Unit -3

Absolutely, here's the classification of visualization systems with proper images for each category:

Based on Purpose:

- **Exploratory Visualization Systems:** These systems are designed for users to play around and discover insights from data. Imagine sifting through sand for seashells - you have to explore to find the treasures!

Opens in a new window www.ibm.com

Person using a data exploration software on a computer screen

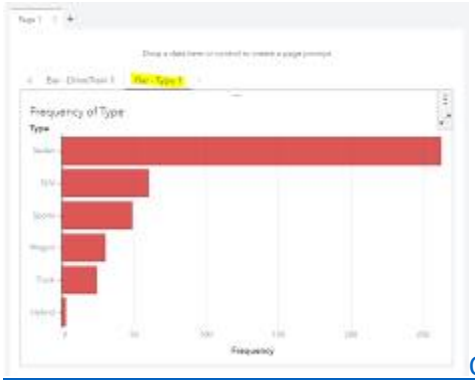
- **Explanatory Visualization Systems:** These systems tell a predefined story with data, like a tour guide showing you specific landmarks.

Opens in a new window slidemodel.com

presentation slide with a clear chart and text explaining the data

Based on Functionality:

- **Static Visualization Systems:** Imagine a printed photograph - it's a fixed image you can't change. Static visualizations are similar.



Opens in a new window support.sas.com

bar chart in a printed report

- **Interactive Visualization Systems:** These systems allow you to zoom in and out of the data, like using a magnifying glass to see details.

Based on Technology Stack:

- **Desktop-Based Visualization Systems:** Imagine a powerful workstation for video editing. Desktop visualization systems are similar, offering advanced features on a local computer.


Opens in a new window www.fusioncharts.com

powerful desktop computer with data visualization software open on the screen

- **Web-Based Visualization Systems:** Accessible from any device with a web browser, like using Google Docs online. Web-based visualization systems offer easy access and collaboration.

Based on Interactivity:

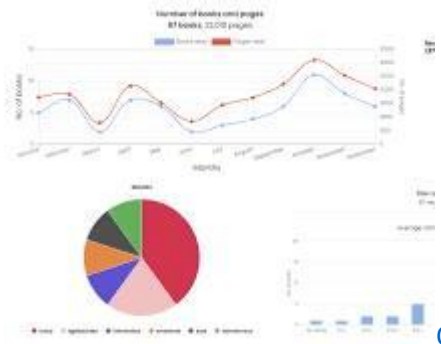
- **Low-Interactivity Visualization Systems:** Imagine a static image in a museum exhibit. Low-interaction visualizations are similar, offering limited zooming or panning.
- **High-Interactivity Visualization Systems:** High-interaction visualizations are like an immersive art installation you can walk around and explore. They allow filtering, sorting, and detailed information pop-ups.

Opens in a new window www.zdnet.com

person wearing a VR headset and interacting with a 3D data visualization

Based on User Interface Design:

- **Traditional Visualization Systems:** Imagine familiar bar graphs and pie charts in a textbook. Traditional visualization systems use these well-understood chart types.



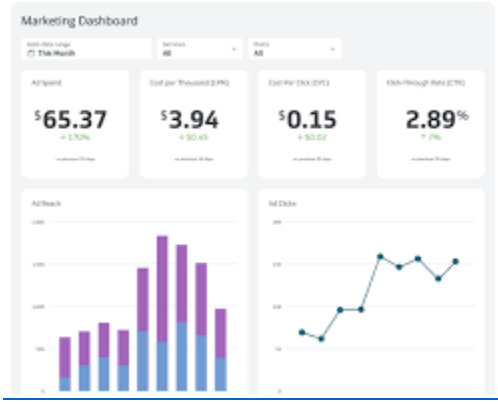
Opens in a new window www.thestorygraph.com

textbook page with a line chart and pie chart

- **Advanced Visualization Systems:** Advanced visualizations are like modern art museums, using creative designs and new technologies. They may use 3D graphics, virtual reality, or augmented reality.

Additional Categories:

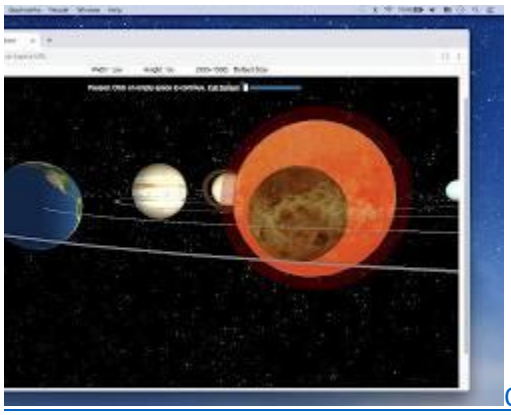
- **Business Intelligence (BI) Visualization Systems:** Imagine a business dashboard showing sales figures. BI systems help analyze business data for decision-making.



Opens in a new window www.klipfolio.com

business dashboard with charts and graphs showing sales figures and marketing performance

- **Scientific Visualization Systems:** Imagine a simulated model of the solar system. Scientific visualizations help understand complex scientific data.


Opens in a new window chrome.google.com

3D simulation of the solar system on a computer screen

- **Healthcare Visualization Systems:** Imagine an MRI scan image. Healthcare visualizations help diagnose and treat medical conditions.


Opens in a new window pngtree.com

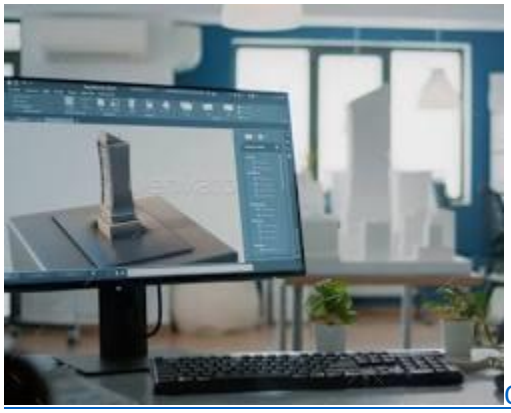
MRI scan image of the brain on a medical display screen

- **2D Visualization Systems:** Imagine a map on a piece of paper. Most data visualizations are 2D, using charts, graphs, and maps.


Opens in a new window interactivegeomaps.com

world map with countries highlighted in different colors

- **3D Visualization Systems:** Imagine a 3D model of a building. 3D visualizations help represent spatial data and complex structures.

 Opens in a new window photodune.net

3D architectural model of a building on a computer screen

I hope this classification with images is helpful!

Remove Your Rose Tinted Glasses: Data Visualizations Designed to Mislead

By Bernardita Calzon in Data Visualization, Jun 8th 2022

Table of Contents

Nobody likes feeling manipulated in any way, shape, or form. But while that may be the case, people are duped by data visualizations every day.

From political issues to sports statistics and the recent report you received on the ROI of your company blog, the internet as well as informational reports are flooded with examples of misleading data visualization.

Bad data visualizations come in many forms, with some more obvious than others. But, by knowing what to look for, you can avoid connecting with metrics that will lead your organization down the wrong path.

The best way to safeguard from misinformation is to arm yourself with tech-appropriate analytical online data visualization tools and evaluative skills that will expose the most oversimplified or malicious data visualizations.

Here, we will present a mix of the most common visual data misrepresentations together with practical tips on how not to fail when presenting data.

Learn how to spot the common tricks used to manipulate data and how to avoid the pitfalls for your own visuals.

Let's get started.

Your Chance: Want to create your own data visualizations for free?
Explore our 14-day free trial & benefit from great visualizations today!

Misleading Data Visualization Examples

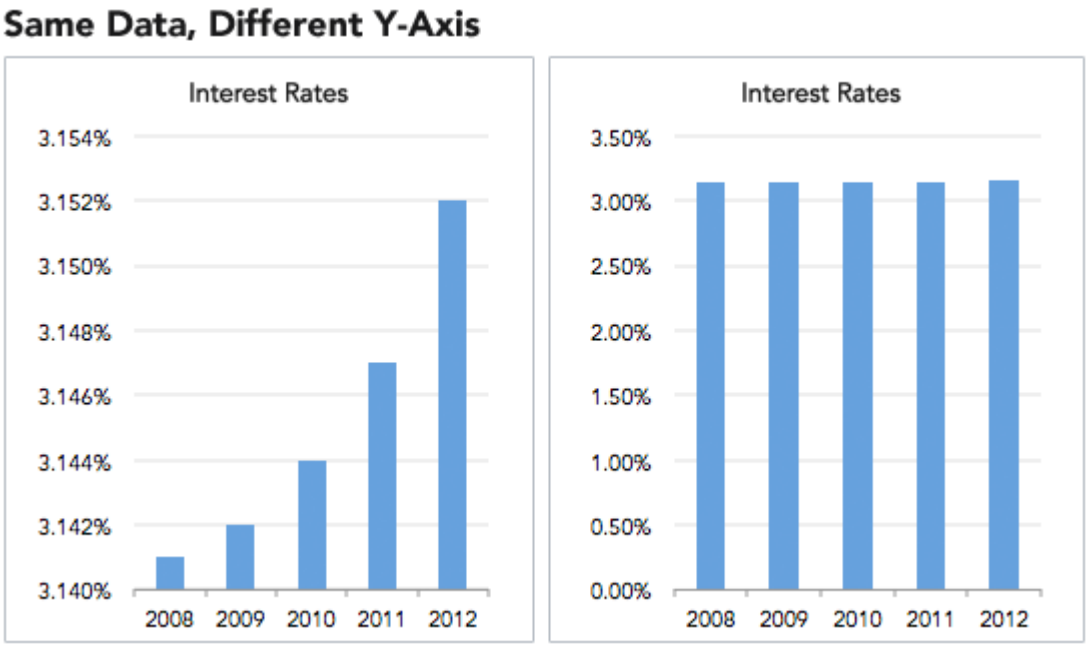
To start our journey, we're going to look at the digital world's biggest misleading data visualization real-life examples.

