COMP9321:
Data services engineering

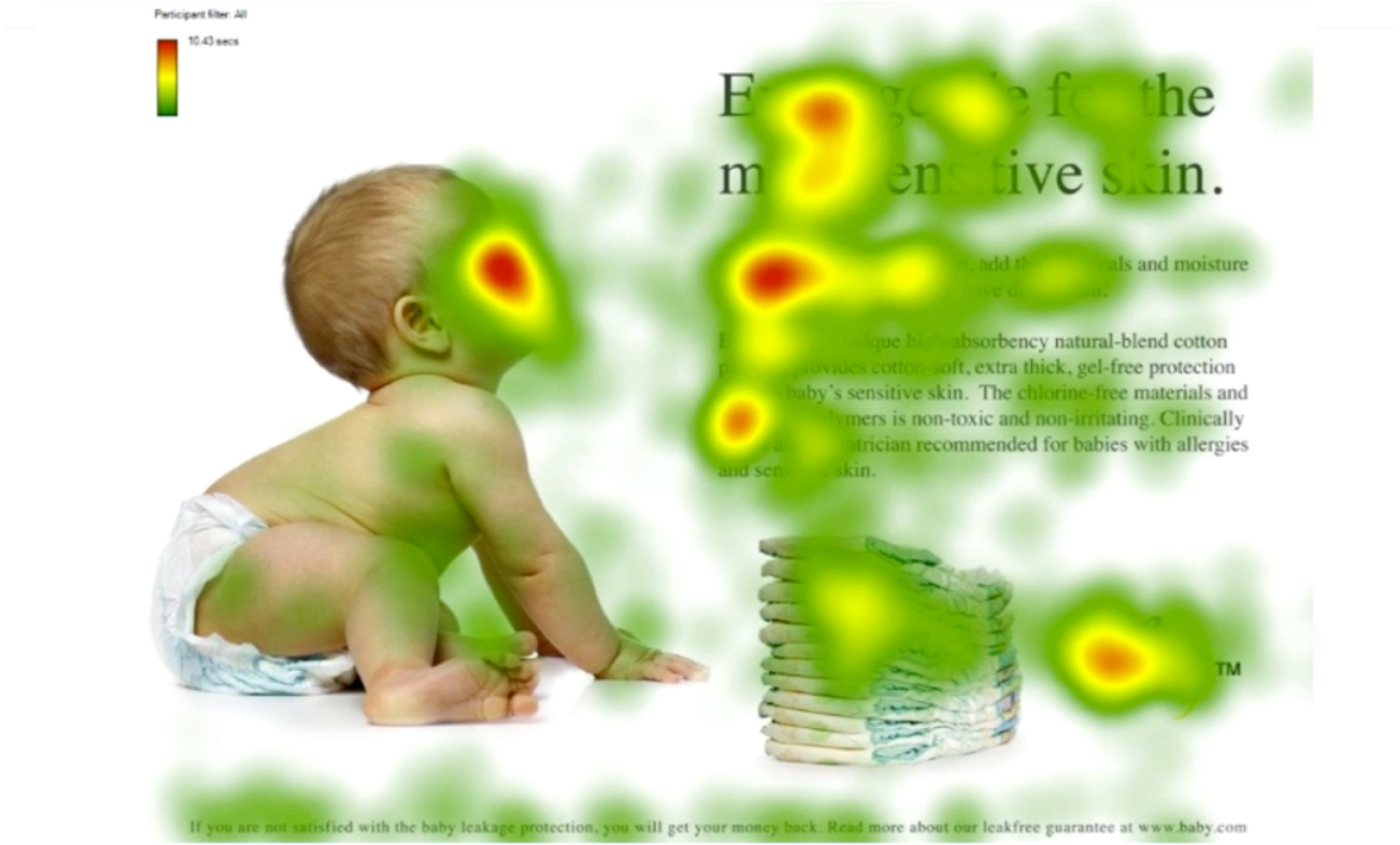
Semester 1, 2018,

Week 5 Data Visualisation (Principles and Basic Techniques)

By Helen Paik, CSE UNSW

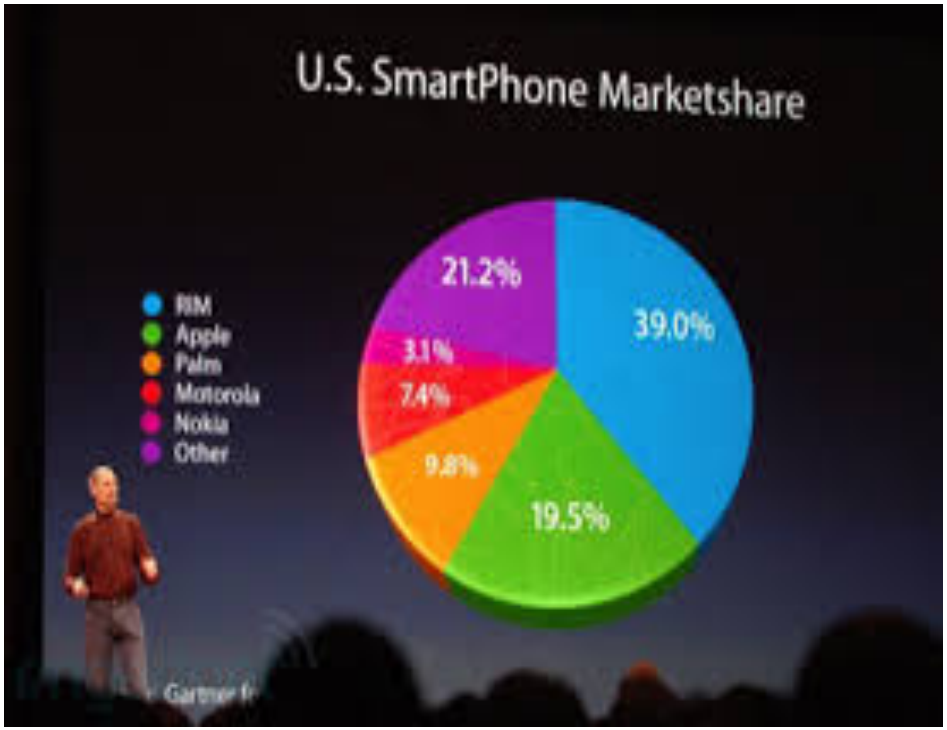
Visualisation isn't just about graphics

Highly competent visualisation "tricks" can affect what you see and what you pay attention to:



When Steve Jobs says …

- Again … playing with human visual perception
- 21.2% versus 19.5% slices of the pie



Apple SmartPhone Market Update, Macworld keynote (2008)

https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=000185

In this lecture …

Often, visualization is considered highly specialized area and getting it right takes skilled knowledge from multi disciplinary areas (statistics, graphic design, understanding of human perception of visual elements, or sometimes understanding of human psychology)

In most cases, the topic will be a course by itself…

Of course, we cannot get to that level of details here … But if you are writing an application that is data-driven, most likely you'd run into some visualization tasks.

What I intend to convey in this lecture is two parts:

First : the basic principles of 'good and competent' visualization

Second: The core concepts in D3 language (what it does), we actually get to see the basic building blocks of visualization techniques which are useful for further exploration of the area if you are interested …

Presentation as a service

Of course, there is another important point to make.

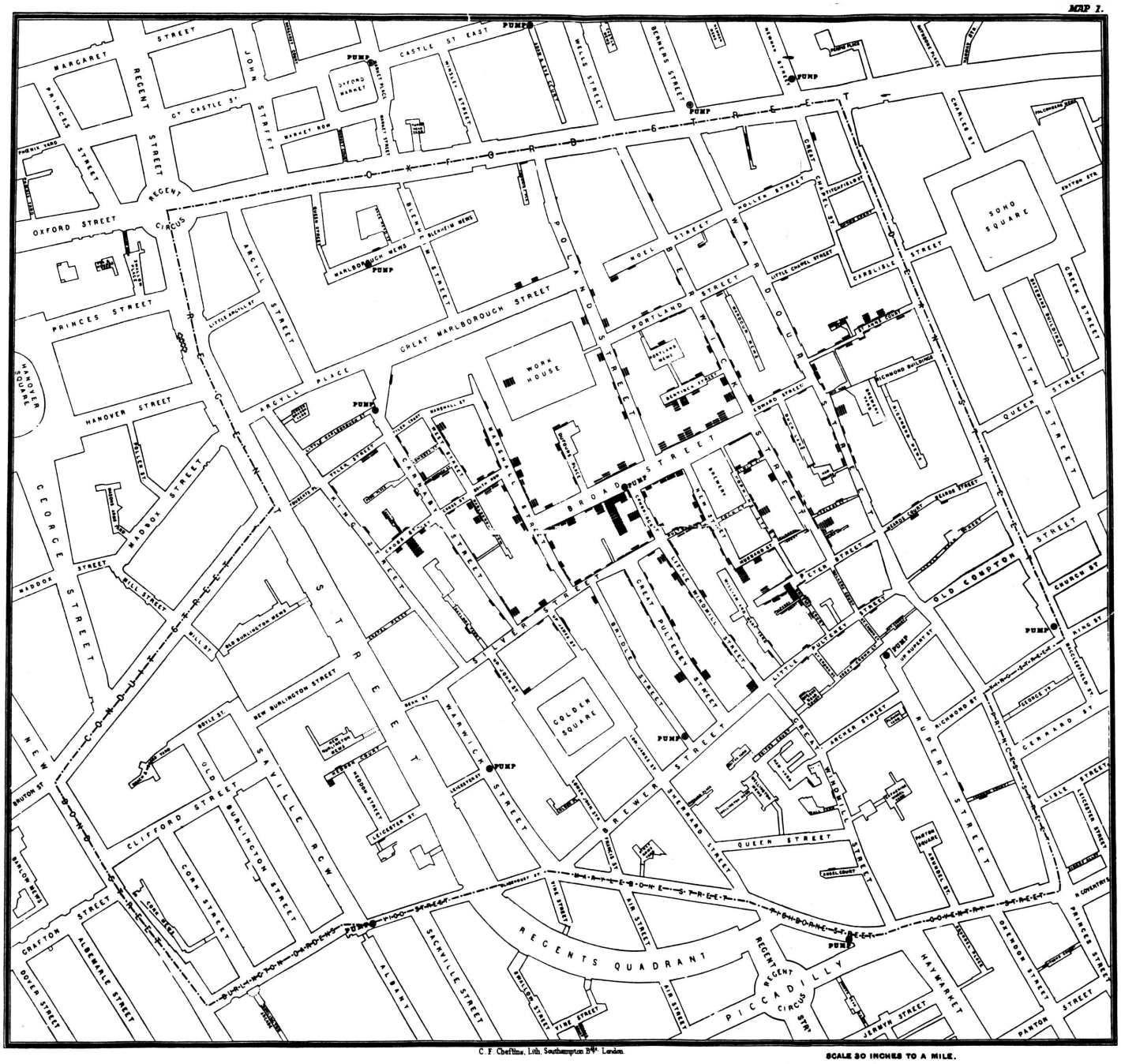
Once you know how to visualise data, what comes naturally after that is to offer that knowledge as a service (API)

It's known as "Presentation" or "Visualisation" as a service.

"Visualisation on the Cloud"

- e.g., https://www.highcharts.com
- e.g., Google Map (good example of presentation as a service)
- On request, its response can contain either "presentation logic + data", or already visualised data in graphics/HTML

So for this course context, the concept of "presentation as a service" is more interesting than the visualisation techniques themselves.



What is Visualisation

Visualization transforms data into images that effectively and accurately represent information about the data.

Insight

https://www1.udel.edu/johnmack/frec682/cholera/ (John Snow Map of Cholera 1854)

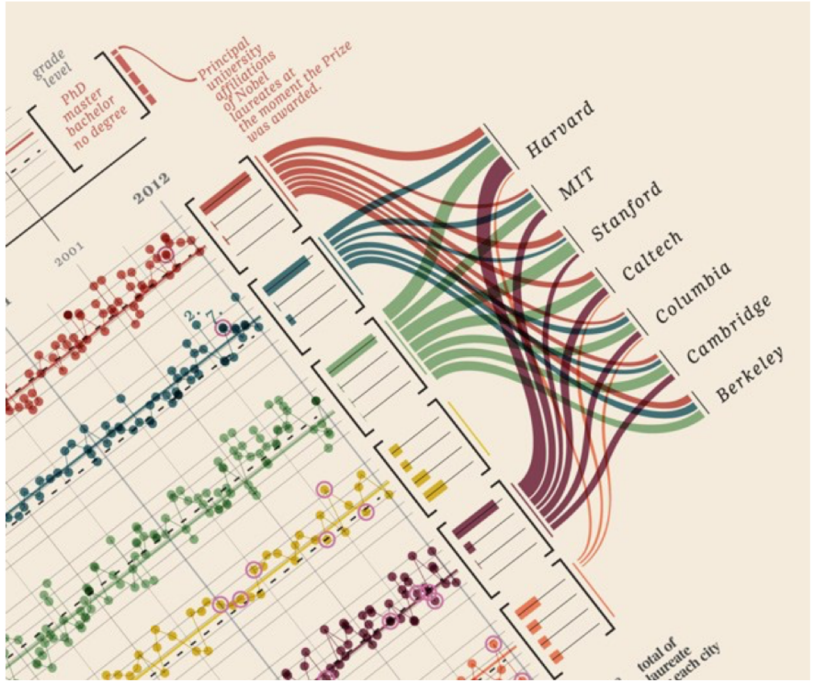
What is visualisation

Three types of goals for visualization – ... to explore

Nothing is known … Visualisation is used for data exploration/discovery …

e.g., A Visual History of Nobel Prizes and Notable Laureates, 1901-2012

https://www.brainpickings.org/2012/11/29/giorgia-lupi-noble-prizes-visualization/



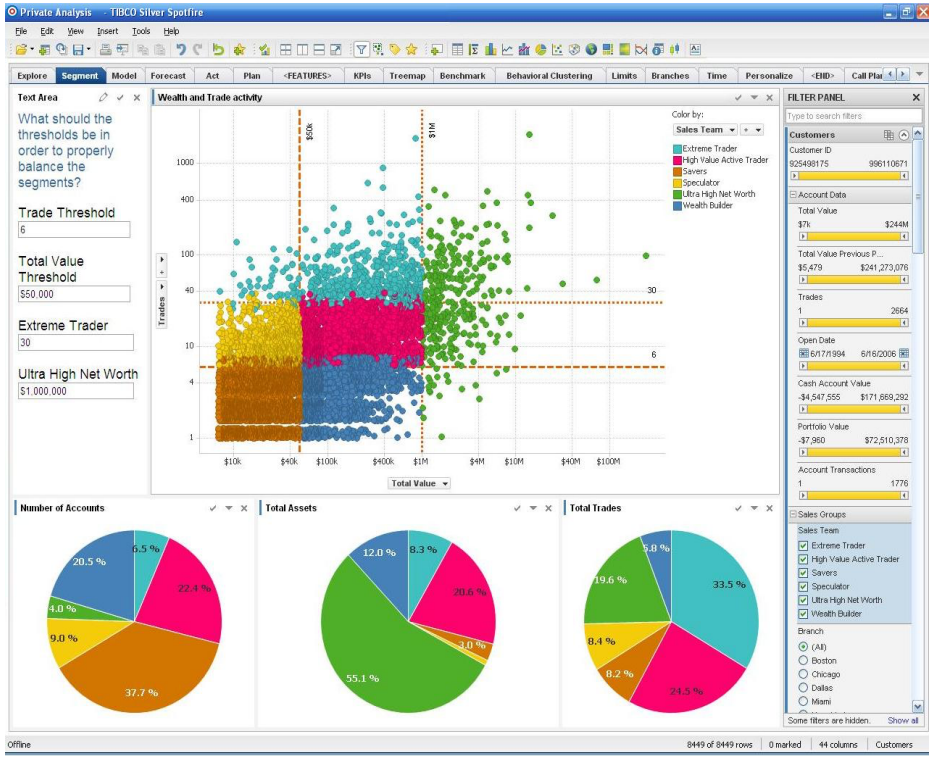
https://www.safaribooksonline.com/library/view/designing-data-visualizations/9781449314774/ch01.html

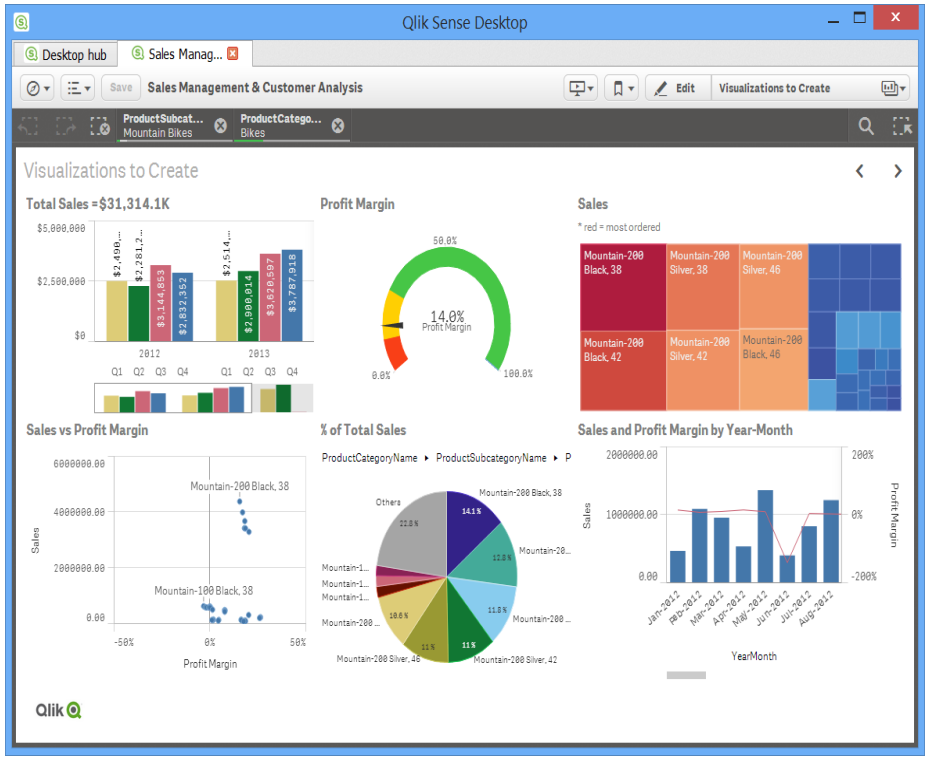
What is visualisation

Three types of goals for visualization – ... to analyse

You have some hypotheses. Visualisation is used for Verification or Falsification

You'd normally use what is classified as "data analysis and visualisation" tools (- a range of data source connectivity, built-in quick visualisation graphs, etc.)



