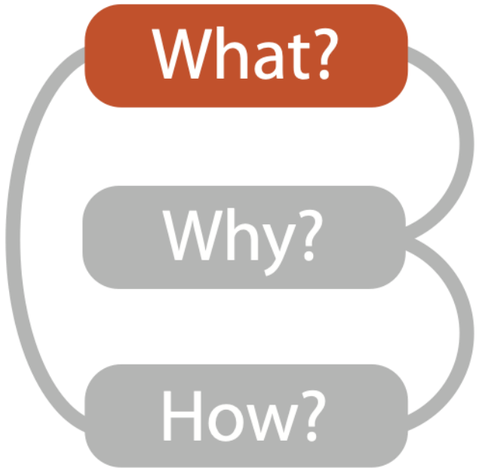
Visualization for Data Science
Visual Marks & Channels



What?

Datasets

Data Types

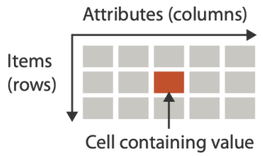
→ Items → Attributes → Links → Positions → Grids

Data and Dataset Types

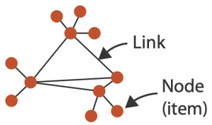
Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

Dataset Types

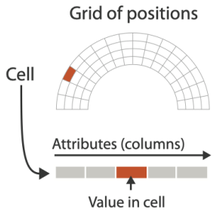
→ Tables



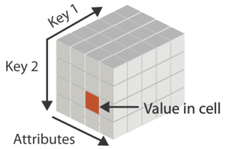
→ Networks



→ Fields (Continuous)



→ *Multidimensional Table*



→ *Trees*

→ Geometry (Spatial)

Dataset Availability

→ Static

→ Dynamic

Attributes

Attribute Types

→ Categorical

→ Ordered

→ *Ordinal*

→ *Quantitative*

Ordering Direction

→ Sequential

→ Diverging

→ Cyclic

Lecture Recap

What kind of questions can we answer?

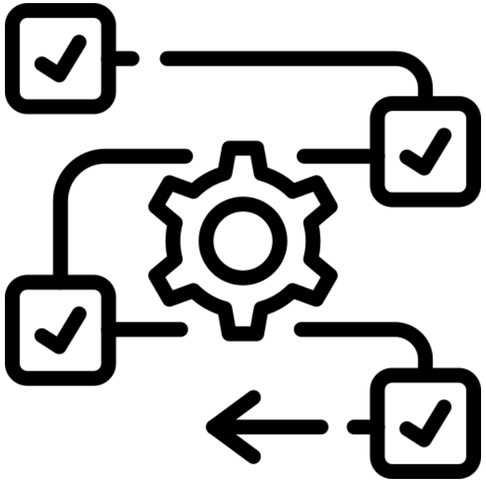
What is the data?

Who might be the audience?

City	Condition	Temperature
Calgary	Mainly Clear	-4°C
Charlottetown	Light Snowshower	-6°C
Edmonton		-7°C
Fredericton	Clear	-9°C
Halifax	Light Snow	-6°C
Iqaluit	Clear	-28°C
Montréal	Mainly Clear	-9°C
Ottawa (Kanata - Orléans)	Mainly Clear	-10°C
Prince George	Fog	-6°C
Québec	Mainly Clear	-15°C
Regina	Mist	-13°C
Saskatoon	Mist	-11°C
St. John's	Mostly Cloudy	-5°C
Thunder Bay	Light Snow	0°C
Toronto	Cloudy	-1°C
Vancouver	Mainly Clear	7°C
Victoria	Mainly Clear	5°C
Whitehorse	Mostly Cloudy	-12°C
Winnipeg	Mist	-6°C
Yellowknife	Light Snow	-13°C