Each of these unethical data visualization examples has the potential to derail your strategic efforts and take you down an informational dead end. Take heed of these poorly arranged visuals and you will know exactly what to look for when analyzing your business's most invaluable data.

1. Truncated Y-axis

Infamous for its overuse in politics, the truncated y-axis is a classic way to visually mislead. Take a look at the graph above, comparing people with jobs to people on welfare. At first glance, the visual dynamics of the graph suggest people on welfare to number four times as many as people with jobs. Numbers don't lie, however, and when analyzed, they point out much less sensational facts than the data visualization would suggest.

This type of misinformation occurs when the graph's producers ignore convention and manipulate the y-axis. The conventional way of organizing the y-axis is to start at 0 and then go up to the highest data point in your set. By not setting the origin of the y-axis at zero, small differences become hyperbolic and therefore play more on people's prejudices rather than their rationality. Notice on the graph below, originally shared by Gizmodo, how much larger the differences look when truncating the y-axis.

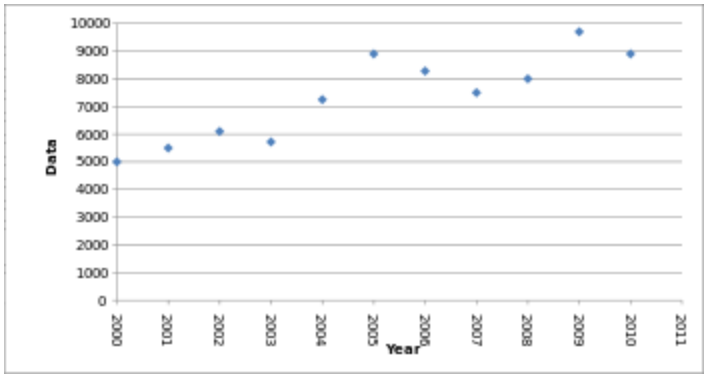


Focus on creating your data visualizations using data with a zero-baseline y-axis and watch out for truncated axes. Sometimes these distortions are done on purpose to mislead readers, but other times they're just the consequence of not knowing how an unintentional use of a non-zero-baseline can skew data.

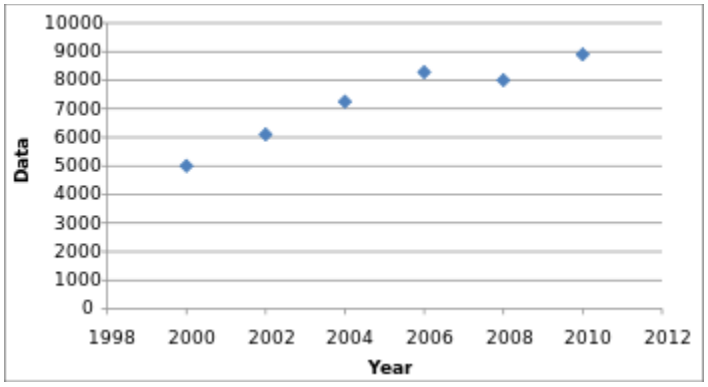
2. Omitting Data

Why lie when you can just omit? By omitting certain data points, trends that don't actually exist can easily be created whereas some existing highlights can go unnoticed. That's because by omitting some data we are missing the context. Leaving out variables can affect how you interpret the data and what conclusions you draw from it. So whenever you're examining a variable and its relationships, carefully consider the context in which that variable exists and deliberately seek out other variables that could affect the one you're studying.

As an example of what happens when you omit some data, be that because you purposefully want to create a misleading data visualization or you simply want to make your work easier, take a look at the scatter plot below. By leaving out some data points, the chart that normally would be filled with dips and spikes looks much smoother and more stable. See these graphs originally published by Cogent Legal.



Vs



By only plotting every second year instead of every year, the graph appears to have a steady increase, while the real data is more volatile. Companies can take advantage of this by omitting years with significant changes in sales to model their earnings to look constant and predictable, masking the true volatility of the market. When evaluating data visualizations be sure to have all the data accessible.

3. Correlating Causation

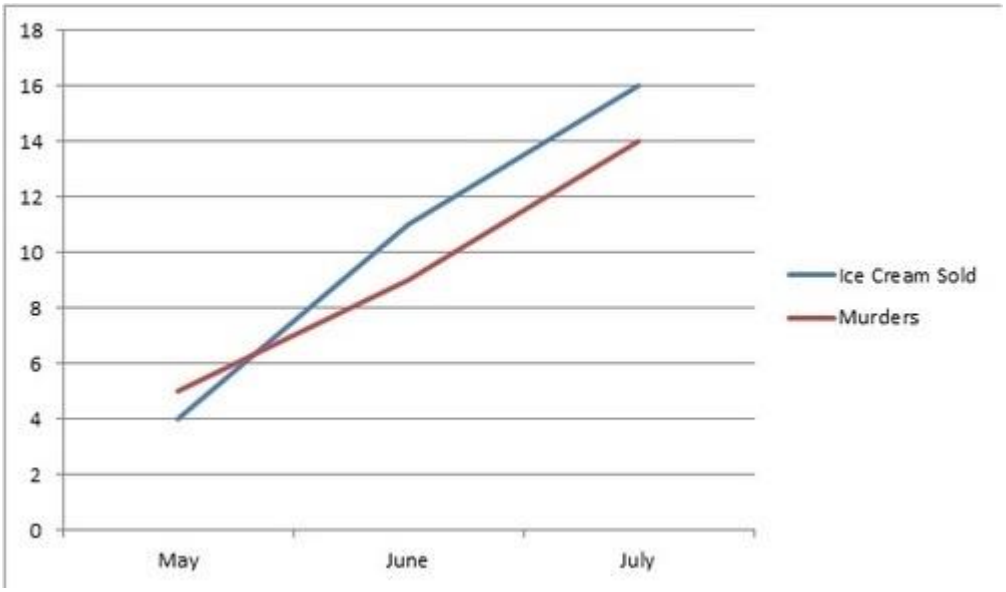
Any high school math class should have covered that correlation doesn't imply causation. But looking at the headlines of the most popular internet articles ("Does X cause Y?") and political statements it is easy to forget. Essentially, the correlation and causation scenario is the assumption that because two variables changed at the same time, one caused the other. As seen in our example below, that shows a relation between a rise in ice cream sales and murders. We are beginning to see correlating causation more and more with big data analyses. Data scientists are finding statistical patterns in data and sometimes care more about correlation rather than causation since figuring out correlation is simply easier.

That said, there are mainly two reasons why correlation doesn't always mean causation. The first one is what scientists call a "confounding variable" or third variable. Meaning an "invisible" or non-identified variable that can be affecting the two other variables and make them seem causally related. The problem with this, is that the third variable might never be discovered, hence, causation can't be confirmed.

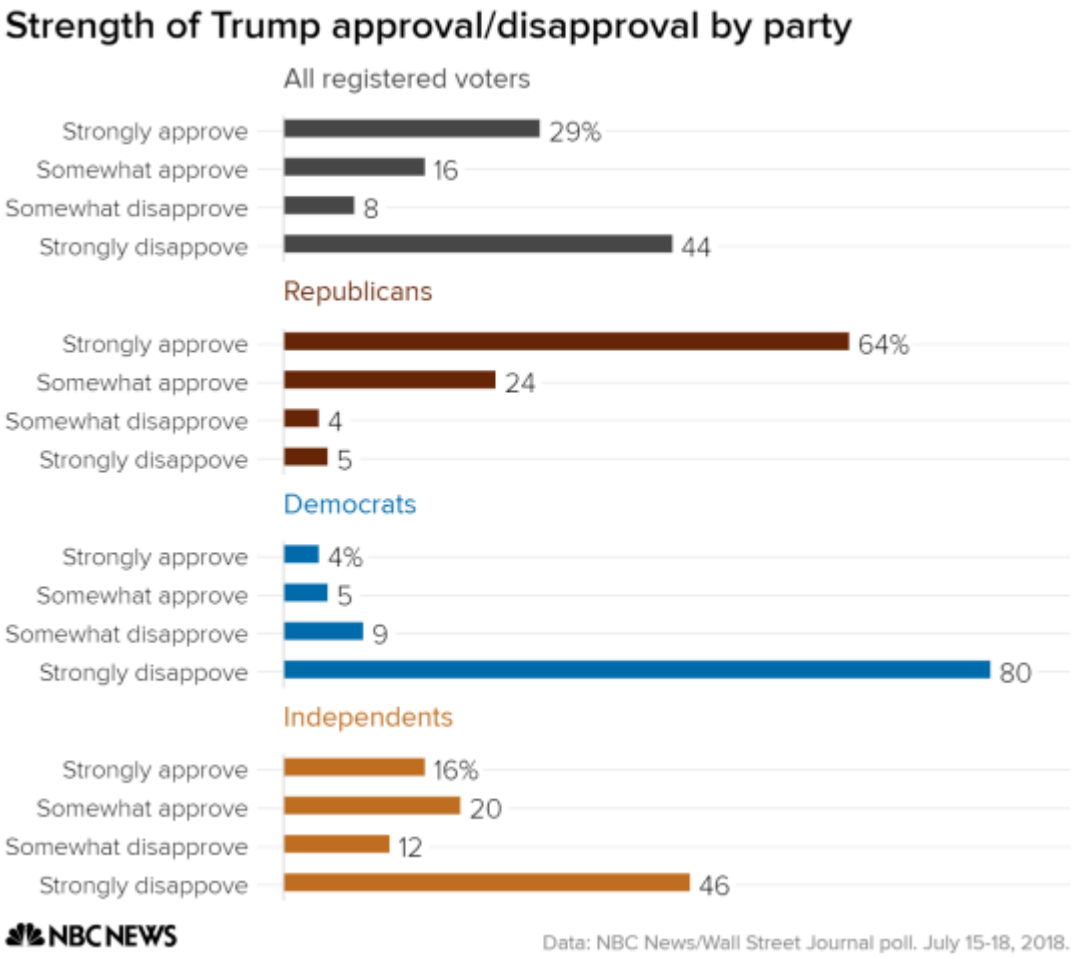
Another scenario in which correlation and causation are difficult to prove is with directionality. This means when two variables might be caused by each other but can't prove which one caused the other. Purposely or not, correlation and causation issues happen mainly in the media, advertisement,

and politics, it is very unlikely to find such situations in scientific papers where analysts are already aware of these threats.