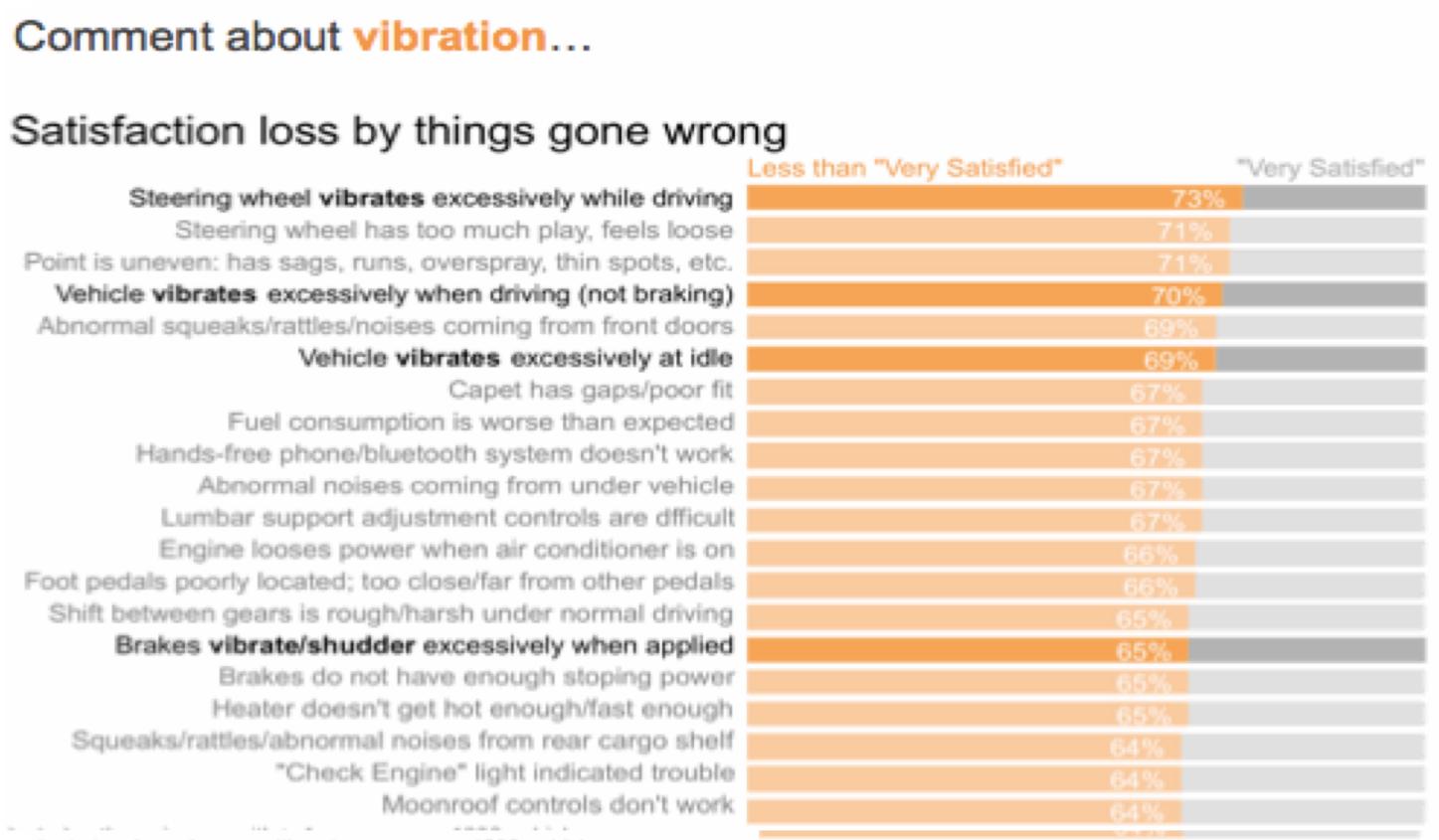


https://www.safaribooksonline.com/library/view/designing-data-visualizations/9781449314774/ch01.html

What is visualisation

Three types of goals for visualization – ... to explain

You do know what the data contains … Visualisation is used for “effective” communication of “results” – Making them clear for the audience



https://www.safaribooksonline.com/library/view/designing-data-visualizations/9781449314774/ch01.html

Data Visualisation

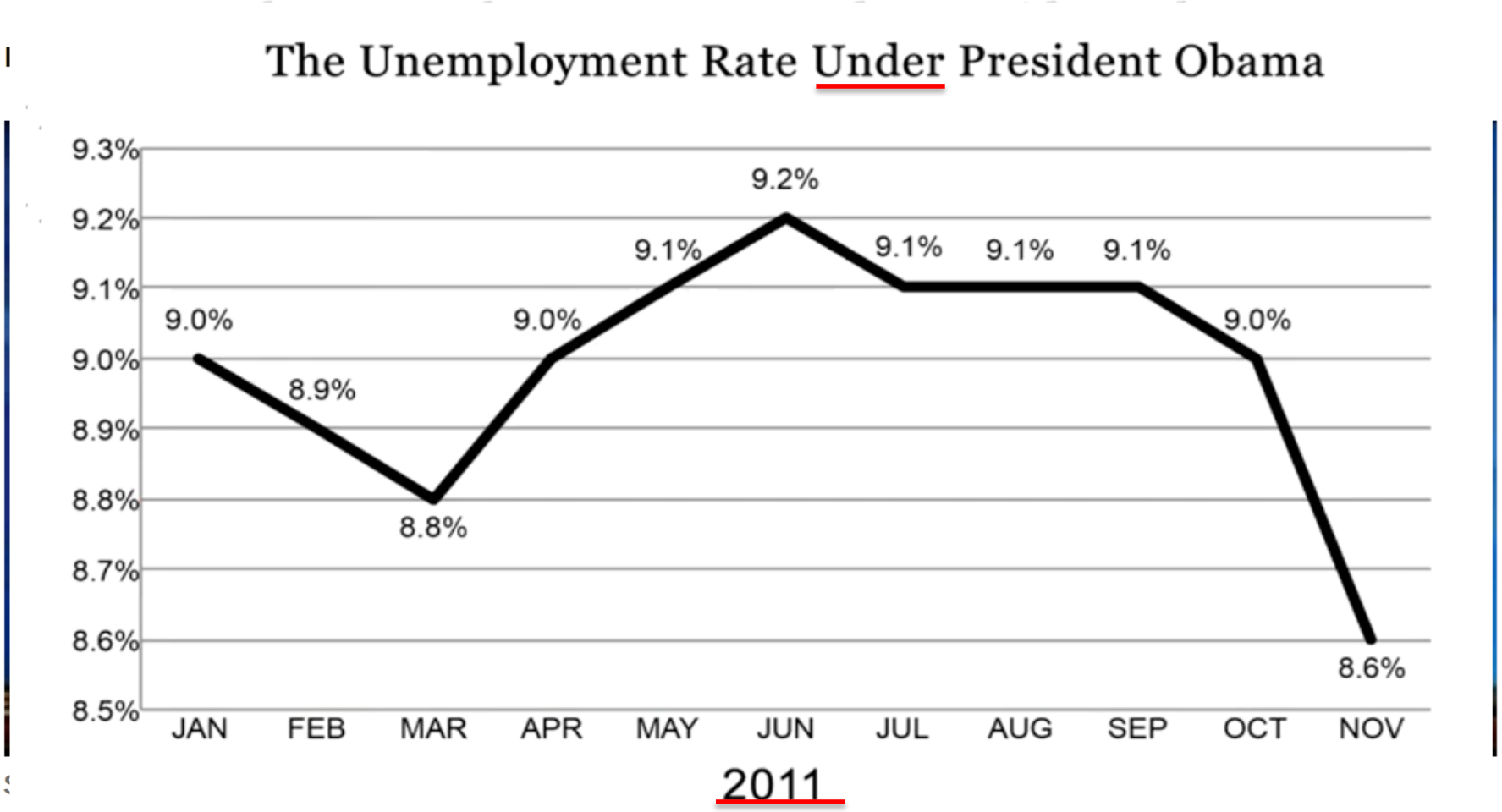
Referring to any visual representation of data that is:

- algorithmically drawn (may have custom touches but is largely rendered with the help of computerized methods);
- easy to regenerate with different data (the same form may be repurposed to represent different datasets with similar dimensions or characteristics);
- often aesthetically simple (data is not decorated); and
- relatively data-rich (large volumes of data are welcome and viable, in contrast to infographics).

https://www.safaribooksonline.com/library/view/designing-data-visualizations/9781449314774/ch01.html

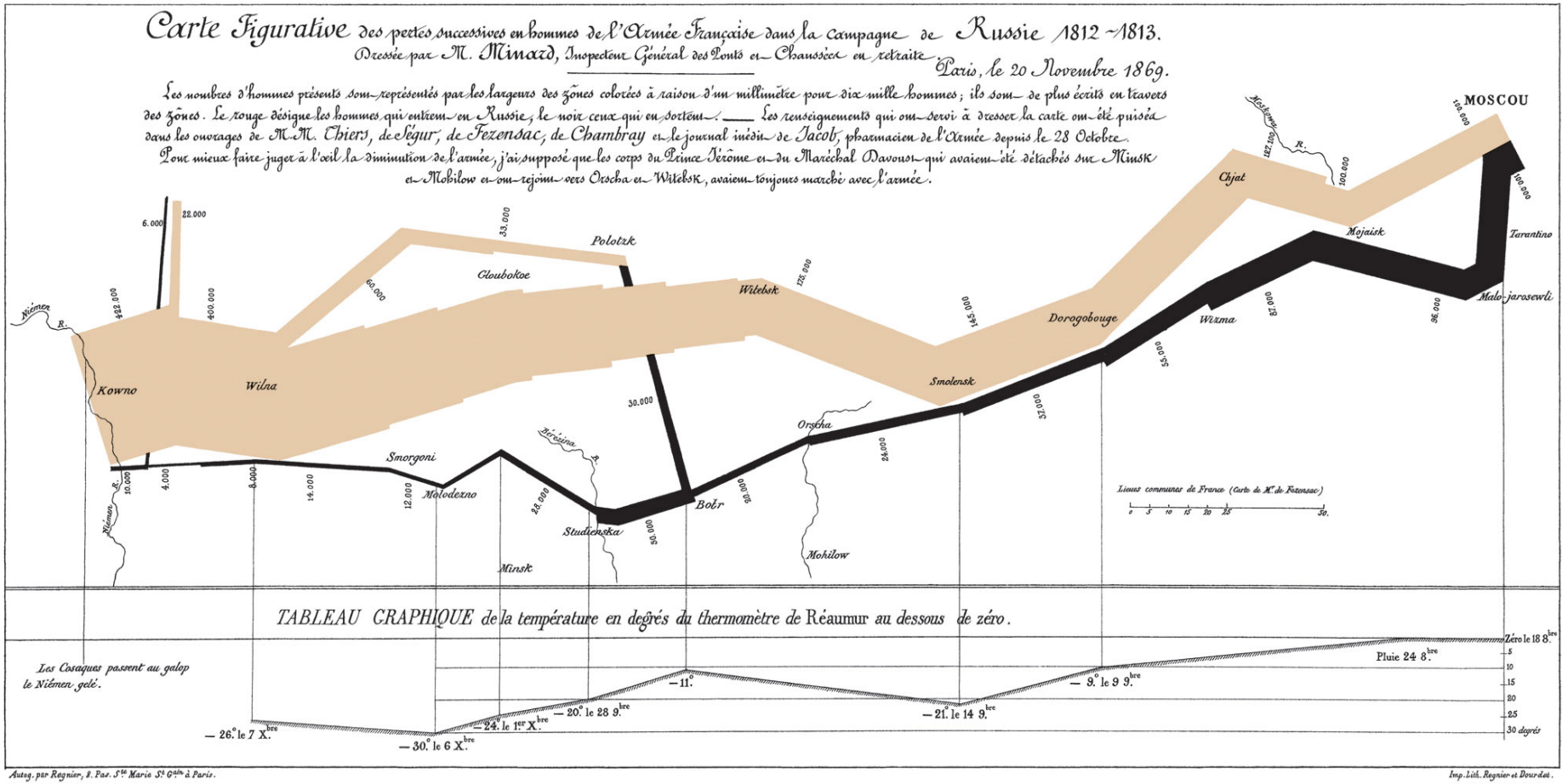
So what makes a good visualisation

Accuracy, Story, Knowledge: Aim to create a visualisation that are accurate, tell a good story, and provide real knowledge to the audience.



So what makes a good visualisation

Accuracy, **Story**, Knowledge: Aim to create a visualisation that are accurate, tell a good story, and provide real knowledge to the audience.



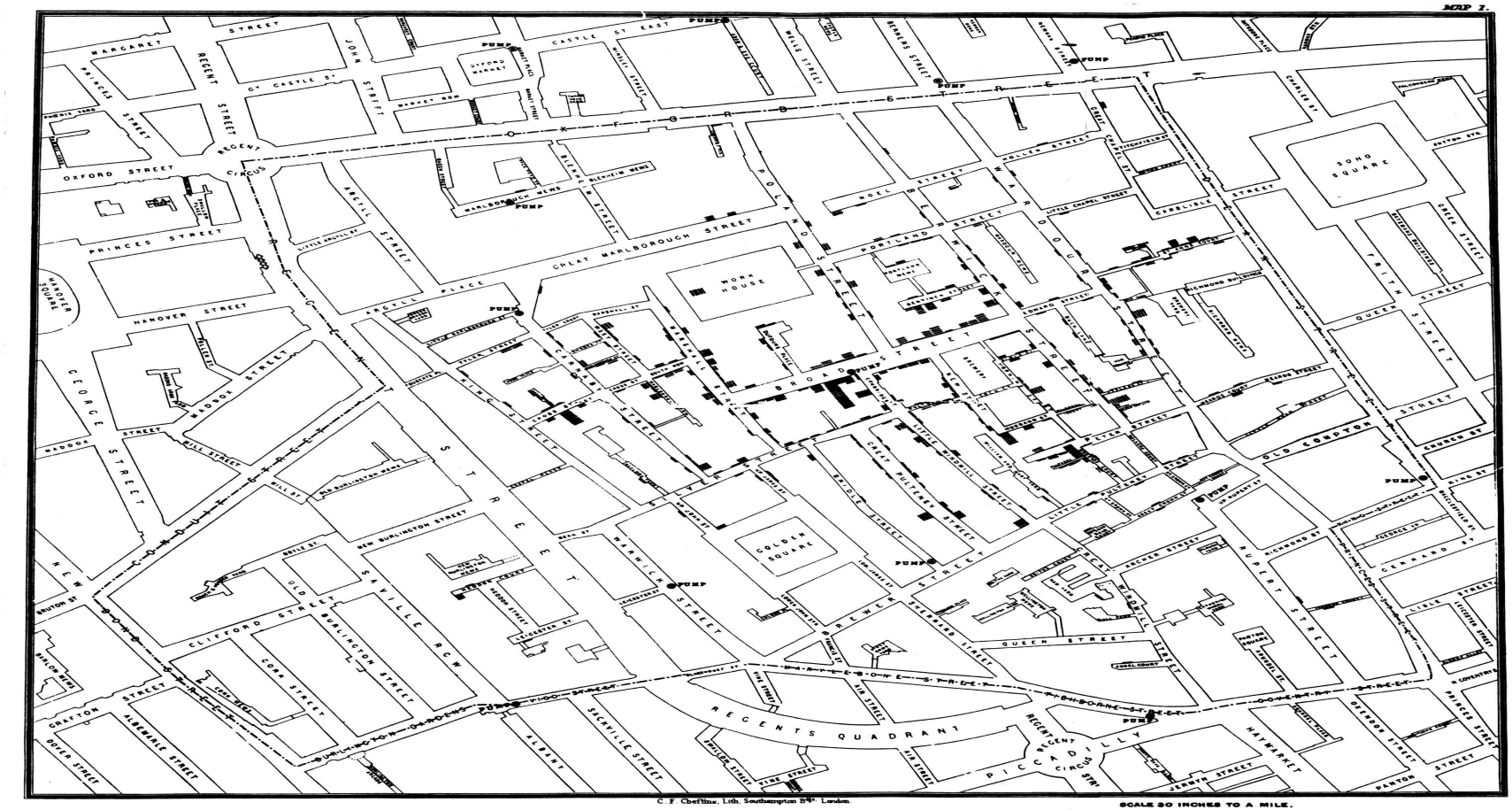
The Minard Map - "The best statistical graphic ever drawn"

https://en.wikipedia.org/wiki/Charles_Joseph_Minard (Charles Minard Map)

Learning Data Visualization by Bill Shanders, Lynda.com

So what makes a good visualisation

Accuracy, Story, **Knowledge**: Aim to create a visualisation that are accurate, tell a good story, and provide real knowledge to the audience.

