Visualization of the information

Carlos Arbonés and Juan P. Zaldivar

GCED, UPC.

Lecture Notes.

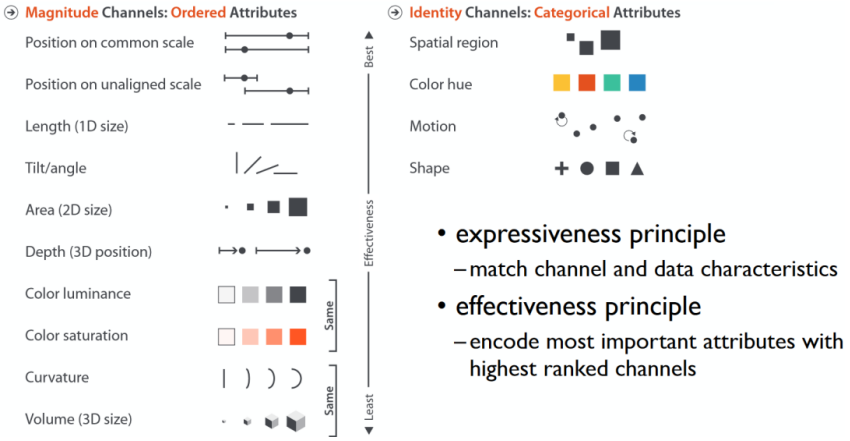
Contents

1 Introduction to Visualization

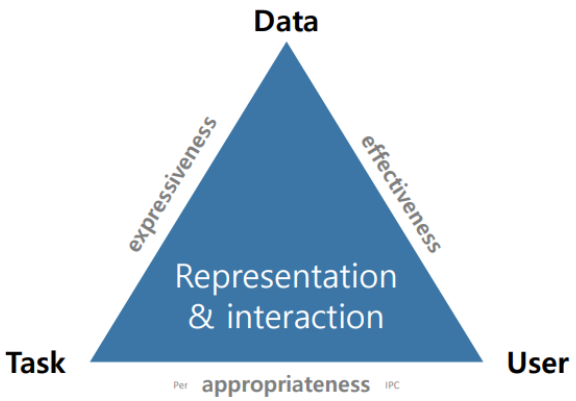
We have to pay attention **users**, in what are their needs, background, work environment, etc. Also focus on the **data**, for example the scale, e.g, quantitative, qualitative and also the type, e.g, 1-dimensional, 2-dimensional, the number of variables, etc. Finally we have to know **what are the tasks**.

In visualization is crucial accomplishing the following:

1. *Expressiveness:* show exactly **the relevant information in the data**, no more no less. A visualization is expressive if and only if it encodes all the data relations intended and no other data relations.
2. *Effectiveness:* take into account the **cognitive capabilities** of the human visual system, depends on the user. Create the visualization to exploit the capacities of the human system and use the space as optimally for the user to understand the data. The message has to be easy to get. We must know which visual variables are better and worse to distinguish elements.



3. *Appropriateness:* is important that design decisions reflect their objective, that is, the way we are encoding the data serves the user to solve tasks. Cost-value ratio that asseses the benefit of the visualization, mainly **time (computation)** and **space (screen-space)** efficiency. There is no need for using the whole screen for plotting extremely simple data for example.



To carry out tasks more **effectively** we need a match between data/task and representation, there are a lot of possible representation and many are ineffective. The chance of finding good solution increase if we understand the full space of possibilities. Representation must be **novel**, enable entirely new kinds of analysis, is the ability to bring something new or innovative to the table. An effective approach should have the capability to enable new types of analysis that were not possible or feasible before.

And **faster**, speed up existing workflows (making the current or established procedures and tasks within a particular system or organization operate more quickly and efficiently). To **validate effectiveness** there are many methods and we have to pick the appropriate one for our context.

Some **inappropriate practices** in data visualization are:

- To have some data set and we **throw it to any chart type**
- Get some **random data and create "visualization"** from it
- Encoding **too much or irrelevant information**
- Using **unsuitable palettes**: we should to able to interpret the visualization without the palette legend
- Using **unreadable text**
- If the chart is published by **Donald J. Trump**
- Use **lots of axis**, difficult to easily interpret
- **Correlation does not imply causation**
- Use **space** prudently
- Be careful with **clutter**
- Use **3D** with caution
- Be careful with **scales**
- Use **color** wisely
- Use **standard axis**

1.1 The Basics

Computer-based visualization systems provide ***visual*** *representations* ***of data sets*** *designed to* ***help people*** *carry out tasks more* ***effectively***. Augment the capabilities of the human rather than replacing it by computation decision making.

Visualization is related to **understanding the underlying data** by helping the user to understand data using their excellent perception capabilities. It **helps the user to carry out tasks more effectively**. If the result is a calculation, we should probably not be using visualization at all. If we know what are we looking for then we do not need it. Putting **human in the loop** is fundamental.

Anscombe's Quartet: Raw Data

	1		2		3		4	
	X	Y	X	Y	X	Y	X	Y
	10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
	8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
	13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
	9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
	11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
	14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
	6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
	4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
	12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
	7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
	5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89
Mean	9.0	7.5	9.0	7.5	9.0	7.5	9.0	7.5
Variance	10.0	3.75	10.0	3.75	10.0	3.75	10.0	3.75
Correlation	0.816		0.816		0.816		0.816	

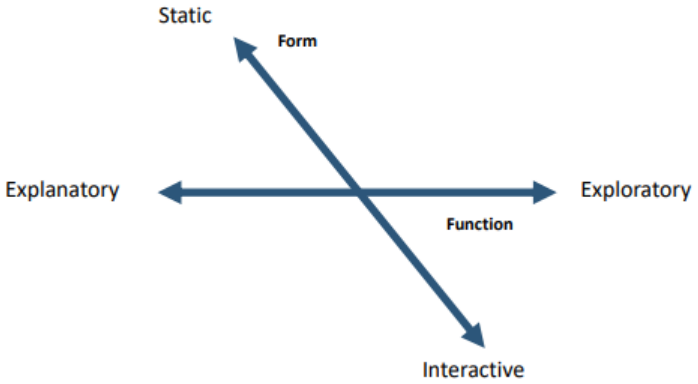
We need to be careful when representing datasets. **Summaries lose information** and **details matter**. Famous example: *Anscombe's quartet*. We do not know how the data is distributed, although they might have the same summaries information.

1.1.1 Main applications of visualization

The main applications of visualizations are:

- *Explanatory:* present the results. Visualization is used for **presentation**. To communicate data and ideas, explain and inform providing evidence and influence and persuade. **Commonly only showing a few variables of the data.**
- *Analysis:* Analyse hypothesis. The typical objectives are **showing many variables**, illustrate overview and detail to **facilitate comparison**. Presentation might choose some parts, **analysis will focus on all of them**.
- *Exploratory:* Inspect the data to learn new things, get **insights.** Visualization is very useful in exploratory data analysis when **we don't know what we're looking for**, **we don't have a priori questions** and **we want to know what questions to ask**.

In conclusion, **exploration** is used for gathering knowledge on the data when nothing is known, **analysis** is used for verification or falsification of hypothesis and **presentation** for communicating the results.



1.2 General Rules

One has to **be honest** with the audience and **check if the data is correct, updated**, etc. Also how it was collected and if it is reliable.

Explain the use of the colors and the legends. If encoding (of colors and shapes) are not standard there has to be a reason. The colors have to take into account the physical conditions of the audience and their background knowledge about the domain.

We need to **make things memorable**, give a certain story when showing the visualizations. The display section is also important, it is not the same to present the visualization in a screen of a phone than a billboard.

1.3 Data, Tasks, Users

1.3.1 Data Types

- **Nominal**: Unordered set of names/observations.

- **Ordinal**: There has to be an specific order in the observations (days of the week).
- **Quantitative**: measured or simulated data.

We can make nominal data ordered by introducing order and quantitative in ordinal by binning. We can make certain transformations but this is something artificial.

1.3.2 Data Structure

Structure	Examples
1-dimensional	Alphabetic lists, source code, texts/documents
2-dimensional	Planar or map data, photos
3-dimensional	Molecules, human body, buildings
Temporal	{start, finish}, e.g., medical records, project management, historical presentations
Multi-dimensional	N attributes -> points in n-dimensional space, e.g., relational databases
Tree	Hierarchies or tree structures, e.g., file directories, business organizations
Network	Connected as graphs, e.g., communications networks, social networks

1.3.3 Tasks

The 4 most important tasks in data visualization are:

- **Overview**: gain an overall knowledge of the data.
- **Zoom**: we can concentrate into a small region in out data. We need to provide this action to people with **interactions**. We can perform zooming in all dimensions. One dimension at a time by moving bar controls or similar functionalities.
- **Filter**: similar to zoom. Focus on some elements, deleting uninteresting elements. What we do is by removing using sliders.
- **Details-on-demand**: we are not going to show all the data we have. At certain points we need to give the users the opportunity to give details, so they click the info they want to see. Select a set of items and get the details when needed.

There are other tasks such as:

- **Relate**: being able to relate items from one view to another. Or look for similar items in one view (e.g "Other countries having this issue...").
- **History**: important to track history. Go back to certain steps of the actions of manipulation of data.
- **Extract**: allow to extract relevant information according to the user.

1.3.4 Users

- **Limitations**: Background, visualizations limitations, etc.
- **Computational Capacity**: Interactive visualizations must be efficient enough. Computer time and memory are limited resources.
- **Human perceptual and cognitive capacity**: Memory and attention are finite resources so we are vulnerable to large changes. Also the length of the presentation.
- **Display capacity**: Not enough space for big visualization. People are not going to use infinite space. Scrolling million of pixels is not doable. Maximizing the amount of info in the space available without overwhelming the user (Ink-ratio[1]). Displaying

[1]It refers to the proportion of ink (or pixels in digital formats) used to represent the actual data in a data visualization, as opposed to the ink used for non-data ink, which includes labels, gridlines, decorations, and other ink that doesn't directly convey information. A high data-ink ratio indicates that a significant portion of

a lot of information in one chart can reduce the need for navigation and exploration, but user may be overwhelmed by visual clutter.

1.4 Visualization as a Design Process

1. **Problem characterisation**: We have some sort of data that can be managed in Ink-ratio. For the same data a lot of possible visualizations. Need to find what the **user wants to solve**. We need to choose some domain experts to help us understand the needs. Sometimes the tasks can not be verbalized, which leads to an **iterative creation of the visualization**, while including more information of the target/task.
2. **Data Abstraction:** Transform the data into a more abstract way (a more generic representation). F.e example gender difference among countries, we can transform the data to adjust to explain that gender difference.
3. **Technique and algorithm design**: Decide how we let the user select and interact with the visualization. Select appropriate visual encoding.
4. **Validation**: we can try to validate if our visualizations are good. If the users can benefit of the visualizations, if the data types are useful to encode the information. We can measure if it is efficient or not. Some of the validation methods can only be applied after the visualization is complete. Do the visual encoding communicate effectively the abstraction? Verify the designed algorithm to visualize and render is faster, takes less memory.

1.5 Visualization Mantra

Schneiderman's Mantra: This mantra underscores the importance of structuring and interacting with information in a way that supports effective task and its understanding.

1. **Overview First:** When working with large datasets or complex information, it's crucial to begin by providing users with an overview or a high-level perspective. This overview should offer a summary of the entire dataset or information, allowing users to grasp its overall structure and context. An overview helps users decide where to focus their attention and provides a sense of orientation.
2. **Zoom and Filter:** After establishing an overview, users should be able to zoom in on areas of interest or apply filters to narrow down their focus. This allows for exploration at different levels of detail, from a broad view down to specific subsets or individual data points. Zooming and filtering enable users to investigate particular aspects of the data more deeply.
3. **Details-on-Demand:** This principle emphasizes the importance of providing users with the ability to access detailed information on-demand. When users identify a point of interest or focus on specific data, they should be able to retrieve additional information or details about that particular item. This "drill-down" capability ensures that users can explore the data at the level of granularity they require.0

By adhering to Shneiderman's Mantra, visualization designers aim to create interactive and user-friendly interfaces that support effective data exploration and decision-making.

---

the ink or pixels in a visualization is dedicated to presenting data, making the visualization more efficient and effective in communicating information. In contrast, a low data-ink ratio implies that a large portion of the ink or pixels are used for non-data elements, potentially reducing the clarity and efficiency of the visualization.

2 Good Practices in Visualization

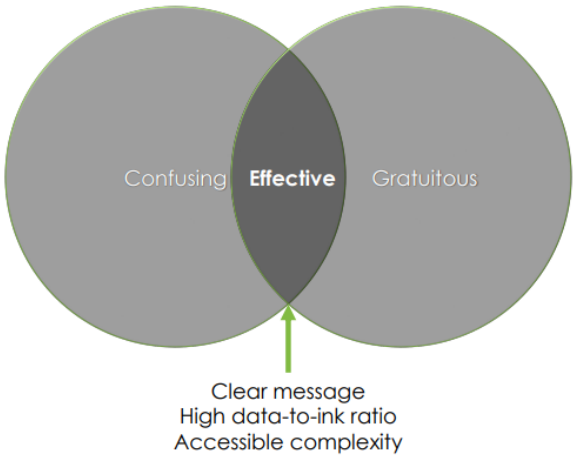
2.1 Effective Visualizations

The main idea is to **communicate a message**. We have a lot of data and we want the reader to get some knowledge from this data. We need that the message is transmitted and the people understand the message. **A visualization is not effective if is too complex or is misleading** (in the sense that the data is not understood).

Some elements that have to be considered in an effective visualization are:

- **Data density**: not too many elements, this could be confusing. Important that we use the minimum amount of data needed to transmit the message. We want information-rich visualizations.
- **Visual mappings**: if we have a lot of elements the user will need too much time to process it. As simpler as we can. Observers must understand the depiction without effort.
- **Amount of information**: keys, labels, etc. helps understanding the data.
- **Color usage**: influences what we can understand and what we can see from a map. If we choose bad colors is very difficult to understand data. We can also use it to guide the user, to mark important things.

To be sure if we are doing a good visualization we need to analyse it and see **if the message is understood**. We will need to analyse the data and consider if this data contains some important parts that we need to highlight and if the user gets it.



The **general principle** we need to follow is that we need to strive to give our viewer the greatest number of useful ideas in the shortest time with the least ink (simple visualizations). We want **information-rich and informative** visualizations.

- **Create visualizations when necessary**, when we need to transmit and explain something. The desire for a figure is not always proportional to its utility.
- **Don't merely display data, explain it**.
- **Know the message in advanced** to plainly the visualization that satisfies the audience. Need to adapt it to who will be looking at it.
- **Strive for clear communication**. Don't hide message with context. Revise and redraw. To explore data, achieve effective encoding. To communicate concepts, use effective design

- **Satisfy your audience, not yourself**. Be aware of bias in evaluating effectiveness of visual forms.
- **Respect visual capacities of humans**.

We will need several attempts to achieve a good visualization. If we want to communicate contents we need a good design, is an **iterative process**. Refine the visualization until we get what we want. We need to consider **legibility**, the idea is that it should be readable in the screen we are going to display it. Our visual ability is not infinite.

2.2 Specific principles

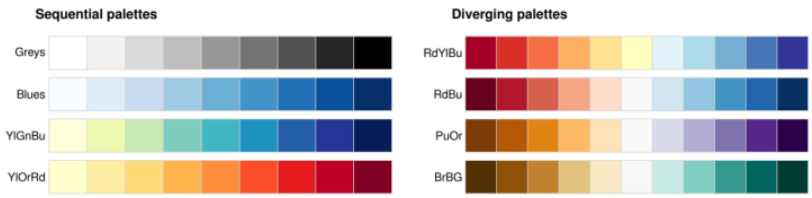
- **Data informs variation:** the visual representation of data should effectively communicate variations and patterns within the data. The visual elements used should reflect the underlying data accurately.
- **Consistency:** consistent design elements, such as color schemes, fonts, and labeling, to make your visualizations more understandable and aesthetically pleasing.
- **Avoid redundancy:** avoid unnecessary duplication of information. Each element should serve a distinct and meaningful purpose.
- **Conciseness:** remove any extraneous elements that don't contribute to the message you want to convey.
- **Remove to improve:** simplifying a visualization by removing unnecessary clutter and distractions can often make it more effective and easier to understand.
- **Focus & emphasis:** use visual cues, such as highlighting, color, and size, to guide the viewer's attention to the most important aspects of your data. Emphasize the key points you want to convey.
- **Attractiveness:** while functionality is crucial, an attractive design can make your data visualizations more engaging and memorable. An aesthetically pleasing presentation can draw the viewer's attention and make the data more accessible.

2.3 Use of color

We can use color to:

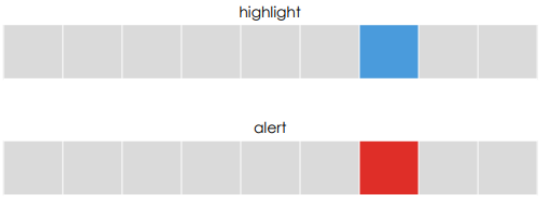
- **Distinguish**: we will use palettes that allows us to represent categorical data, use colors different to the other but not have any other implications.

- **To encode values**: to encode quantitative data. We can use sequential palettes and diverging palettes.



  - Highlight Trends: Sequential palettes allow viewers to easily perceive trends and variations in data as values increase or decrease.
  - Clear Reference Point: The neutral color at the midpoint provides a clear reference point, making it easy to identify values above or below this reference.

- **To highlight** elements/values.



2.3.1 Tips for Color Selection

- Use color **only when needed** to serve a particular communication goal.
- Select **suitable color palettes**.
- Non-data components should be displayed just visibly enough to perform their role (e.g., light grey).
- Avoid using a combination of red and green

2.4 Comparison

- Zero baseline if possible.
- Choose the most effective visualization.
- Place elements to facilitate comparison.
- Tell the whole story. Omitting data may be misleading, but extra data can also be misleading

Many visual depictions may communicate the same data correctly. But some are more difficult to understand than others. Need more time/cognitive effort. Always select the most effective ones in terms of time, space, cognitive effort...