Here's our favorite correlating causation data visualization, but for more check out Buzzfeed's: [The 10 Most Bizarre Correlations](#).



4. Cherry picking



Source

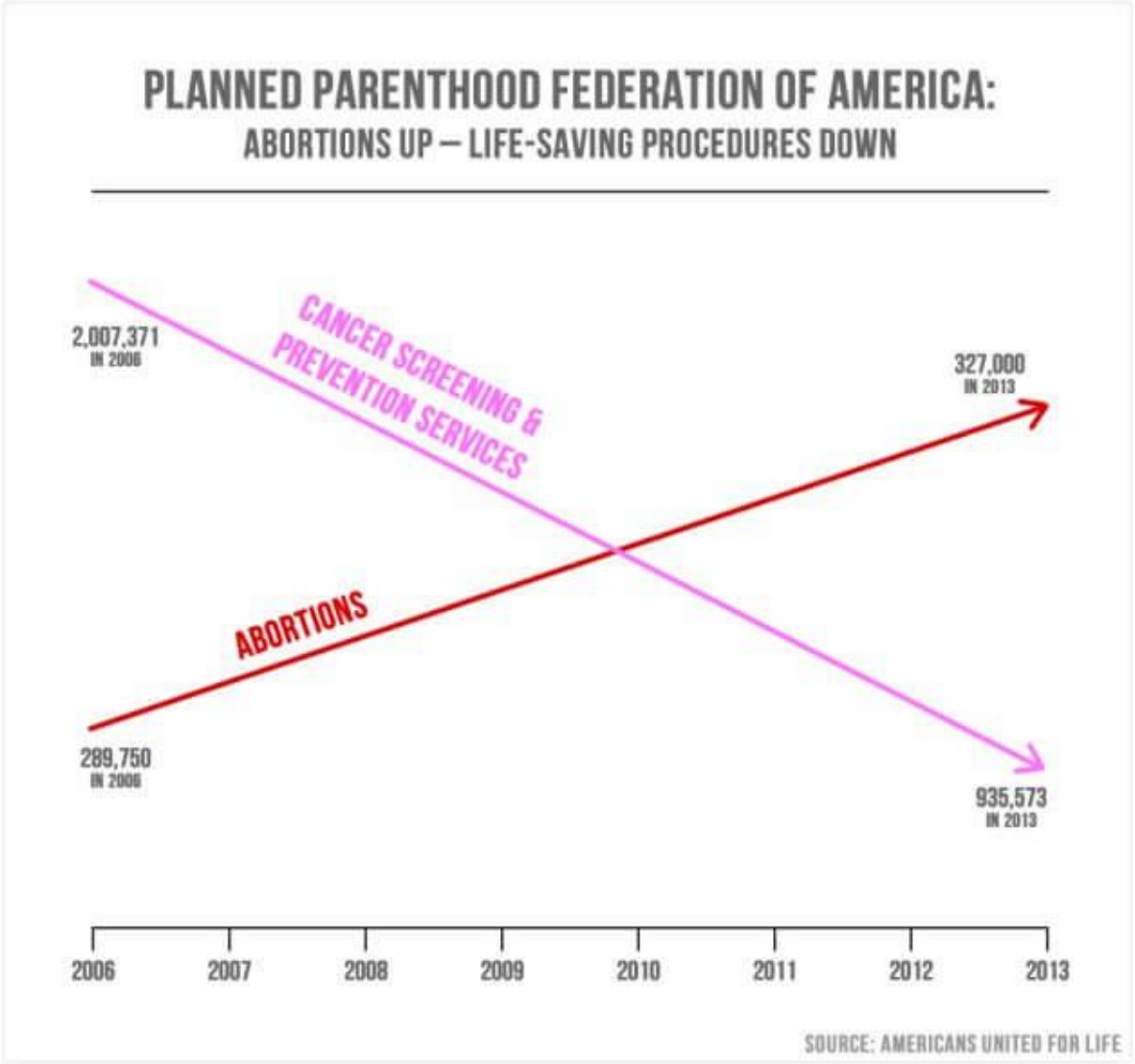
Cherry-picking is a form of data-driven deception where certain sets or sources or information are omitted from a survey, study, chart, or graph. The primary reason for cherry-picked visuals is an aim to offer clean, predictable results that fit into a neat trend, pattern, or box. The issue with cherry-picking is that it doesn't paint an honest, objective picture, offering results that are inaccurate or missing out on vital segments of knowledge.

This visual is particularly deceiving as the lines are exaggerated. In addition to this, while the results appear to be expressed as a percentage, not every result totals 100. As such, this visual is disproportionate and doesn't offer an accurate representation of the data at hand.

One small piece of information can completely change the perception, and ultimately the outcome derived from a KPI or metric, which means that cherry-picked visuals not only offer little value - but will serve to make a decision that harms your business.

Remember: The data you analyze from digital charts and graphs should be completely objective. If your information comes from cherry-picked sources, you will never drive the organization forward.

5. Dualing data



Source

While dual-axis charts often prove incredibly effective at comparing two related informational sets that don't share a unified scale when under pressure, they can become misleading visuals when specific markers aren't set correctly or missing entirely.

In the above example from the 2015 Congressional Hearing based on Planned Parenthood, we can see clear-cut results showing a rise in abortions and a drop in cancer-related health services. This visual is deceiving as it doesn't contain any values within its axis', merely showing a loose trend or pattern without any real context. As such, the true meaning of the data is skewed, making it unbalanced and potentially harmful. When you use dualing data without cleared marked values, they're essentially meaningless and can be manipulated to display a result that is not representative of the truth - avoid this mistake at all costs.

6. Misleading pie chart

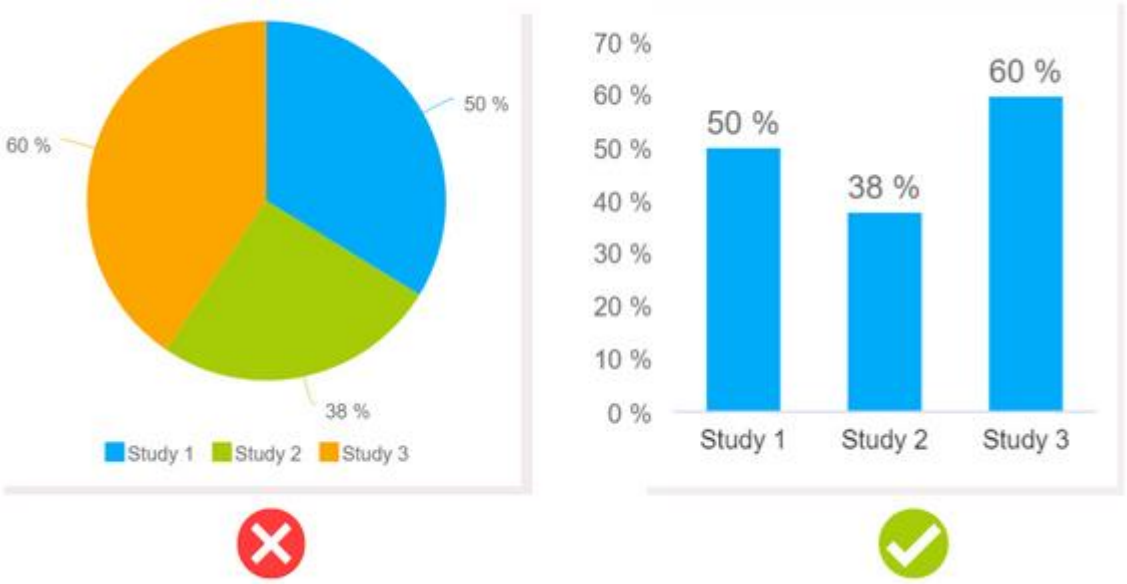
Source

When it comes to bad data visualization examples, misleading pie charts are without doubt one of the most common. Pie charts by their very nature are proportional and as such, show values that typically amount to 100% (or the entire segment of pie). But, when you're presented with a mix of choices or variables where a survey respondent can choose more than one answer, the results can appear confusing or deceiving.

The above example accurately displays results amounting to 100%. But, in a case where 100 survey respondents, for instance, are able to choose more than one of the geographical regions based on a specific question ('where would you relocate to for economic gain?' for example), the pie chart instantly becomes inaccurate because many respondents may have picked both North America and

Australasia. In that case, you're not seeing things as they really are - and to avoid this level of inaccuracy, using a Venn diagram would be a better choice.

7. Using the wrong chart type



A misleading data visualization issue that plagues business intelligence (BI) across industries is making the wrong chart choice.

As you can see from the example above, from these charts that were generated with datapine's online data analysis tool, the pie chart is a poor choice as the visuals don't represent the numbers in their best or most accurate format. Typically, a pie chart should amount to a whole (100 expressed as a percentage) - but that's not the case in this instance. As you can see, a neat, balanced bar chart is the best choice as it presents the information in its truest, most digestible form.

Rather than cases of people manipulating data for personal gain, selecting the wrong chart type is typically down to human error or a lack of analytical experience. While your metrics and insights may be 100% trustworthy and accurate, making a poor visual choice can either dilute your results or offer outcomes that can lead you to draw the wrong conclusions.