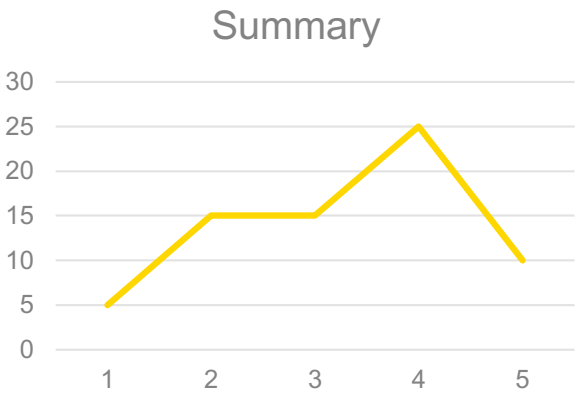


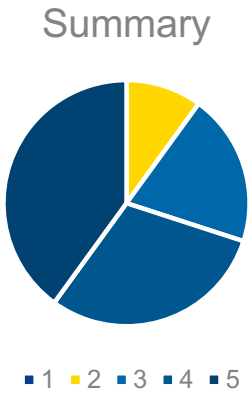
Widely attributed to creating the field of Epidemiology

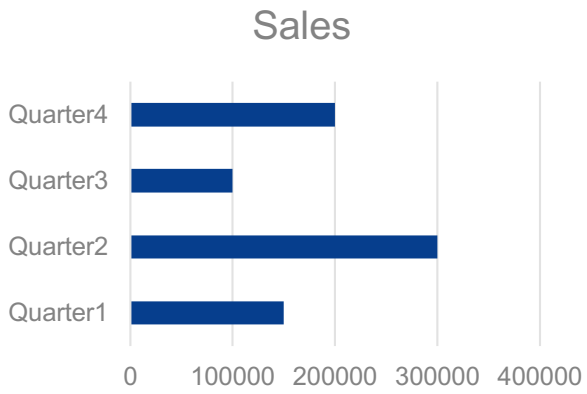
Some of the basics … “Charts vs. Graphs”

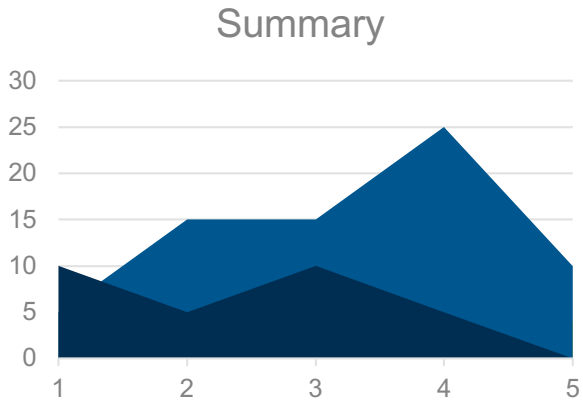
Often used interchangeably, but they are different in terms of the “visualisation techniques involved”.

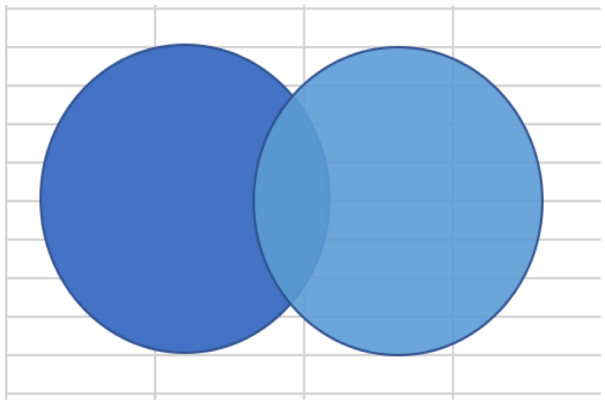
	Lights	Desk	Table	Monitor
Exhibit 1	✘	✓	✘	✘
Exhibit 2	✓	✓	✓	✘
Exhibit 3	✓	✘	✘	✓
Exhibit 4	✓	✓	✓	✓
Exhibit 5	✓	✓	✘	✘
Exhibit 6	✘	✘	✘	✓
Exhibit 7	✓	✓	✘	✓
Exhibit 8	✓	✘	✓	✘
Exhibit 9	✓	✘	✓	✘
Exhibit 10	✓	✓	✘	✘
Exhibit 11	✓	✓	✘	✓
Exhibit 12	✓	✘	✓	✓
Exhibit 13	✘	✘	✘	✘





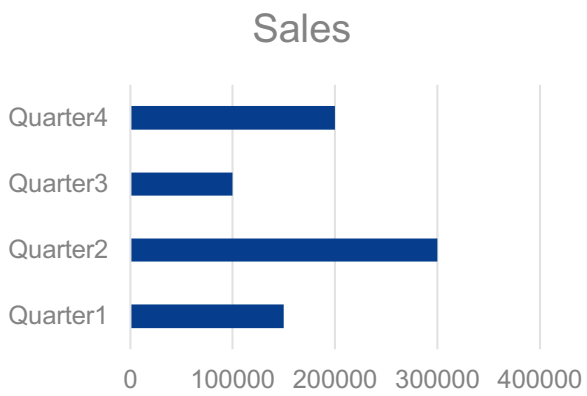






Some of the basics … “Charts vs. Graphs”

	Lights	Desk	Table	Monitor
Exhibit 1	✘	✓	✘	✘
Exhibit 2	✓	✓	✓	✘
Exhibit 3	✓	✘	✘	✓
Exhibit 4	✓	✓	✓	✓
Exhibit 5	✓	✓	✘	✘
Exhibit 6	✘	✘	✘	✓
Exhibit 7	✓	✓	✘	✓
Exhibit 8	✓	✘	✓	✘
Exhibit 9	✓	✘	✓	✘
Exhibit 10	✓	✓	✘	✘
Exhibit 11	✓	✓	✘	✓
Exhibit 12	✓	✘	✓	✓
Exhibit 13	✘	✘	✘	✘



Both rely on an established, repeated pattern to show data

e.g., Bar: repeating equal width rectangles along a scale of information

e.g., Comparison Chart: repeating Tick/Cross along a scale of information

Graphs: rely on X or Y or both axes to make sense. At least one of these axes is numeric. Graph draws correlation between these axes by plotting points along the grid

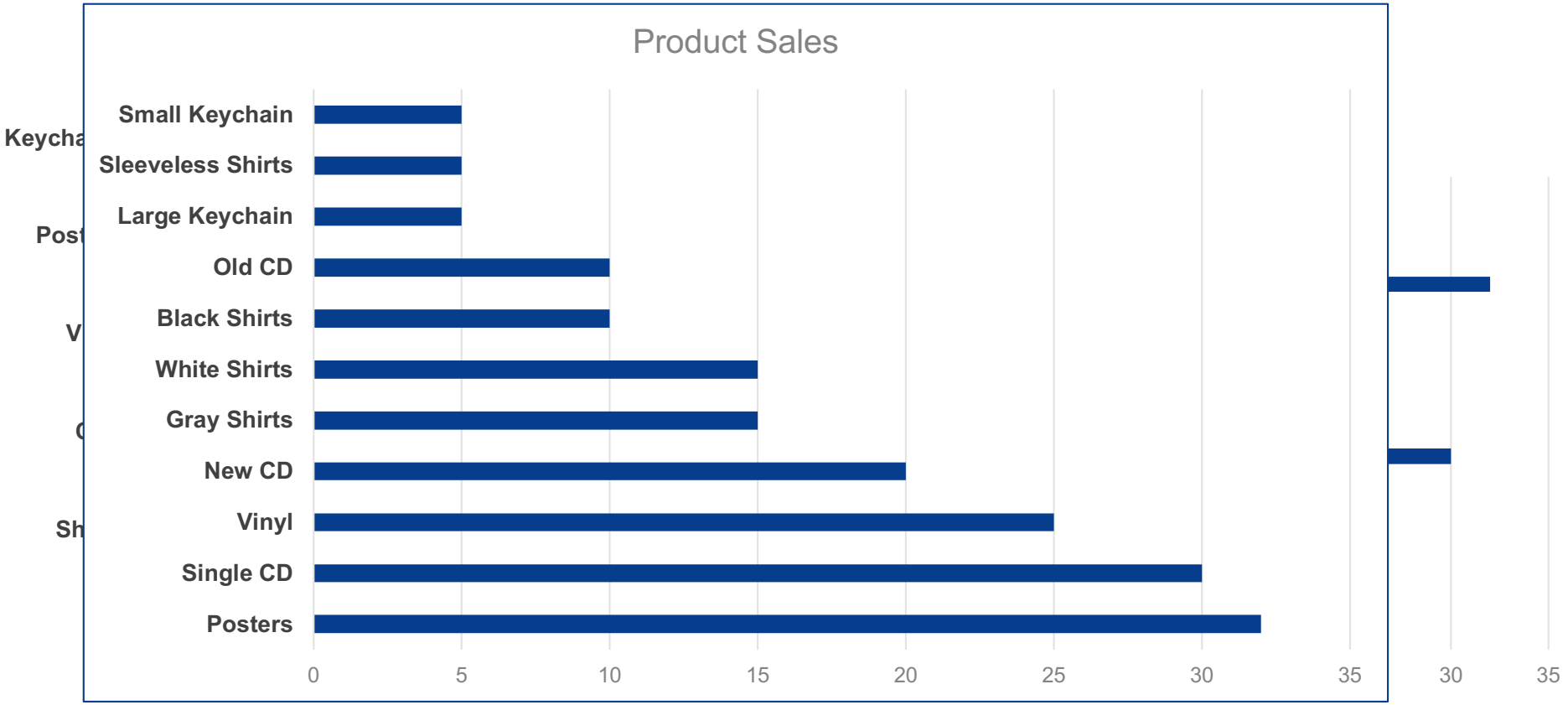
Charts: not restricted by X/Y axes, not necessarily numerical

Which ones are charts? Which ones are graphs?

Some of the basics … Organising data

Imagine a merchandise sales figure by an artist:

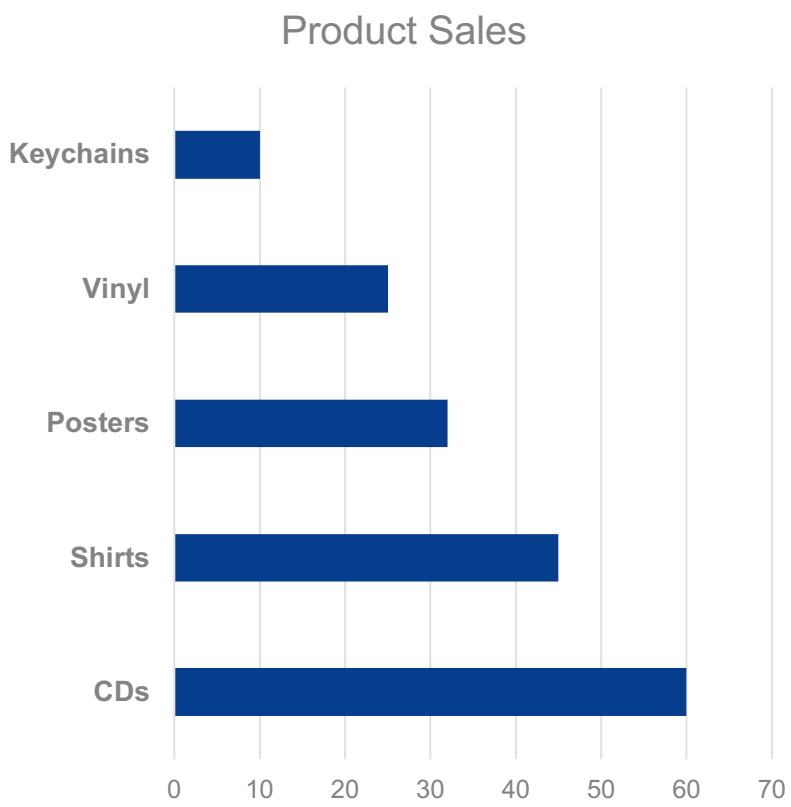
T-shirts: 45, CD: 60, Vinyl: 25, Posters: 32, Keychains: 10

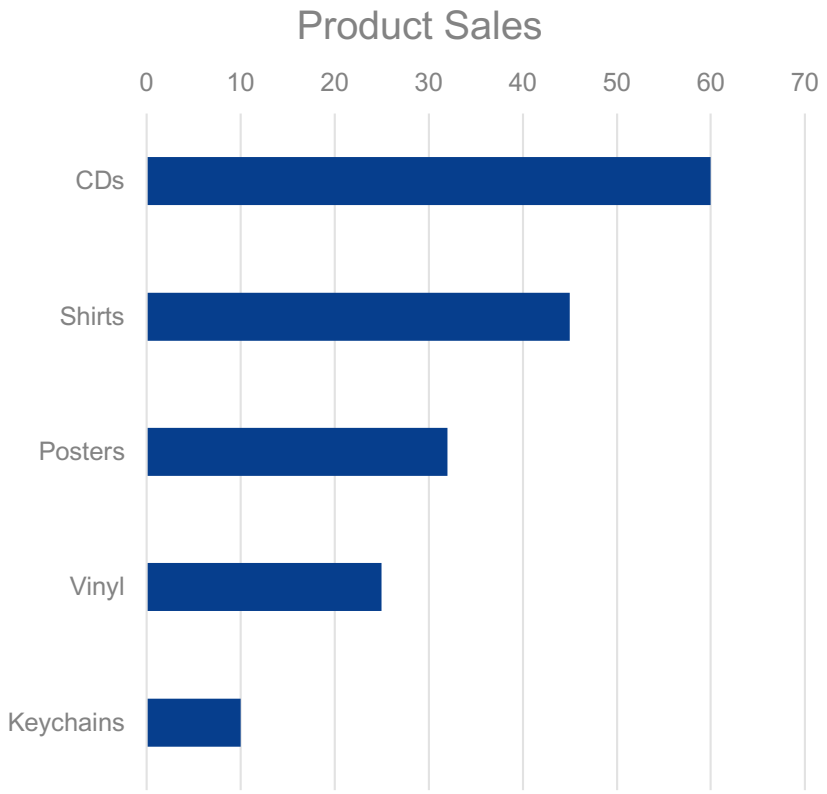


Data Visualization Best Practices: Killer Infographics by Amy Balliett,

Some of the basics … Organising data

When possible always order the data … but perception could be also tricky …



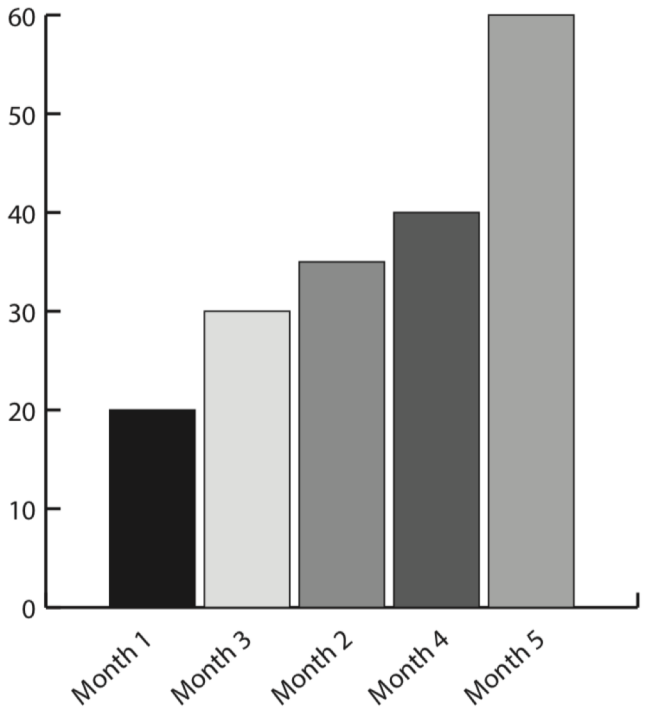


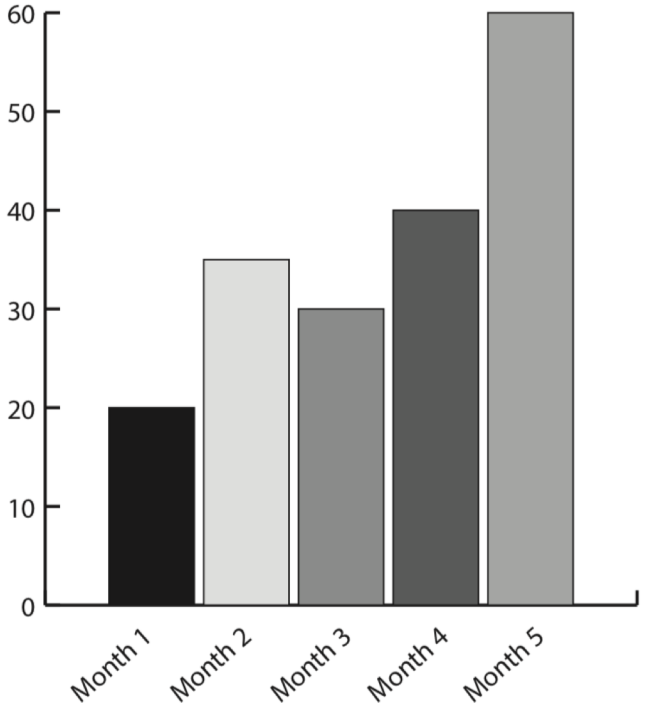
Subjective interpretation: most people read left to right …

Data Visualization Best Practices: Killer Infographics by Amy Balliett,

Some of the basics … Organising data

When possible always order the data … but perception could be also tricky …



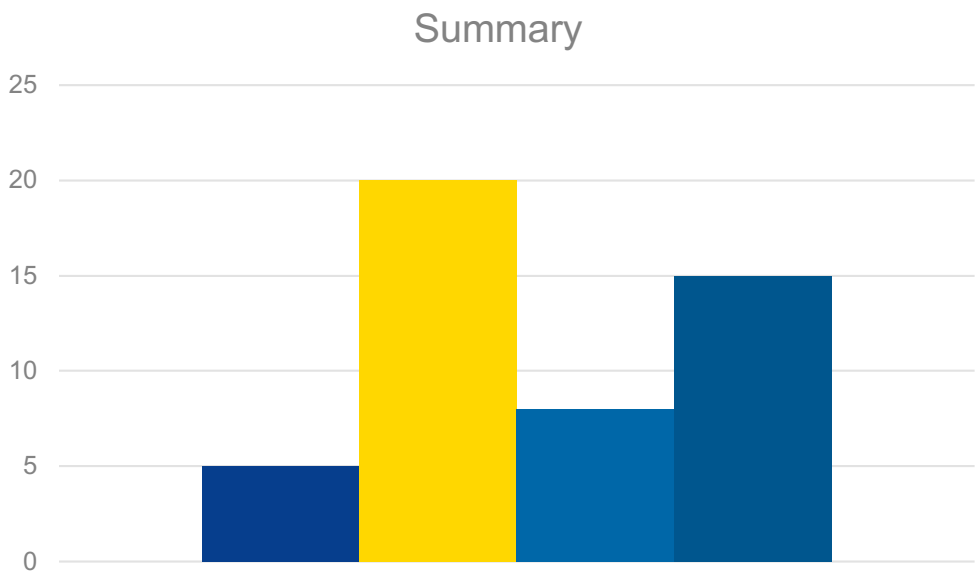


When a line follows a "timeline", stick to the timeline …

Data Visualization Best Practices: Killer Infographics by Amy Balliett,

Some of the basics … Colours important

Putting “Form (Prettiness)” before “Function”