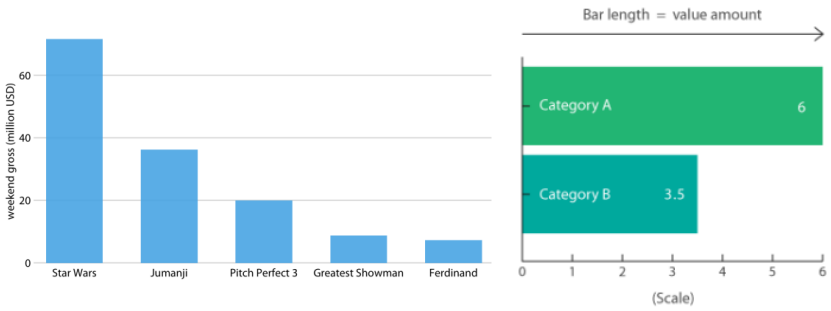
3 Visualization techniques

3.1 Display quantities

3.1.1 Bar charts

Are used to **compare/lookup (really easy)**. Can scale to hundreds of elements.

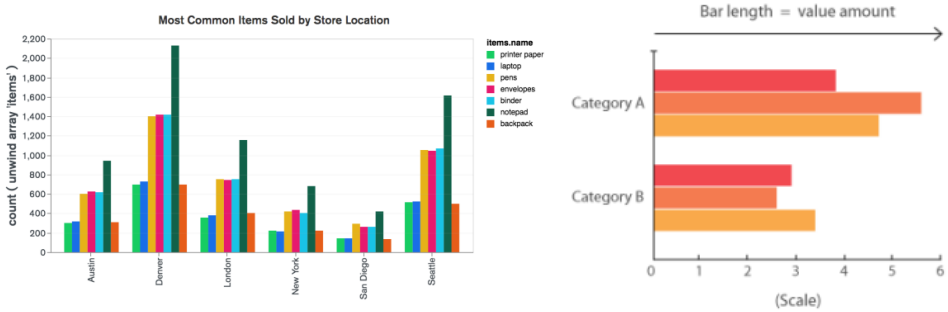
- They always **must start at 0**. If not, the proportion of the quantities is lost, causing misunderstanding to the user.
- **Labels easy to read**. Think of the orientation of labels (horizontal if possible). Labels of the bars should not be too long.
- **Order based on data or labels**. Alphabetical order if we want to make label search easier. By quantity if we want to facilitate value search
- **Neutral colors** better, something related to gray or just gray. Other colors to emphasize.
- **Grid lines** needed if we are looking to have a **precision**.
- If data is **ordered in time**, better use **line chart**.
- **Don't use hundreds of bars**



3.1.2 Paired bar charts

Usualy used to compare. Easy to identify specific data in the same category, but not between different categories. **Length** is used to express quantity. **Color** is used to separate values in each category. **Spatial regions** separate categories.

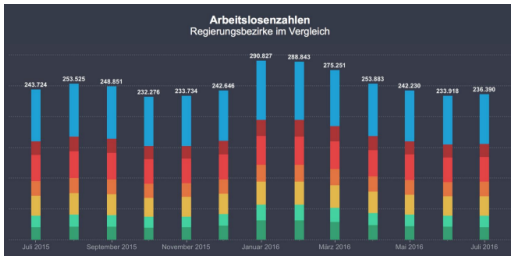
- They always must **start at 0**.
- **Bar chart guidelines** apply.
- **Don't use them** if one category is **time**.

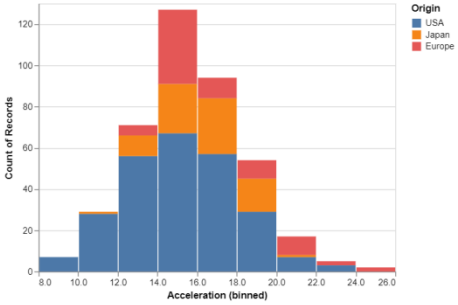


3.1.3 Stacked bar chart

Contains the same information as *Paired bat charts*, but now bars are stacked vertically. More **difficult to compare different groups**, also **difficult** within the same category. The **total quantity of the stacked bar has to make sense**, not only the divisions.

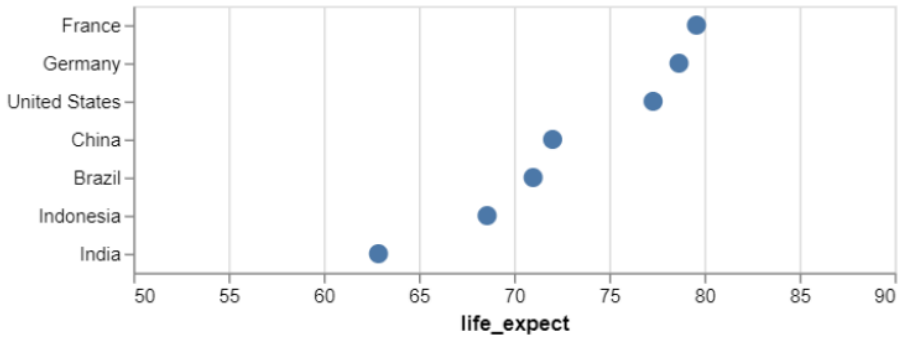
- **Start at 0**.
- Same **guidelines** as **bar charts**.
- Difficult to compare between groups
- Difficult to compare within groups
- Don't use when total quantity does not make sense
- Use few categories





3.1.4 Dot plot

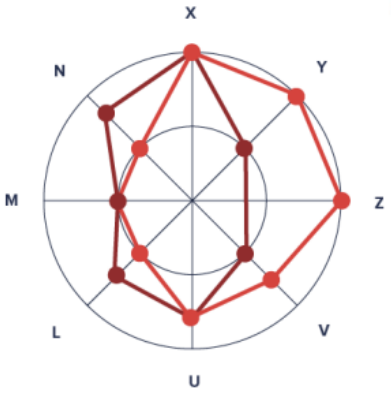
- **Don't need** to start at **0**
- Must be **ordered by quantity**, the opposite makes the chart difficult to read.
- Suitable when **small differences** must be shown. Bar charts might lead the attention away from those differences.
- If values are relevant, **label axes suitably**

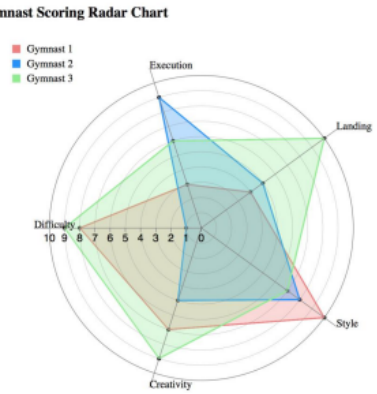


3.1.5 Radar chart

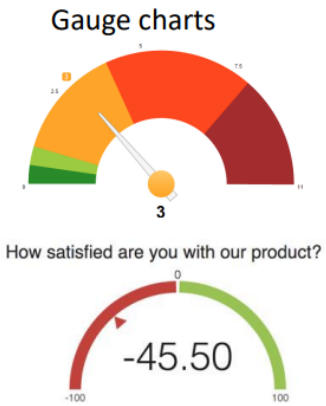
Instead of bars, it shows the data in a radar way. It is **analogous to paired/grouped column charts**. Easy to compare the values in the category.

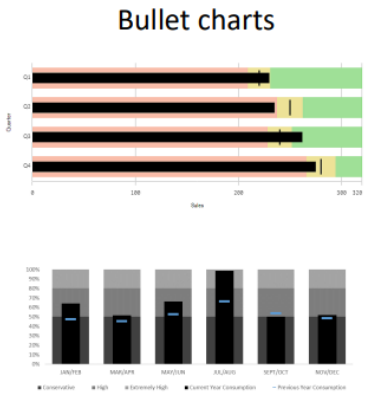
- Multiple dimensions
- Space efficient
- Different designs (points, area ...)
- Do not scale very well
- Can be small





3.1.6 Gauge & Bullet chart





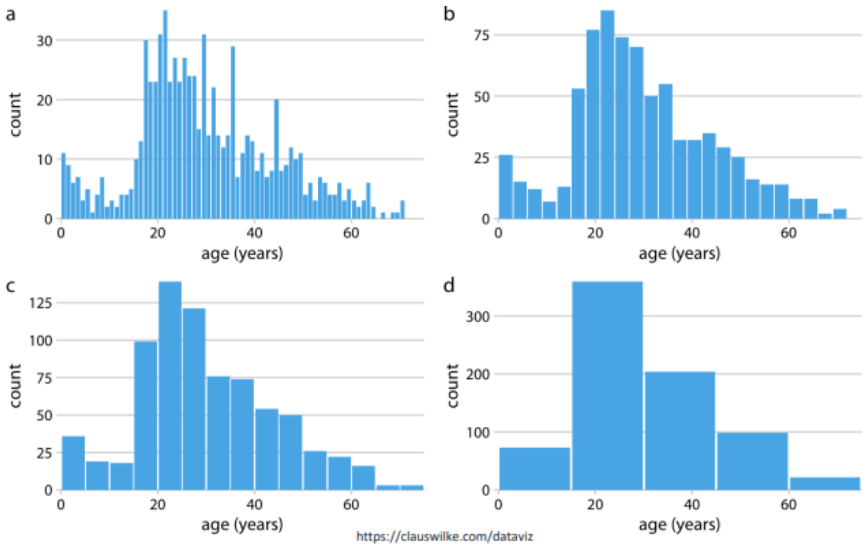
- Adaptation of **real gauges**
- Very common in **business analytics** (used to display KPIs)
- Current value (front) vs reference (background)
- Using **angle** to encode values (less optimal)
- Use too much space
- Commonly include the data in text too

- Version of gauge charts using bars
- Using the background of the bar chart to encode reference value(s)
- Space efficient (may encode multiple values in the same space)
- Better for perception (comparing lengths instead of angles)

3.2 Display distributions

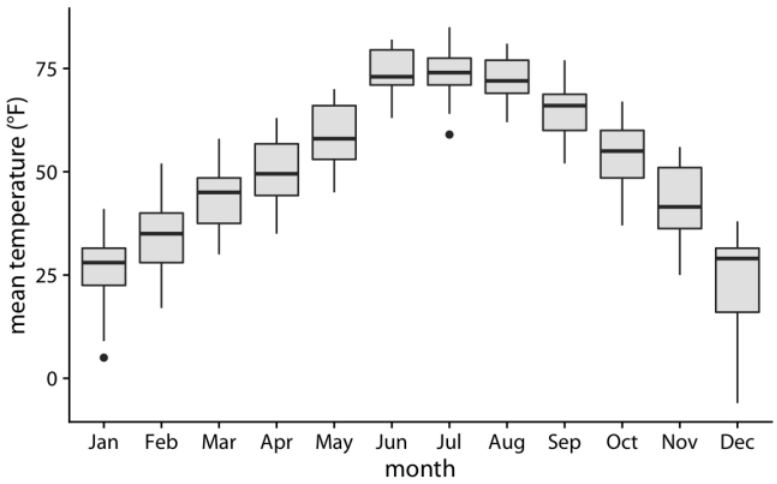
3.2.1 Histograms

Not focused on the values itself but in the **distribution/trend** of the whole set of values. Complicated but important to **choose the number of bars** for the distribution (might complicate the interpretation and the visualization of the distribution).



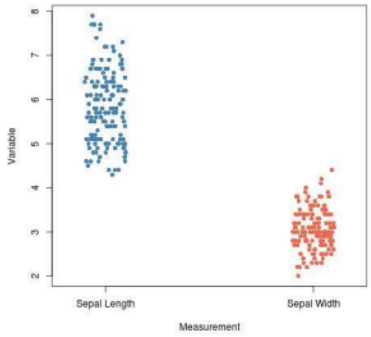
3.2.2 Boxplot

- Useful for **several distributions** at the same time
- Gives insights on data distribution (median, minimum, maximum, outliers...)
- Box plots **hide/abstract** too much data
- Hidden information may be relevant
- Can use **alternative charts (violin, streep...)** to show the internal distribution



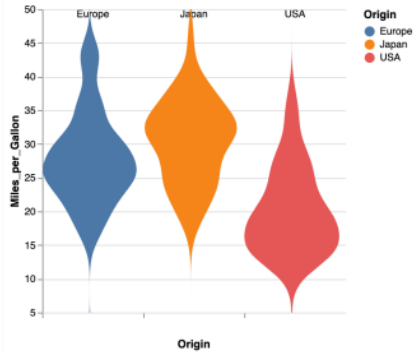
3.2.3 Strip chart

- Shows all the data points (revealing the distribution)
- Difficult to interpret
- Use random positioning in one axis to avoid operlapping.

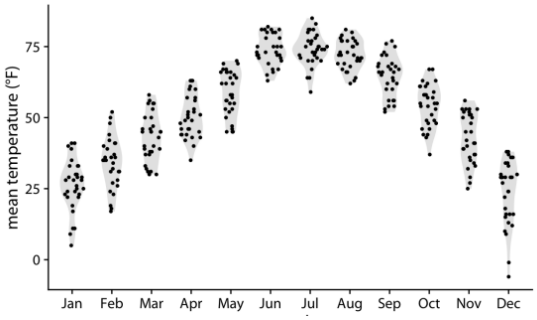


3.2.4 Violin plot

- An accumulation in the horizontal axis to illustrate the distribution (density chart)
- Reflected for aesthetic purposes.
- We lose the statistical properties
- Need to calculate the shape
- May still hide some data

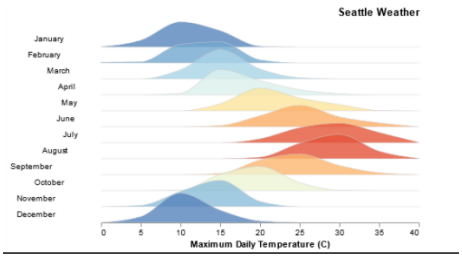


3.2.5 Strip chart + Violin plot



3.2.6 Ridge plot

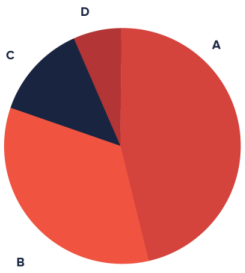
- Like a half violin plot in horitzontal
- Allows more data
- Allows overlapping if done carefully (can be a problem)
- No accurate value estimation possible



3.3 Display proportion

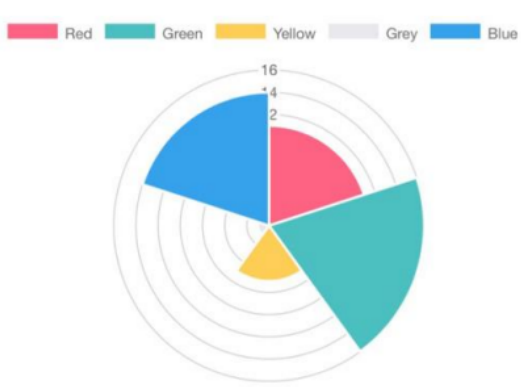
3.3.1 Pie chart

- They are proportions, should add to 100%
- **Angle** is used to display **quantity**
- Use **few categories**
- Start at 12:00 and sort in descending order (clockwise)
- Similar values will be **difficult to appreciate** visually
- Very influenced by the color palette chosen
- Too much space for the low information shown
- Certain key proportions ($1/4^{th}$, half) may be easier to read
- Diffcult to get them well
- The community **hates** them



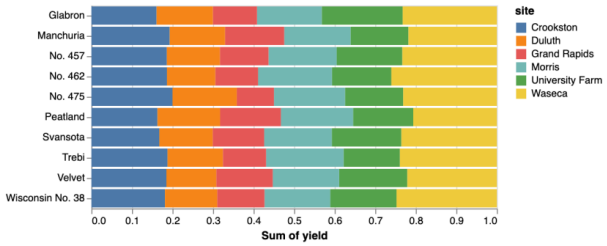
3.3.2 Polar area chart

We encode the value with the area of each of the sector. They still occupy a lot of space but it is easier to compare/estimate the values and distribution. Allows to stack categories but that makes everything harder.



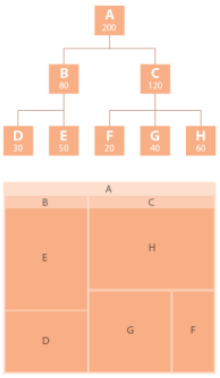
3.3.3 Normalized stack bar

Easy to compare the different categories. When many categories, the ones that are not adjacent are difficult to compare.



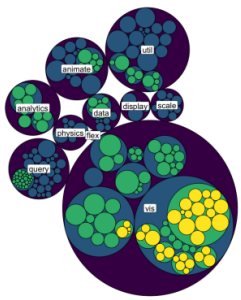
3.3.4 Tree maps (enclosure diagrams)

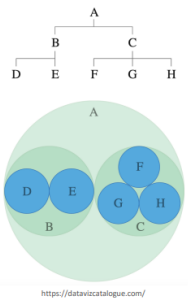
Areas are much more difficult to compare than bars. But we can use them to show hierarchycal data.



3.3.5 Circle packing

Allows to show hierarchycal data with circles rather than rectangles. Same problems as Tree maps.