https://weather.gc.ca/canada_e.html

Data abstraction: Three operations

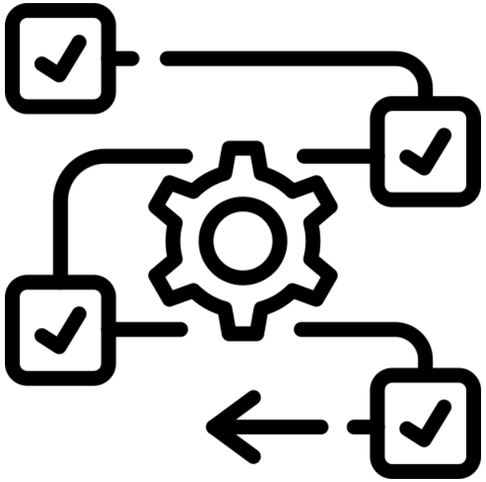


- translate from domain-specific language to generic visualization language

- identify dataset type(s), attribute types
- identify cardinality
  - how many items in the dataset?
  - what is cardinality of each attribute?
    - number of levels for categorical data
    - range for quantitative data

- consider whether to transform data
  - guided by understanding of task

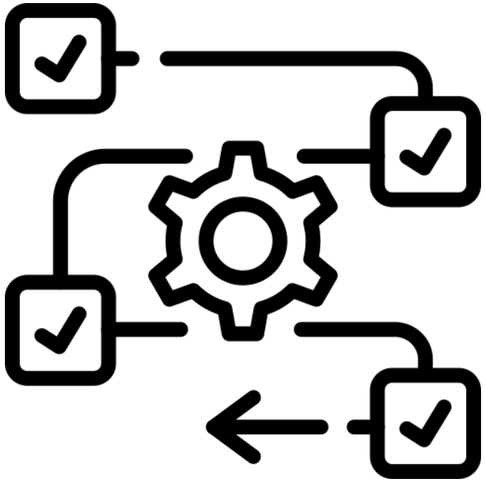
Data vs conceptual models



- data model
  - mathematical abstraction
    - sets with operations, eg floats with * / - +
    - variable data types in programming languages
- conceptual model
  - mental construction (semantics)
  - supports reasoning
  - typically based on understanding of tasks [stay tuned!]

- data abstraction process relies on conceptual model
  - for transforming data if needed

Data vs conceptual model, example



- data model: floats
  - 32.52, 54.06, -14.35, …
- conceptual model
  - temperature
- multiple possible data abstractions
  - continuous to 2 significant figures: quantitative
    - task: forecasting the weather
  - hot, warm, cold: ordinal
    - task: deciding if bath water is ready
  - above freezing, below freezing: categorical
    - task: decide if I should leave the house today

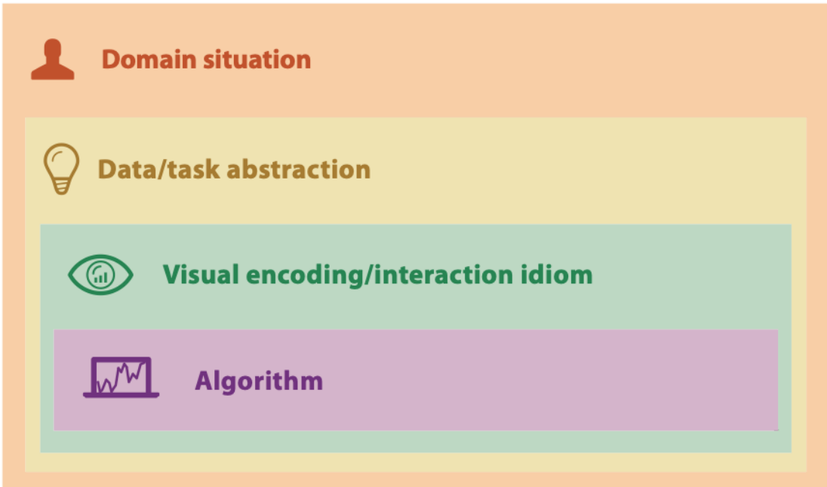
Visual encoding

How many ways can you **visually** represent two numbers, e.g.? 75 and 37

A. 1
B. 2
C. 5
D. 10
E. 15

Design Task (5 mins)

Thinking outside the box, sketch all the different ways you can visualize two numbers: 75 and 37