Another way poor chart or graph choices can have a negative impact on organizational outcomes is because in these cases, the results sometimes fail to make sense at all. As a result, time becomes wasted, resources become sapped, and your data becomes redundant. Before you make your selections, be sure you carry out ample research and ensure your chart will represent your information in the most optimal way possible.

These are a few of the common ways in which visuals can mislead, this goes hand in hand with the misuse of data which is also a common practice in the media, politics, and other industries. If you want to go deeper into the topic, take a look at our misleading statistics blog post.

Your Chance: Want to create your own data visualizations for free?
Explore our 14-day free trial & benefit from great visualizations today!

How to Avoid The Pitfalls of Misleading Data

If you want your data to tell the whole truth and nothing but the truth, implement these practices to avoid generating misleading data visualizations.

1. Follow Convention

By using the standard model for visual models, you can avoid misleading your reader. If your high school math teacher would mark you down on an exam for your methods of data representation, think twice.

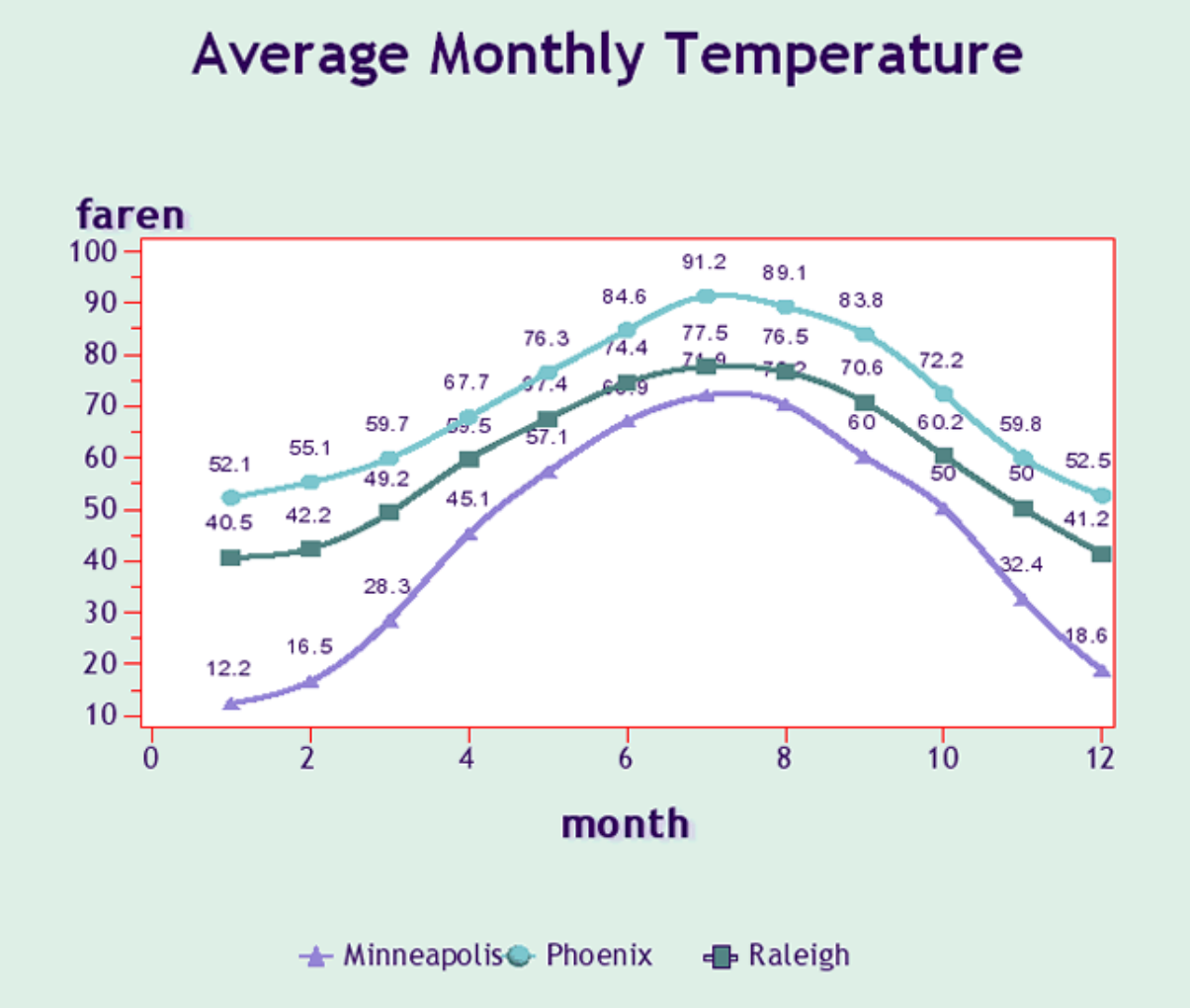
Start your y-axis at 0 to avoid making small differences look large. The exception to this practice would be if these small differences actually mean something significant. In global climate change data, often a global average temperature increase of 1 degree per year can have dire consequences. Thus a small increase in temperature is important and needs to be highlighted. However, when looking at something where small changes don't correlate with a big impact, start your y-axis at 0.

Following convention also goes for pie charts as well. People are conditioned to look at pie charts as equaling 100% of the data; don't mislead people by only giving them a slice of the pie's data.

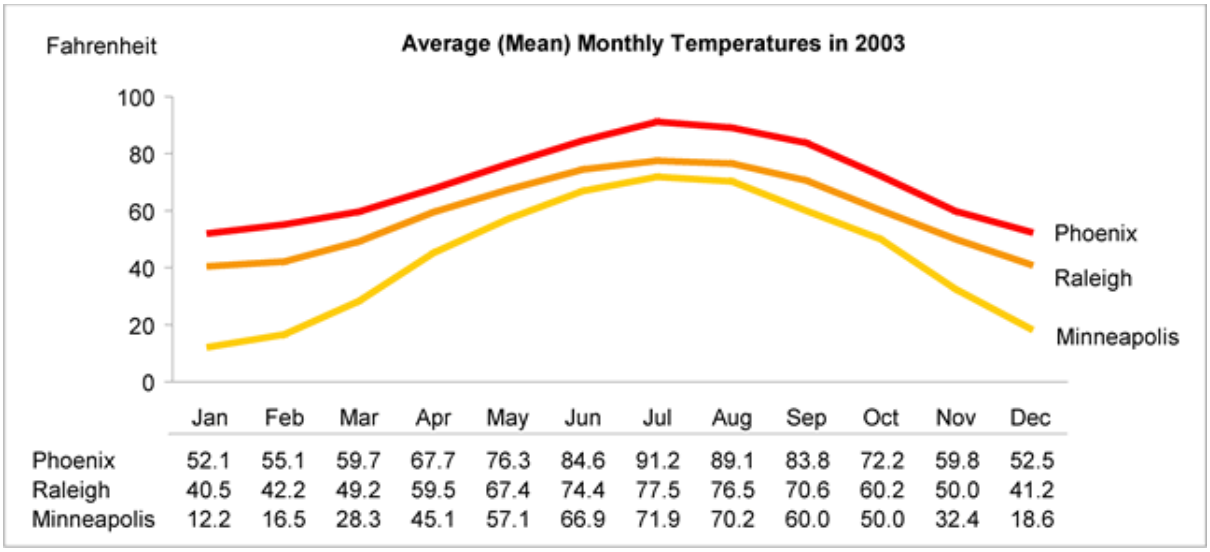
2. Make Your Data Visualization Clear and Easy to Understand

This one should be a no-brainer, but it is often neglected in a world inundated with flashy and intricate graphs. Below are two graphs mapping out the same data. As you can see, the second graph is much easier to identify trends and more aesthetically pleasing. Don't let sloppy visuals negate the credibility of your hard-earned data.

Unclear data visualization



Improved Data Visualization



Visualization guru Edward Tufte explains, "excellence in statistical graphics consists of complex ideas communicated with clarity, precision, and efficiency". Turn this into your mantra every time you sit down to create data visualizations.

3. Give up on PowerPoint

PowerPoint is a tool of the past. Now dashboards are in. The new dashboard software offers works with real-time data and allows you to turn numbers into visuals with only a few clicks. On the other hand, in PowerPoint performing SQL queries, and exporting the results to create charts before importing them onto slides can take a long time. With extra bells and whistles – colorful headlines, text, images – the process can take even more time. With easy-to-use SQL query builders, a drag and drop interface, and a metric builder, data visualization tools can produce results in mere minutes. The less time spent on creating data visualizations, the more time you have to understand the data.

4. Make smarter color choices

When examining misleading visualization examples, people often gloss over the more intricate design particulars. But, when it comes to representing visually-based metrics in an accurate, reliable, and digestible format, making smart color choices is essential.

To ensure you use colors that don't detract from the core information and represent your discoveries logically, here are some best practices to consider:

- Avoid using too many colors as this will confuse the viewer and make your message difficult to absorb
- Don't use colors that are too similar and offer little contrast as this could make your results appear misleading or less definitive
- Try not to use tones or shades that are likely to confuse people that are colorblind

5. Choose the right chart type

Earlier, when we examined examples of misleading data visualizations, we talked about the negative impact of choosing the wrong chart type. To avoid such an informational calamity, you should decide whether your insights are quantitative or qualitative - doing so will help you make an informed visual decision.

If you're working with qualitative information, you should (that describes or categorizes), you should opt for pie charts, Venn diagrams, or pie charts and if your findings are quantitative (insights that are directly measurable), line or trend charts or histograms are usually better suited. Really dig into your metrics as well as what they mean in a real-world context and you will land on the right chart type. BI dashboard software such as datapine, provides multiple charting options and templates to help you make the right decisions when it comes to visuals.

6. Don't present too much data

Another significant factor in misleading charts, graphs, or KPIs is cramming too much information into a small space. When you present your insights visually, the core idea is to create a narrative and tell a story. When you add too many sources or variables into the mix, you will make your results look messy, diluting the information in the process.