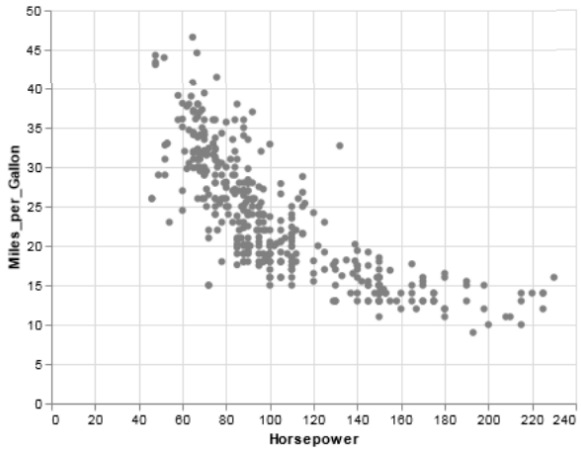




3.4 Display Relationships

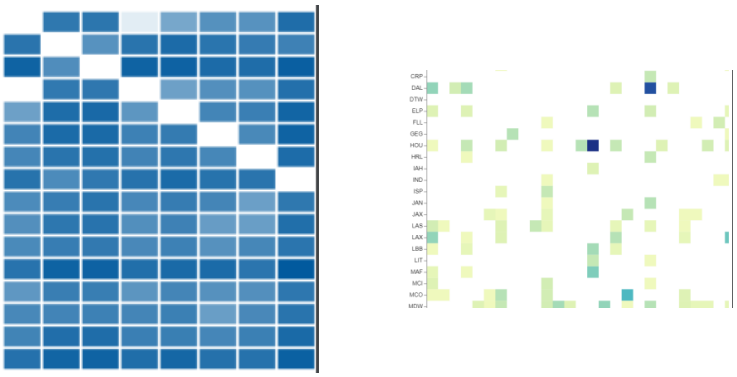
3.4.1 Scatterplots

- Typically represents data without keys. We have two quantitative values represented in points encoded by position.
- Useful to find correlations or outilers. If there are many points, may be difficult to understand when the clutter of points is dense.



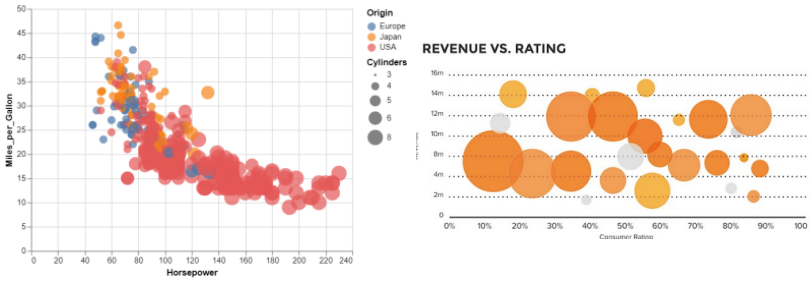
3.4.2 Heat Maps

- It is typically represented with an array of rectangles.
- They represent 2 categorical variables and we encode a 3rd quantitative value using color.
- The marks is the area located in a matrix encoded with the 2 categorical attributes.
- For finding outliers and clusters. Categorical variables have no order, so x and y can be ordered as we wish. *Reordered matrix*, with the objective of reordering to find clusters. (Not possible when a category is time).
- Commonly used in bio to encode gen expressions and so on.
- Color palette should not be continuous but discrete to notice the difference between values.



3.4.3 Bubble Charts

- Enhance/Increase the number of variables of scatter plot.
- Cluttering problem will appear much more earlier than a normal scatter plot. I can have overlapping depending on the distribution of the data. Common technique is not defining the points with `opacity = 1`, by making them semi-transparent to see if points overlap.
- Don't use ordered colors when representing categories like in Figure on the right.

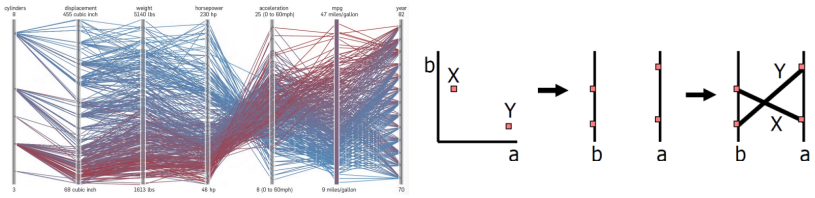


3.4.4 Scatterplot Matrices

- Combine different scatterplots. Provide the tool to analyse pair-wise relationships. Find relationships/correlations.
- Normally the same plot is repeated with a change of axis. Waste of space, waste of half of the matrix.
- Visualization is small due to the dimensions.
- Let the user some interaction tasks; *zoom out, ...*

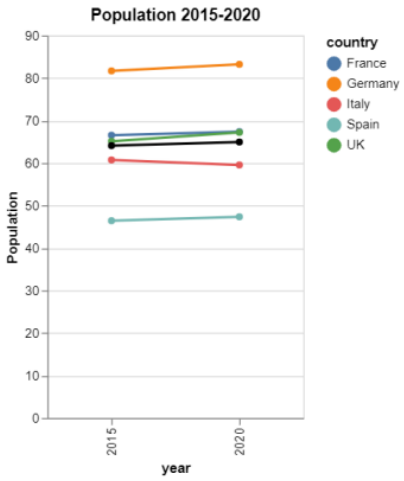
3.4.5 Parallel coordinate plots

- Show multiple variables. Let you see whether variables are correlated.
- Keys can be quantitative or categorical. Axis are scaled to the min/max values.
- Same behaviour for similar correlations. If there is a positive correlation the lines will be parallel and if negative, the lines will cross.
- Not easy to see correlation of not adjacent dimensions. We need to provide interactive tools so users can change the order of the dimensions, highlighting lines, etc.
- Not hundreds of elements. Can represent a large amount of points using special techniques like transparency.
- Scale really well



3.4.6 Slope charts

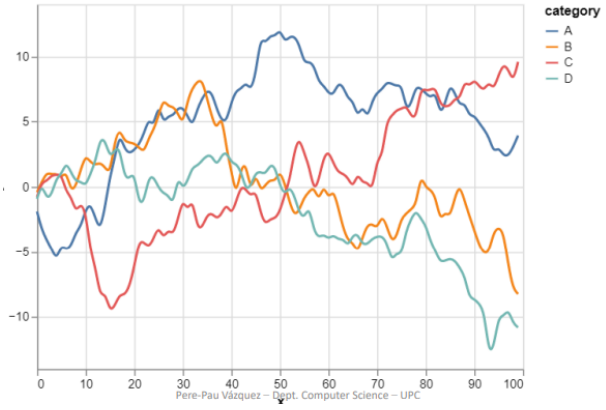
- Often encode two values. Normally of time instances.
- Intended to show increase/decrease of 2 data points along time.
- Really simple but useful. Lets us compare really quick.



3.5 Display Time Series

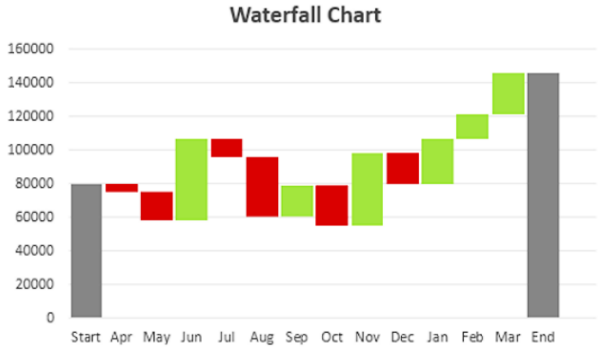
3.5.1 Line Charts

- The most common representation.
- One key-one value. Data is quantitative.
- Marks are points. Encode the quantity by the position of the points.
- Lines used to show trends.
- Different from bar charts. Lines can go into negative values, meaning that the axis does not have to start at 0.
- Do not scale very well. More than 10-12 lines will have several issues such us color palette and overlapping.



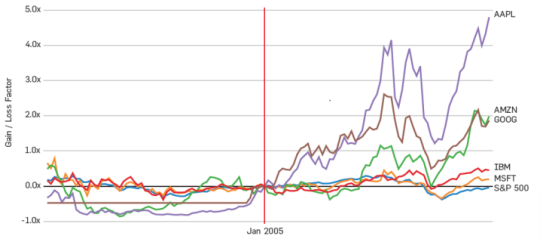
3.5.2 Waterfall Chart

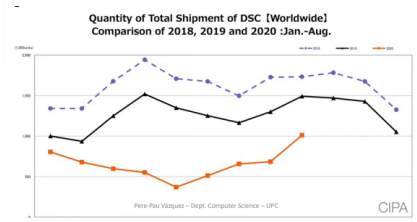
- Used in very specific scenarios. Mostly in business to represent cash flows (money entering and being spent).
- Don't start at 0. Start at the end of the value of the previous one.
- At the end we have another reference bar to encode the final value.
- Usually, use of two colors (green/red). Optionally a third color to the reference bar.



3.5.3 Index Chart

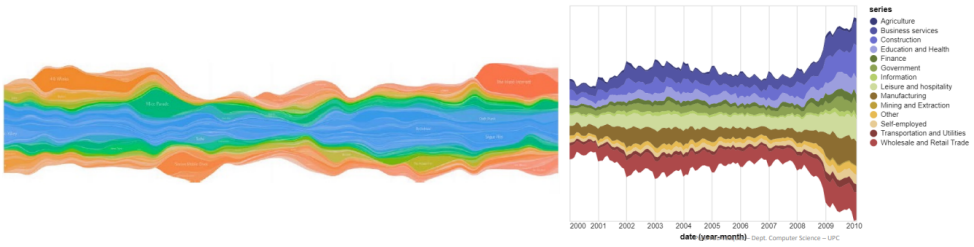
- Solve the problem of time data not being able to be compared evolution-wise. Instead of using as absolute values the values we want to encode, we use a reference.
- The value is the *now value* - *reference value*. Lets see the evolution. Positive values show an increase and negative decrease.
- Can be indexed in different ways, like in time (see Figure in right). We show different years in the same chart in order to compare easily how the different years have evolved.
- Used when we need to compare values in time fairly by setting starting points.





3.5.4 StreamGraph

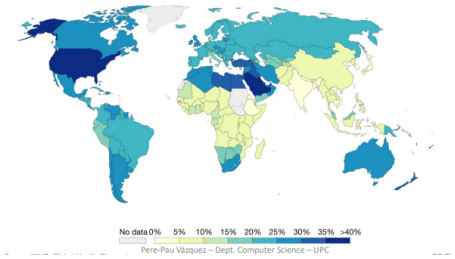
- Like stacked bars, with a very thin width. Bars can start at both directions of the axis.
- Grasps of trends. Identify the most important aspect and growth.
- Provide zooming, lenses, blah, blah...
- Try to communicate how things have evolved in time. We typically need to recalculate how the geometry is laid out because we want it to be smooth.
- Can have a lot of keys. They scale much better than stacked bars (altough stacked bars do not scale well) .
- Does not support negative values.
- Can not be used with information that can not be added such as temperature.
- Since elements are not aligned is difficult to interpret values and estimate trends.

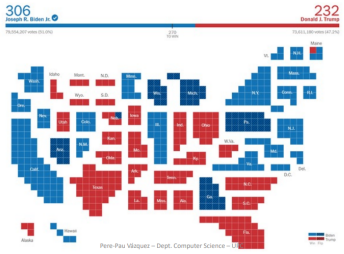


3.6 Display Geospatial Data

3.6.1 Choropleth Maps

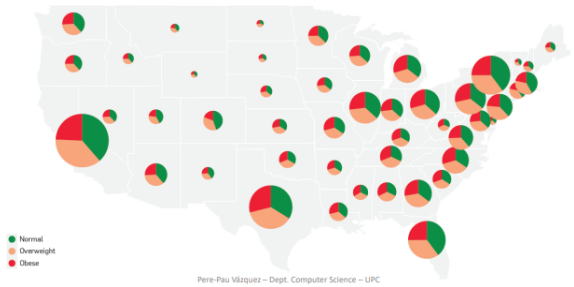
- We use maps when the geographic information is relevant.
- Problems:
  - The bigger the size of the area, the more it attracts our attention.
  - The position (things on top are thought to be more important than things on the bottom.
  - Ignoring the density, we usually have to normalize.
- Sequential discrete color palette.





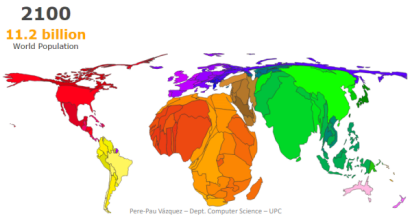
3.6.2 Graduated Symbol Maps

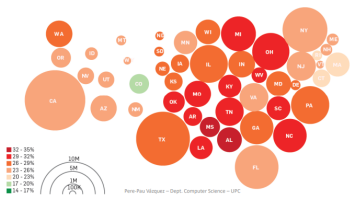
- Places symbols on the underlying map
- Avoid the problem of the geography interfering with the interpretation of the data.
- Enables visualizing more dimensions



3.6.3 Cartograms

- Modifying geometry is complex. Depending on the distortion, the countries can loose its shape and identification.



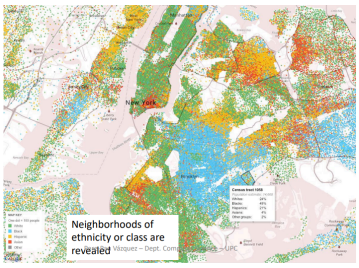


3.6.4 Dot maps

To represent data from a census, dot maps are a common way to visualize geospatial data, where each point on the map represents an observation or a set of data related to a specific geographical location.

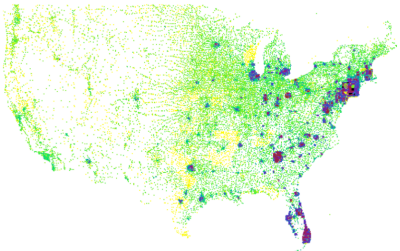
Issues:

- If the **size of the symbol is used to represent a quantitative** parameter, **scaling may present perception issues**. Variations in symbol size can make it difficult to make precise comparisons between different locations on the map.
- Perception of size also depends on the local environment of the points on the map. **Points close to each other may give the impression of being a group or a single entity**, which can be misleading in terms of the actual data distribution.
- **If color is used to represent a quantitative parameter, issues related to color perception may also arise**. People's ability to distinguish and comprehend differences in colors varies, which can lead to incorrect interpretations of the data.
- **When working with large datasets, there may be issues of overlap or overplotting of points on the map.** This occurs especially in densely populated areas, where multiple points overlap and make precise visualization challenging.
- **Areas with low population may appear virtually empty on the map**, which can result in a biased perception of data distribution, as individual points in scattered areas may not be clearly visible.



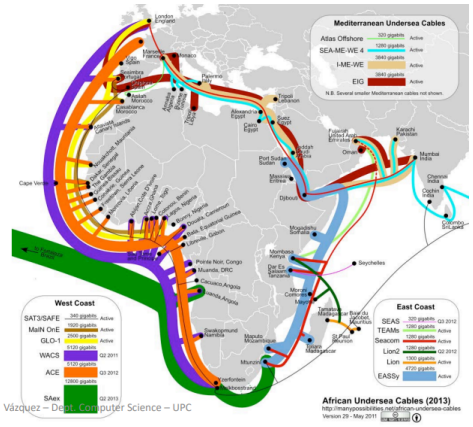
3.6.5 Pixel maps

- Repositions pixels that would otherwise overlap.
- Does not aggregate the data.
- Avoids overlap in the two-dimensional display.
- Provides quite an intuitive result.
- The main idea of the repositioning is to recursively partition the dataset into four subsets containing the data points in four equally-sized subregions.



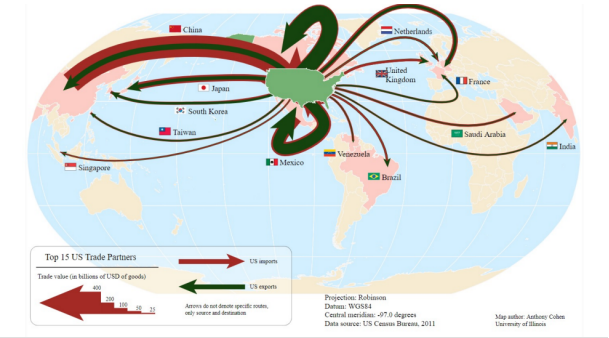
3.6.6 Lines in Geospatial maps

- Map attributes in the map to lines
- Limited application opportunities.



3.6.7 Flow maps

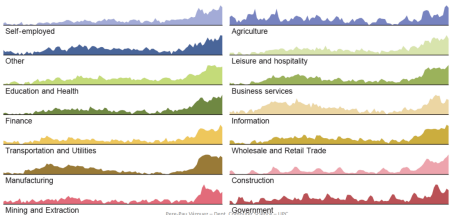
- Depicts movement of a quantity in space.
- Implicitly represents time.
- Can encode a large amount of multivariate information, including path points, direction, line thickness, color, and more.
- May require subtle distortion of the map.



3.7 Other maps

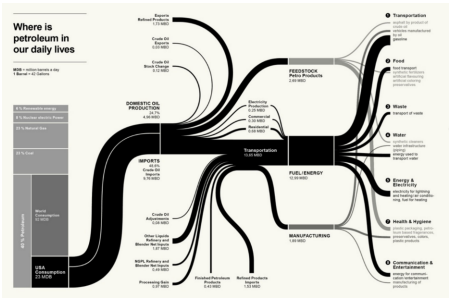
3.7.1 Multiple variables. Small multiples

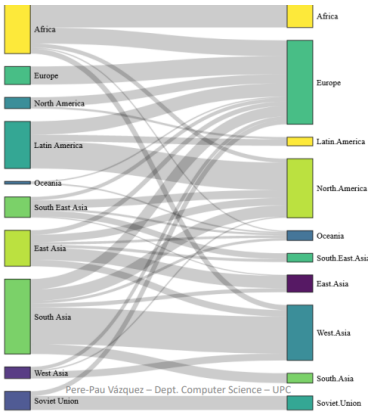
- Grid with axes of smaller charts: This approach is used to facilitate comparison. By arranging smaller charts on a grid with shared axes, it becomes easier to compare different data sets or visualizations side by side.
- The same can be done for time series: Instead of overlapping multiple time series on a single plot, creating small multiples is a useful technique. Small multiples involve creating separate, smaller charts for each time series, making it visually easier to identify trends, seasonal patterns, and other insights in the data.