Lots of colours could be confusing, distracting from the information

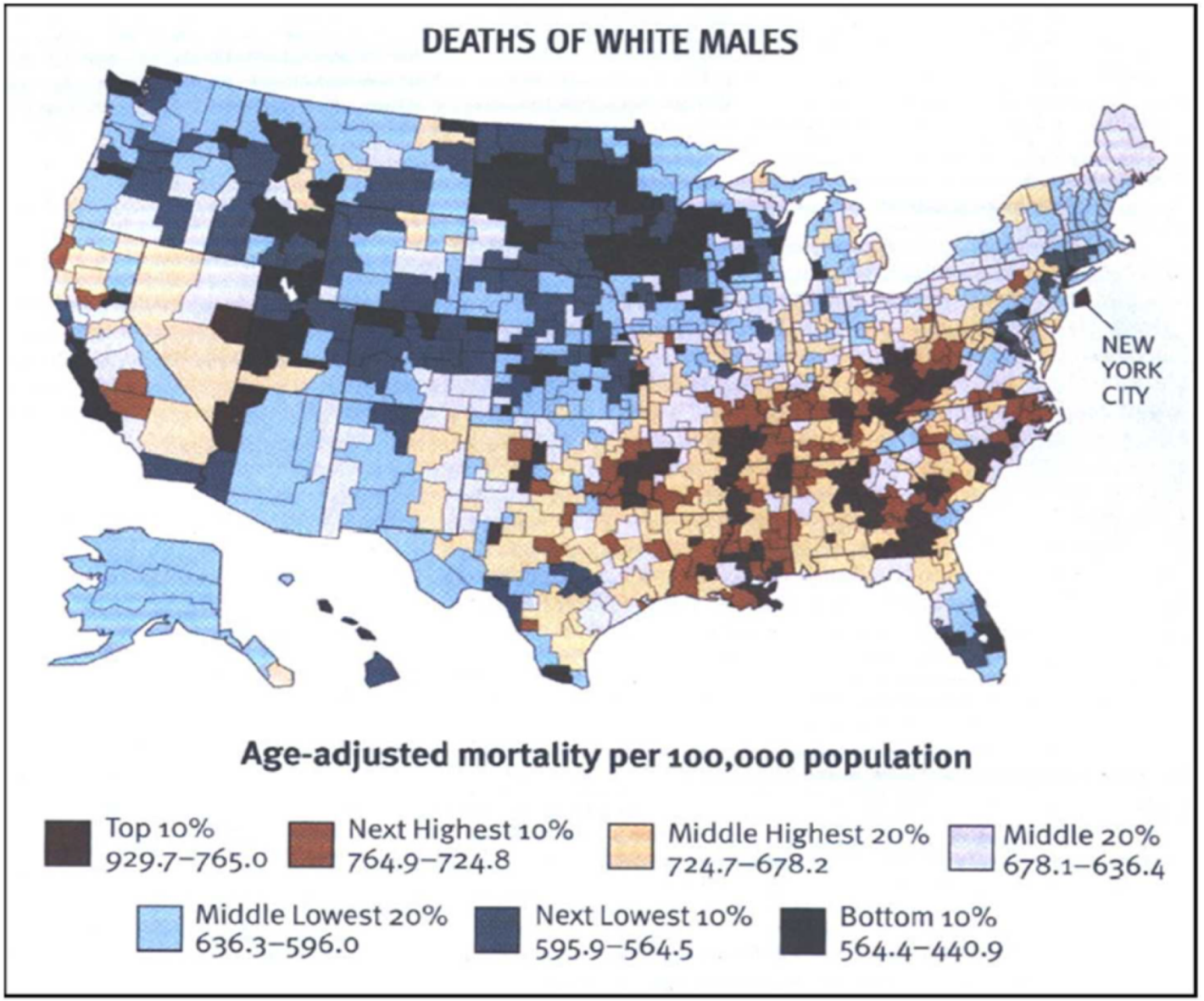
Just because you have millions of colors to choose from

doesn't mean you must use them all •••

Some of the basics … Colours important

Not just about limiting colours …

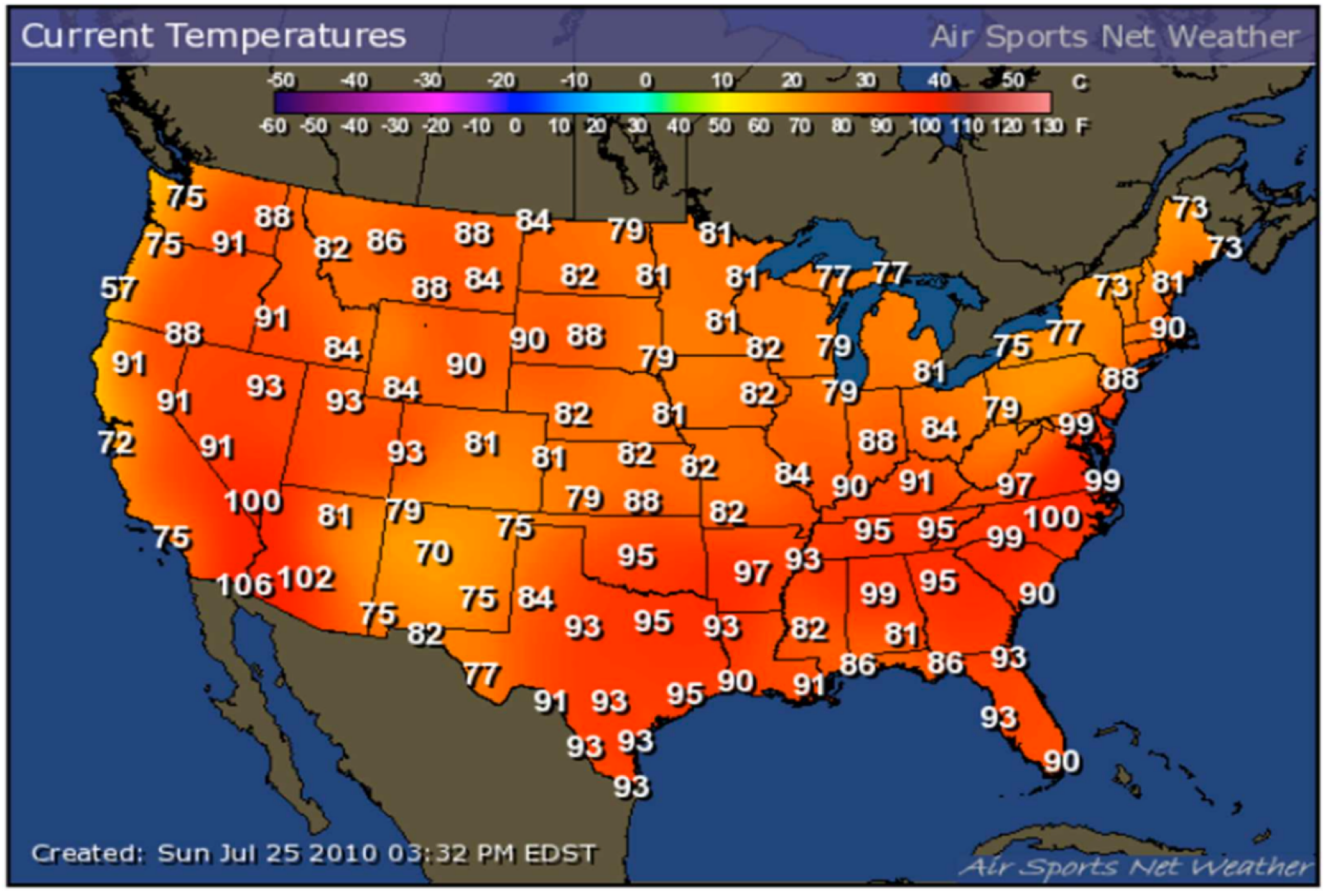
What's wrong with this colour map?



http://www2.cs.uh.edu/~chengu/Teaching/Fall2017/Visualization_fall2017.html

Some of the basics … Colours important

Not a bad choice of color scale,
but the Dynamic Range needs some work



http://www2.cs.uh.edu/~chengu/Teaching/Fall2017/Visualization_fall2017.html

Some of the basics … Colours important

Common advice on using colours:

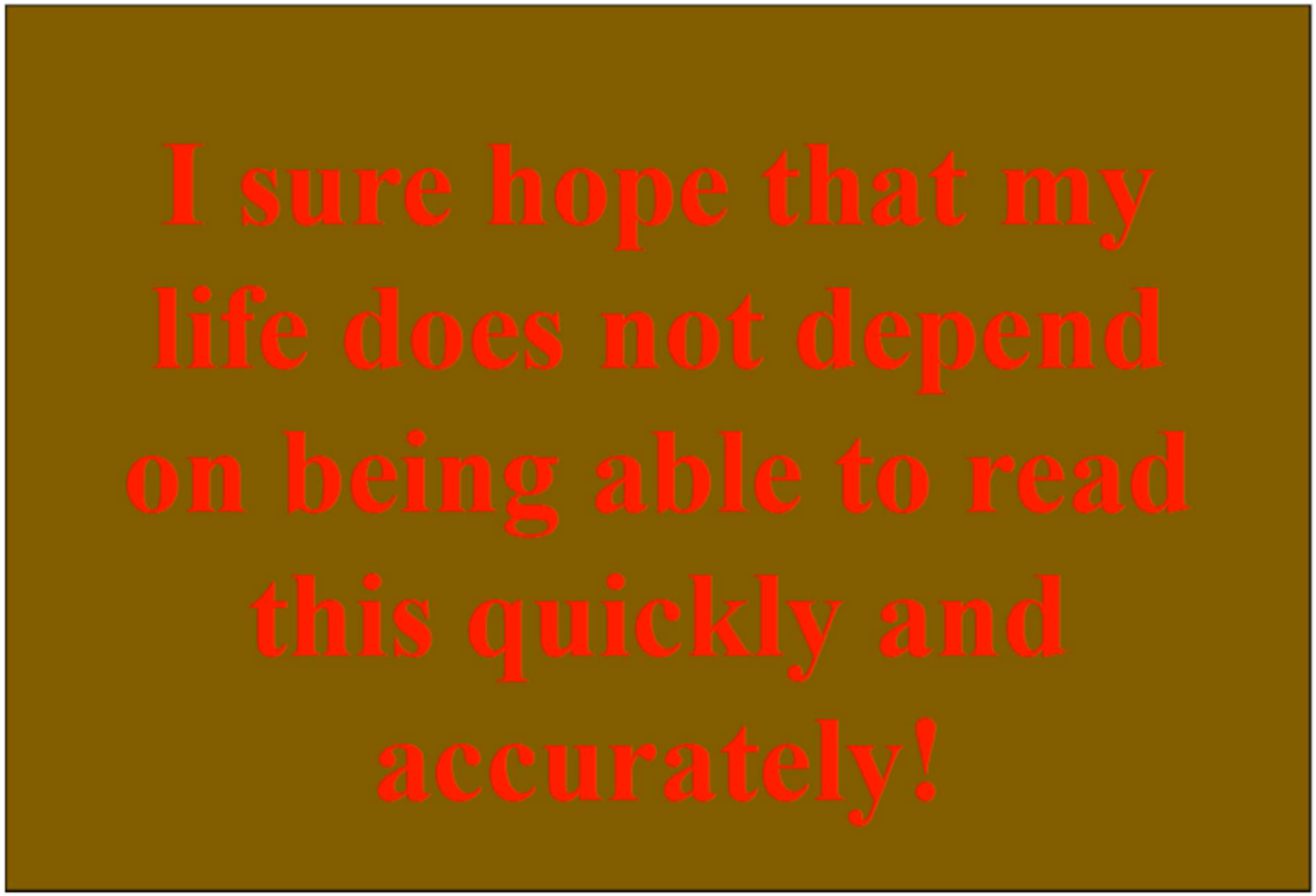
most people have strong association with pre-established colour meanings. Don't go against them.

Red	Green	Blue
Stop	On	Cool
Off	Plants	Safe
Dangerous	Carbon	Deep
Hot	Moving	Nitrogen
High stress	Money	
Oxygen		
Shallow		
Money loss		

http://www2.cs.uh.edu/~chengu/Teaching/Fall2017/Visualization_fall2017.html

Some of the basics … Colours important

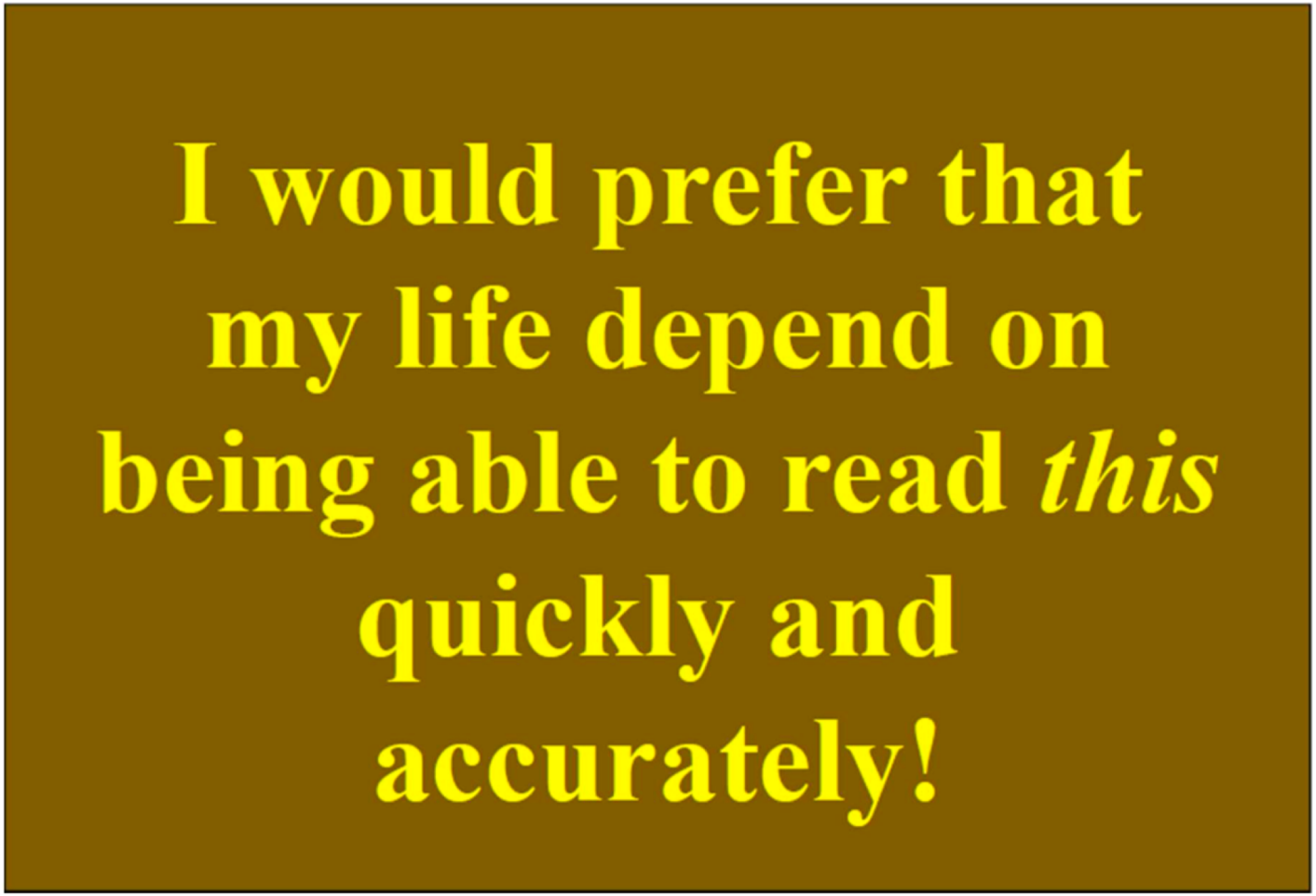
Just to make things interesting …. Colour alone is not the whole picture.



http://www2.cs.uh.edu/~chengu/Teaching/Fall2017/Visualization_fall2017.html

Some of the basics … Colours important

What about correct contrast choice?



http://www2.cs.uh.edu/~chengu/Teaching/Fall2017/Visualization_fall2017.html

Some of the basics … Colours important

Colour blindness is more common than we think … (close to 10% of generic population)