Before you set about collecting and presenting your insights, it's vital to think about data storytelling and choose only the variables that are directly relevant to your objectives. If you find yourself with too many variables or too much information, be ruthless. Go back to the drawing board and trim the informational fat until you have data that is clear, concise, and you know will result in a quality, accurate, and impactful visualization.

7. Think for Yourself, Question Authority

With your new skills of sniffing out faulty data and misleading information, make sure to review each data visualization with a skeptical eye before you present it.

More importantly, don't let your data become infected with these cheap tricks. Create a beneficial dashboard culture in your company to be sure that every piece of data and every visualization has been scrutinized before it goes public.

The Impact Of Bad Data Visualizations

Now that we've explored common instances of misleading visually-based metrics and considered how to avoid making these common mistakes, we're going to look at the impact of working with unethical information or making poor choices.

Fragmented narratives

Informational storytelling is one of the biggest components of a successful BI strategy. One of the biggest mistakes people make when working with visual information is working with an image, or images that only paint fragments of an entire picture.

When this happens, key insights and discoveries are missed. In addition to the data becoming less persuasive, it breaks up a narrative into illogical parts which, in turn, will make your BI efforts redundant. When this happens, you will suffer from a consistently poor return on investment (ROI) while creating inefficiency throughout the organization.

Poor decision-making

Whether you're talking about the wrong scale, poor chart choices, or bad design, working with misleading metrics often results in poor decision-making.

When your insights aren't telling the truth, you will make ill-informed decisions that will essentially:

- Drain your time and budget
- Lead to ineffective sales and marketing campaigns
- Put a strain on your internal processes
- Drive down staff engagement and loyalty
- Stunt your organizational growth

Operational and financial inefficiency

Working with insights that are inaccurate or misleading will also result in poor operational as well as financial efficiency.

Without concrete visual insights that tell you exactly what is working and what isn't, it's unlikely that you will ever get to the heart of the issues that sap your budgets or drive down productivity. As a result, your operational, as well as fiscal processes, will be poor and you will fail to reach your organizational key goals or milestones.

Key Takeaways On Misleading Visuals

We've explored the most misleading data visualization examples and offered practical advice on how to achieve consistent informational enlightenment in the digital age. Now it's over to you.

To ensure you connect with the best visual metrics for your business and enjoy continual organizational growth in a cutthroat commercial world, start your free 14-day datapine trial. We will help you level up the company's analytical strategy free from the perils of misleading visualizations.

- **NumPy:** For handling mathematical operations without any hassle.
- **Matplotlib:** Library will help implement the **CDF plot** (we will discuss more on this).
- **Seaborn:** A Data visualization library built on top of matplotlib; we will use this one more widely in this article.
- **Pandas:** DataFrame manipulation library, we will use it for creating **DataFrame**.

```
d1 = np.loadtxt("example_1.txt")
d2 = np.loadtxt("example_2.txt")
print(d1.shape, d2.shape)
```

Output:

```
(500,) (500,)
```

Inference: Reading two text-based datasets from **the load txt** function of **NumPy** and looking at the output, one can point out that there are **500 data points** in each dataset.

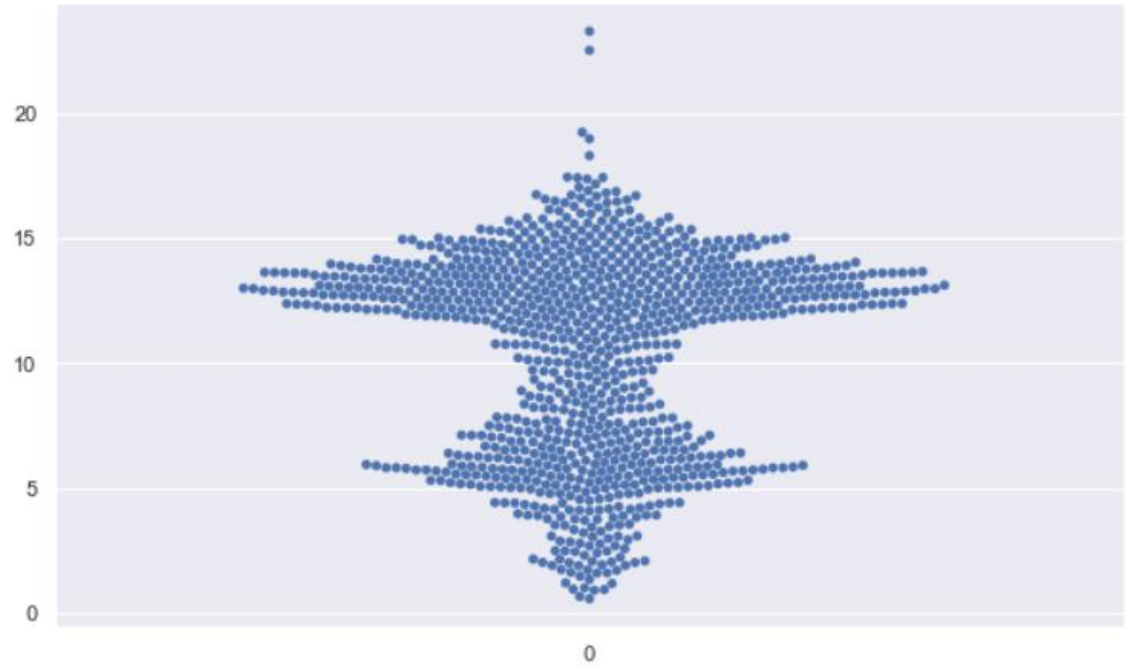
Bee Swarm Plots in Data Visualization

Starting with the **Bee Swarm Plots**, The best thing about this plot is that along with showing the **distribution of the dataset,** we can also **see each data point** by managing the **size** of those data points; make sure you provide a suitable size; otherwise, for bigger data point distribution will tend to get out from canvas while for the relatively **smaller size** of data

Inference: Swarm plots are like the scatter plots (which show the complete data representation) but with **categorical data** in the axis, which differentiate them from the general **scatter plots**. Hence, for that reason, we are first creating **DataFrame,** which has some values and its categorical type.

```
sb.swarmplot(data = dataset["value"], size=5);
```

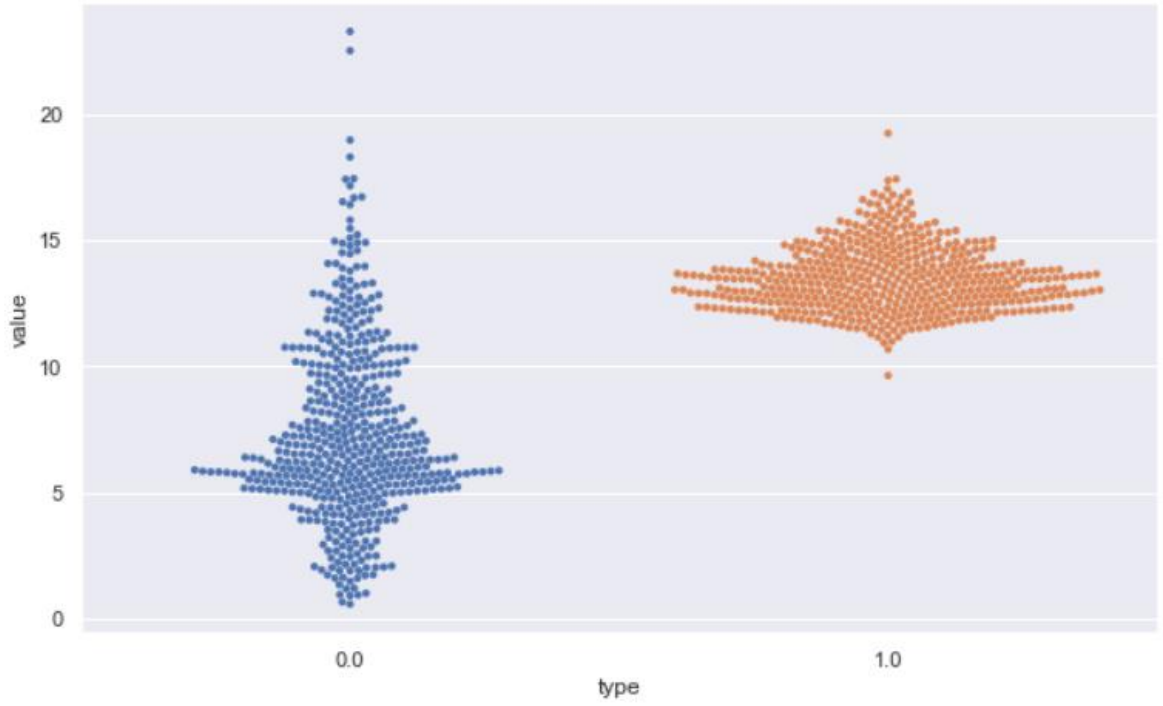
Output:



Inference: To explain the swarm plot and keep the simplicity in this example, we are plotting only **one categorical type**. In the above plot, we can see where the data points are more (denser) and where the data points diverge **away from the center** (both left and right).

```
sb.swarmplot(x="type", y="value", data=dataset, size=4);
```

Output:



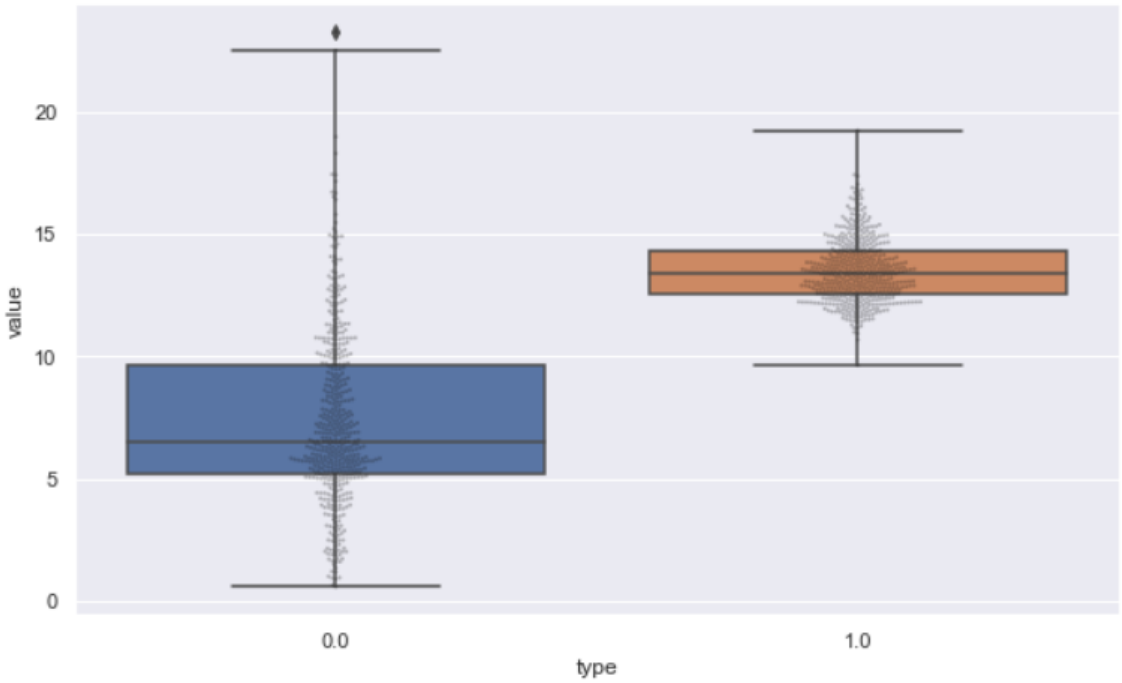
Inference: In the previous graph, we plotted only one category, this time, we are planning two, which are visible on the X axis labeled as **0.0 and 1.0,** and the concept is the same, i.e., the more the density, the more diverge will be data points from the center to both **left and right**.

Box Plots in Data Visualization

Now it's time to move forward and discuss another type of data visualization graph, i.e., **Box plot,** which is also widely known as box and whiskers plots as both the end has **cat-like whiskers**, along with that it put forward a wide variety of inferences in the table that we will discuss now.

```
sb.boxplot(x="type", y="value", data=dataset, whis=3.0);
sb.swarmplot(x="type", y="value", data=dataset, size=2, color="k", alpha=0.3);
```

Output:



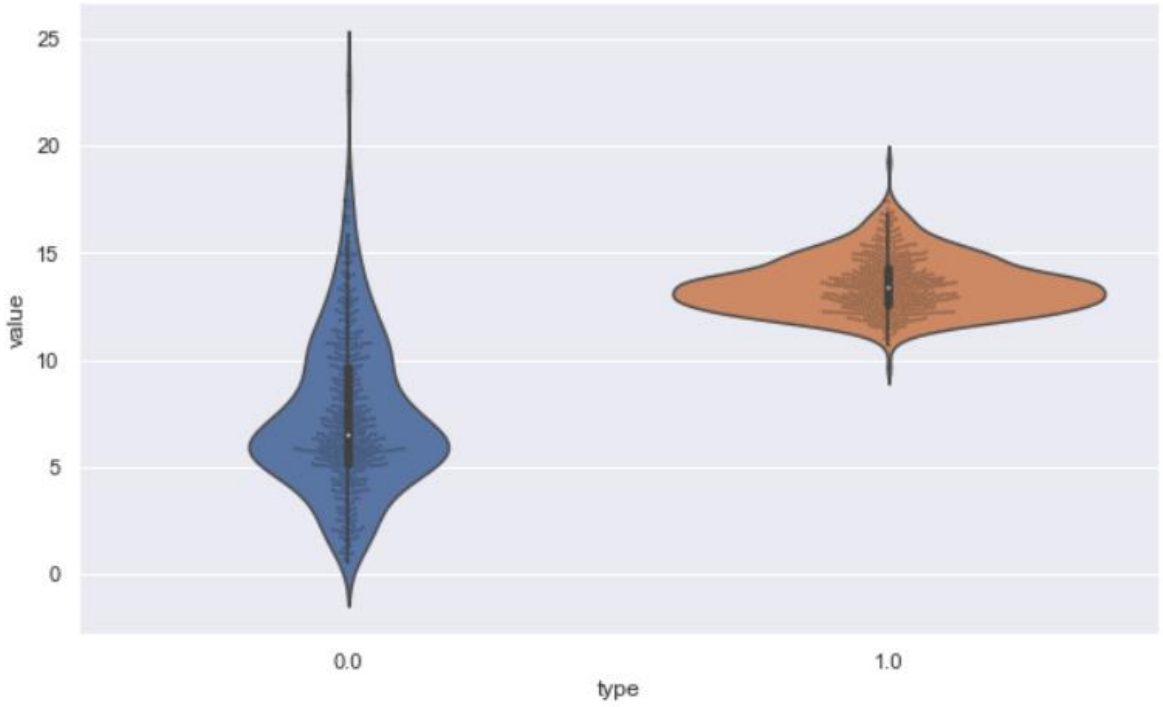
Inference: BoxPlot is yet another visualization plot best suited for **1-D** data, As it demonstrates a variety of statistical measures such as **outliers, median, 25th percentile, 50th percentile, and 75th percentile**. One can notice that along with the boxplot, we are also imputing the swarm plot on top of that, keeping the **alpha value less** so that we can see more of the boxplot and the swarm should not overshadow the same. This is a perfect combination where we can see the **accurate distribution** of the dataset along with its **density**.

Violin Plots in Data Visualization

Okay, so a box plot doesn't display too much, does it? What if we want a hint more information? **The answer is** – We can use **the Violin plot** for the same; we summoned this plot because the boxplot could not show the complete distribution. All it was showing is a summary of statistics, but in the case of **the violin plot,** it is best in the case of **multimodal data** (when data has more than one peak).

```
sb.violinplot(x="type", y="value", data=dataset);
sb.swarmplot(x="type", y="value", data=dataset, size=2, color="k", alpha=0.3);
```

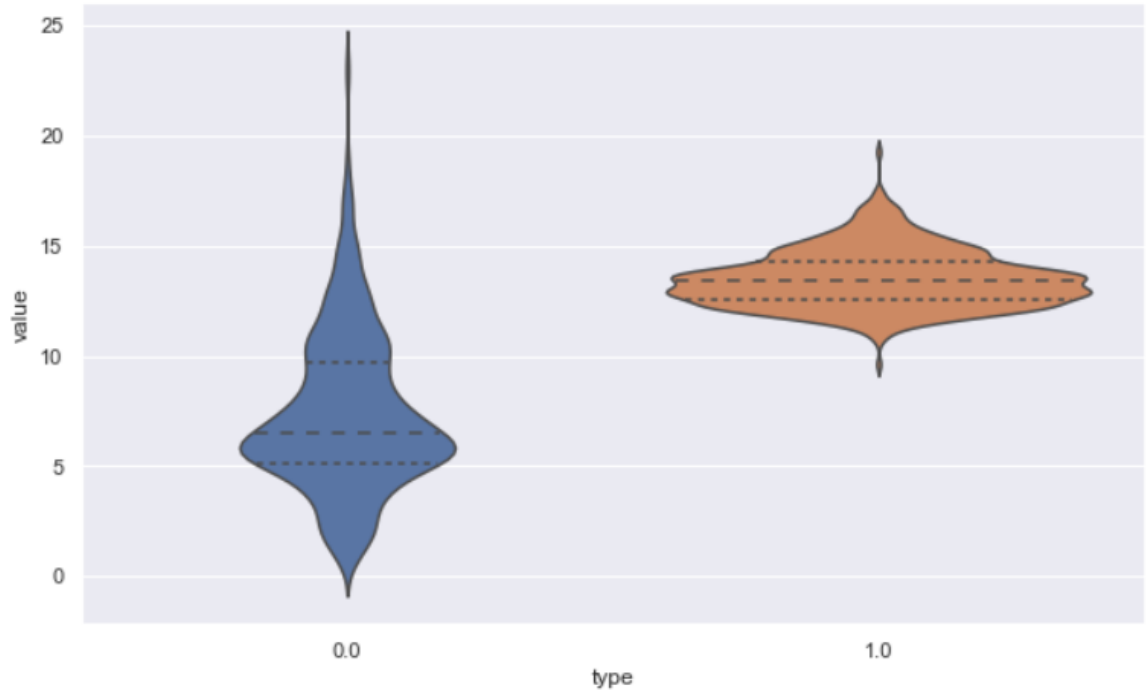
Output:



Inference: We have used a Swarm plot with a Violin plot where the swarm is doing its job, i.e., showing the **data representation** from each data on the other side violin plot shows us the **complete distribution** for both types we have in our self-created DataFrame.

```
sb.violinplot(x="type", y="value", data=dataset, inner="quartile", bw=0.2);
```

Output:



Inference: In **seaborn or matplotlib,** we always have the option to change the plot based on our needs by providing the required parameters. In violin plot also we have that flexibility, and for that, we have used two different parameters:

- **Inner**: This will stimulate how the data points will be represented in the violin plot (internally). We have multiple options like **box, quartile, point**, etc. A quartile is a value we chose, and the output is visible acc to us where the plot has internally changed the look and feel.
- **BW:** This one is to manage the **kernel bandwidth** and how **rigid and smooth** we want the corners to be in our violin plot.