3.7.2 Sankey Diagrams

- It is a specific type of flow diagram.
- The width of the arrows is proportional to the flow quantity.
- Emphasis is placed on the major flows in the system, allowing for the identification of dominant contributions to the general flow.
- To improve clarity, the diagram minimizes arrow crossings. This may involve omitting or downplaying small or weak flows.
- The relative positioning of nodes is also crucial, as the diagram may become unreadable if there are too many nodes.

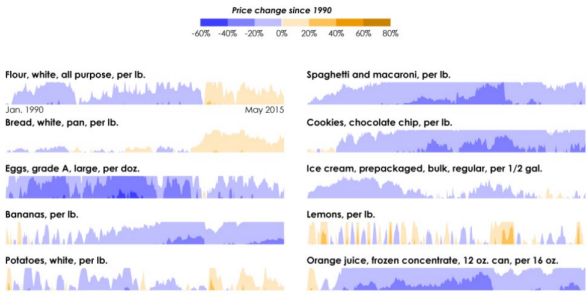




3.7.3 Horizon graphs

Increase data density by overlapping, while keeping the resolution of the graph.

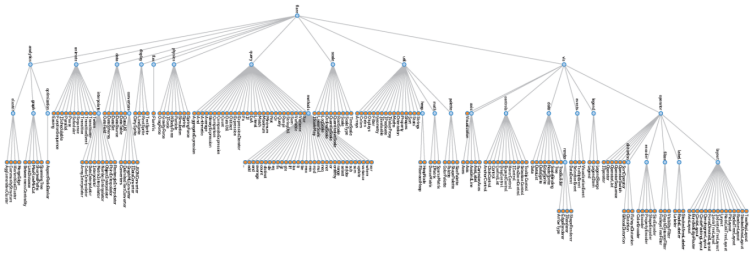
- Start with an area chart
- Mirror negatives to the positive side
- Divide the chart into bands, and mirror again
- Divide the chart into bands, and mirror again
- Result 25% less of vertical space with same resolution



3.8 Hierarchy: Node-link diagram

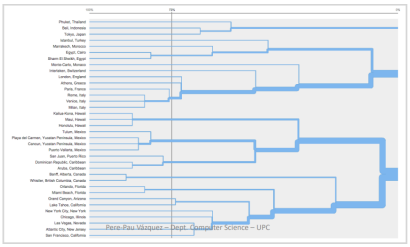
Visualization of hierarchycal relationships to find relationships, groups.

- Line crossing can be a problem
- Might cause a waste of space



3.8.1 Hierarchy: Dendograms

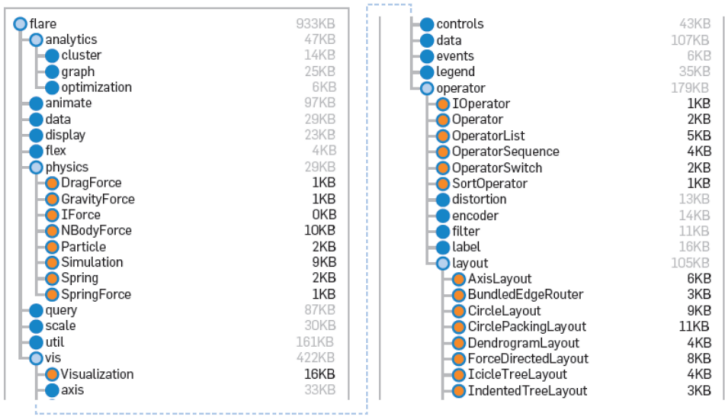
All the leaves are at the same level.



3.8.2 Hierarchy: Indented trees

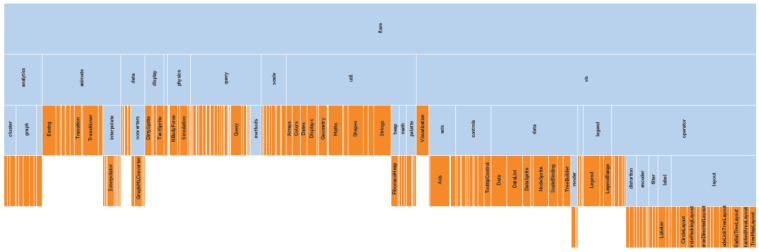
Use in OS to depict the directories.

- Requires a large amount of vertical space.
- Efficient interactive exploration of the tree (to find a specific node).
- Multivariate data shown adjacently (size of file, date, etc.).



3.8.3 Hierarchy: Adjacency diagram

- Space-filling variant of the node-link diagram.
- Nodes are drawn as solid areas.
- The placement of nodes relative to adjacent nodes illustrates their position in the hierarchy.
- Length can be used to encode an additional dimension of information.

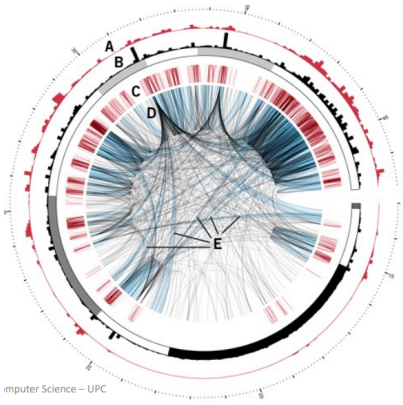


3.8.4 Networks

Contain information about relationships. Information such as who is connected to, who is a central player (connected to many nodes), groups, cliques. Must place related nodes close and unrelated far away. Reduce crossings may facilitate legibility. ***Note**: Node-link diagram is an example of a network. Its circle representation is better but implies a lot of crossing*

- Drawing is highly complex
- Collinear edges for a large number of nodes
- Very long edges and "meaningless" edge length

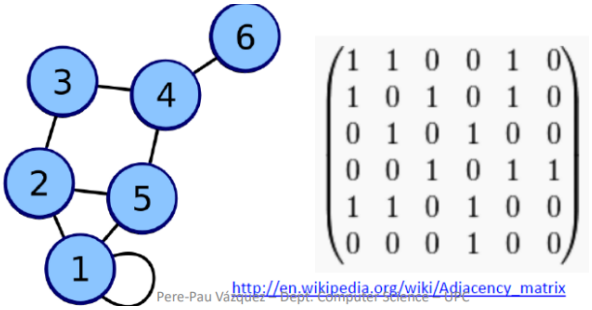
- Strong regularity can obscure inherent structures
- Very dense drawings for complex graphs
- As a circular layout, the external space can be used for more data.
- Highly regularized and tidy visualization
- Ordering of nodes possible
- Edges or nodes never overlay other nodes
- Easy to visually proceed along edges



3.8.5 Networks: Adjacency matrix

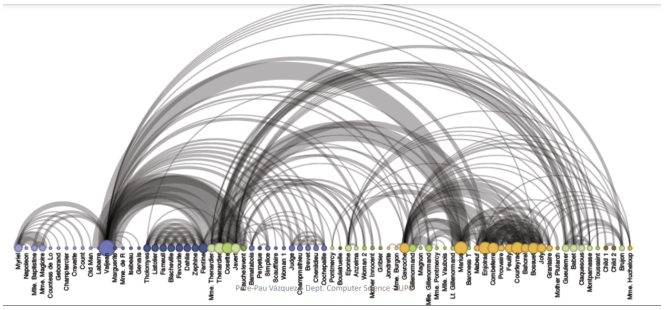
Uses adjacency matrix of the graph. *nxn* adjacency matrix of graph G with n nodes. Use of color might facilitate the interpretation of the links.

- Difficult to follow "paths" of connections.
- Reordering is expensive
- Crossings are impossible
- Ordering can reveal clusters and bridges (might be done interactively)
- Useful for dense graphs
- Visually scaleable



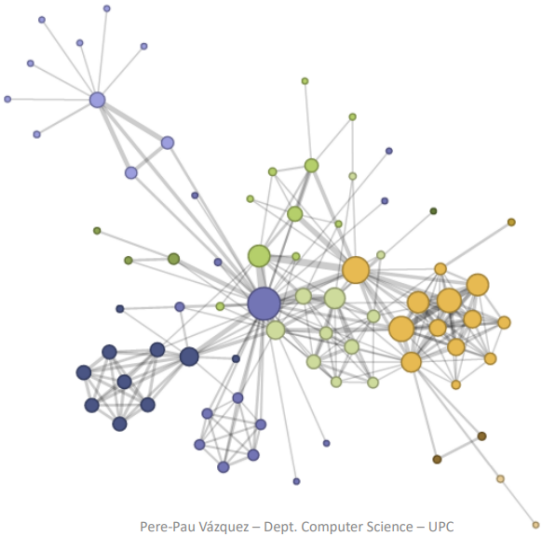
3.8.6 Networks: Arc diagram

- Lays the nodes in one dimension.
- Circular arcs represent links.
- Good ordering of nodes helps in identifying cliques and bridges.
- Problems may arise with the sorting of the data, known as seriation.
- Multivariate data can be displayed alongside nodes (for example: size, colour, etc)



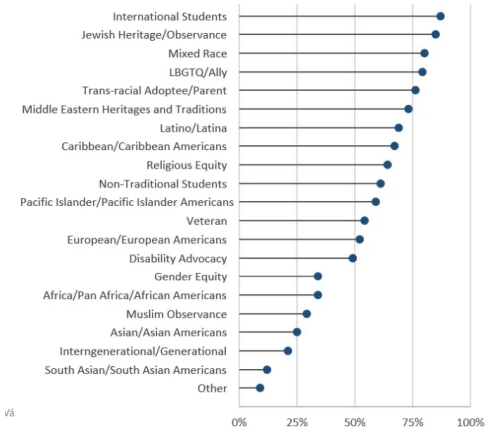
3.8.7 Force-directed layout

- Nodes are represented as charged particles that repel each other.
- Links are modeled as springs that pull related nodes together.
- The layout algorithm uses physical simulation of the forces between nodes to determine their positions.
- Interaction can be added to the visualization to disambiguate links and provide a more dynamic experience.

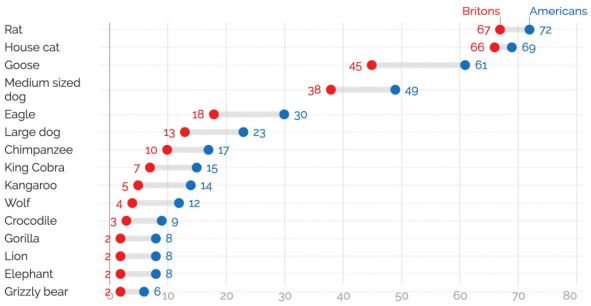


3.8.8 Lollipop

Might not start at 0.

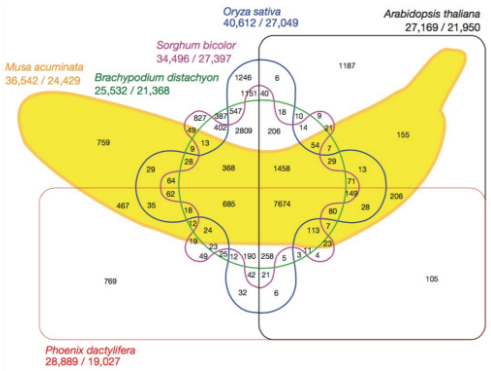


3.8.9 Dot plot with two values



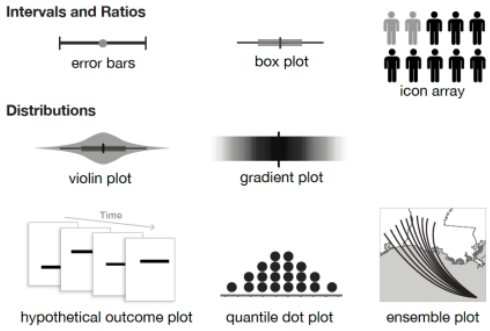
3.8.10 Intersection of sets

Generalization of Venn diagrams. Using Venn (aka Euler) diagrams for more than 3-4 sets is a bad idea. MANY possible intersections. **Perceptually efficient** visual encodings. Shows **combinations of intersections**. Can add attributes about the intersections



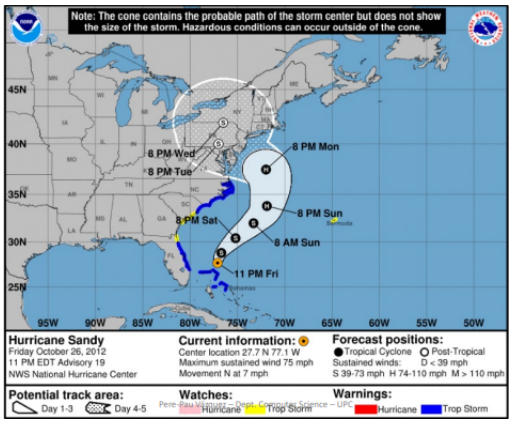
3.9 Uncertainty

Data is (very commonly) uncertain to a certain degree. We need to communicate this uncertainty. Regular users are not used to uncertainty visualization, because it may lead to the readers considering your data is flawed. Some common uncertainty methods are violin plots and line ranges.



The only uncertainty communication method that has gone to public is hurricane visualization:

- Cone **contains probable path**
- Uncertainty grows with time
- Forecasting models exhibit different behaviors (paths)



Problems with cone-based hurricane forecasting visualization:

- **Cone size != hurricane size**, hurricane impact outside cone

Problems with spaghetti plots forecasting visualization:

- Some are not models, some are old (e.g., 12 hours old), some are statistical models useless for tracking...

4 Perception

We need to understand which elements (distribution of data points, colors, geometric shapes) are involved in how the user perceives the data. The way we design the visualization determines our understanding of the data. In the context of the task we need to solve, we must provide visual tools and affordances.

Affordances: Being able to identify elements that assist you in interacting with the visualization.

- **Simple case scenario**: Low number of dimensions. Small data points and minor attributes. Nearly any visualization will work, but it may lack perceptual properties. With many data points, we might not be able to represent all the data in the visualization.
- **Worst case scenario**: A high number of dimensions that we cannot all represent, and we have to select the most important dimensions. Many visualization techniques may not work properly, even with the best selection of our dimensions. Perhaps some interaction techniques could work.
  - With three or more dimensions, things start to get a bit more complicated (overlapings, ...)
  - With a higher number of dimensions, the position in the matrix and colors can differentiate the dimensions, according to our visualization.

The perception properties vary for each visualization.

4.1 Preattentive Processing

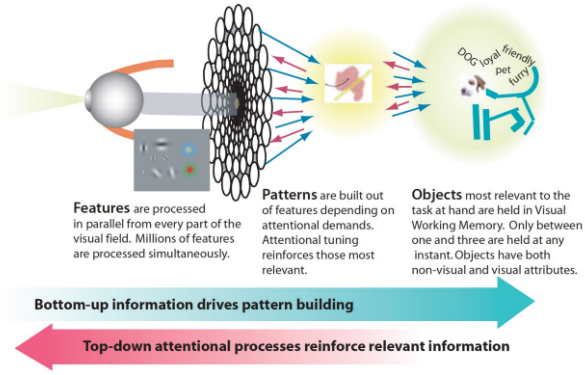
When creating a visualization it has to be simple and easy to understand. **Preattentive processing** can be described as the processing of sensory information that occurs before the conscious mind starts to pay attention. When talking about visuals, preattentive processing extracts basic visual features of the complete visual field.

Create something that can be **understood as quickly as possible** with **minimal cognitive effort**. Understand how human visual perception and information processing work. Very related to psychology.

Simplified 3-stage model:

1. The brain can process in **parallel to extract low-level properties**. Features are processed simultaneously. Here we detect things like shape, position, orientation, etc. This phase is made in unconscious way, independently of what is our tasks, we can not control it. This first stage is the preattentive processing.
2. **Extract some structures** via pattern perception. We construct pattern ans start recognizing objects. This process is not made in parallel (serial) so it takes longer.
3. Perform a **sequential search** to find the desired information. From everything we have seen the brain extracts a small set of objects that are the ones we really remember.

Certain variables can be easily identified (fast identification) before beginning to scan the entire visualization. This is important because the user can find what they were looking for and avoid distraction. Identify which features can be perceived rapidly and in what context to use that to your advantage in the visualization.



You have already seen the information before you start searching for it and we know this occurs before your consciousness does by measuring the response time. If it is below a certain threshold, we know that the visualization aid was useful.

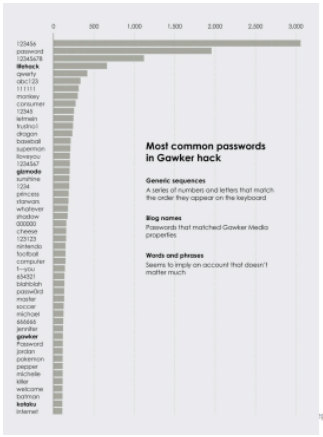
The conjunction of several variables is typically not preattentive (e.g., a mix of squares and circles with different colors when looking for the red circle). It requires a more time-consuming serial search for each dimension.

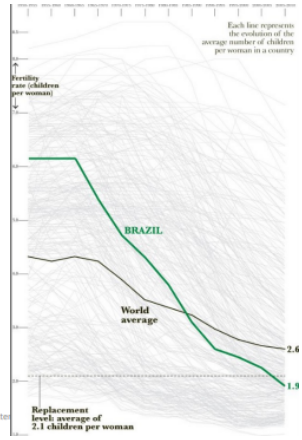
Conjunction search therefore involves looking for an object that is defined by a combination of distinct visual characteristics, such as color and shape, rather than a single feature. It is a more complex and challenging type of visual search compared to *feature search*, where the target is defined by a single, distinctive feature. There might be distractors. Only add color based on necessity to ensure that we don't have distractors. **Conjunction of 2 properties is usually not preattentive**.