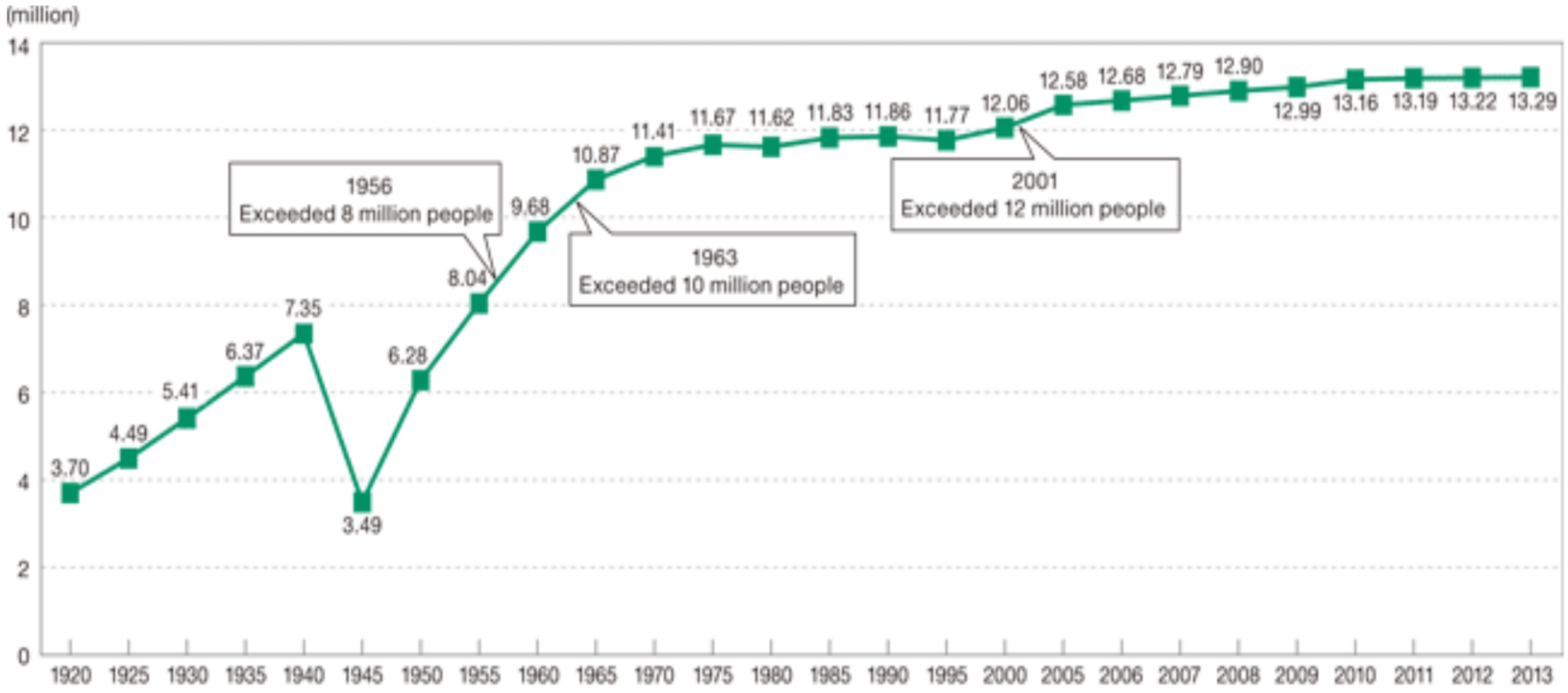
There are resources you could utilise:

e.g., http://colorbrewer2.org/

Tips on increasing accessibility of your visualisation:

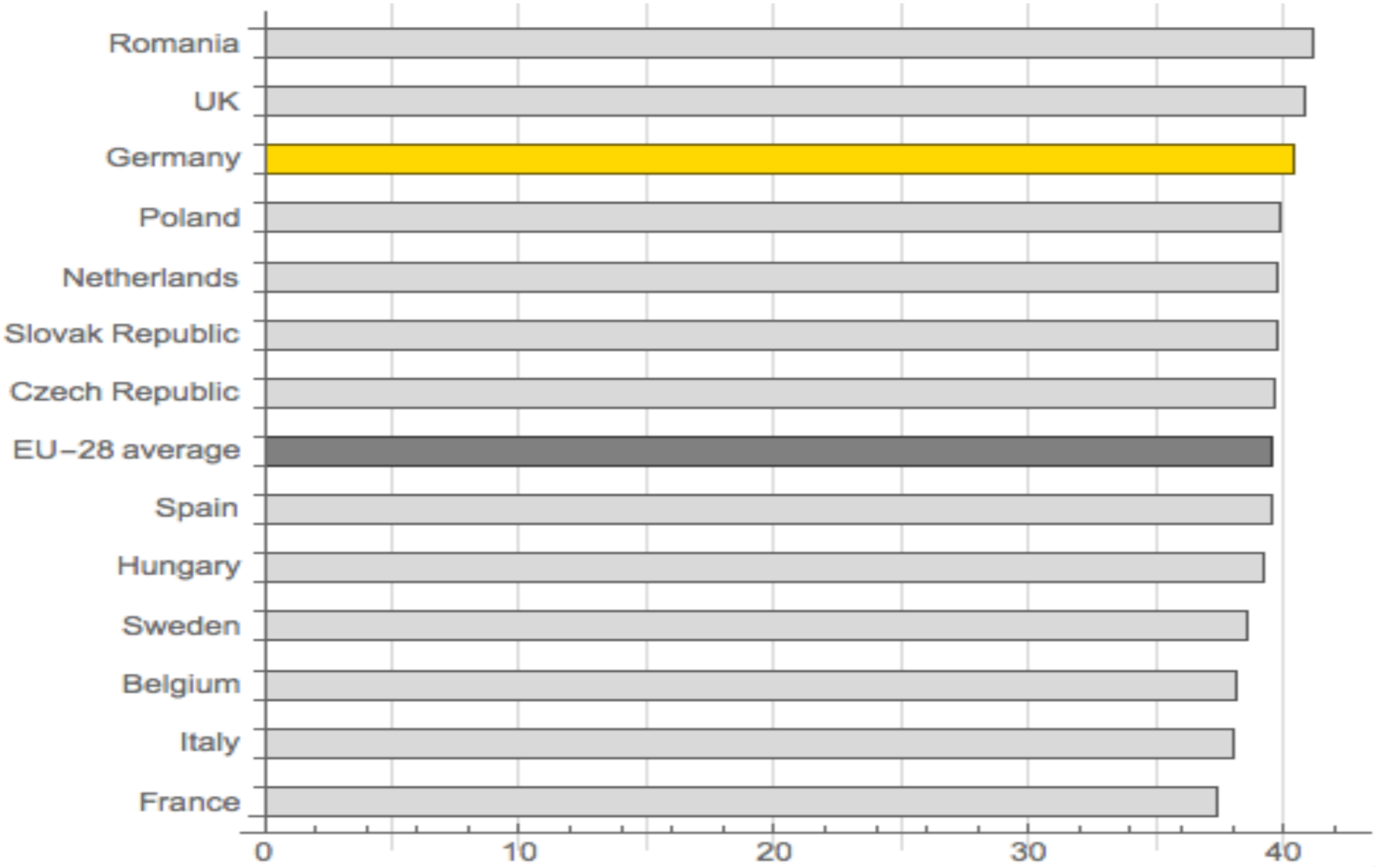
http://blog.usabilla.com/how-to-design-for-color-blindness/

Some of the basics … Keep Scales consistent



https://www.livestories.com/blog/five-ways-to-fail-data-visualization

Some of the basics … Keep scales consistent

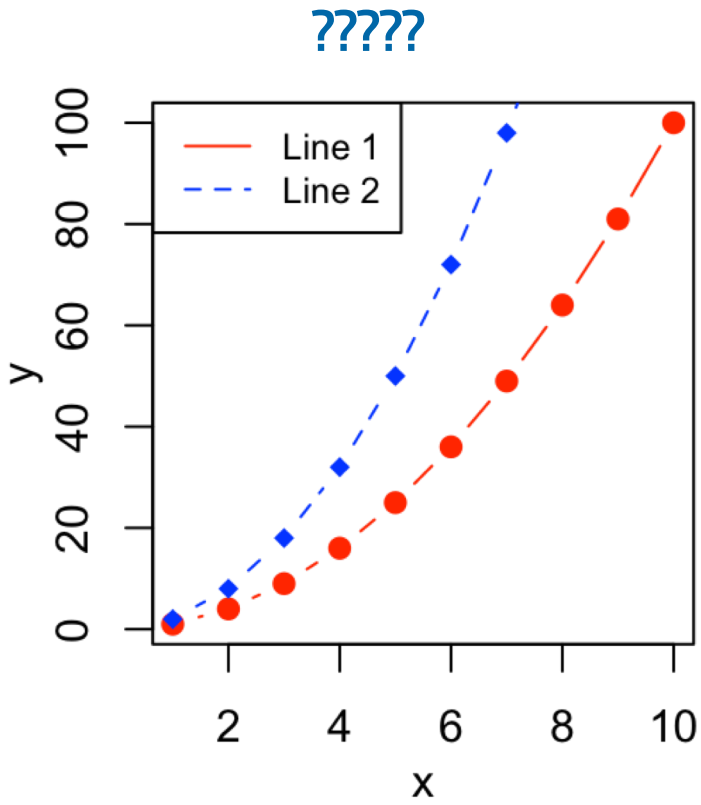


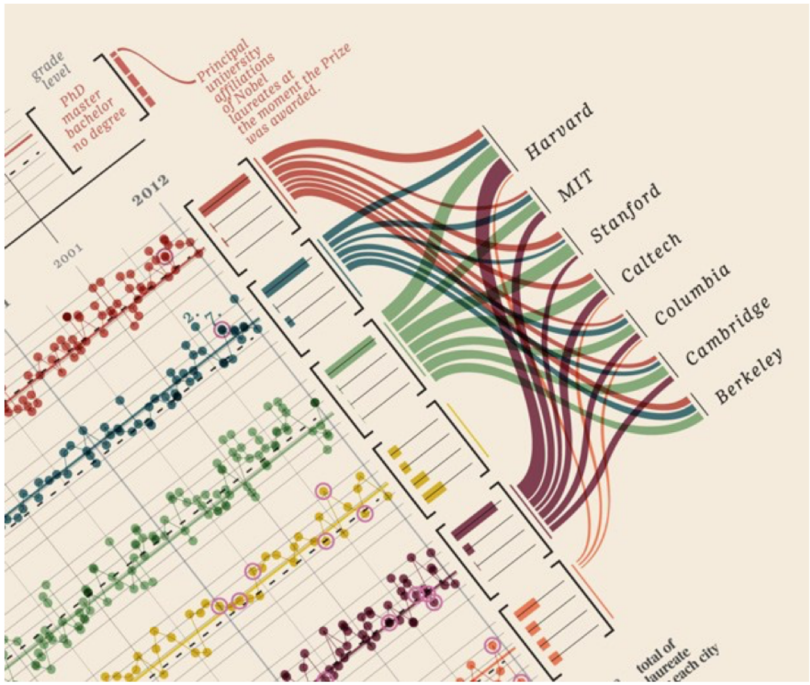
http://callingbullshit.org/tools/tools_misleading_axes.html

Some of the basics … Legend and Sources

It is absolutely necessary to include the sources of your data and correct legends to interpret your visualisation.

Without the legends, your visualisation is basically a “pretty picture” with no meaning





Preparing the data for visualisation

Data almost never comes in the exact form that you need it in

One of the common parts of the data visualisation tasks is to clean and convert the data

Some of the common data adjustments:

- Calculating indexes and ratios
- Calculating percentile
- Aggregating
- Regrouping
- Converting from Excel/CSV to JSON/XML/SQL

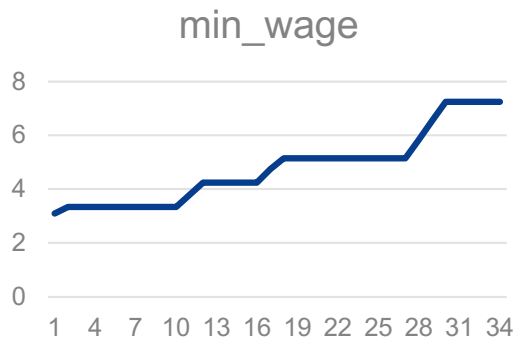
You may need simple tools, or database SQL scripting, programming scripting solutions for these.

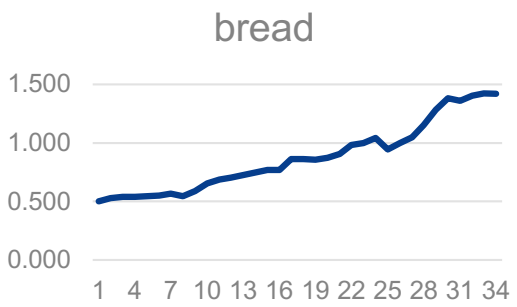
- Excel (available!)
- Tableau (commercial)
- Direct data manipulation with SQL or programming language

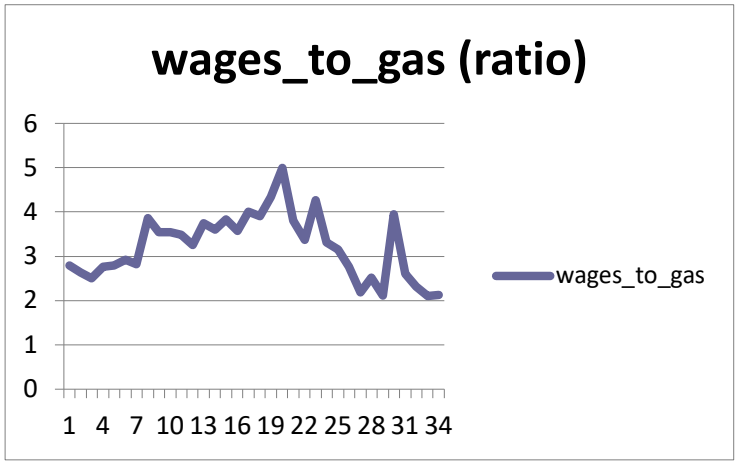
Preparing the data: Indexes and Ratios

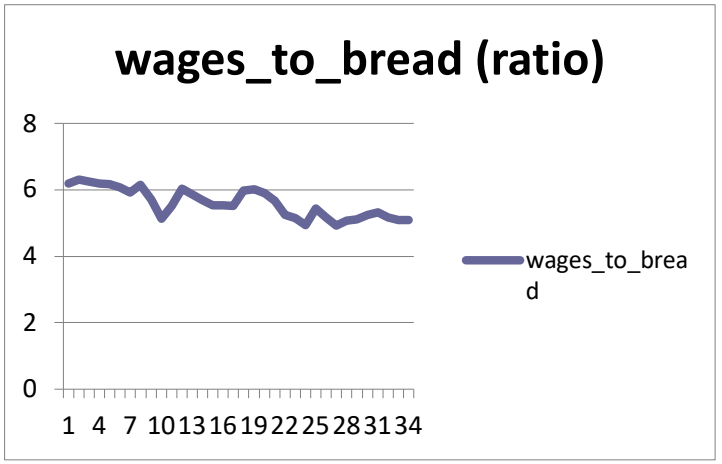
Are we comparing apples with apples (or are they apples and oranges?)

* Indexes and Ratios allow you to convert the data in a way that makes it easy to look at data side-by-side that is not necessarily easy to do in the original form









min_wage/gas
min_wage/bread

Preparing the data: Calculating Percentile

Calculating percentile makes it easier to compare numbers to each other as part of a whole
-- where you stand compared to the rest of the herd, ***relative standing***

	A	B	C	G
1	Country Name	2016	Rank	Percentile
2	United States	$ 18,624,475,000,000	1	99%
3	China	$ 11,199,145,157,649	2	99%
4	Japan	$ 4,940,158,776,617	3	98%
5	Germany	$ 3,477,796,274,497	4	98%
6	United Kingdom	$ 2,647,898,654,635	5	97%
7	France	$ 2,465,453,975,282	6	97%
8	India	$ 2,263,792,499,341	7	96%
9	Italy	$ 1,858,913,163,928	8	96%
10	Brazil	$ 1,796,186,586,414	9	95%
11	Canada	$ 1,529,760,492,201	10	95%
12	Korea, Rep.	$ 1,411,245,589,977	11	94%
13	Russian Federation	$ 1,283,162,985,989	12	94%
14	Spain	$ 1,237,255,019,654	13	93%
15	Australia	$ 1,204,616,439,828	14	93%
16	Mexico	$ 1,046,922,702,461	15	92%
17	Indonesia	$ 932,259,177,765	16	92%
18	Turkey	$ 863,711,710,427	17	91%
19	Netherlands	$ 777,227,541,581	18	91%
20	Switzerland	$ 668,851,296,244	19	90%

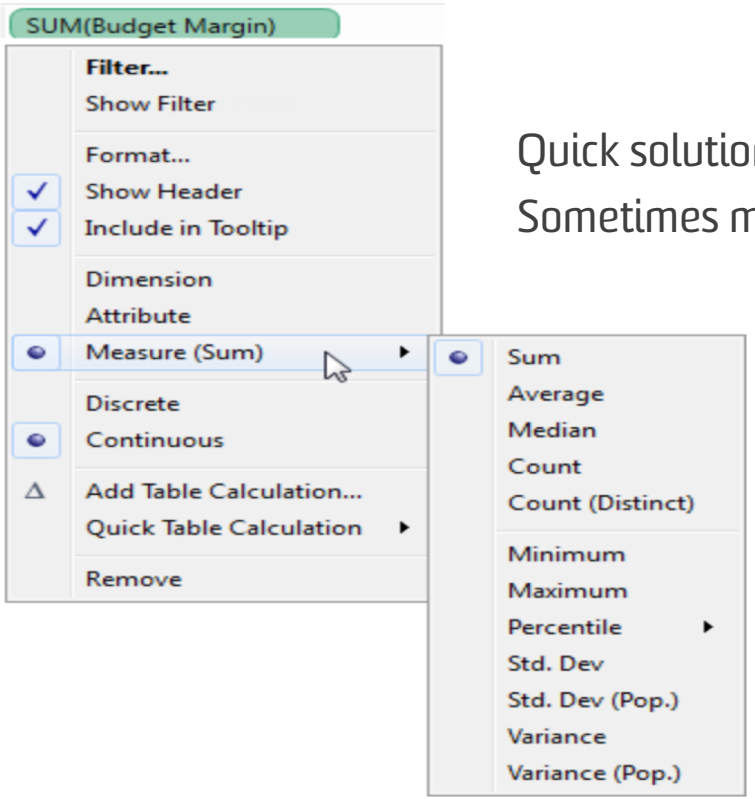
1-(ranking/total county)

1-(1/191)
1-(2/191)
1-(3/191)
...