Empirical Cumulative Distribution Functions

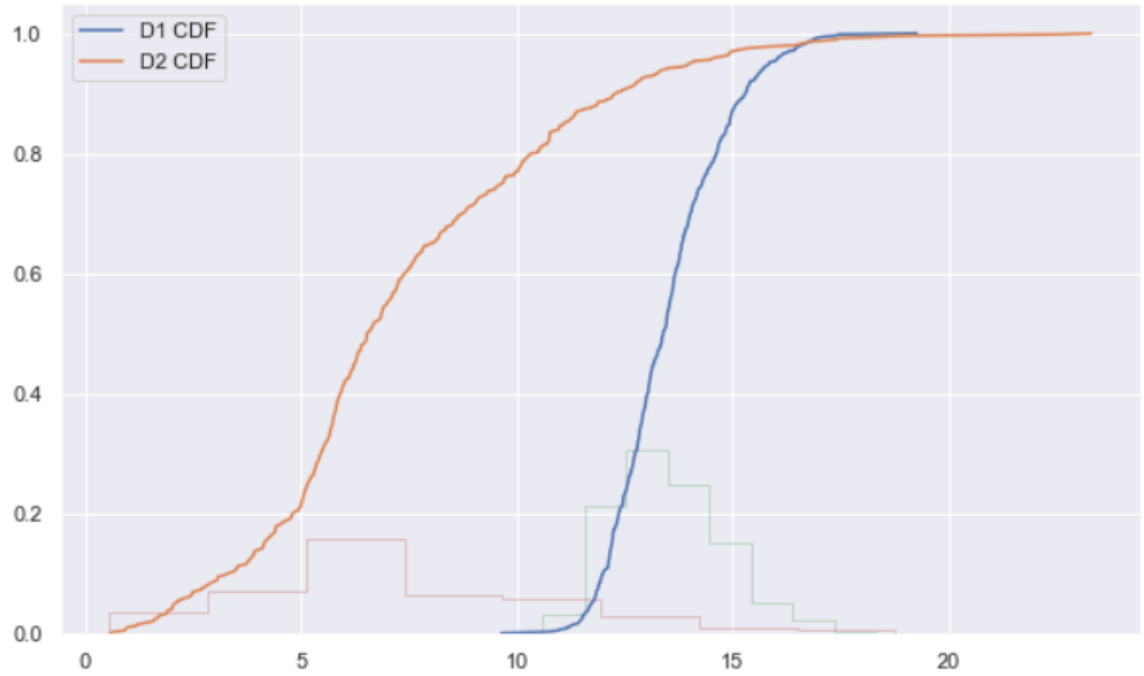
When you form a histogram, the fact that you have to bin data means that the looks can change significantly when you change in size. And each bin has statistical uncertainty. You can get past that using a CDF. It's harder – visually – to see features in the PDF when looking

at the CDF; however, it's generally more useful when you are trying to make quantitative comparisons between multiple distributions. We'll get on that later.

```
sd1 = np.sort(d1)
sd2 = np.sort(d2)
cdf = np.linspace(1/d1.size, 1, d1.size)

plt.plot(sd1, cdf, label="D1 CDF")
plt.plot(sd2, cdf, label="D2 CDF")
plt.hist(d1, histtype="step", density=True, alpha=0.3)
plt.hist(d2, histtype="step", density=True, alpha=0.3)
plt.legend();
```

Output:



Inference: CDF plots are not usually recommended for extracting insights for business needs. They are a bit complex to understand, but they can be a good resource for further analytics. There is an exciting relationship between **Histograms** and **CDF plots** **where** Histograms represent the **area of the bar** because of its nature of representation. At the same time, **CDF is cumulative** as it returns the **integral of PDF**.

Conclusion

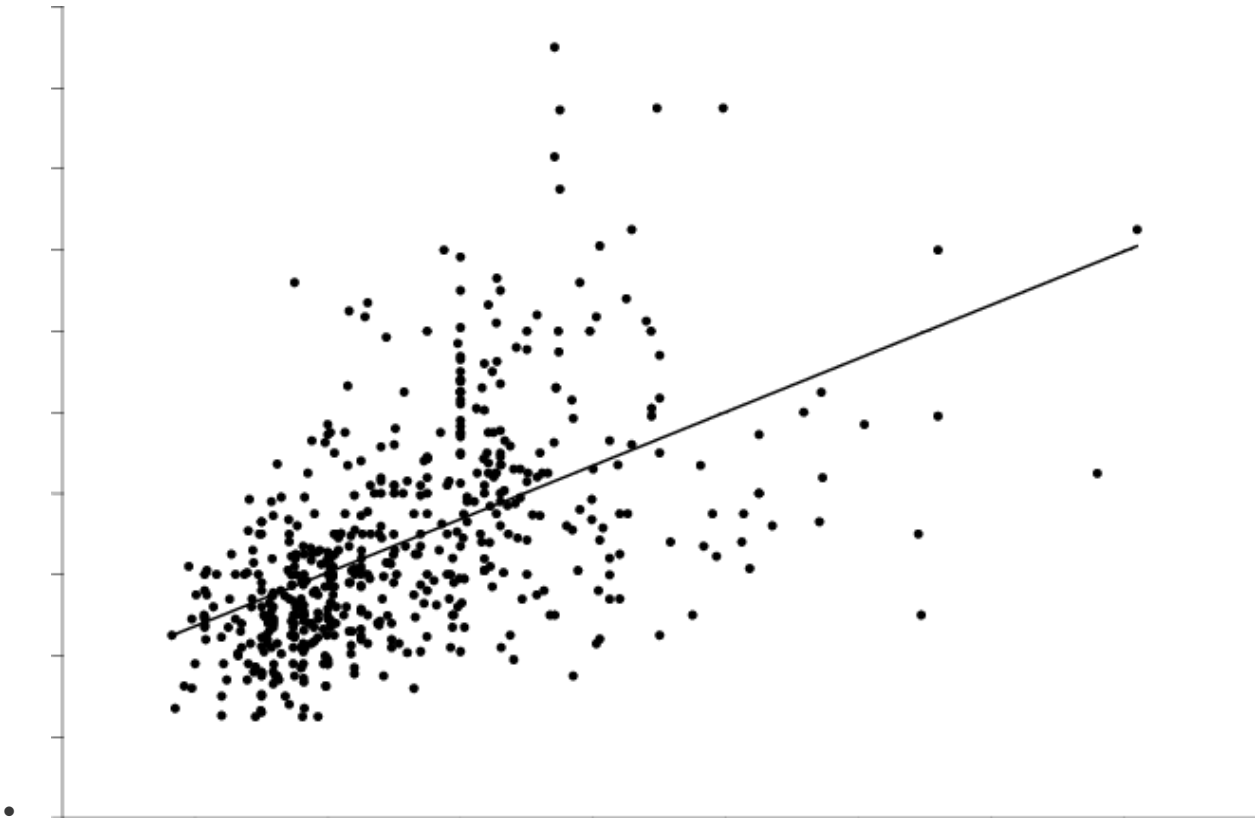
We are in the end game now. This is the last section of the article, where we will point out all the stuff that we have learned so far about **visualizing the data when it is in format of one-dimensional**. In a nutshell, we will give a brief explanation to get you to know the flow and key takeaways from the article.

1. Things started with **the Bee Swarm Plot,** where we learned about the perks and cons of this plot which helped us know **when to use** this plot to represent data distribution.
2. Then comes the **box and violin** plot, which echoed similar functionalities and output. Here we learned how the violin plot could be a better alternative when showing the **data distribution completely** and not only **summary statistics**.
3. The last plot was **Empirical Cumulative Distribution Functions (CDF),** where we saw the usage of this plot and also came to know the interesting relationship between **Histograms and CDF.**

Data visualization on two dimensional data

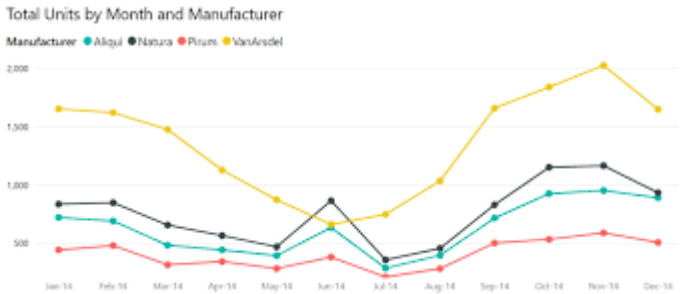
Data visualization is a powerful tool for exploring and communicating patterns, trends, and insights within two-dimensional data. Here are several common types of visualizations you can use for two-dimensional data:

1. **Scatter Plots:**
    - Scatter plots are effective for displaying the relationship between two continuous variables.
    - Each point on the plot represents an observation, with one variable on the x-axis and the other on the y-axis.