Find preattentive attributes

1. Measure response time for tasks
2. Check weather time is smaller than certain threshold

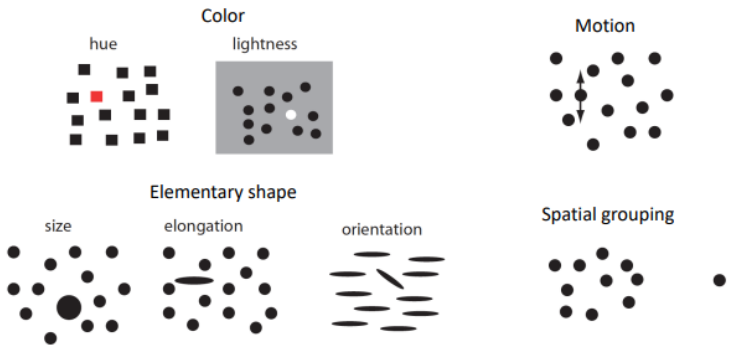
A limited set of basic visual properties are processed **preattentively.** This includes **information that "pops out"** and is amenable to parallel processing by the low-level visual system, which occurs prior to conscious attention. Understanding preattentive processing is important for designing effective visualizations. It involves questions like: **What features can be perceived rapidly? Which properties are good discriminators? What can mislead viewers? How can information be designed so that it pops out?**





Preattentive processing works when we know what we are looking for. The brain **prioritizes cells sensitive to the element we are searching for**, giving them more relevance, while partially silencing others. Preattentive processing is **highly sensitive to distractors**, which can interfere with efficient processing of the desired element. Interestingly, training does not seem to have any significant influence on this process. **Movement ALWAYS attracts our attention**.

Basic pop-out channels



4.2 Perception Laws

Depending on the visual organization of the data, we can perceive some information and not others. Once identified, it is difficult to stop seeing that information. In general, our brains try to create meaningful things from what we see, even if it is not real. The brain tries to extract some information. The structure of the visualization helps the brain create those ideas to be true.

4.2.1 Pragnänz Law

We tend to perceive simpler shapes. Relates to the organization of visual elements in a way that simplifies complex scenes or patterns.

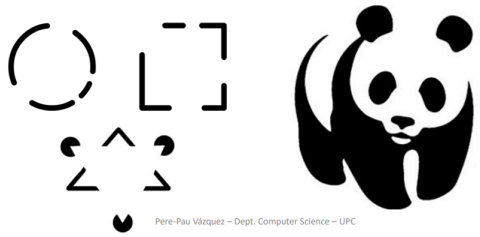
When applied in visualization and design, the Prägnanz Law can be advantageous in several ways:

1. **Clarity:** By simplifying complex information and reducing it to its essential elements, you can make data and messages more accessible and understandable to your audience.
2. **Reducing Cognitive Load:** Minimizing visual complexity reduces the cognitive load on the viewer, making it easier for them to absorb and retain information.
3. **Enhancing Engagement:** Simple and well-organized visuals are more engaging and can hold the viewer's attention for longer, leading to better comprehension and retention.

4. **Aesthetics:** Clean and straightforward designs are often considered more aesthetically pleasing, which can enhance the overall user experience.

4.2.2 Law of Closure

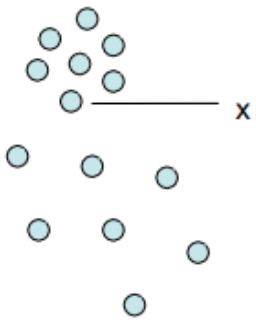
The division of complex elements enables us to detect simpler elements/shapes. The mind sees the complex element as a combination of simpler shapes. The Law suggests that when presented with a set of separate elements, individuals tend to mentally complete or close gaps in those elements to perceive a whole, complete object or shape.



1. **Design with Intention:** Consider where you want viewers to perceive closure in your design. This may involve leaving strategic gaps or incorporating open-ended lines and shapes to guide the viewer's interpretation.
2. **Balance Simplicity and Complexity:** While closure simplifies complex shapes, you should strike a balance between simplicity and the need for conveying necessary information. The design should not become too abstract or obscure.

4.2.3 Grouping by Spatial Proximity

Even small differences lead to a different perception of the distribution of the elements. Things close together form a group. Increasing the space between them makes the user think that the two groups are not related. Grouping by similarity is another variant.

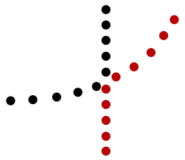


4.2.4 Law of Continuity

There is a tendency to make trends continue. The shape of the edge could create the illusion that two elements are connected.

1. **Consider Data Flow:** Think about the natural flow of data or information in your visualization. Use the principle of continuity to guide viewers along this flow, whether it's a chronological sequence, a process, or a narrative.

2. **Maintain Consistency:** Ensure that the elements you want to be perceived as continuous share similarities in terms of color, shape, or other visual attributes. This consistency reinforces the perception of continuity.
3. **Balance Complexity:** While continuity can simplify complex information, be mindful of maintaining a balance between simplicity and the need for conveying necessary details. The design should be informative and not overly abstract.



4.2.5 Law of Common Fate

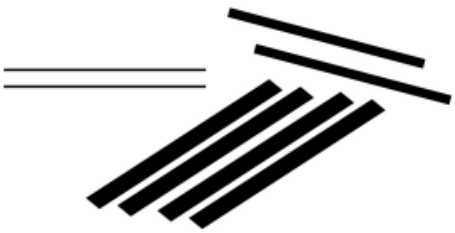
There is a tendency to group elements that move in the same direction. This law suggests that when elements in a visual display share a common direction or movement, they are perceived as belonging to the same group or category.

1. **Understand Data Relationships:** Consider the data or information you're presenting and identify relationships or categories that can be represented through shared movement or direction.
2. **Use Consistent Movement:** Ensure that elements that belong to the same group or category share consistent movement patterns. This could involve having elements move in the same direction, at the same speed, or with similar animations.
3. **Balance and Moderation:** While movement can enhance visualization, be cautious not to overuse it. Too much motion can be distracting and overwhelming. Find the right balance to maintain clarity and effectiveness.

4.2.6 Principle of Parallelism

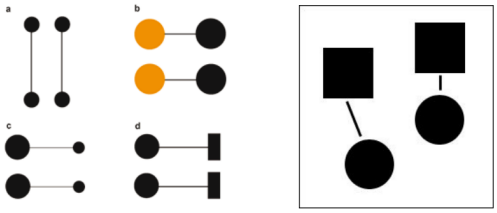
Similar to the law of common fate, but here the groups are based on the orientation of the elements. The principle that emphasizes the use of parallel lines or patterns to create a sense of order, structure, and organization in visual design.

1. Balance and Symmetry: Use parallelism to create balanced and symmetrical compositions, which are pleasing to the eye. Achieving balance through parallel lines can help maintain order and clarity.
2. Flow and Direction: If your visualization has a narrative or sequence, consider how parallel lines can guide the viewer's eye along the intended path. This can help convey a story or process more effectively.



4.2.7 Principle of Connectedness

Elements being visually connected are perceived as more related than unconnected elements. This is a strong principle that can override the other principles. Visual connections can make us reconsider the initial grouping. The line of connection does not have to be strictly connected to the elements (there can be space between the element and the line).



For example, circuit designs are mainly understood by following the connecting lines.

1. **Use Visual Links:** Create visual connections between related elements using lines, arrows, or other visual cues. The style of the connection should be consistent and easily distinguishable from other design elements.
2. **Visual Hierarchy:** Use the Principle of Connectedness to establish a clear visual hierarchy. For example, you can create a central node that connects to various sub-nodes, indicating their hierarchy and relationship.
3. **Balance and Clarity:** Be mindful of balance and clarity in your design. While connected elements provide organization, avoid overcrowding and ensure that the connections enhance, rather than obscure, the information.

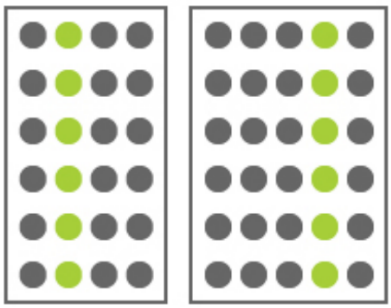
4.2.8 Law of Symmetry

Elements are grouped based on their symmetry. Symmetrical images are perceived collectively, even in spite of distance.

1. **Aesthetic Appeal:** The use of symmetry often results in aesthetically pleasing and harmonious designs. Visual elements that are balanced and symmetrical tend to be more attractive to viewers, which can enhance the overall appeal of your visualization.
2. **Clarity and Organization:** Symmetrical compositions provide a sense of order and organization. This can be particularly advantageous when dealing with complex data or information. A well-balanced design makes it easier for viewers to understand the structure and relationships within the visualization.
3. **Visual Hierarchy:** Symmetry can help establish a clear visual hierarchy by creating a focal point or central element. This can guide the viewer's attention to specific information or data points within the visualization.

4.2.9 Principle of Common Regions

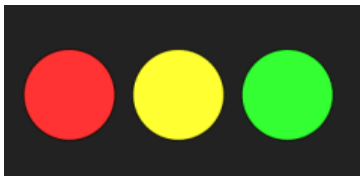
Principle of common region: Elements located in the same closed region are perceived as a group (containment). It's based on the boundary lines that contain the elements, even though the elements inside the grouping can have evident differentiation.



4.2.10 Principle of Previous Experience

Our past influences how we perceive the elements. Take this into account to avoid misleading the user and save time by making the visualization more intuitive (e.g., red for negative and positive values).

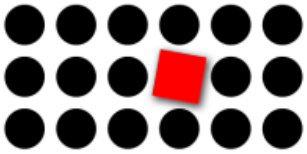
- There is a tendency to distinguish and detect which elements are in the foreground and which are in the background. We don't want to create elements that are wrong or confusing and do not distinguish between the foreground and background.



4.2.11 Principle of Focal Point

Emphasize the viewer's attention on a specific element of the visualization that serves as an entry point into the visualization. Sometimes, we can enhance it with captivating techniques.

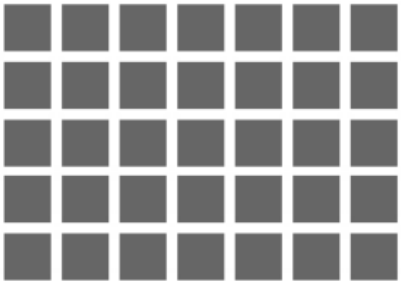
1. **Guiding Attention:** The focal point serves as a visual anchor, directing the viewer's gaze to a specific element or area within the visualization. It helps ensure that the most important information or message is noticed first.
2. **Creating Visual Hierarchy:** The principle of focal point helps establish a clear visual hierarchy. It allows you to differentiate between primary and secondary information, making it easier for viewers to understand the significance of each element.
3. **Enhancing Engagement:** A strong focal point can capture the viewer's attention and make the visualization more engaging and memorable. It encourages viewers to explore the content in more depth.



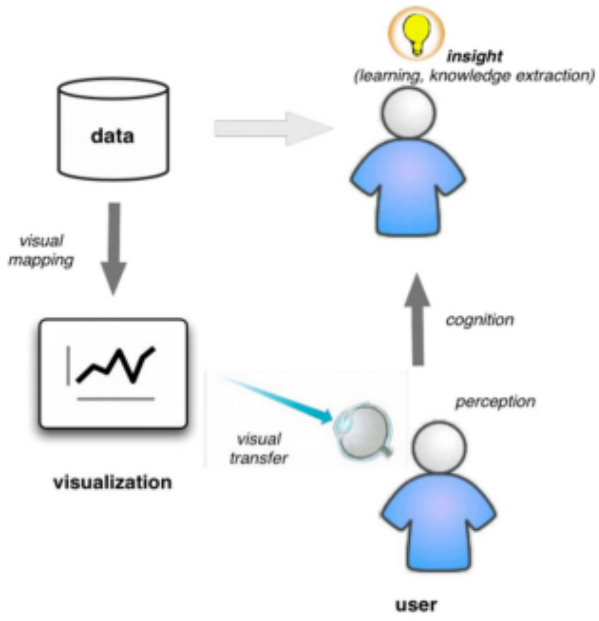
4.2.12 1 + 1 = 3 Effect

A perceptual phenomenon that occurs when elements that are not present appear in our visual system because the organization of the visualization allows it. For example, the small light gray squares in the corners of the big squares.

- We don't want visualizations that allow this principle. For instance, in maps with labels, the labels can be enclosed inside rectangles.



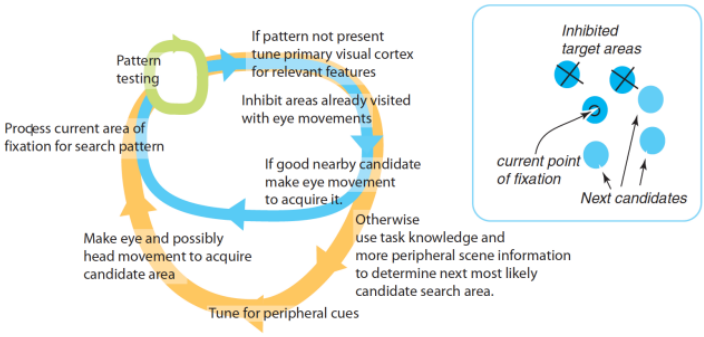
4.3 Application of Perception



4.3.1 Feature Hierarchy

Decide on the hierarchy in which you want to communicate the features, as visual search is hierarchical. Clearly separate them to solve the problem.

- The first scanning process gives an initial understanding of the main elements.
- Eye movements are used to look for new feature candidates, with a limited amount of storage in memory.
- Test whether the extracted features are the ones you were looking for.



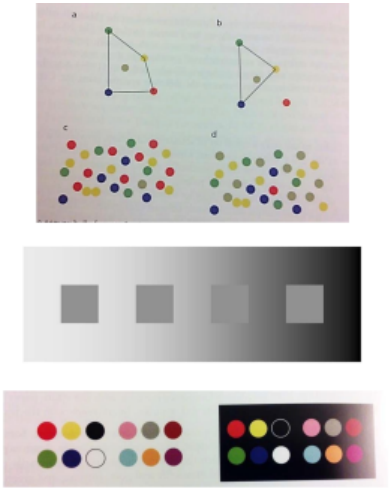
Applying feature hierarchy in visualization involves organizing the layout from large to small while maintaining a structured approach at each level. This not only enhances the visual appeal but also significantly improves search efficiency. It's crucial to place the most important piece of information using the most sensitive channel to ensure its prominence. However, there will always be competition between different visual channels, and when dealing with conjunction searches, it's essential to consider factors like the availability of free channels, size, and the surrounding environment or background.

4.3.2 Visual variables

Each visual variable is distinct, characterized by different properties. For instance, when using colors, various attributes like lightness, hue, and saturation can be employed. Selecting a visual variable to encode information involves considering several factors, such as the number of distinct levels and its interaction with other visual elements. Depending on the specific visual variable in use, different considerations come into play.

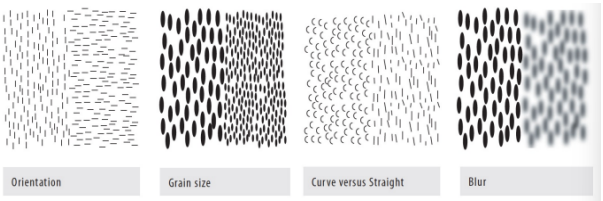
When dealing with color choice, some factors to consider are:

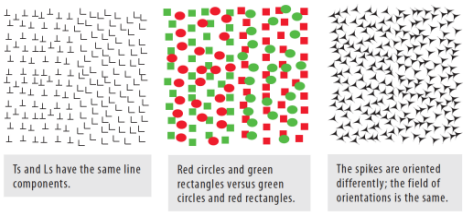
- **Distinctness**: capacity of distinguish different colors.
- **Unique hues**: if we have to use tonalities that are unique (different for each one) or tonnes that do not need to be recognized unequivocally.
- **Contrast with background**: choose colors to contrast with the background.
- **Number $\implies$ Difference:** if we want to distinguish them the number of colors must be small.
- **Field Size**: used space
- **Color blindness**: possibility of elements to be distinguished or not.
- **Conventions**: for example green or red cannot be used in the place of the other.



4.3.3 Texture

Visual variable that can also be used. For example orientation, grain size, blur, shapes, contrast, etc. They might add a lot of noise to the visualization. We can combine methods, orientation and colors, directions, etc.





4.3.4 Glyphs

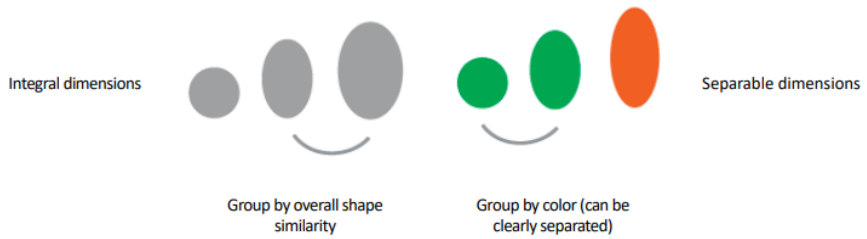
Used when we want to represent a lot of variables in an small space. Glyphs are geometrical forms that intend to show many attributes. It will be a success if we achieve to encode this variables in forms that can be distinguished in a good manner.

Visual representation that encodes data in a specific shape. Encodes attributes of data that can change, so f.e. we can encode two variables with the width and the length of a rectangle. One or more quantitative data attributes are mapped in a systematic way to the different graphical properties of an object.