Preparing the data: Aggregating, Converting Data

Data aggregation is the process where raw data is gathered and expressed in a summary form for statistical analysis

For example, raw data can be aggregated over a given time period to provide statistics such as ***average, minimum, maximum, sum, and count.***



Quick solution by a tool like Tableau/Excel …

Sometimes manual SQL scripts necessary

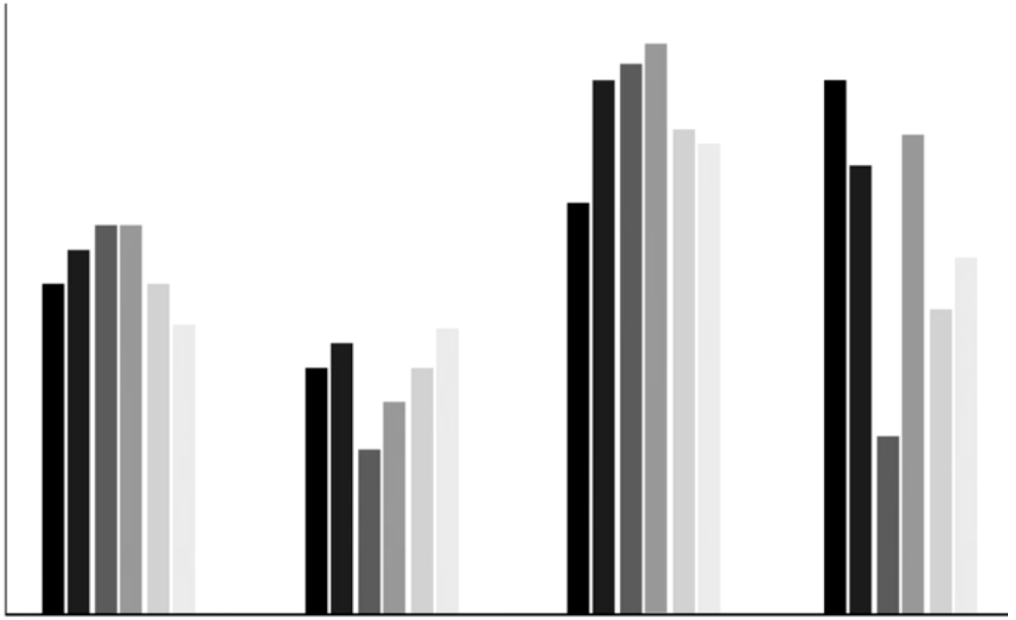
The right paradigm

One of the most difficult things to do is figuring out which charts/graphs to use in which situation.

I'd say you need to have the basic competency here – and then be aware of good alternatives.

The good old BAR graphs:

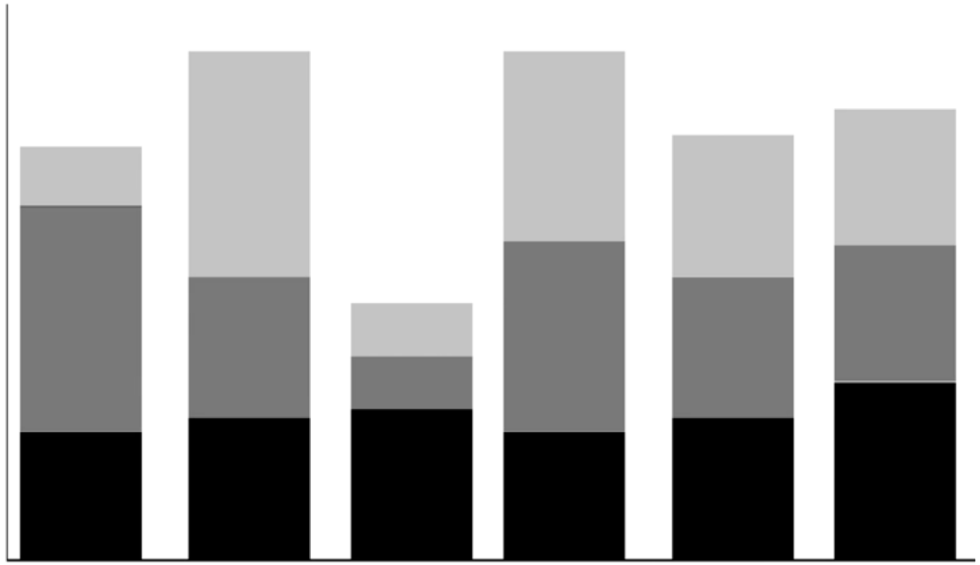
- Highly effective in terms of 'parsing the information'
- Experts say human brain is wired to differentiate these rectangular shapes
- In fact, you should start by asking "why isn't a bar graph enough here?"
- But when there are more variables, many "grouped" bars don't look good



The right paradigm

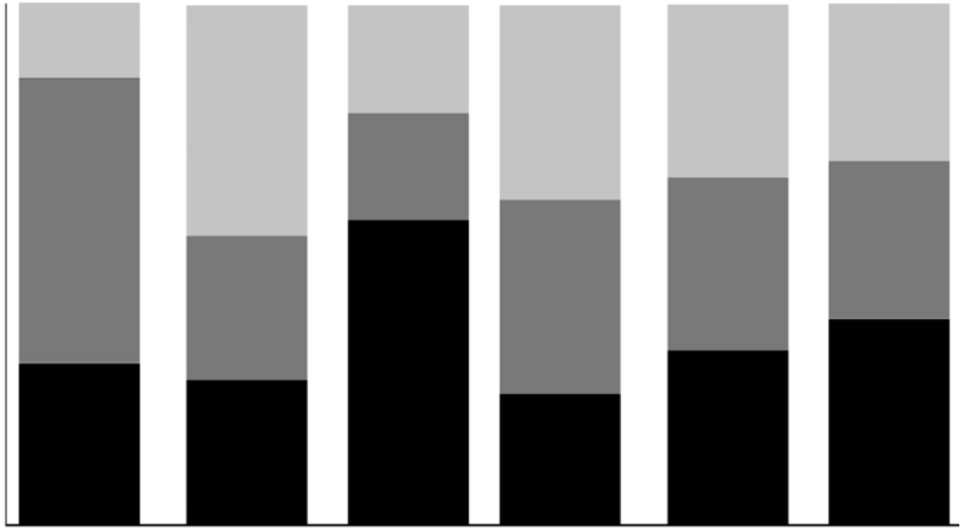
STACKED Bar graphs:

- Compare data within groups
- Whole bar represents the total value of that group, and each segment represents the value within the group



STACKED Percentage Bar graphs:

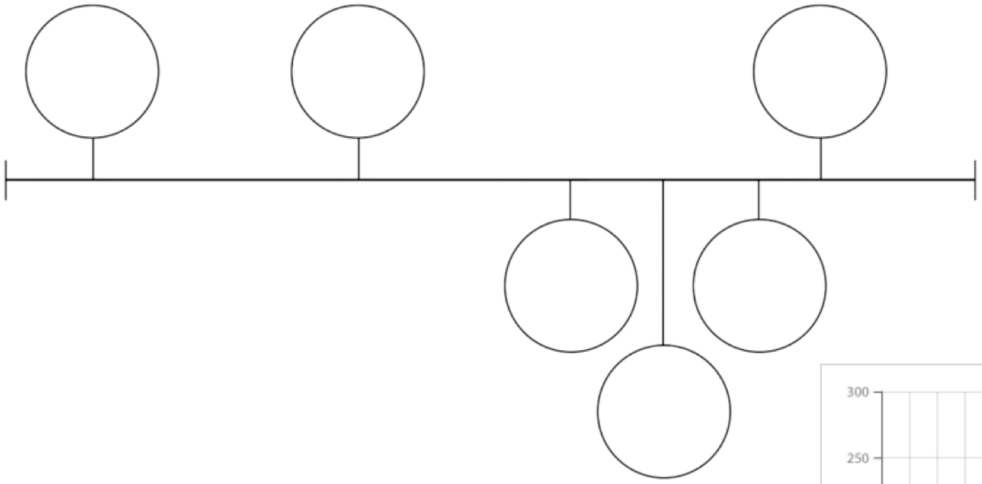
- If you want to compare relative contribution of each category to the whole ….
- Whole bar = 100%
- Showing relative strength of each category within the whole

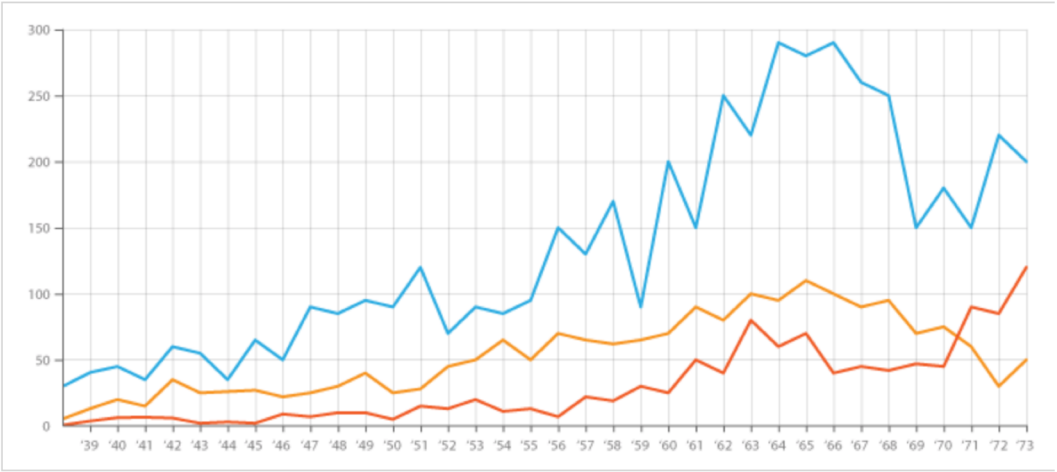


The right paradigm

Line graphs:

- To show values “over time” or a continuous interval
- Stories over “Timeline”

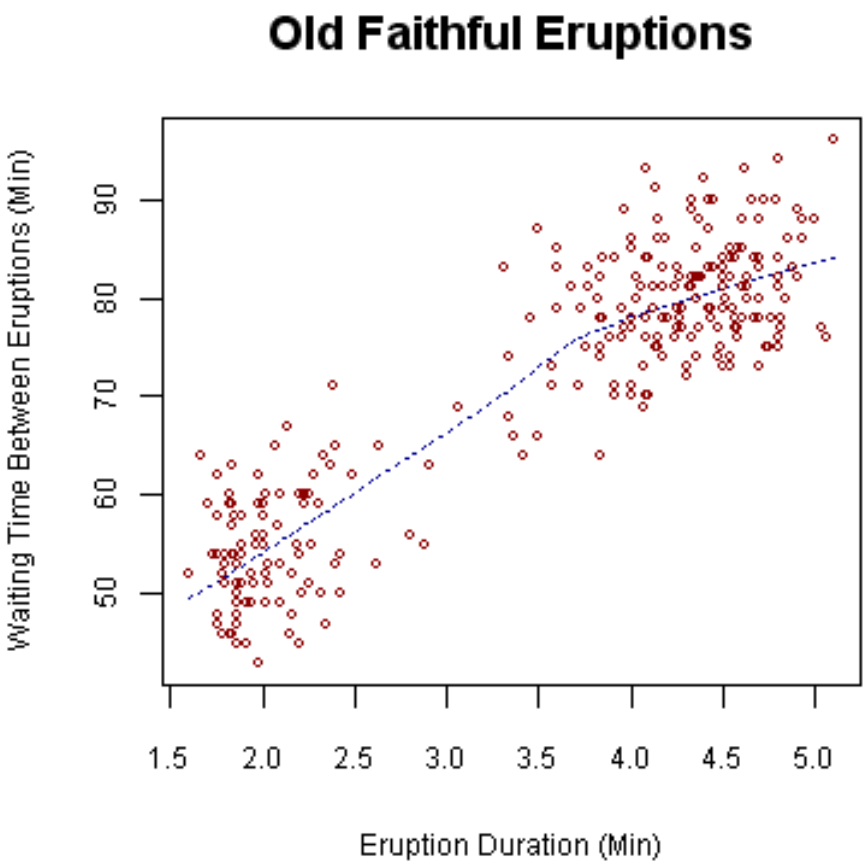




The right paradigm

Scatter Plot graphs:

- To show two variables and their correlations (i.e., X axis vs. Y axis)



Waiting time between eruptions and the duration of the eruption for the Old Faithful Geyser in Yellowstone National Park, Wyoming, USA. This chart suggests there are generally two "types" of eruptions: short-wait-short-duration, and long-wait-long-duration

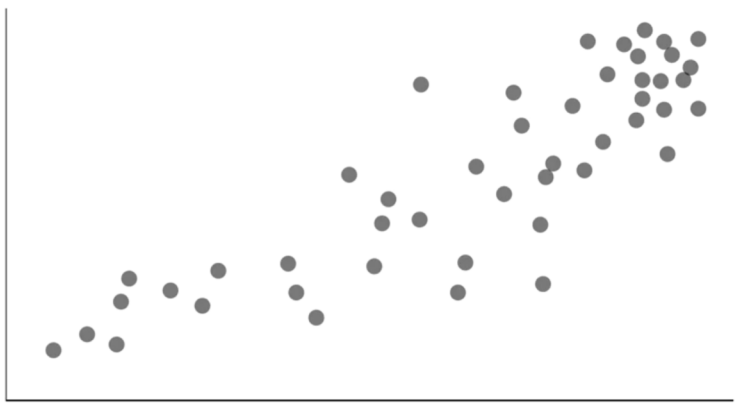
https://en.wikipedia.org/wiki/Scatter_plot

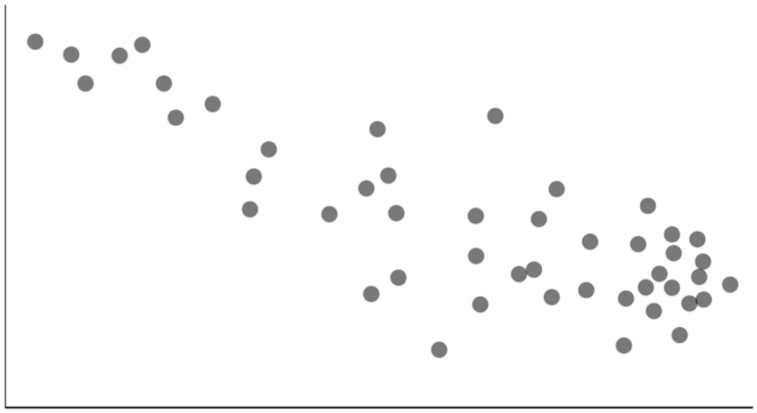
Learning Data Visualization by Bill Shanders, Lynda.com

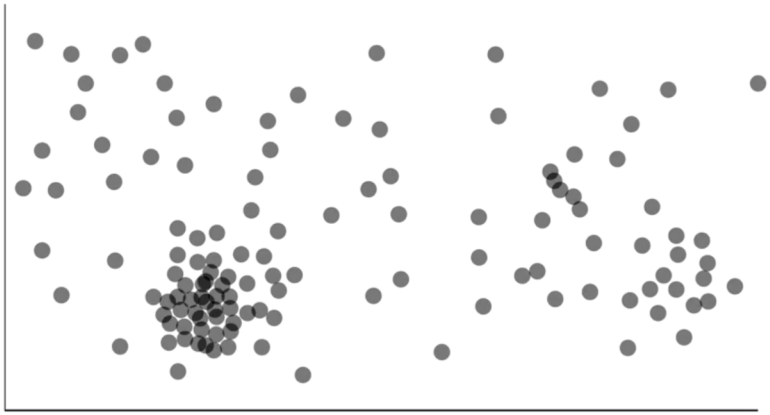
The right paradigm

Scatter Plot graphs:

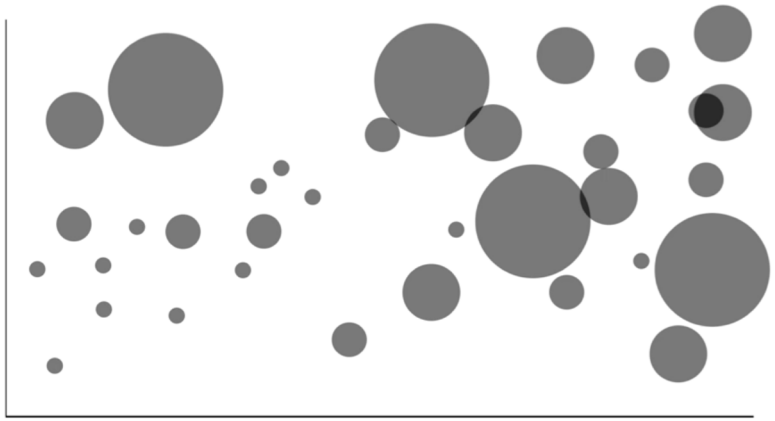
- To show two variables and their correlations (i.e., X axis vs. Y axis)







No correlations, but discernible patterns

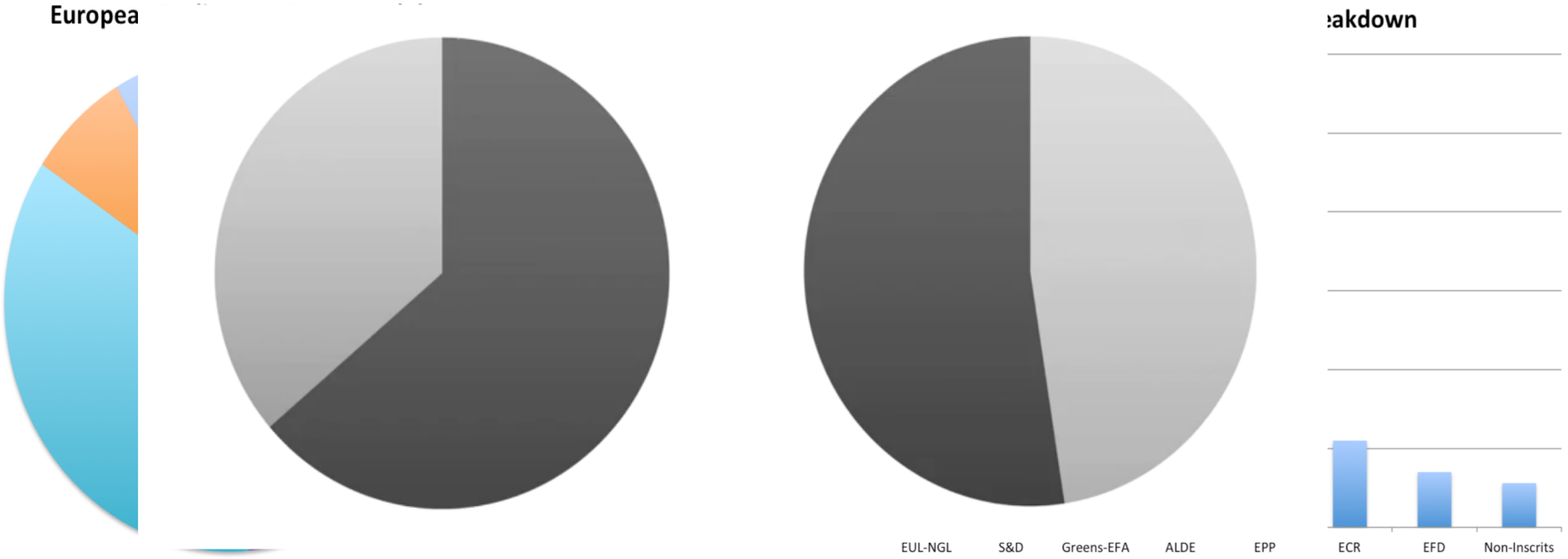


Size of the dots – third variable

The right paradigm

Pie Charts:

- Although commonly used, not considered an effective form. Human brain is not wired to parse round shape areas and arcs
- Normally other graphs can do the same job (e.g., BAR graphs)
- Maybe OK when showing two variables (<, >, similar, etc.)