Lessons Learned from the exercise

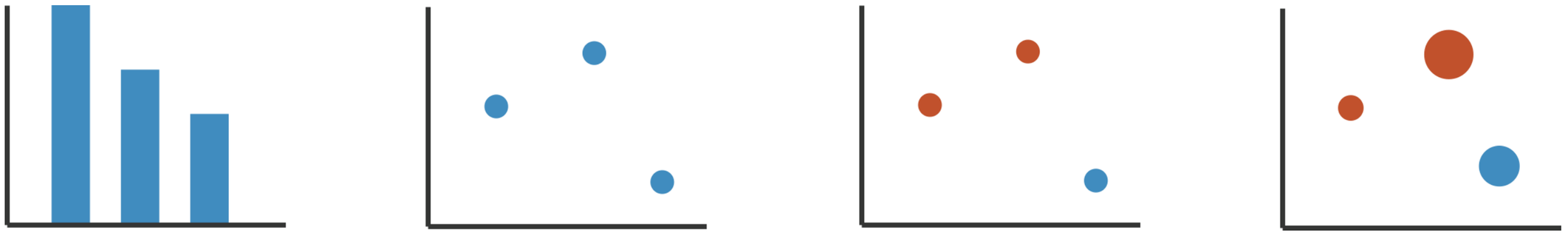
- Some vizzes are scalable to larger lists of numbers
- Some vizzes focus on the relationship between the values, while others focus on each value independently
- Information visualization is a language that has its own grammar
  - Combinatory: composition of marks
  - Generative: serve as building blocks to other vizzes
  - Expansive: hard to count all options
  - Creative

Learning Outcomes

- Describe the basic visual primitives of visualizations (marks and channels)
- Differentiate between a mark and channel
- Describe a visualization using appropriate viz grammar
- Use makes and channels to create a bubble chart in Altair
  - Chart Object
  - How to attach data to the Chart
  - How to specify each channel and describe which attribute encodes

Visual encoding

How to systematically analyze idiom structure?



- Marks: represent items or links (e.g., line, point, circle)
- Channels: change appearance of marks based on attributes (position, color)

