it appealing to your audience
- label axes and data where appropriate
- chose colours supportive of your topic
- With narrative: Stick with your story. Make additional info optional through interactivity.
- If you made assumptions or predictions, clearly indicate that

**Don’t**

- use too many colours or distracting elements
- Add bias
- try to make it look “cool” when it distracts from the point of the data
- mislead
- clutter

MONASH University

# Visualisation and Story telling

Visualisation combined with Storytelling is a powerful tool to get an audience to **gain insight** and knowledge about a topic.

Visualisation enables people to **understand data**. Storytelling enables people to **understand backgrounds**, remember content and learn.

Visualisation and Storytelling together get the big picture across.

Simple visualisations like charts and graphs can be done in Excel. More complex ones should use dedicated software and tools.

# Creating visualisations

- Data wrangling may or may not be necessary.
- Different types of graphs are required for different sets of data.
- Colour and style should support the topic and be appealing to the audience. Simple is generally better, because
- Keep it as simple as possible, while maintaining all *relevant* information.
- Label correctly, include a legend if necessary.
- Don’t mislead, don’t include bias.

# Presentation

Learn which tools or libraries are available for you to create the visualisation. Don’t reinvent the wheel.

In the narrative, key points or events should be supported with visualisations or imagery. Too much additional information will clutter the presentation. That can be made *optional via interactivity* if the user wishes to get more background info.

It can help to play around with different colours, types of shapes, etc. to get a feel for the visualisation. It should be supportive to the topic of the narrative. For example, bubbly cute icons are not appropriate for a topic like Covid-19.

# Resources

**Visualisation / Storytelling**

- Live Online Training Course: Data Visualisation & Storytelling | Informa Corporate Learning  video at the bottom with a webinar about creating visualisations
- https://www.tibco.com/events/series/harvard-business-review-data-visualization-webinar-series  several vis and storytelling webinar videos
- SWD community | storytelling with data
- Information is Beautiful blog, many vis examples, (paid) online trainings
- 7 Deadly Sins of (Academic) Data Visualisation – Spatial.ly
- The Data Visualisation Catalogue Which Visualisation to use for which data, lots of examples and guides
- From data to Viz | Find the graphic you need

**Excel**

- https://www.excel-easy.com/

# Resources

**Web**

- https://github.com/abcnews/scrollyteller ABC news JS scrolly teller ReactJS component "oddyssey"
- A beginner's guide to scrollytelling Scrollytelling examples
- 11 Javascript Data Visualization Libraries for 2019

**3D Visualisation**

- Knowledge in 3D: How 3D Data Visualization Is Reshaping Our World
- Fundamentals of Data Visualization

**Tools and Software**

- Python : Anaconda https://www.anaconda.com/distribution/ nteract : https://nteract.io/
- Google Colab https://colab.research.google.com
- Unity3D https://unity3d.com/
- WebVR : https://webvr.info/ A-Frame https://aframe.io/
- PreVis Prototype : https://mivp-dws1.erc.monash.edu:3000/