The right paradigm: hierarchical data

To show the “connections” between and the hierarchy of objects.

A tree diagram:

http://mbostock.github.io/d3/talk/20111018/tree.html

- Default for showing hierarchy (e.g., org chart)
- Any situation where you a parent, which has children (and grand children)

A node link diagram:

http://mbostock.github.io/d3/talk/20111116/force-collapsible.html

- Showing a lot of links between objects

Tree map:

http://mbostock.github.io/d3/talk/20111018/treemap.html

- Size of each category

Chord Diagram:

https://bost.ocks.org/mike/uberdata/

Complex data -> very difficult to parse the information (e.g., between dots far apart) Interactivity could help parse (https://bost.ocks.org/mike/fisheye/)

The right paradigm: showing data on maps

On an existing map API like Google API …

Place markers (https://www.latlong.net)

- specific location (e.g., building), centre of a region

Layers (data associated with the regions on a map)

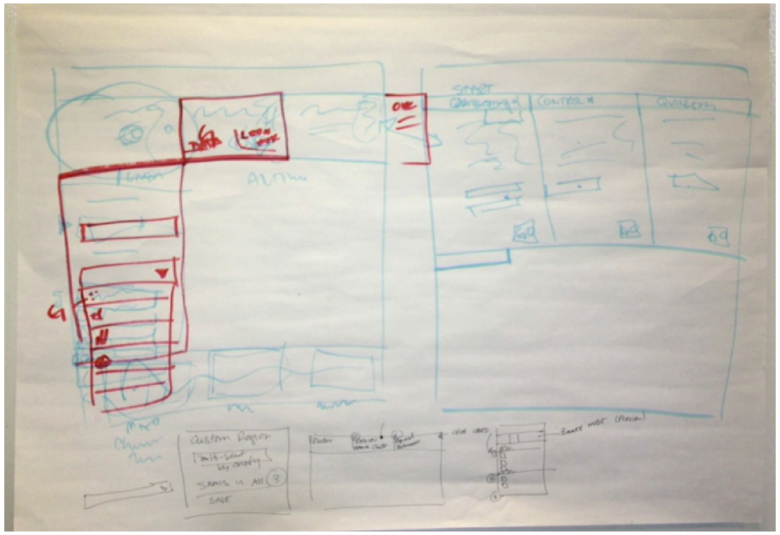
- Point clustering (http://bl.ocks.org/andrewxhill/raw/8360694/)
  - display aggregated number/data points per region
- Choropleth map (http://leafletjs.com/examples/choropleth/)
  - display divided geographical areas or regions that are coloured, shaded or patterned in relation to a data variable.
- Heat map (https://onemilliontweetmap.com/)
- Flow map
  - show the movement of information or objects from one location to another and their amount (thickness of lines, colours)
  - https://datavizcatalogue.com/methods/flow_map.html
  - https://www.iom.int/world-migration

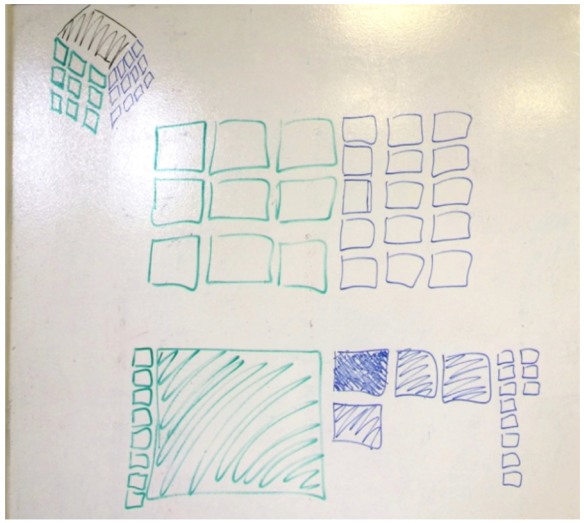
To put it together …

So … there are many many options for visualising data (including a lot of fancy and interactive ones from the latest tools and libraries)

But let's try to have some basic competency on this:

- Accuracy is important, having a clear story to tell is important
- You need to be ready to do some basic data prep and pre analysis before visualisation
- Knowing the right paradigm (form) to use for the story
- Aware of your own limitation as 'non-expert' (visualisation is not easy)

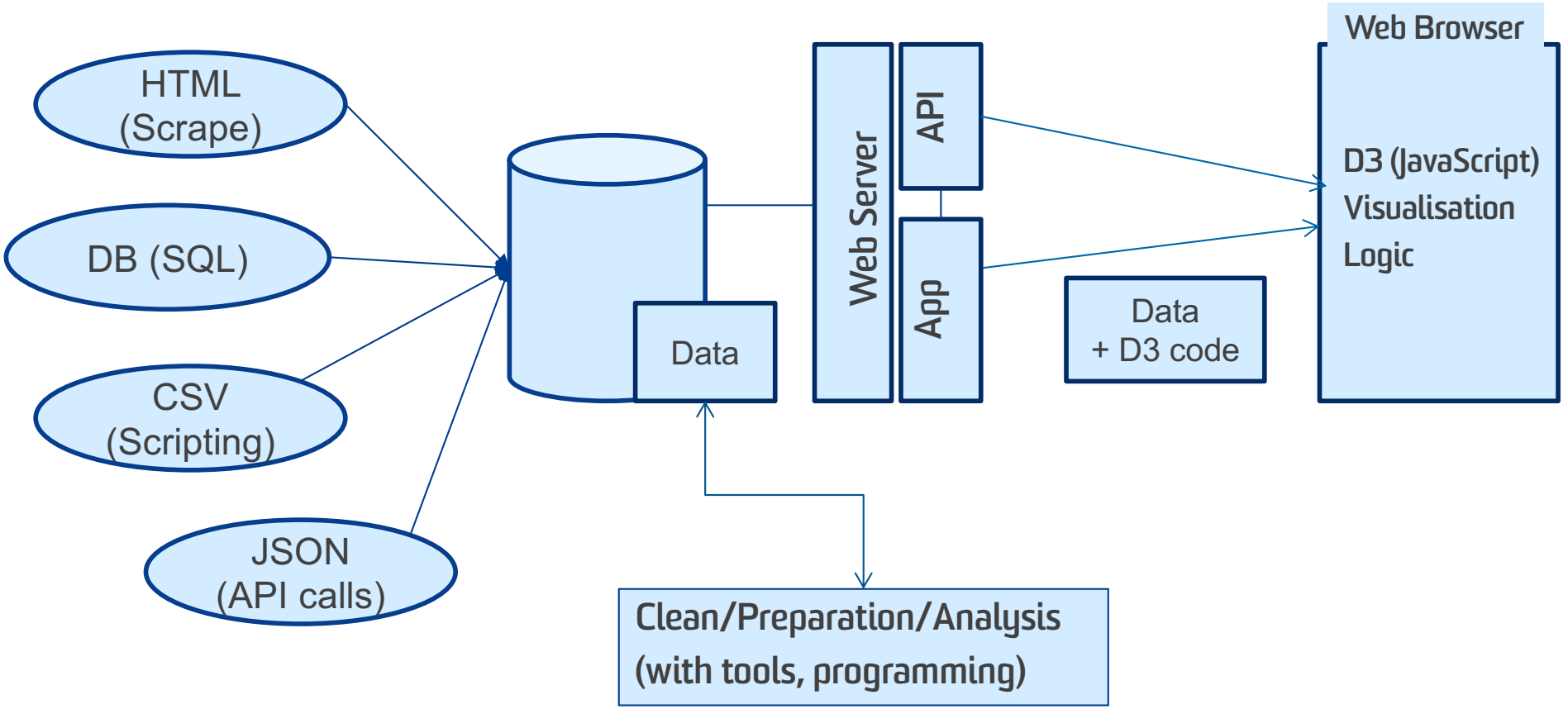




Actually, a lot of experts recommend "sketching the idea out" with pen and paper.

D3 – JavaScript Library for DataVIz

Where does this sit?



D3 – Data Driven Documents

- D3 a JavaScript library for manipulating Web documents (i.e., DOM) based on data – “manipulation” is to do with adding visual elements to the documents.

(INSIDE INDEX.HTML)

```
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <title>D3 </title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/style.css">
</head>

<body>

    <!-- Viz Area -->
    <div class="container">
        <div class="row">
            <div id="visualisation-area"></div>
        </div>
    </div>

<!-- D3 JS libraries -->
<script src="js/d3.min.js"></script>
<!-- Your D3 Code -->
<script src="js/main.js"></script>

</body>
</html>
```

(INSIDE JS/MAIN.JS)

```
var svg = d3.select("#visualisation-area").append("svg")
    .attr("width", 400)
    .attr("height", 400);

d3.json("datafiles/ages.json", function(error, data){
    if (error) throw error;

    data.forEach(function(d){
        d.age = +d.age;
    })

    var circles = svg.selectAll("circle")
        .data(data);

    circles.enter()
        .append("circle")
            .attr("cx", function(d, i){
                return (i * 50) + 25;
            })
            .attr("cy", 25)
            .attr("r", function(d){
                return d.age * 2;
            })
            .attr("fill", function(d){
                if (d.age < 25) {
                    return "blue";
                }
                else {
                    return "red";
                }
            });
});
```

D3 – Data Driven Documents

- D3 work with web standards (HTML, CSS and SVG) for visual elements - gives you the full capabilities of modern browsers without tying yourself to a proprietary framework
- Very well defined “process” for manipulation
  - Step 1: Load data into the browser’s memory
  - Step 2: Bind data to “virtual” elements to the document
  - Step 3: Transform the elements, set various visual properties and make them “appear” on the document
  - Step 4: Transition the elements between states to create dynamic visual effects

- The “D3 process” is the core idea … everything else is detailed techniques that enhance the core idea

SVG (Scalable Vector Graphics)

D3 is most useful when used to work with visual elements that are 'Scalable Vector Graphics (SVG)

- W3C Recommendation (version 1.1)
- More reliable, visually consistent across browsers, and faster (small size), vector-based (do not lose quality when resized)

SVG is a text-based image format … Each SVG image is defined using markup code (like XML/HTML), and it is included directly within any HTML document.

All modern browsers "support" (i.e., understand and can render the text format properly).

Every SVG element is a "canvas" on which your visual elements like circles are rendered.

```
<svg width="500" height="500">


</svg>
```

```
<svg width="500" height="500">
  <rect x="50" y="20" width="150" height="150"
    fill="blue"/>
</svg>
```

SVG (Scalable Vector Graphics)

```
<svg width="400" height="60">

</svg>
```

Let's look at a few examples of basic shapes (and SVG "canvas" - coordinates)

https://www.w3schools.com/graphics/svg_circle.asp

https://www.w3schools.com/graphics/svg_rect.asp

https://www.w3schools.com/graphics/svg_polygon.asp (can build maps)

https://www.w3schools.com/graphics/svg_text.asp (can draw text)

Core D3 concept …

- Placing and selecting visual elements onto the document
- D3 select – returns a page/document element

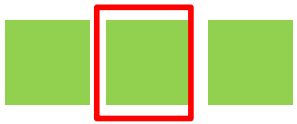
(INSIDE INDEX.HTML)

```
<svg width="400" height="60">
    <rect class="outside" x="0" y="0" width="50" height="50" fill="green"></rect>
    <rect id="center" x="60" y="0" width="50" height="50" fill="green"></rect>
    <rect class="outside" x="120" y="0" width="50" height="50" fill="green"></rect>
</svg>
```

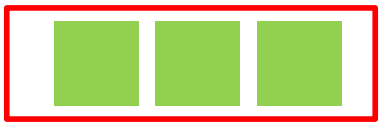
(INSIDE D3 Code)

d3.select("rect")

d3.select("#center")



d3.selectAll("rect")



d3.selectAll(".outside")

Browser …

https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept

- Placing and selecting visual elements onto the document
- D3 Append – add visual elements to the selected document element

(INSIDE INDEX.HTML)

```
<svg id="canvas" width="500" height="500">
</svg>
```

(INSIDE D3 Code)

```
var svg = d3.select("#canvas")

var rect = svg.append("rect")

rect.attr("x", 25)
rect.attr("y", 0)
rect.attr("width", 150)
rect.attr("height", 60)
rect.attr("fill", "blue")
```

```
// conventional ...method chains
var rect = d3.select("#canvas")
      .append("rect")
            .attr("x", 25)
            .attr("y", 0)
            .attr("width", 150)
            .attr("height", 60)
            .attr("fill", "blue")
```


https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept …

- Placing and selecting visual elements onto the document
- Now we can add the SVG (canvas) and some visual elements into INDEX.HTML

(INSIDE INDEX.HTML)

```
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <title>D3 </title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/style.css">
</head>

<body>

    <!-- Viz Area -->
    <div class="container">
        <div class="row">
            <div id="visualisation-area"></div>
        </div>
    </div>

<!-- D3 JS libraries -->
<script src="js/d3.min.js"></script>
<!-- Your D3 Code -->
<script src="js/main.js"></script>

</body>
</html>
```

(INSIDE JS/MAIN.JS)

```
var svg = d3.select("#visualisation-area").append("svg")
    .attr("width", 400)
    .attr("height", 400);

var circle = svg.append("circle")
    .attr("cx", 100)
    .attr("cy", 250)
    .attr("r", 70)
    .attr("fill", "grey");
```


https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept -- data joins/binding

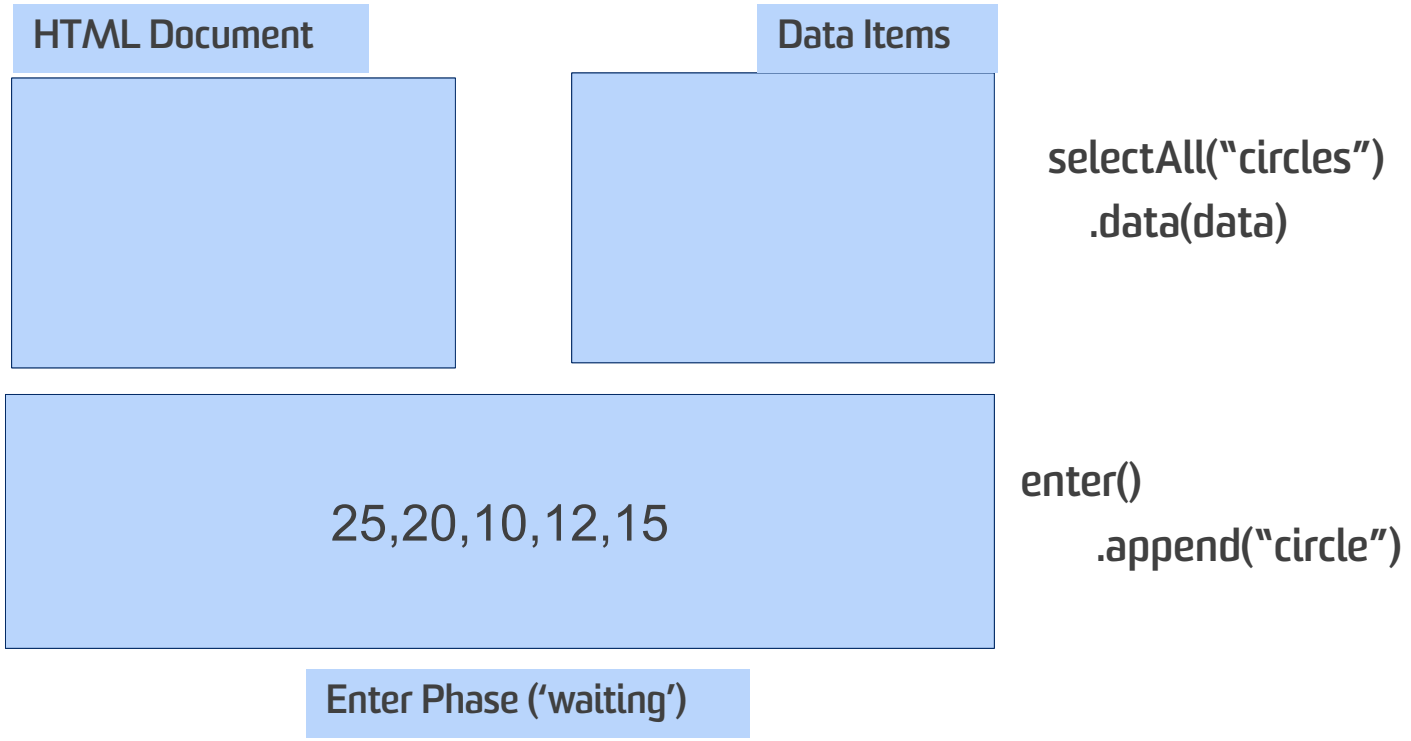
- Associating visual elements with data (the whole point of D3!)
- D3 specific concepts: selectAll ”virtual elements”, enter() function

```
var data = [25, 20, 10, 12, 15];

var svg = d3.select("#visualisation-area").append("svg")
    .attr("width", 400)
    .attr("height", 400);

var circles = svg.selectAll("circle")
    .data(data);

circles.enter()
    .append("circle")
        .attr("cx", function(d, i){
            return 25;
        })
        .attr("cy", function(d){
            return 25;
        })
        .attr("r", function(d){
            return 25;
        })
        .attr("fill", "red");
```


https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept -- data joins/binding