Marks for items

- basic geometric elements or primitives

Points

0D

Lines

1D

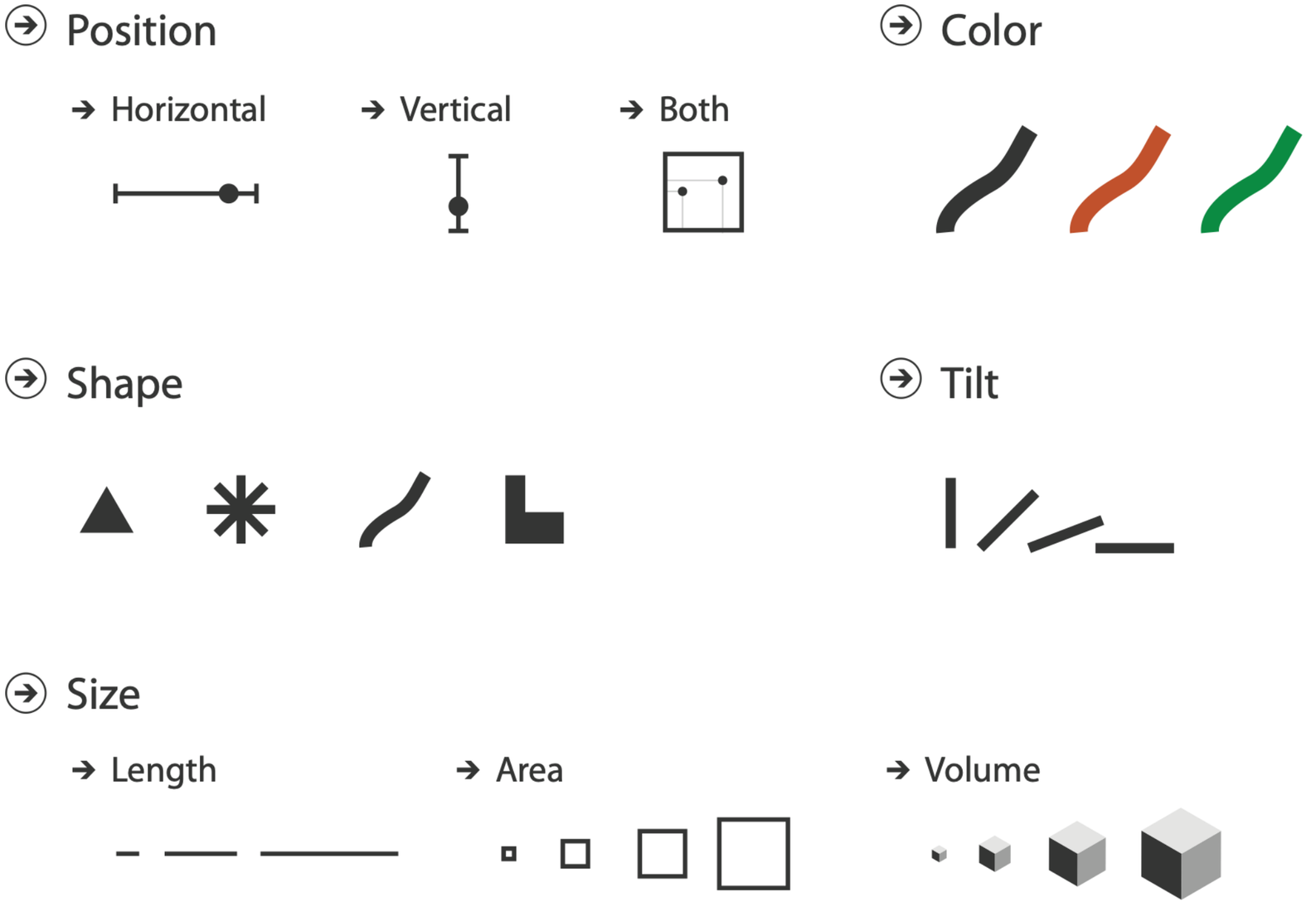
Interlocking Areas

2D

- 3D mark: volume, rarely used

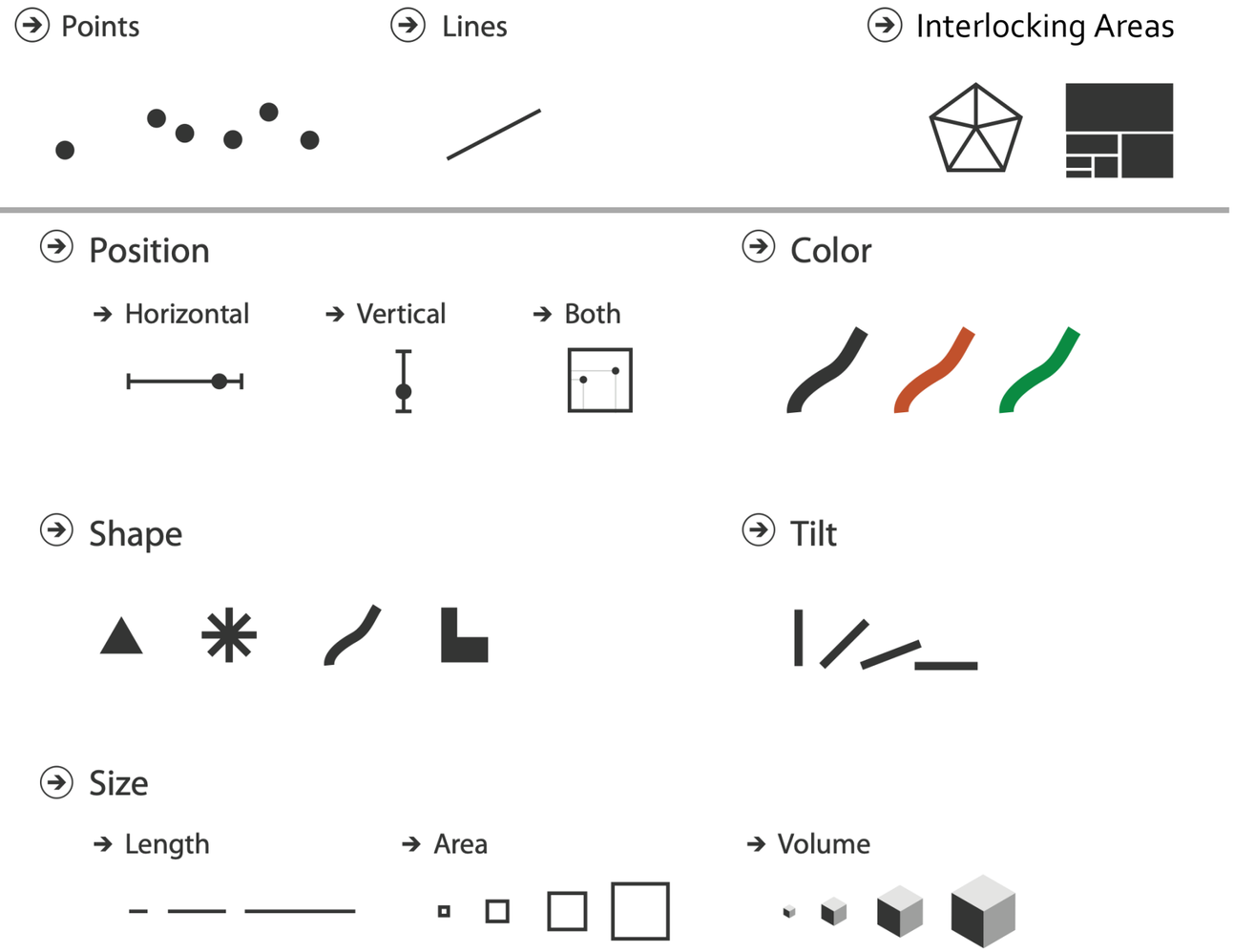
Channels

- control appearance of marks
  - proportional to or based on attributes
- many names
  - **visual channels**
  - visual variables
  - retinal channels
  - visual dimensions
  - Visual encodings