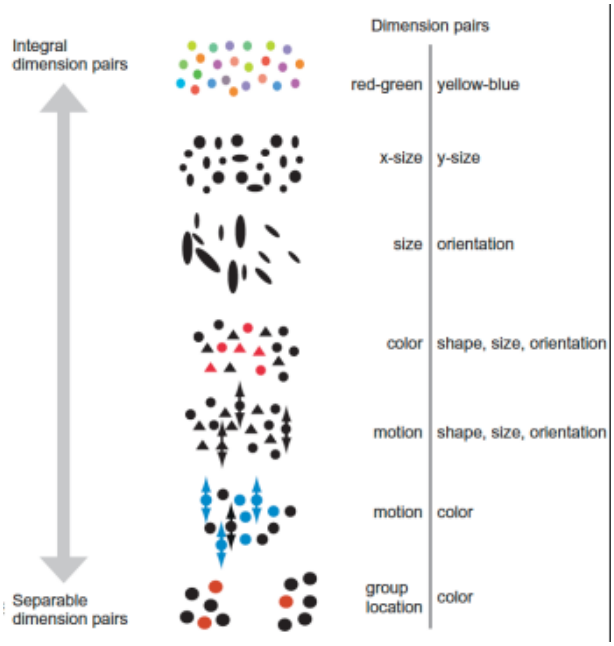
Challenges of encoding can appear when we cannot distinguish/concentrate on only one dimension (separate dimensions visually). In the rectangle example we can not distinguish/separate both dimensions (length, width).

Perceptual independence of the display dimensions:

- **Integral dimensions:** refers to dimensions that we can not separate if we do not do an strong cognitive effort. It is not possible to perceive or attend to only one dimension without attending to the other. For example, a rectangular shape is perceived as a combination of its width and height.
- **Separable dimensions:** it is really easy to identify if 2 variables encoded in that way are 2 different variables. It is possible to perceive or attend to only one dimension without attending to the other. For instance, you can independently perceive the size and orientation of a line or the size and color of a ball.



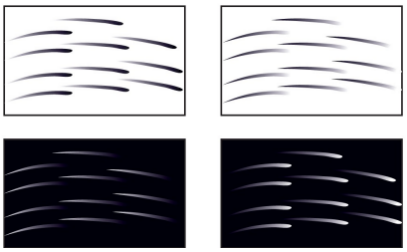
- If we want users to respond **holistically** (taking everything into account), use integral dimensions.
- If we want users to respond **analytically**, understanding one variable at a time, use separable dimensions.



4.3.5 Direction and orientation

It is not something easy to do, because the intuitive thought of arrows can occupy a lot of space. Another methods can be used are opacity and width of the direction line. Things to consider:

- Critical points have to be well represented.
- Depending on the task adjust the visualization.



4.3.6 Transparency

For encoding one thing on top of the other. The main problem when using transparency is that some colors when overlapping can create different colors or other colors that appear. For reducing clutter we can use it sometimes.

Not only to solve problems but to encode information. Be aware if the colors are the adequates and the contorns are not lost by the interceptions of elements. In general there is a lot of interference, we have to select good color palette. The mixture of texture and color can be better to represent multiple intersections of elements.

Laciness: Conditions in which image is perceived as two distinct layers instead of one fused (the result of combining both layers) and we cannot intepret that there are 2 layers.

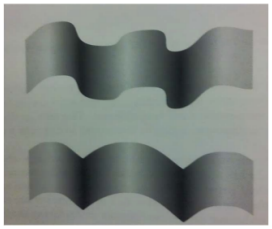
4.4 Pattern learning

Users are really good detecting patters, but this patters have to be learned. Some patterns are already known. We have to **make use of those instead of creating new ones**. When we are creating complex charts we may use examples, teaching the people to read our charts. If we do not teach the people we can not afford difficult examples. We can reinforce the visualization with new elements like legends.

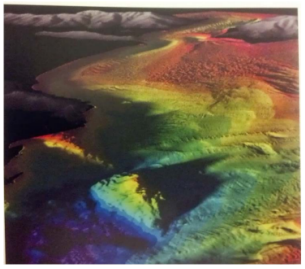
4.4.1 Complex surfaces

Some methods can help the user to perceive shapes in a visualization. In general we should not abuse the use of shades. If we want to mimic reality when using visualization

we will need a lot of shapes and more elements, not recommended since it can hide the real data. Generating shadows can be dangerous and using textures is right but not really promising.



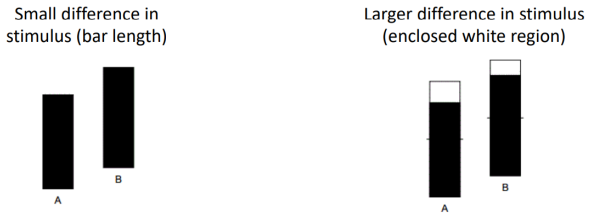
Shading and contours



Shading models - Lighting

- Simple lighting model should be normally used
- Inter-reflection must be avoided
- Specular reflection is useful to reveal fine details
- Shadow casting can be used only if they don't interfere with other information
- Surfaces may be textured, but low contrast to avoid interference with shading information

4.4.2 Relative judgements



The detection of difference in the bars is not visually instant when the bars are large.

Weber's Law describes the relationship between the magnitude of a physical stimulus and the perceived change in sensation or perception. It states that the just-noticeable difference (JND), or the smallest perceptible change in a stimulus, is a constant proportion of the original stimulus intensity. In other words, the JND is not a fixed amount but a constant fraction or percentage of the initial stimulus.

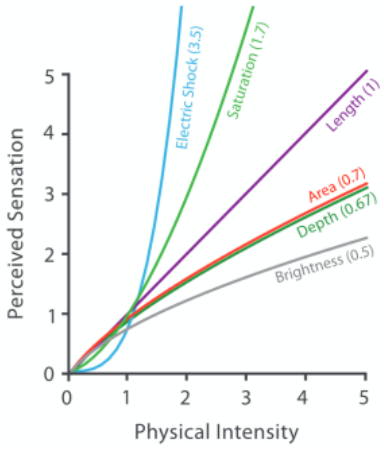
$$JND(k) = \Delta I/I$$

ΔI represents the change in stimulus intensity (the JND). I represents the initial stimulus intensity. k is a constant that depends on the sensory modality and the specific nature of the stimulus.

The perception of this difference can be highlighted with the encoding of the difference visually with an enclosed region or similar method. This means that for comparison we might interested in adding reference to make the comparison easier. This reference affect the way of understanding the visualization, because they compete with the data for our visual attention.

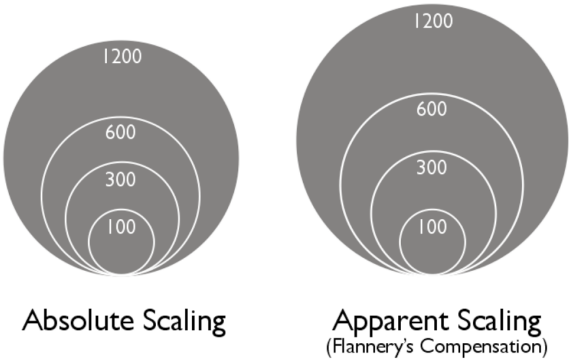
1. **Color Differentiation:** In data visualizations that use color to represent data categories or values, the JND plays a crucial role. If the difference in color between two categories is too subtle (below the JND), viewers may have difficulty distinguishing them.
2. **Line Thickness and Length:** In line charts, bar graphs, and scatterplots, line thickness and length can convey information. Lines that are too thin or too short may not be distinguishable from one another.
3. **Data Point Separation:** The distance between data points can affect how viewers perceive relationships and trends. If data points are too close to each other, viewers may have difficulty distinguishing individual points, especially in dense scatterplots. Adequate separation is required to ensure that data points are distinguishable.

We have to know how accurate we are when estimating values. When estimating values we are more accurate when estimating lengths than areas (and volumes). The higher the dimensions more difficult to estimate. **Steven's psycho-physical power law**: states that the length is very efficient at transmitting information, since estimation is practically imminent.



Flanner's compensation

Assign a bigger area that should be to compensate visual perception.



4.4.3 Tell truth about data

Try not to distort the perception of the user of the visualization. The lie is the distortion of the graph according to the actual data. The **lie factor** is a measure of how much a data visualization distorts the data it represents, particularly in terms of the size of the effect or change being displayed. It is calculated as the size of the effect shown in the visualization (e.g., the difference between two data points) divided by the actual size of that effect in the data.

$$\boxed{\text{Lie Factor} = \frac{\text{size of the effect shown in graphics}}{\text{size of the effect in data}}}$$

If the lie factor is equal to 1, it means that the visualization accurately represents the data, and there is no distortion. However, if the lie factor is significantly greater or less than 1, it indicates that the visualization either exaggerates or understates the data's actual effect.

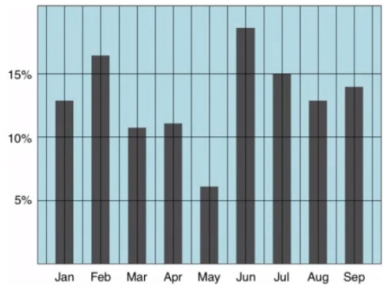
We need to be aware that the visualization must match the actual data. The impression of the visualization stays longer in mind than the actual data.

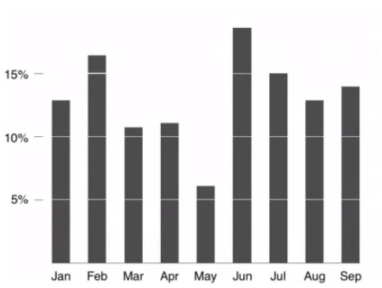
4.5 Data-Ink ratio

Signal to noise ratio is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. A ratio higher than 1:1 indicates more signal than noise. The primary goal of communication is to maximize the signal while minimizing noise.

When converting this measure to visualization, it's important to consider the concept of **data-to-ink ratio**. Keeping the design simple enhances perception, and you can further enhance information by using redundant coding and highlighting. To effectively reduce noise, **it's crucial to eliminate unnecessary elements from the visualization.**

$$\text{Data-Ink Ratio} = \frac{\text{Ink used to represent data}}{\text{Total Ink used in the graphic}}$$





4.5.1 Innovative charts

We need to ensure people get it. If we create some visualization that uses not standard technique to represent some concept we have to ask somebody what they interpret.

The are 2 ways of analyzing these charts. The first thing is try to guess what the chart is meaning without the labels. Then reading the labels try to see if our interpretation changes. The text content of the visualization can give a previous expectation on what the visualization is about.

4.6 Comparison

To facilitate comparison thought visual design:

- **Identify the elements to compare**: target can be explicitly/implicitly defined.
- **Identify the elements that make the comparison difficult**: When the number of items increases, the complexity increases and this will have to be taken into account for the design of comparison. The complexity of the items itself can be a problem itself, f.e. the shapes of two countries, atom molecules, etc.
- **Proper comparative strategy**: We can select a subset of the whole data.
- **Craft a design that facilitate comparison**: When we know the elements we want to compare we have to make something where the user can estimate quantities, difference. The are multiple ways.

5 Multiple Views

When we have multiple data, a single detailed visualization can be difficult to create and understand by the user. We can rather represent multiple views of the same chart.

- Multiple views need a bigger cognitive effort since we have to remember we have seen. Remember all the previous details. It is better to have them near.
- We can use them also to give context, comparison, detail, etc.
- The division of space into multiple views can result in problems. We need to asses if the multiple views are useful for this space waste.
- Makes th user to understand better the data and we have different ways to represent the data.

5.1 Juxtaposition

Usually one next to the other,

- Comparison is easy. - To emphasise different datasets and their similarities. - If the views have no common elements, the comparision will only introduce noise.

5.2 Superposition

In the same plot.

- We normaly use different color to differentiate more datasets. - Only when we dont have may items, they will be easyly separetad.
- The eye movement is less than in juxtaposition.
- Interaction can result efective.
- Map are a common example of multiple layers in the same visualization.

5.3 Explicit encoding

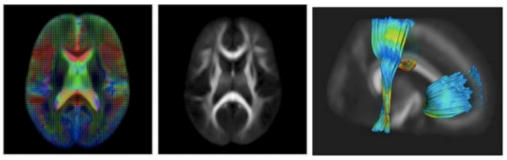
Some operation between the data and have another way of comparison from the data.

6 Data Reduction

One approach to reducing dimensionality involves:

- Subdividing the data over time to display different slices.
- Faceting the data across multiple views.

The choice of method depends on the nature of the data. While aggregation is always an option, averaging poses a complex problem as it often diminishes the strength of the signal and can be problematic.



Various representations of the entire brain may require rotating images for navigation through different slices.

The most common techniques for data reduction include filtering, aggregation, and using dimensionality reduction algorithms.

- **Filtering:** Involves removing some information.
- **Aggregation:** Merges elements to provide an overview of the data, sometimes created temporally.

- **Overviews:** Generated temporarily for a specific data view, allowing a detailed focus on the data.

6.1 Filtering

Filtering can be spatial or non-spatial. It can be implemented in two ways: allowing users to navigate freely or constraining their movements to enhance efficiency and focus. Techniques like **panning** (movement through visualization) or **rotating** (especially in 3D) can be employed. Zooming, either geometric or semantic, offers additional ways to explore data.

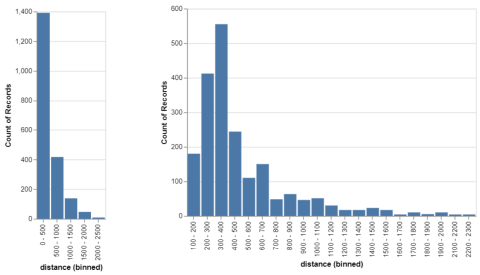
6.1.1 Non-spatial Item Filtering

This approach disregards item locations and focuses on item attributes. Dynamically changing views and restricting items based on queries help maintain user attention by minimizing visualization delays.

6.2 Data Aggregation

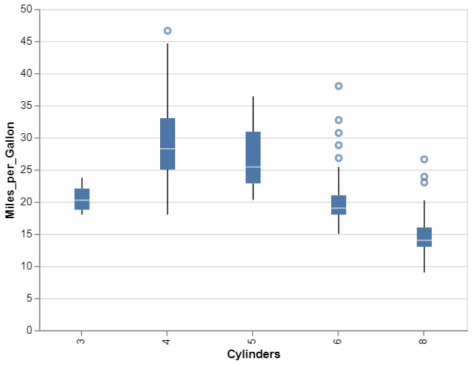
Aggregation involves combining items to reduce their number in the visualization. To prevent users from losing the perception of the original item count, new visualization motifs can be created. Combining items instead of eliminating them can introduce new attributes (e.g., max, min), but it may lead to signal averaging issues.

Count of elements is a common aggregation method, represented as bar charts segmented into typical ranges. Determining the ideal number of bins is crucial, as too few can hide important data, while too many can distort the representation.



Varying histograms with different bin sizes can dramatically alter the representation, emphasizing the importance of choosing appropriate bin sizes.

Boxplots provide a statistical summary of the distribution.

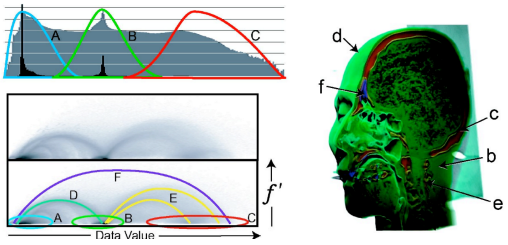


However, they lack detailed distribution information. Alternatives such as violin plots or strip plots can address this limitation.

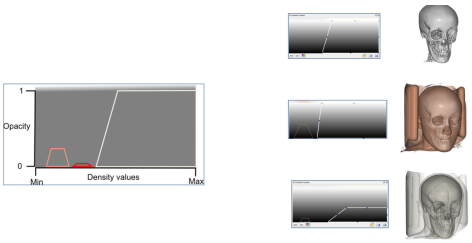
6.3 Data Reduction Techniques

6.3.1 Classification

For volumetric data (e.g., medical data), which encodes volumes in space, classification is a common technique. It involves defining a **transfer function** mapping volumetric values to a 2D range, independent of their 3D positioning.



In this example, different classifications highlight specific aspects of the data, such as showing only bones, introducing skin with a distinct color, and combining both bone and skin information.



6.3.2 Dimensionality Reduction

Given a N-dimensional dataset, i might be interesting in reducing nito a smaller number of dimensions that allow me to represent the same data.

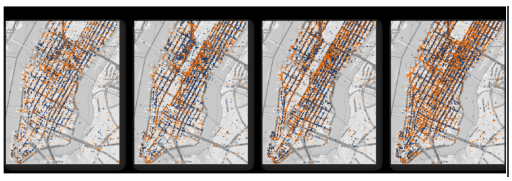
Depending on the data, sometimes is better to use small multiples instead but there is not a specific rule to follow a unique strategy.

6.3.3 Levels of Detail

Design levels of detail. Data can be very complex like in map. If we see the earth form fare we can represented as a globe, and closer in 2d, if we are closer we can repsretent the cities, then the roads, if closer a lot of details of the streets.

Basically we create algorithms thath depending on have much space in the screen and how complex is the data generatesdifferent vis thath can be unsderstood. Aggregarion or geometric representaiton can be used.

Pickups (blue) and dropoffs (orange) in Manhattan on May 1st from 7am to 11am. If we put all the point into a single view, it will not be visuable. Instead here we use interaction to define the level of detail to extract the relevant information.



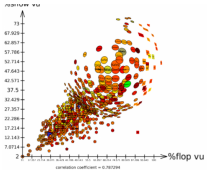
6.3.4 Navigation

In order to achive all this changed we need to interact with data, this can be done with the mouse or some widgets of the app. Unless the user is expected to use some intereactiont, we need to update the vis really fast. WE also need to create the most intuitive nav system fot he user. If user expecting that scroll is going to do somthg we have to do it.

6.3.5 Focus and Context

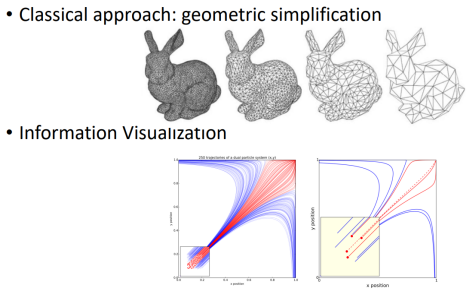
Give more detail in one region than the other. **Virtual lences** a lence tha distorts the data in one part (the area of interest), it can be spatial or semantic.