- Associating visual elements to data (the whole point of D3!)
- D3 specific concepts: ”virtual elements”, enter() function



Core D3 concept – loading data (external)

- Loading external data (from external files, including URL – AJAX API calls)
- the same origin policy applies …
- Most common: CSV, TSV or JSON,

```
d3.json("data/mark.json", function(error, data){
    if (error) throw error;

    console.log(data);

    data.forEach(function(d){
      d.mark = +d.mark;
    })
```

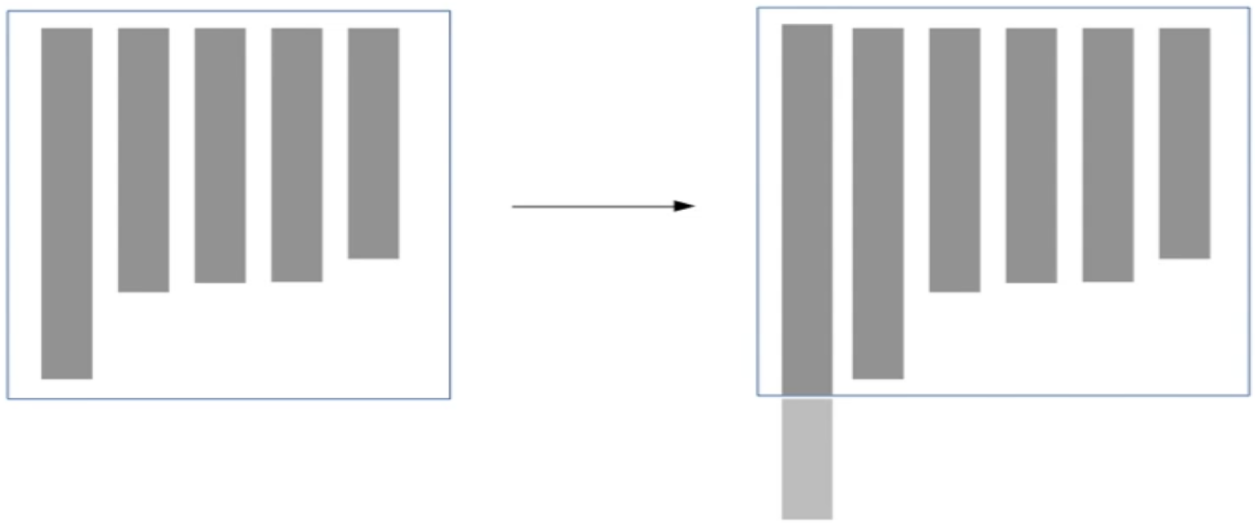
```
d3.csv("data/mark.csv", function(error, data){
    if (error) throw error;
```

```
d3.tsv("data/mark.tsv", function(error, data){
    if (error) throw error;
```


https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept – Scales

- Load data, bind them with visual elements using selectAll/enter, place them on the document … then?



Adding more data items?

Data items with bigger max values than you anticipated ?

Data items are not likely to be fixed (not static) …

How do you cater for the values so that when they are visualised, they do not end up out side of the 'canvas' range?

Scales are functions that map from an input domain to an output range …

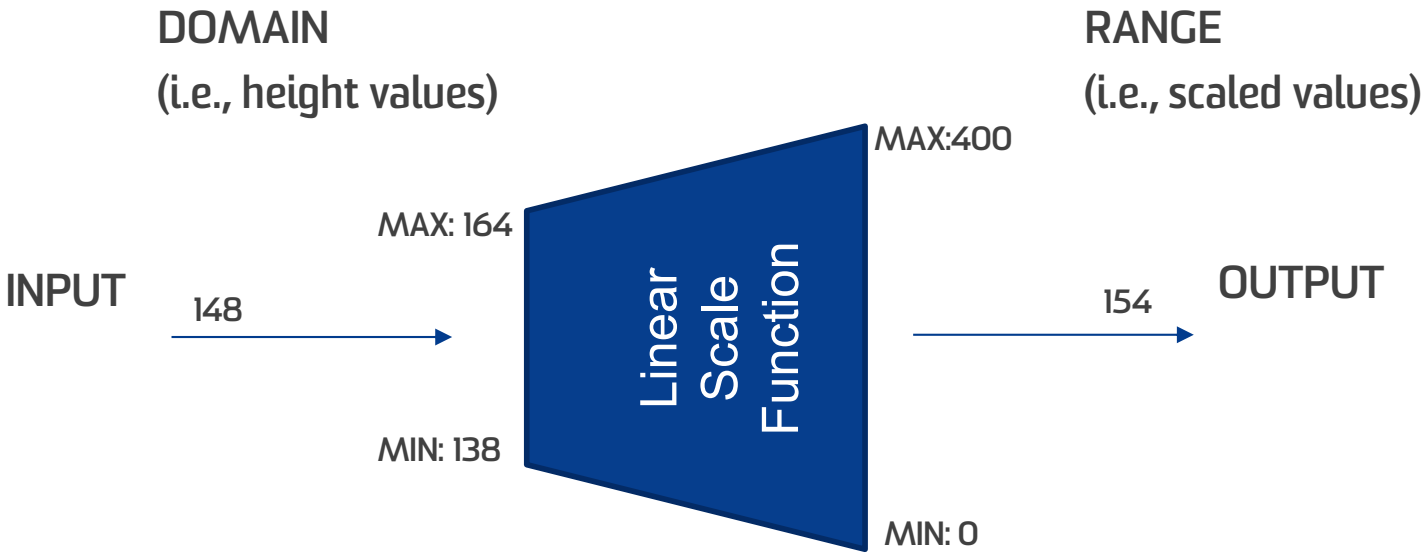
Create a scale function in D3, then pass a data value to it, it returns a scaled output value

Core D3 concept – Scales

- Let's look at the most common scale – Linear Scale

```
var data0 = [
        { name: "Tom", height: 138 },
        { name: "Jessica", height: 153 },
        { name: "Annie", height: 148 },
        { name: "Richard", height: 164 },
        { name: "John", height: 162 },
        { name: "Andrew", height: 143 }
]
```

Input Domain (range of possible input values)
Output Range (range of possible output values)



https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept – Scales

```
[
    {
        "name": "Tom",
        "height": 138
    },
    {
        "name": "Jessica",
        "height": 153
    },
    {
        "name": "Annie",
        "height": 148
    },
    {
        "name": "Richard",
        "height": 164
    },
    {
        "name": "John",
        "height": 162
    },
    {
        "name": "Andrew",
        "height": 143
    },
    {
        "name": "Sean",
        "height": 101
    }
]
```

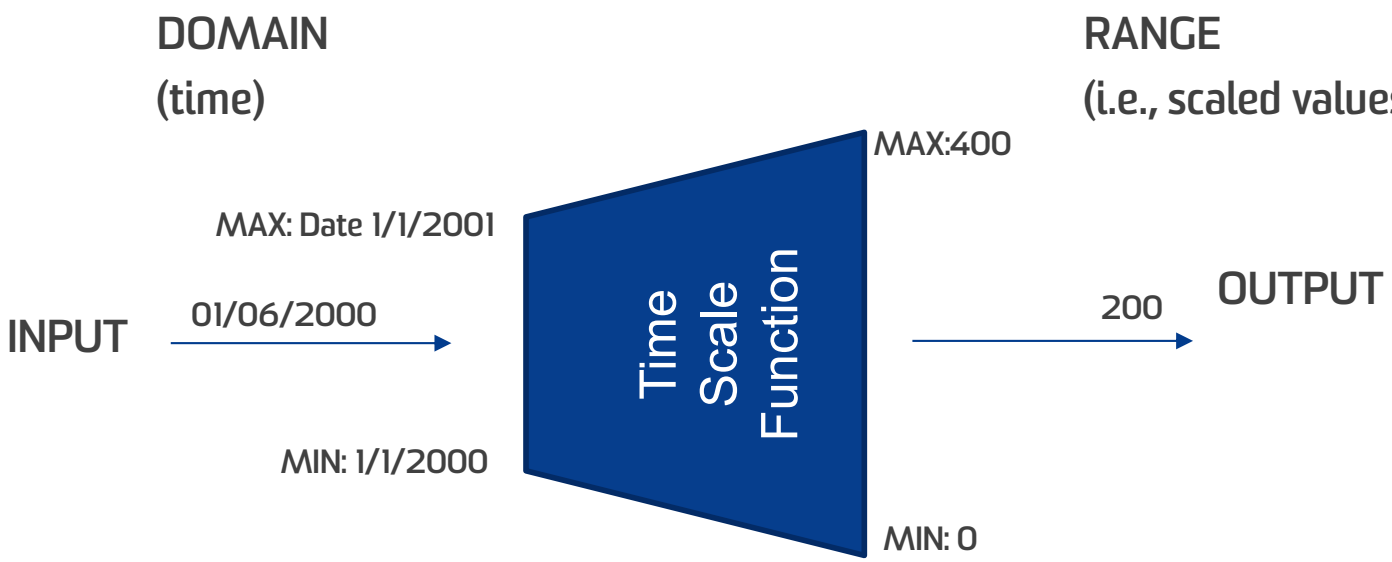
```
var svg = d3.select("#visualisation-area")
    .append("svg")
        .attr("width", "400")
        .attr("height", "400");

d3.json("data/people.json", function(data){
    console.log(data);

    var y = d3.scaleLinear()
        .domain([0, d3.max(data, function(d){
            return d.height;
        })])
        .range([0, 400]);

    console.log(y(138)) // 336.58
    console.log(y(101)) // 236.34
    console.log(y(162)) // 395.12

    console.log(y.invert(336.58)) // 137.99
```

Later, when you set the height of a rectangle, use "y(d.height)"

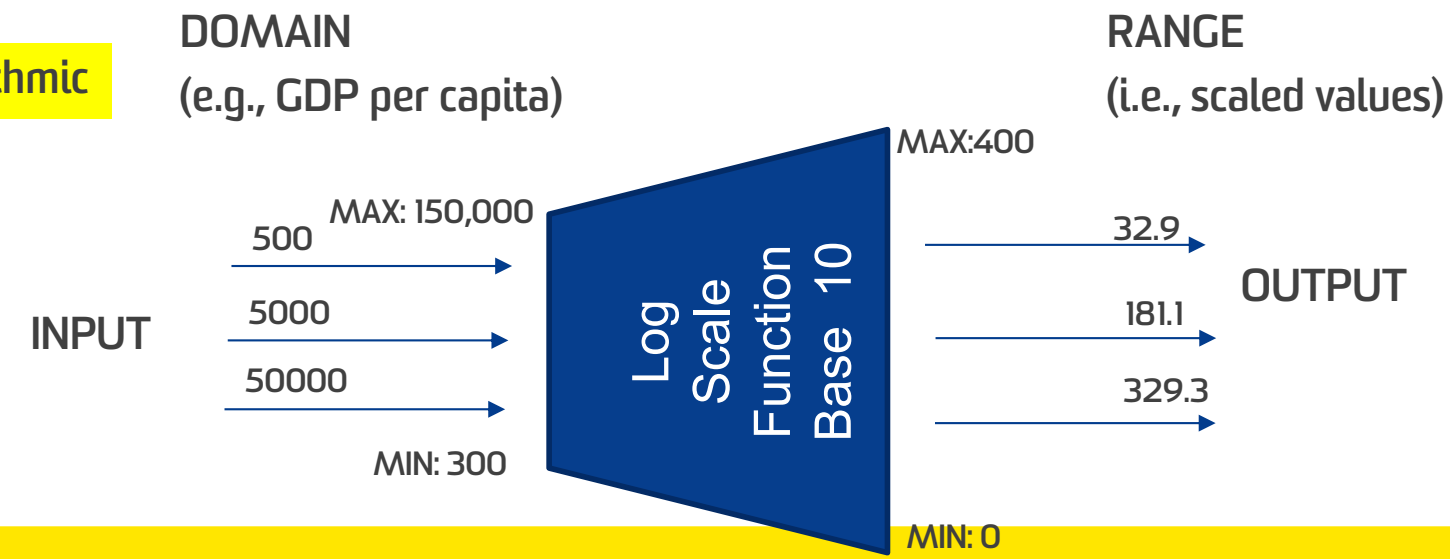
https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept – Other common scales

Time



Logarithmic



https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept – Other common scales

For Discrete domain input and discrete output range
e.g., when you want to associate a category to a colour

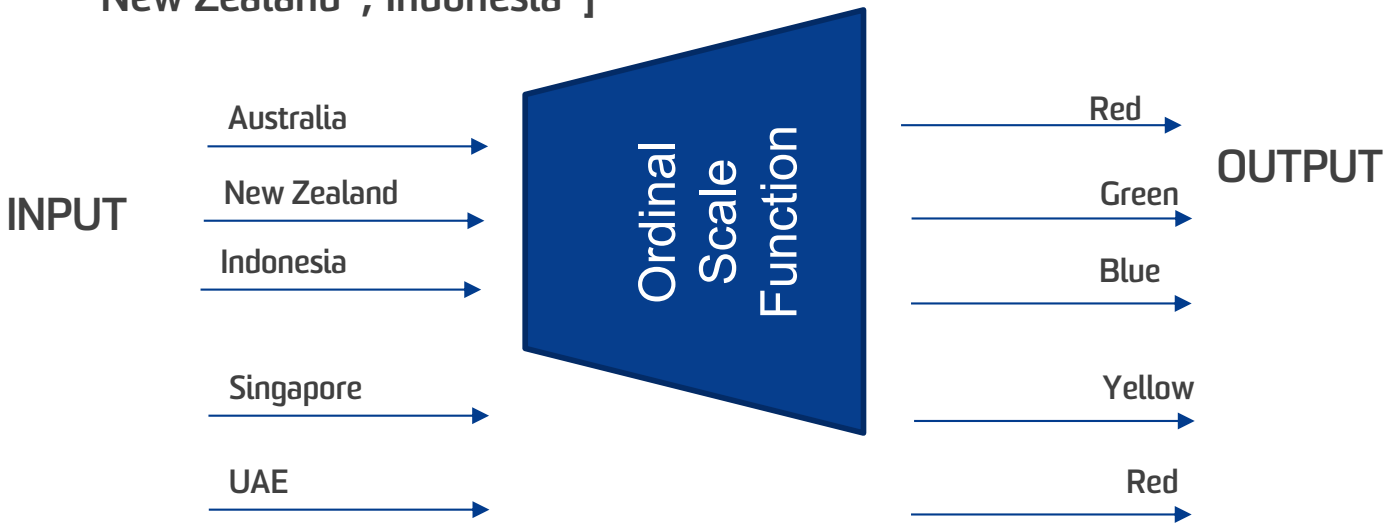
Ordinal

DOMAIN
(e.g., Country Names)

["Australia","Brazil","China",
"New Zealand","Indonesia"]

RANGE
(i.e., colours)

["Red","Orange","Indigo",
"Green","Blue", "Yellow"]

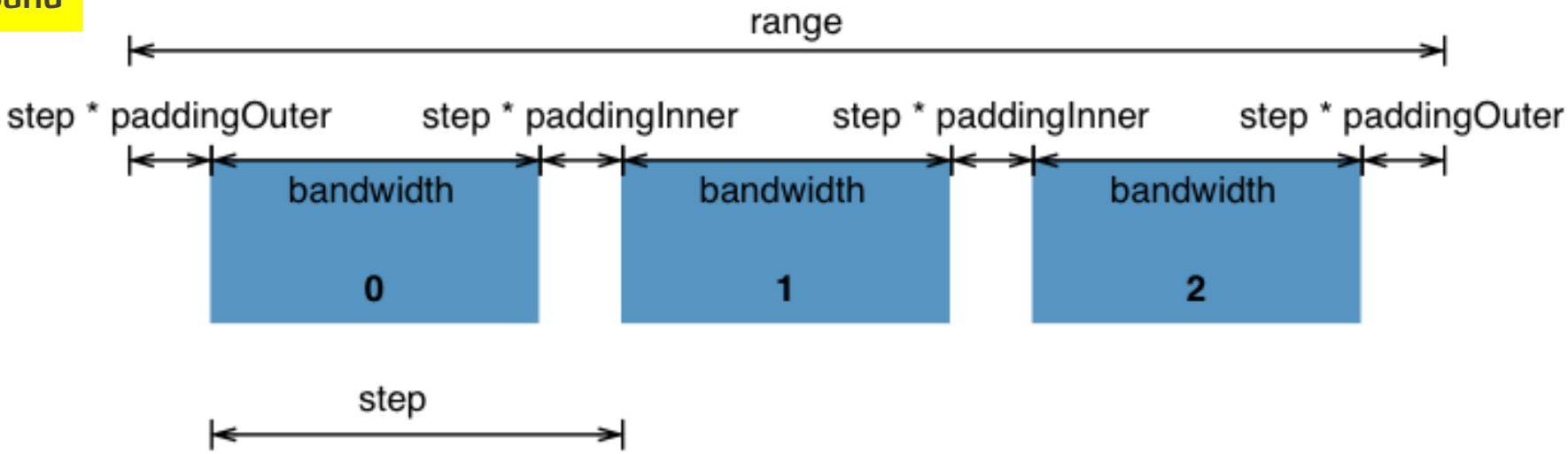


https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept – Other common scale

How about making X-axis grow dynamically?

Band



DOMAIN

(e.g., Country Names)

["Australia","Brazil","China",
"New Zealand","Indonesia"]

Given the input domain, the band scale outputs continuous output range into uniform bands.

Band scales are typically used for bar charts with an ordinal or categorical dimension

https://github.com/d3/d3-scale#band-scales

Core D3 concept – Final Product So Far

https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787

Core D3 concept

From here …

- Adding Axes and labels (very similar to Scale concept)
- you can dig deeper into D3 topics that allow you build more ‘interactive’ and ‘dynamic’ visualisations …
- More important D3 concepts like D3 Updates and D3 Transition …

https://medium.com/@adamjanes/5-crucial-concepts-for-learning-d3-8ca460f18787