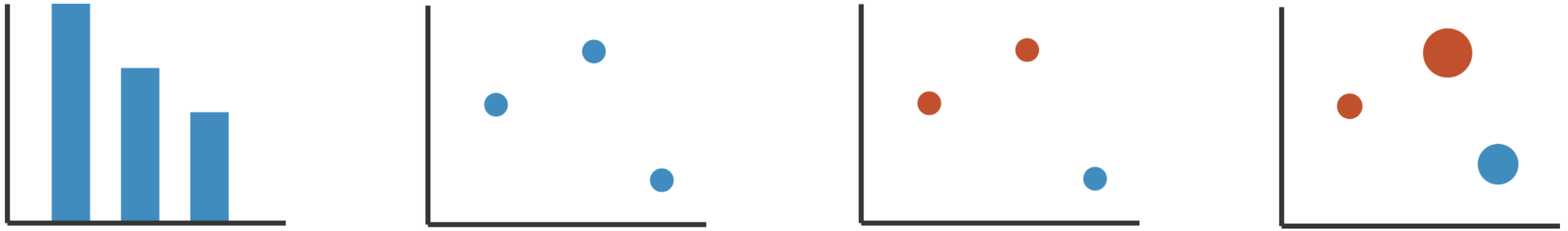
Definitions: Marks and channels

- marks
  - geometric primitives
- channels
  - control appearance of marks
- channel properties differ
    - type & amount of information that can be conveyed to human perceptual system



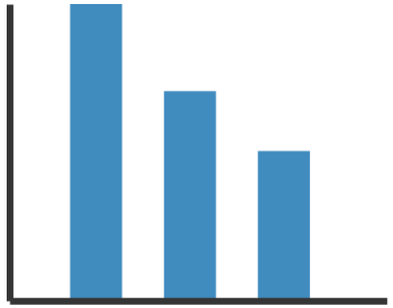
Visual encoding

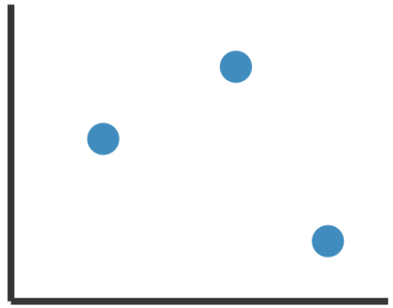
analyze idiom structure as combination of marks and channels