- magnifying glass - magic lences - distortion in x, y axis - fish-eye lence



6.3.6 Geometric Simplification

Other way to generate overviwes is performing the equivalent to geometric simplification. Translate complex information/shapes into simplier ones.



The way to simpleificy is to have lines that represent other lines. Not easy. Find a good similarity function for sitributions that do not hide to much data. An estreme case would be havind a single line representing (maybe diagonal), and if we have multiple attributes (like blue and red) we can not do it. Simplificating in this way is not simple.

The general the problem is that we need more pixels.

6.3.7 Dimensionality Reduction

Reduce dimension more common into 2D space. Keeping/explain as much variance as possiblea andgive a good impression of the real high dimensional data. (*PCA is a linear tecnique*).

Take into account that there is an information loss depending on the importance on the variables and dimensions.

In many cases we might not be intereseted in counting how much variance we lost, or the distance bwetten the high ald low dimensions are high, but seeing if the structures are the same, grouping is similar.

SLicing: losing informaction, trying to get a sense that where is behind, the elements of the intersection.

Projection: in this case we mimic what is done in X-rays with **MIP** and we try to better comunicate the distances using a color encoding.

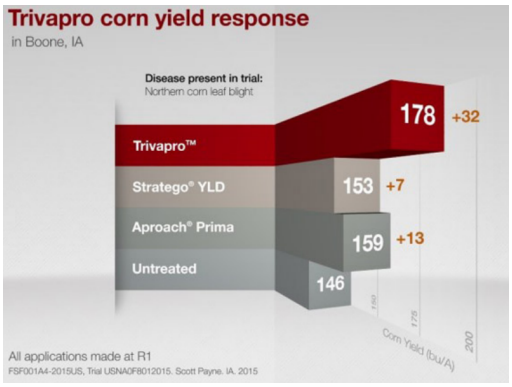
Multidimensional Projections: helps understand the data

- **PCA**: lineal algorithm. If we do a scatter plot we see the result altough the numbers are written completely different when we project them the result is thath many overlap, completely mixed togother.

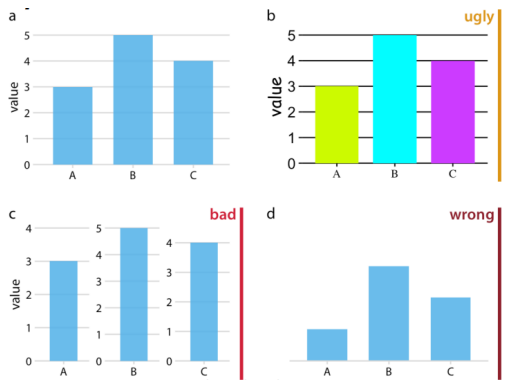
To avoid this problems (the big problem is using linearlity) we can use non linear reduction tecniques.

- **NonLinear Aproaches**: in order to create a projection in 2D they first build distance matrix in a high dimenisons, then the project the points in lower dimenions and do again the matrix of distances. The goeal is to make this 2 matrices of distances as similar as possible. All algotithms work the same, with differences in the distance functions used. The loss function will also be different. ALso different acceelerantion structures and inicialization.
- **T-SNE**: ... **perplexity** ... . The position of the clusters is different, the positon of the clusters do not have any meaning, and the distances between the clusters also makes no differences, closer clusters do not mean they are equal, not in T-SNE.
  - Not deterministic, same number of iterations will not gave the same result. Different runs may result in different executions.
  - If we apply tsne to a random noise dataset tsne may crette some similartiries that do not exist.
- **UMAP**: cluster size means nothing. The distances betwween cluesters might not mean anything but global position is manteined.
  - Hyper:
  - n neighbors:
- **PACMAP**:

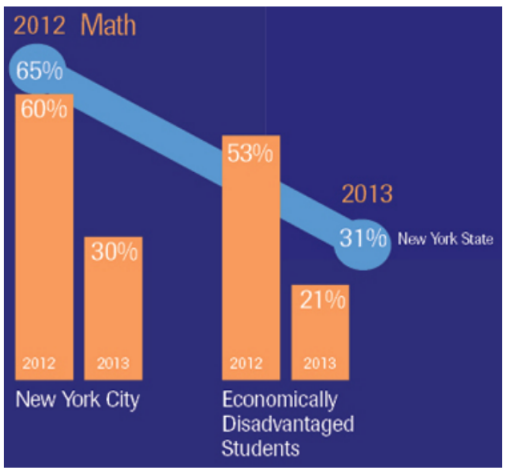
7 Analysis of visualizations



Corn yield response: **Use of 3D for no reason**, the information is so simple there is no need for complex plots. The **axis are not clear**, where is the 0? There is **redundant information**, the quantities are displayed on the axis, with numbers and also the numbers next to the plot is just so much information. The **color palette** is not the best one. Also the **title is not informative**, if we want to emphasize that *Trivapro* is the best treatment the title should be something like *Tivapro outperforms other treatments comparison* however, the title only says *Trivapro corn yield response*, when other treatments are also displayed in the plot.



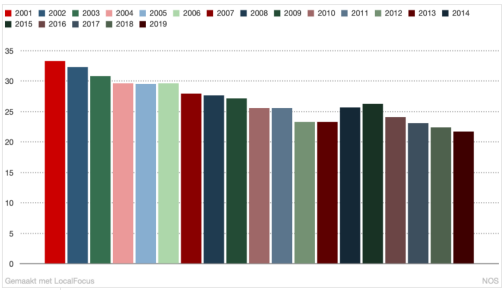
Same data different charts: Number C has three different scales, what it beats completely the purpose of visualization, so we cannot quickly understand what are this values. We need to read the labels and make a cognitive effort to understand and compare. Number B has very light colors, it is very saturated. The thing here is that colors are unnecessary because the bar charts are already separated in space. Unless we need to say something to the user with this colors this is unnecessary. The black grid lines are also so visible we need to concentrate on the data not on other things. Number D, what probably happens is that they cut the y-axis, what is completely wrong because bar charts encode quantity by the length. Therefore if we clip the, we are not encoding. This is the most dangerous.



Math Grades: The bad thing here is that the 31% is associated with 2013 because is close to that but 2013 is also orange so it could also refer to the bars. The line chart and the bar chart does not coincide on the layers for the axis. The bar chart has two names and the line chart has years. It is not even clear if both charts are together or meant to be separated.

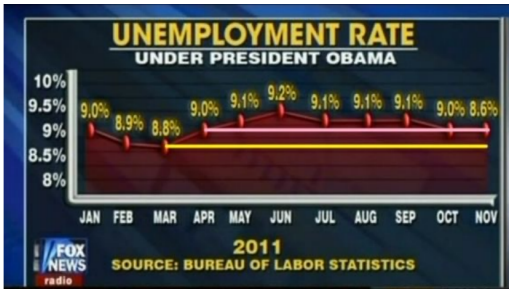
We need to know what to compare, here is complicated because we might be interested in compare NY city (then this should not be a bar chart it should be better a line chart with 2 points), if we want to compare NY city with Economic Disadvantaged is dubious. If we want to compare something and we have a reference we can use a bullet chart. WE do not know if it is a only plot or three separated plots.

Gene Comparison (22 - Good Practices) At the left there is no message. Is beautiful but we cannot understand anything. To improve it we can use that as context for out visualization, we use it at the background and we highlight some connections. And if we only want to explain the message because the context is not important we can make a simpler diagram that is much better to explain the message. Depending if we only want to transmit the message or also give context (inexpert public) will be different. For scientific purposes the last will be better.

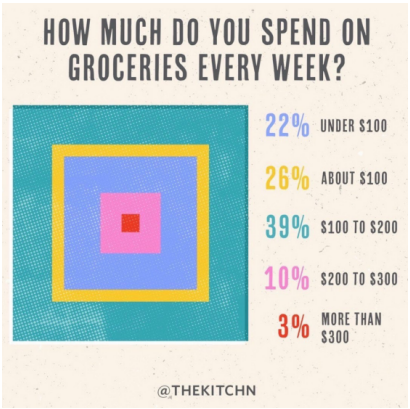


Percentage rokers in Neder: Labels of the years are not in the x-axis. Colors are not necessary because we are only showing one category and besides the color palette is "horrible". It shouldn't be a bar plot, but a line plot showing the continuous path with the years.

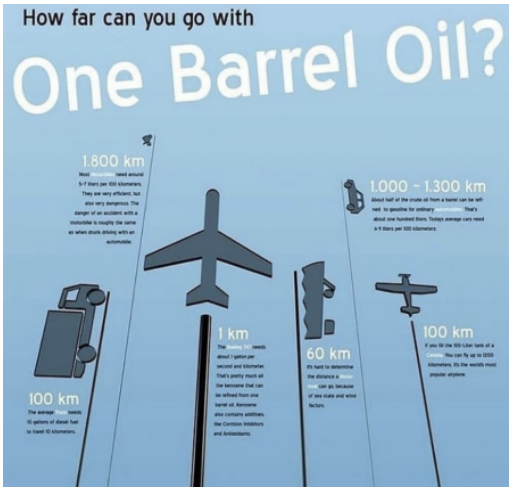
Arthur 51: The proportion of the bars is not proportional to the values, which suggest that the scale does not start at 0.



Unemployment rate: The y-axis does not start at 0 (since it is not a bar chart is not that important). The last values does not correspond to the percentages.



Groceries: It is difficult to compare since they are areas. The colors does not help and they do not directly correspond to the values.



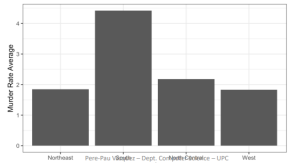
One barrel oil: The length of the bars dont correspond proportionaly. The width of the bars are different but iut doesnt specify what for.

- **We have a dataset with driving tickets. In some cases, we have missing values in the amount of the fine. Comment on the following two strategies to deal with missing values. Discuss strengths and weaknesses:**
  a. Erase the rows
  b. Encode the value with a particular number, such as a negative, e.g. -10000

Exercise:

- **a.** You know what you have is real information. If there are a lot of missing rows if we delete we may lose information not only in data but in context may be useful. Since we don't know the reason of the value missing we cannot calculate other variables like total count, frequency, date. So that 1 row has 1 missing key does not mean that it does not contain important data in it.
- **b.** We keep the information of the other variables, that may be relevant. Disadvantages are that if we use this values to calculate things we may end up with wrong conclusions.

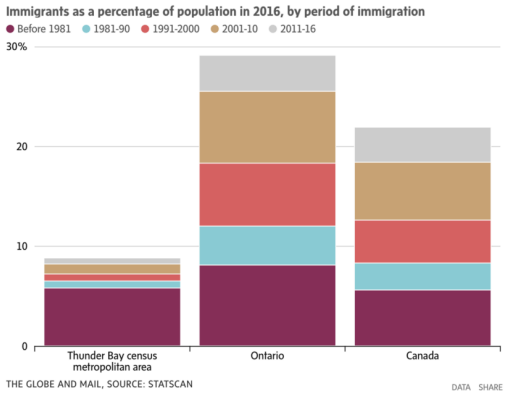
- Say we are interested in comparing gun homicide rates across regions of the US. We see this plot:
  - What region would you consider safer? Why? Discuss its design.



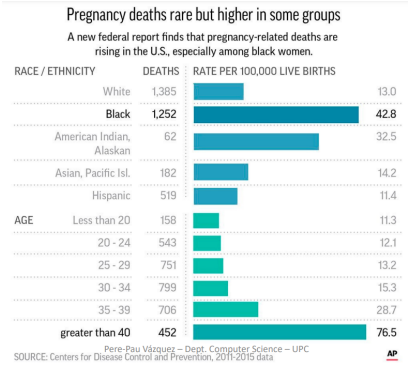
Representation Exercise: We cannot know because we don't know the population. The population should not be factored out.

- **We want to compare the evolution of the energy consumption at UPC with the price of the electricity along one year. Do you think a line plot with dual axis could work?**

Representation Exercise: Dual axis is intended to show 2 variables that are different like *energy consumption (watts)* and *price of electricity (euros)*. If we use a dual axis plot we first need to know that the variables are correlated and select correct ranges. We can use this plot to analyse the correlation.

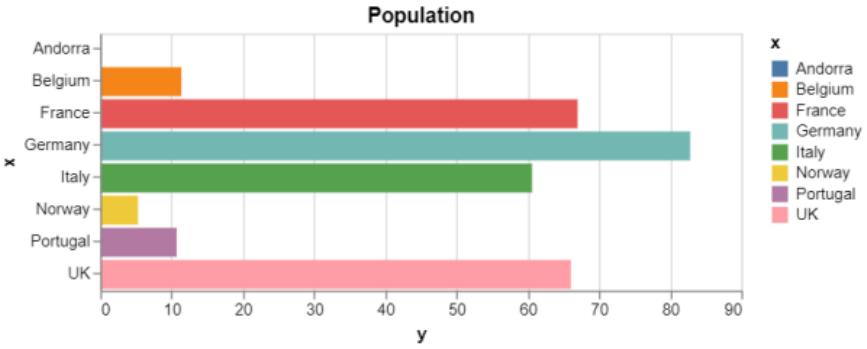


Representation Exercise: Time data is shown in different layers of the stack bars (would be better if time was on the x-axis). Comparing elements that are not aligned is difficult. The time period intervals are not equally distributed. *Thunder bay* is in **Ontairo** and *Ontairo* is in *Canada*. We don't know the population density of the three regions so it can be misleading. Maybe the colors are not adequate because some colors are similar.



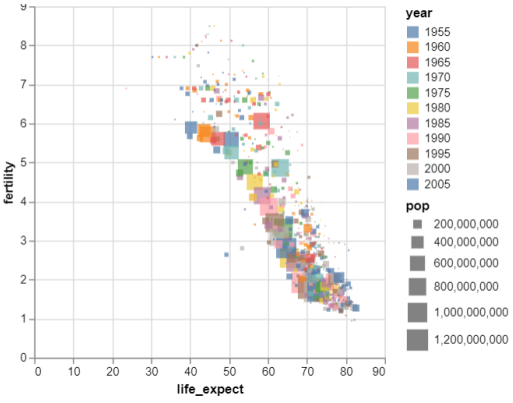
Representation Exercise: The differentiation of *race* and *age* is not evident. We could tend to group the two charts and we should make clear that they are separate. The

proportion of the intervals are not the same. Mis-match between the death quantities with the length of the bars, they both indicate different things but since they are closer together make ine think that they are related. We dont know the population of each *race* or *age* and that can lead to some wrong conclusion. Is the rate of birth the same for each *race* or *age*? The emphasis of the color of the bar is not necessary since the *negrita* of the text is enough.



Exercise 20: The legend is not needed, as well as the color, as they do not provide any extra information. The axis are inverted (bottom should be x), they could be denoted with a more informative label. Andorra is not shown due to the value, maybe a logarithmic scale could work instead.

```
import altair as alt
import pandas as pd
from altair import datum
from vega_datasets import data

df = pd.DataFrame(data.gapminder())

alt.Chart(df).mark_square().encode(
    x = 'life_expect:Q',
    y = 'fertility:Q',
    color = 'year:N',
    size = 'pop',
)
```



Exercise 22: It's a bubble chart but with squares where the position of the points depends on (life_expectancy, fertility). The color of the squares represents the year of the observation and the size the population.

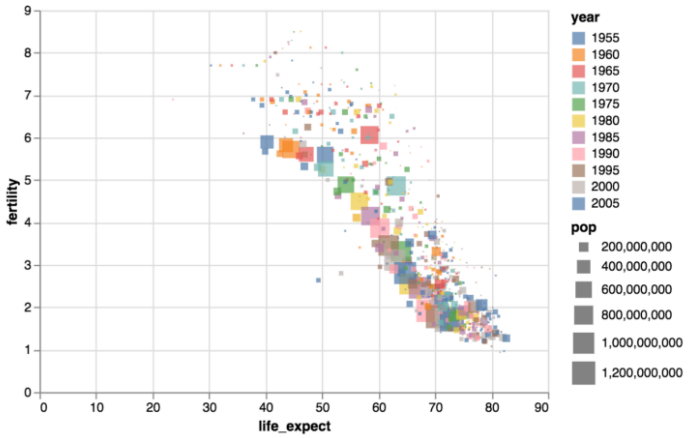
```
import altair as alt
import pandas as pd
from vega_datasets import data

df = data.cars.url

alt.Chart(df).mark_bar(fill = 'red', opacity = 0.0
    ).encode(
    x='Cylinders:O',
    y='count(Cylinders):Q'
)
```

Exercise 23: The chart displays how many cars have each number of cylinders (that's why is ordinal) with a bar chart with red bars. The opacity should be increased if we want the bars to actually appear.