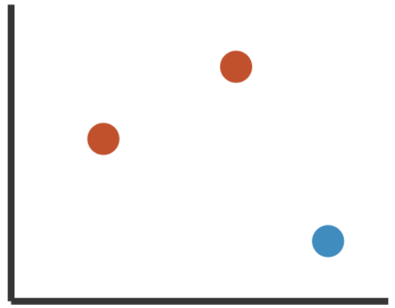


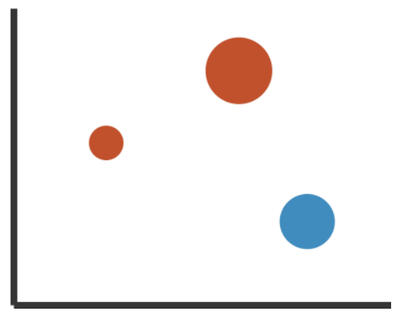
Visual encoding

analyze idiom structure as combination of marks and channels









Channels

vertical position (y)

vertical position (y)
horizontal position (x)

vertical position (y)
horizontal position (x)
color hue

vertical position (y)
horizontal position (x)
color hue
size (area)

Marks

line

point

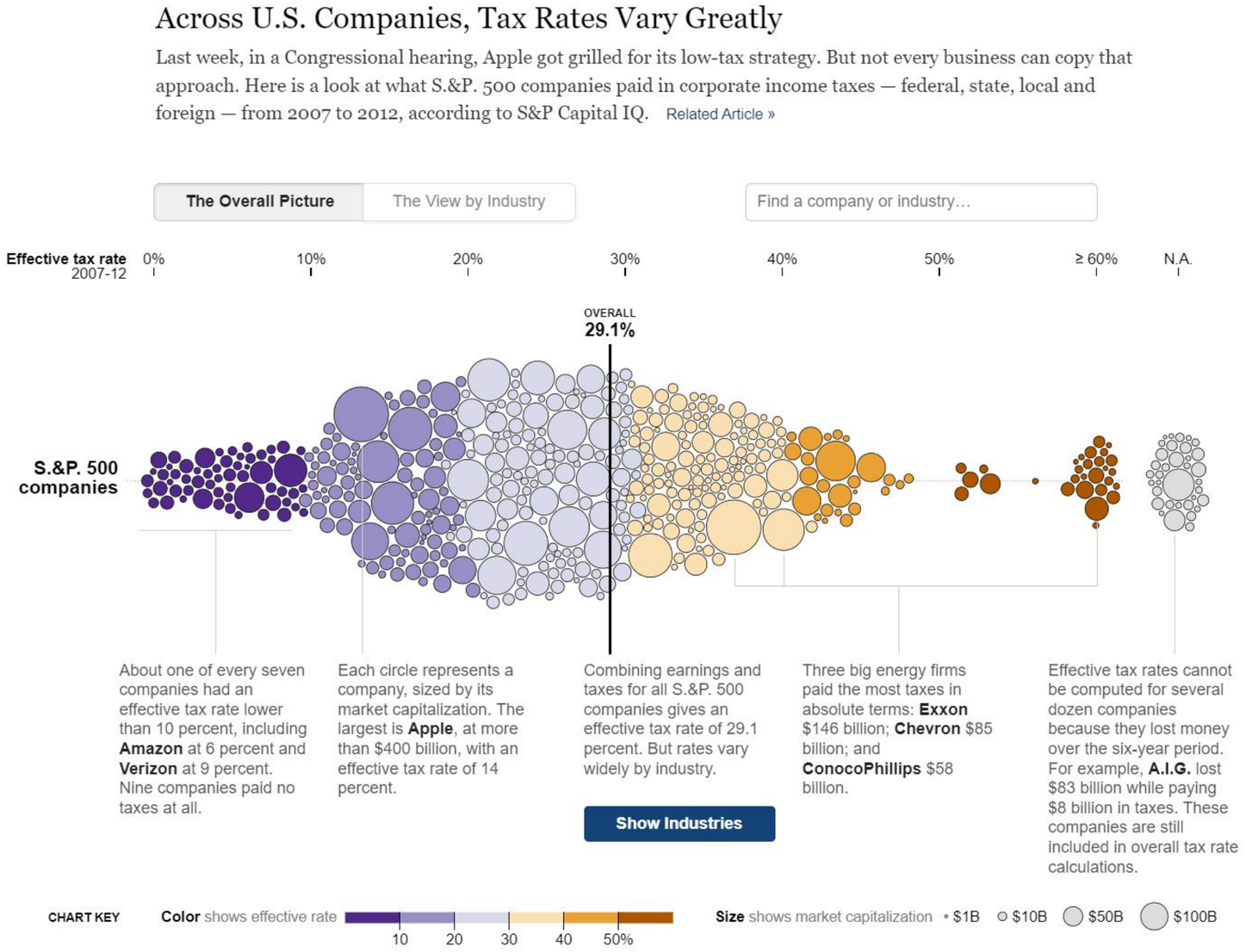
point

point

In groups, deconstruct this visualization

What are the data attributes being encoded?
What marks are used?
What channels are used?

Data Semantics	Attribute Type	Channel	Mark



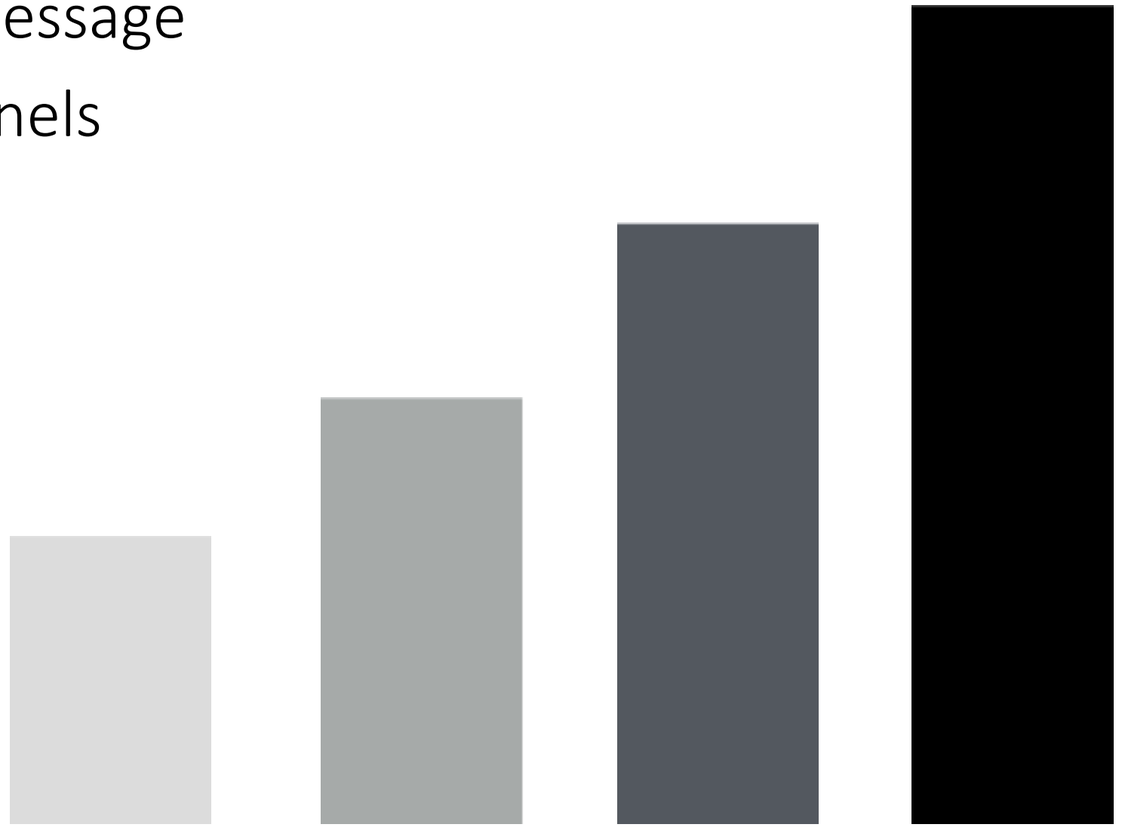
https://archive.nytimes.com/www.nytimes.com/interactive/2013/05/25/sunday-review/corporate-taxes.html

Data Semantics	Attribute Type	Channel	Mark

Redundant encoding

Multiple channels

- sends stronger message
- but uses up channels



Channels

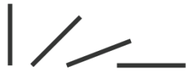
Magnitude Channels: Ordered Attributes

Position on common scale

Position on unaligned scale

Length (1D size)

Tilt/angle



Area (2D size)

Depth (3D position)

Color luminance

Color saturation

Curvature

Volume (3D size)

Identity Channels: Categorical Attributes

Spatial region

Color hue

Motion

Shape

Grammar of Graphics

A framework which follows a layered approach to describe, design, create visualizations in a structured fashion.