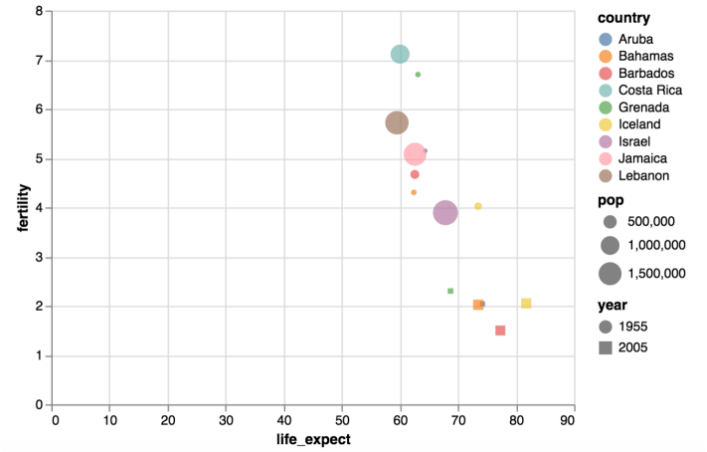
```
import altair as alt
import pandas as pd
from vega_datasets import data

df = data.stocks.url

alt.Chart(df).
    ).encode(


).transform_calculate(
    year='year(datum.date)'
    ).transform_filter(
                        )
```

Exercise 24:

```
alt.Chart(df).mark_area().encode(
x='stock:Q',
y='year:O',
color='company:N'
).transform_calculate(year='year(datum.date)'
).transform_filter(year >= 2006)
```



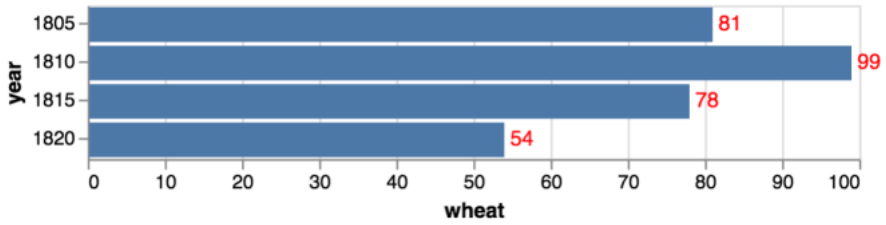
Exercise 37: Population is encoded with size and year with color. 1955 and 2005 have really similar colors and there is a lot of clutter. There is a lot of space not used, life expectancy may start at 30 so the data is better visualized.



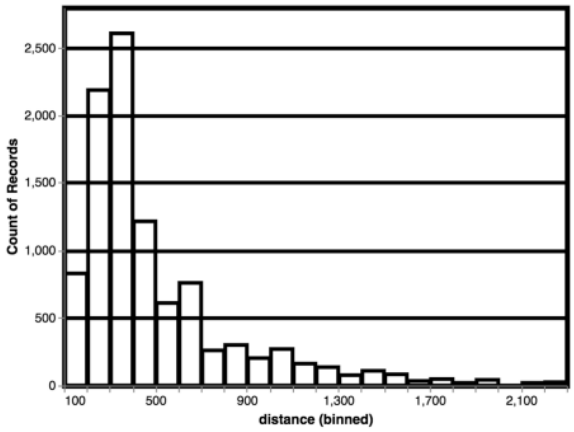
Exercise 38: This chart is expressing the difference in fertility and life expectancy of some countries within 50 years of difference. Again, so much space is not used. The encoding of the years is not appropriate, a better approach could be the use of a two dot plot or a slope plot. Furthermore, the encoding of the 'pop' may seem at first that it does not refer to the size of the squares as well, which makes the user dedicate more effort in depicting the size of the squares. The chart is not effective neither appropriate.

```
import altair as alt
from vega_datasets import data
import pandas as pd


df = data.seattle_weather()


alt.Chart(df).mark_line(color='crimson', clip=True).encode(
    x=alt.X('month(date):T'),
    y=alt.Y('average(temp_max):Q', scale = alt.Scale(domain=(10,20))),
).transform_calculate(
    year='year(datum.date)').transform_filter(alt.datum.year == 2014)
```

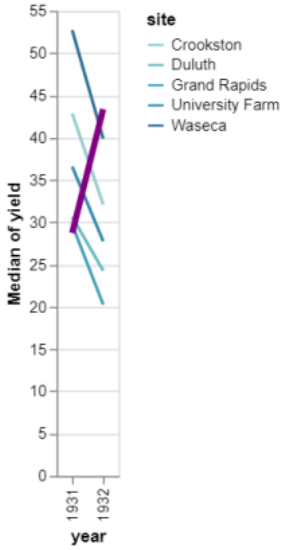
Exercise 39: If the temperature is outside the domain it will not be shown.



Exercise 40: The red numbers are redundant, they encode the same information as the x axis. If the number labels are needed, they should be encoded with a lighter color, like gray to not disturb the first impression. Year should be in the x axis and wheat in y axis. And year should be in ascending order. For encoding temporal data a bar chart is not correct, better use a line chart.

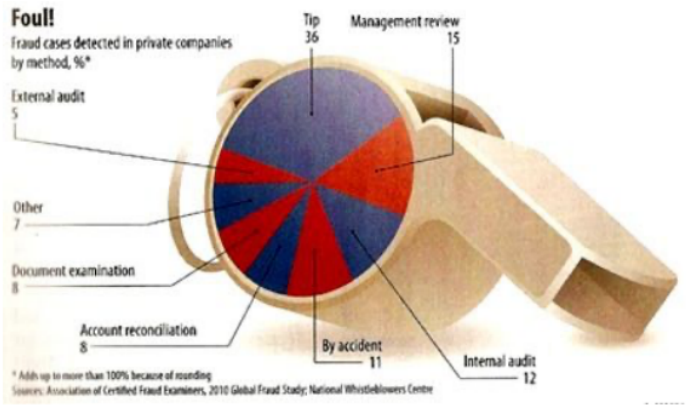


Exercise 41: The grid lines should as thick as the bar lines, if precision is important this lines should not be so visual, the color also should be light grey or something like that. Optional comment: *The grid lines should also not interfere with the bars, since they disrupt with the important data we want to show.*

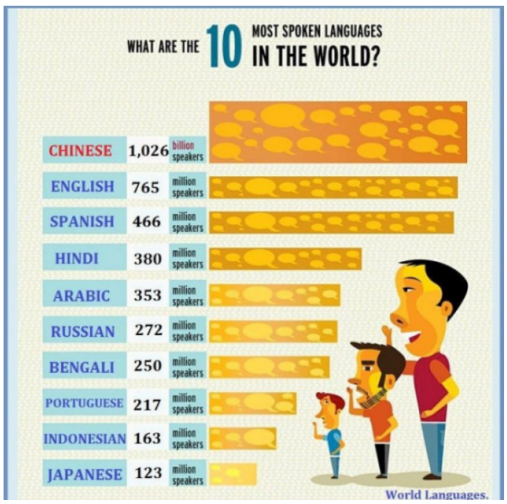


Exercise 42: It's so thin, we would need to do a wider plot. The color palette is not suitable. The purple line what the fock represents that.

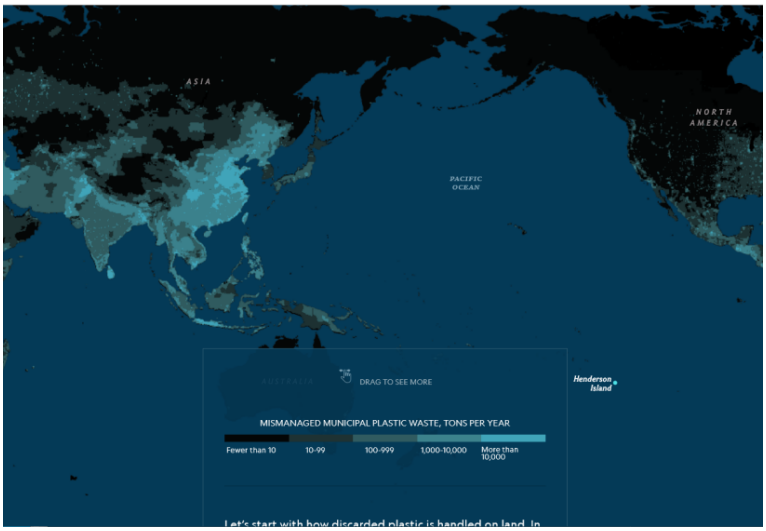
Exercise 43: The perspective of the pie chart can result confusing to the user. The parts that are closer seem to be bigger than the rest, since one could tend to include the extra space from underground of the chart. The proportion 4840 seems to be equally as big as the one with 7260. Even the figures on top of the pie, like the machinery hide some proportions. The figures can get our attention instead of the data we want to encode.



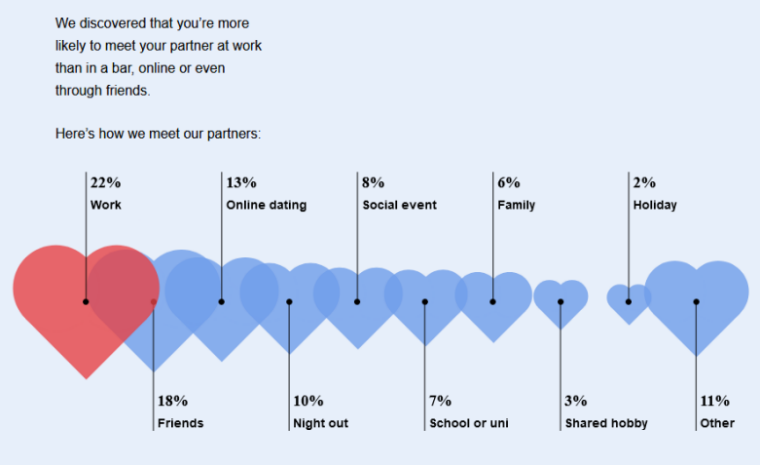
Exercise 44: The proportions are easily distinguishable but the colors could distract the user, since one could tend to group the different classes.



Exercise 45: The label in red of the Chinese speakers can result confusing since it is in billions and the rest of measurements are in millions. It would be more adequate representing all the measurements in millions. If the information is encoded via area it's obviously that is not right since for example Chinese is so much bigger. Also if is encoded in length is not proportional.

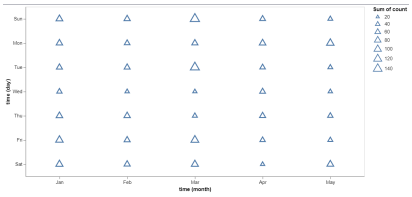


Exercise 46: The color palette is well selected but the colors should be inverted as black is usually encoded for bad things.



Exercise 47: Higher the dimensions more difficult to compare, maybe encode with length with bar charts, although having the percentage helps the user compare. A lot of lower space not used. The title should not explain the result, should explain what the visualization is representing.

```
import altair as alt
from vega_datasets import data

source = data.github.url

alt.Chart(source).mark_point(shape='triangle').encode(
    x='month(time):O',
    y='day(time):O',
    size='sum(count):Q'
)
```



Exercise 49: for each month we will have 7 points (one for each day of the week) in the y axis with a triangle representing the number of commits of all the years of that day with size.

```
import altair as alt
from vega_datasets import data

source = data.flights_10k.url

alt.Chart(source).mark_bar().encode(
    x=alt.X('delay:Q', bin=alt.Bin(step=20)),
    y='count():Q',
    color = 'destination:N'
)
```

Exercise 49: Stacked bars with the x axis the delayed binned (ordered quantitative) and the y axis the number of times each binned delayed occurred. Stacked bars with the color encoding which destination was the delay.
What are the difference between overview+detail and focus+context: Overview should not leave anything behind, is in general complete and focus+context is a way of exploring the data and the focus is intended to focus on the demanded zone.

In focus and context we only see the focus, the context serves as contextualization of where we are, so the user does not get lost.
What perception principle we get benefit from when using node-link diagrams: We can see the connectednes principle, which is really relevant. Even for object that are not the same if we have a connection between those it seems they are linked. VERY VERY STRONG.
Can these two plots come from the same dataset: The smoothness comes out from a different sampling frequency. This is the similar case of histograms with different bins.
What is preattentive processing: Detected previous to conscience. It can be helpful in visualization when we want to highlit something. If we ensure this in encoded for example with a different color of the whole representation. We can use this also to deemphisize other things.

If we want to encode categorical/ordinal/quantitative information, which visual variable would youuse? Why? Color is useful under certain circumstances (number of categories), but rather position is a better mark to encode categorical variables.

8 Altair Basics

```
import altair as alt
from vega_datasets import data

alt.Chart(data.cars()).mark_circle().encode(
    x='Horsepower:Q',
    y='Miles_per_Gallon:Q',
    color='Origin:N'
)
```

8.1 Data Types

- **(Q) Quantitative:** a continuous real-valued quantity
- **(O) Ordinal:** a discrete ordered quantity
- **(N) Nominal:** a discrete unordered category
- **(T) Temporal:** a time or date value
- **geojson:** a geographic shape

8.2 Marks (Altair Documentation)

- **mark_arc**: Represents a circular arc or sector in a visualization.
- **mark_area**: Fills the space between a line or lines and the axis, creating an area chart.
- **mark_bar**: Represents data using bars or columns, suitable for bar charts and histograms.
- **mark_boxplot**: Used for creating box plots to visualize the distribution of a dataset's values.
- **mark_circle**: Represents data points as circles in a scatter plot or other visualizations.
- **mark_errorband**: Represents uncertainty or variability in data as a shaded band.
- **mark_errorbar**: Displays error or confidence intervals for data points in a visualization.
- **mark_geoshape**: Used for creating geographic shapes or maps, often for visualizing spatial data.
- **mark_image**: Displays an image in a visualization, allowing you to integrate images with your data.
- **mark_line**: Represents data using lines, suitable for line charts and time series plots.
- **mark_point**: Represents data points as points in a scatter plot or other visualizations.
- **mark_rect**: Draws rectangles in a visualization, often used for heatmaps or grid-based charts.
- **mark_rule**: Draws horizontal or vertical lines, often for reference lines or gridlines.
- **mark_square**: Represents data points as squares in a scatter plot or other visualizations.
- **mark_text**: Allows you to add text labels or annotations to a visualization.
- **mark_tick**: Represents axis ticks or labels in a chart, helping with data annotation.
- **mark_trail**: Connects data points with lines, often used for visualizing sequences or trajectories.

8.3 Channel configuration (Altair Documentation)

- **Color:**
  - **color**: Default color of the mark
  - **fill**: Color that fills the mark (has higher precedence than color)
  - **fillOpacity**: Float indicating the opacity [0..1]
  - **filled**: Boolean indicating whether the mark is filled
  - **opacity**: Float indicating the overall opacity [0..1]
  - **strokeOpacity**: Float indicating the stroke opacity [0..1]
- **Shape and Position:**
  - **height**: Height of the marks
  - **shape**: For point marks, shape can be circle, square, cross, diamond, triangle up, triangle down, triangle right, or triangle left
  - **Other shapes**: arrow, wedge, triangle
  - **A custom SVG path**: Defined in a rectangle between -1 and 1
  - **size**: The size of the shape. For point, circle, and square, it will be the pixel area of the marks.
  - **x**: X coordinates of the marks, or width of horizontal bars (and area marks).
  - **y**: Y coordinates of the marks, or height of vertical bars (and area marks).
  - **x2**: X2 coordinates for ranged shapes (area, bar, rect, and rule)
  - **y2**: Y2 coordinates for ranged shapes (area, bar, rect, and rule)
  - **width**: Width of the marks.
- **Other Properties Referred to the Strokes:**
  - **stroke**: Default color for the stroke. It has higher precedence than default color (defined using config.color)
  - **strokeDash**: An array of alternating stroke and space lengths, for creating dashed or dotted lines, that may depend on the encoding.
  - **strokeWidth**: The width of the stroke, in pixels.
  - **thickness**: Thickness of the tick mark.
  - **tooltip**: Tooltip text to show upon mouse hover over the object.

8.4 Data Transformations (Altair Documentation)

Transformation	Function	Description
Aggregate	`transform_aggregate()`	Create a new data column by aggregating an existing column.
Bin	`transform_bin()`	Create a new data column by binning an existing column.
Calculate	`transform_calculate()`	Create a new data column using an arithmetic calculation on an existing column.
Density	`transform_density()`	Create a new data column with the kernel density estimate of the input.
Extent	`transform_extent()`	Find the extent of a field and store the result in a parameter.
Filter	`transform_filter()`	Select a subset of data based on a condition. `'year(datum.Year) > 1975'`
Flatten	`transform_flatten()`	Flatten array data into columns.
Fold	`transform_fold()`	Convert wide-form data into long-form data (opposite of pivot).
Impute	`transform_impute()`	Impute missing data.
Join Aggregate	`transform_joinaggregate()`	Aggregate transform joined to original data.
LOESS	`transform_loess()`	Create a new column with LOESS smoothing of data.
Lookup	`transform_lookup()`	One-sided join of two datasets based on a lookup key.
Pivot	`transform_pivot()`	Convert long-form data into wide-form data (opposite of fold).
Quantile	`transform_quantile()`	Compute empirical quantiles of a dataset.
Regression	`transform_regression()`	Fit a regression model to a dataset.
Sample	`transform_sample()`	Random sub-sample of the rows in the dataset.
Stack	`transform_stack()`	Compute stacked version of values.
TimeUnit	`transform_timeunit()`	Discretize/group a date by a time unit (day, month, year, etc.).
Window	`transform_window()`	Compute a windowed aggregation.

9 Altair Questions

1. Describe the difference between keys and values.

   Keys, also known as encoding channels, are the visual properties of a plot that define how data is mapped to the graphical elements.

   Values are the actual data that you want to represent in your visualization. They are the specific numerical or categorical values from your dataset that are mapped to the keys or encoding channels. Values are what determine the visual appearance of the elements in the plot.

   The encode() method builds a key-value mapping between encoding channels (such as x, y, color, shape, size, etc.) to fields in the dataset, accessed by field name.
2. How many keys and values has a typical bar chart?

   Key Encoding Channels: x: The horizontal position of the bars. This encoding specifies the categorical variable that determines the position of the bars along the x-axis.

   y: The vertical position of the bars. This encoding specifies the numerical or quantitative variable that determines the height of the bars along the y-axis.

   Value Encoding Channel: color (optional): The color encoding allows you to differentiate bars based on a categorical variable. This is often used to represent different categories of data with different colors.
3. Can you encode multiple keys in a bar chart?

   Yes, by adapting a normal bar chart into a stacked bar chart we can encode multiple ...
4. What are marks and visual variables in altair graph?

   Marks in Altair represent the basic geometric shapes or graphical elements used to visualize data. Altair supports a variety of mark types, such as points, lines, bars, area, text, and more. Marks are specified in Altair using the mark_*() methods.

   Visual variables in Altair are the aesthetic attributes that you can use to encode data onto the marks. They allow you to map data to properties like position, color, size, shape, opacity, and more.