Grammar of Graphics – Leland Wiklinson

Layered Grammar of Graphics – Hadley Wickman (R visualization package ggplot2)

Vega is a *visualization grammar*, a declarative language for creating, saving, and sharing interactive visualization designs. Vega-Altair – statistical visualization library for Python.

Others: Cicero, Atlas

Grammar Building Blocks

Vega (altair)

- Data
- Graphical marks
- Transformation
  - Aggregate, Bin, Density, Filter, Flatten,
- Scales
- Map projections
- Axes
- Legends

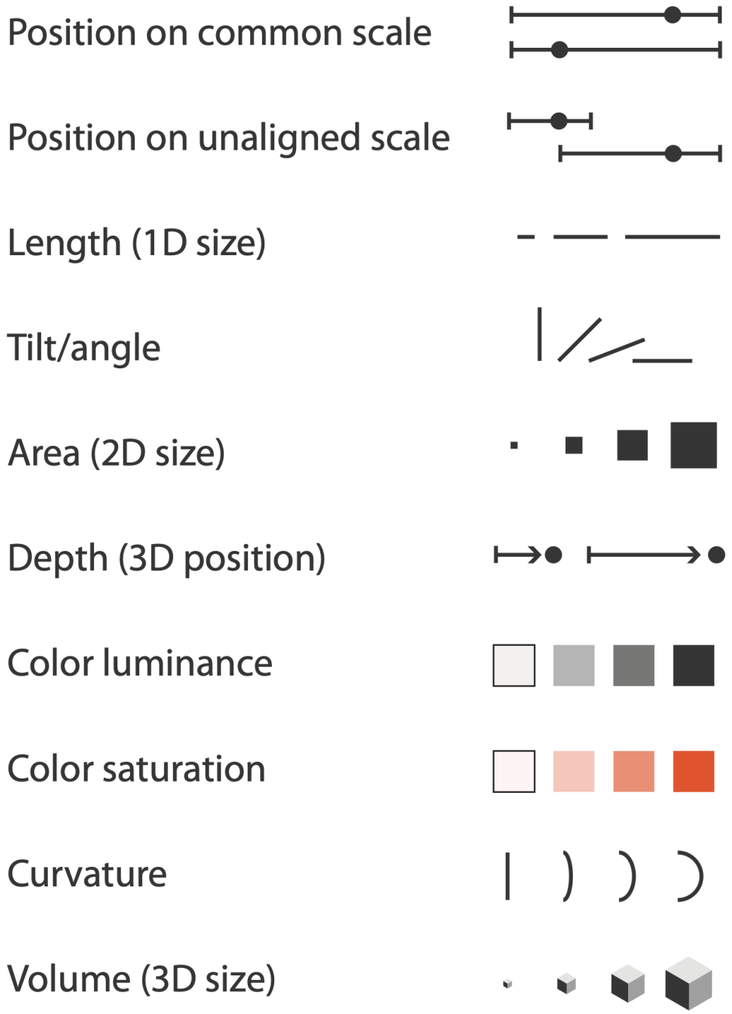
The key idea is that you are declaring links between *data columns* and *visual encoding channels*, such as the x-axis, y-axis, color, etc.

Layered Grammar (ggplot2)

- Data
- Layer
  - Data
  - Mapping
  - Geometric object
  - Stat (Statistical Transformation)
  - Position
- Scales
- Facet specification
- Coordinate system

From Theory to Tooling

Magnitude Channels: Ordered Attributes



Altair Channels

- x: Horizontal (x-axis) position of the mark.
- y: Vertical (y-axis) position of the mark.
- angle:
- color: Mark color, specified as a legal CSS color.
- fill
- opacity: Mark opacity, ranging from 0 (fully transparent) to 1
- radius
- shape: Plotting symbol shape for point marks.
- size

Altair Marks

- arc
- area
- bar
- circle
- line
- point
- rect

Altair Basics

Create a Chart Object

Attach data to the Chart Object

Specify the mark type

```
alt.Chart(data).mark_bar().encode(
        channel_1 = 'column1',
        channel_2 = 'column2',
)
```

Specify each channel and what data it encodes