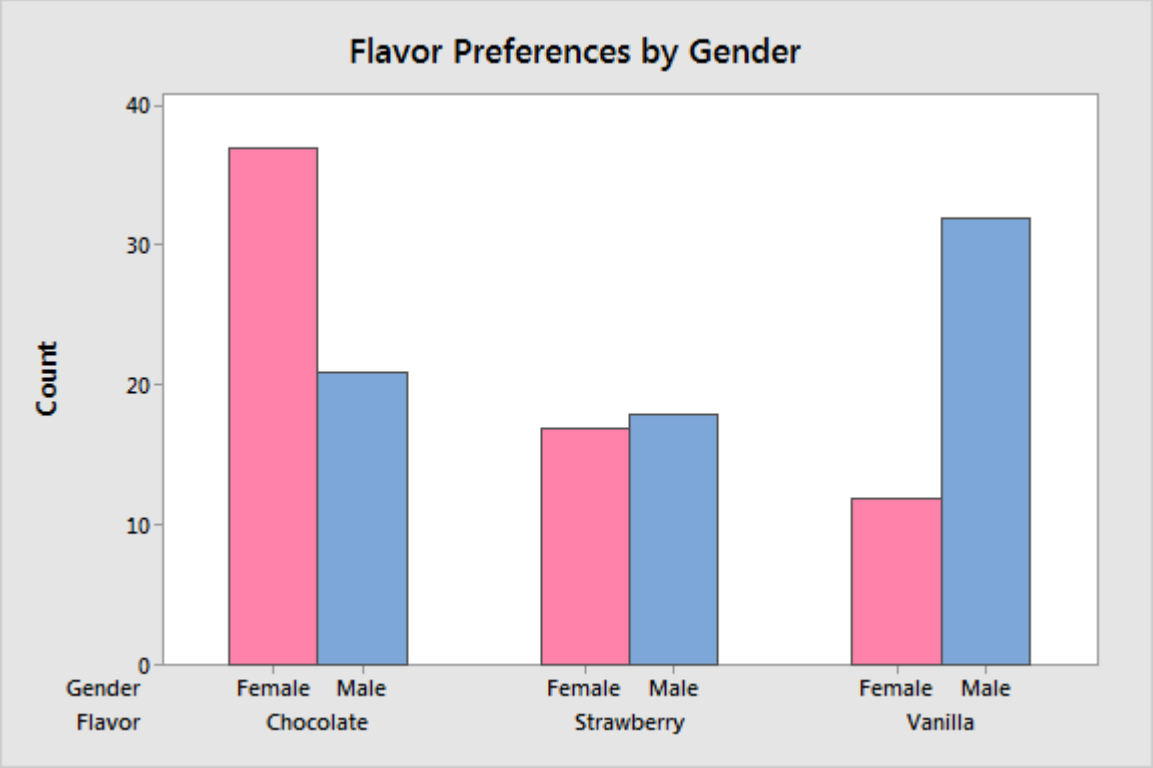
2. **Line Charts:**

    - Line charts are useful for showing trends or patterns over time.
    - They connect data points with lines, making it easy to observe changes in the two variables.



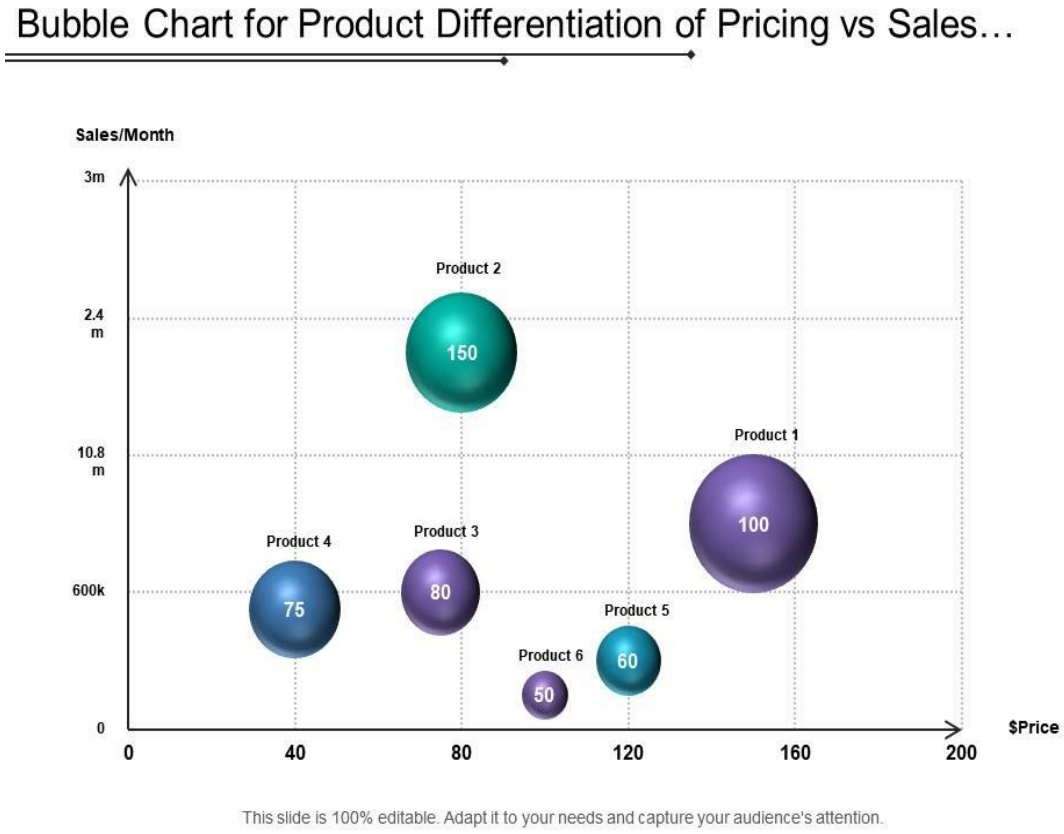
3. **Bar Charts:**

    - Bar charts are suitable for comparing the values of two variables across different categories.
    - The length of each bar represents the magnitude of the variable.



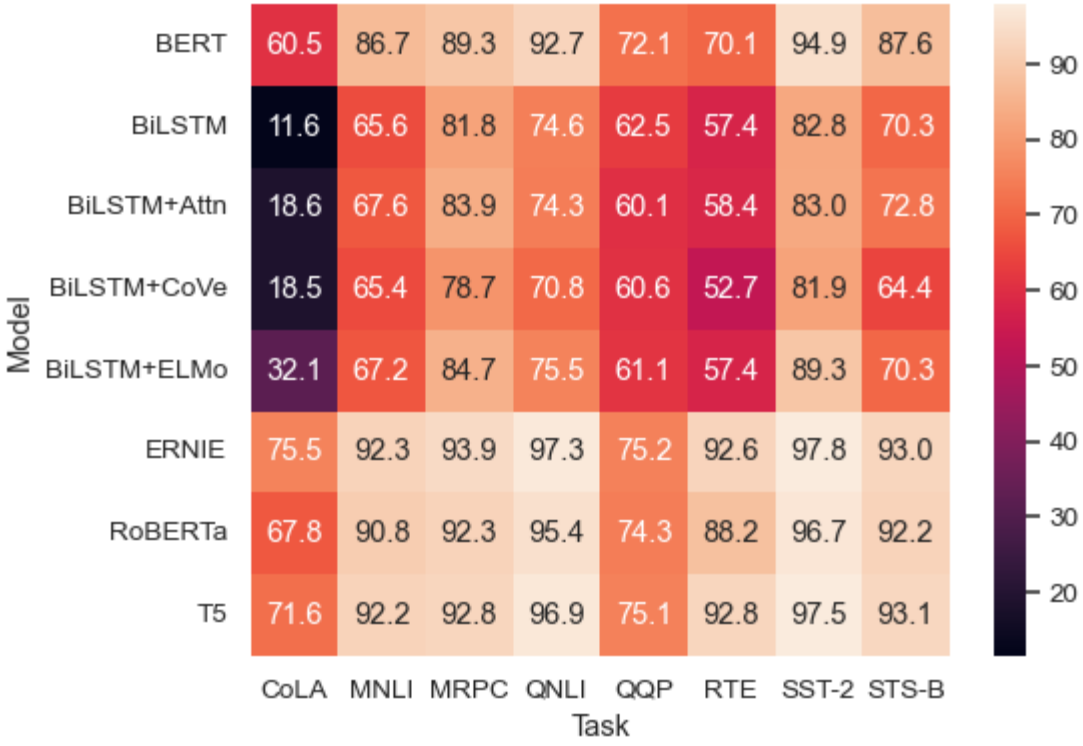
4. **Bubble Charts:**

    - Bubble charts extend the concept of scatter plots by adding a third dimension: the size of the bubble.
    - The size of each bubble corresponds to a third variable, providing additional information.



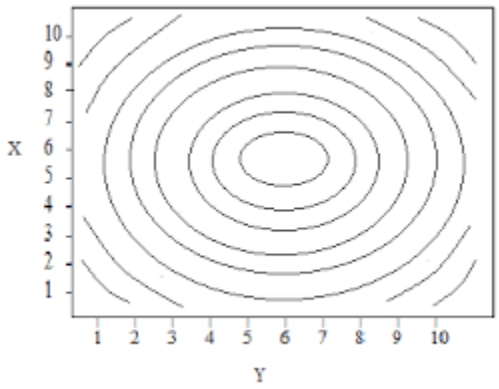
5. **Heatmaps:**

   - Heatmaps are effective for visualizing the intensity of a relationship between two variables.
   - Color gradients represent the magnitude of the relationship.



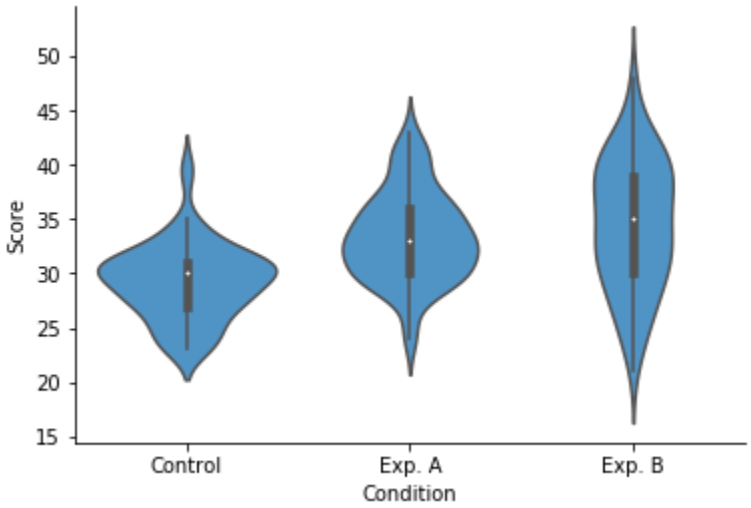
6. **Contour Plots:**

   - Contour plots are helpful when you want to show the distribution of a third variable over a two-dimensional space.
   - Contour lines connect points of equal value for the third variable.



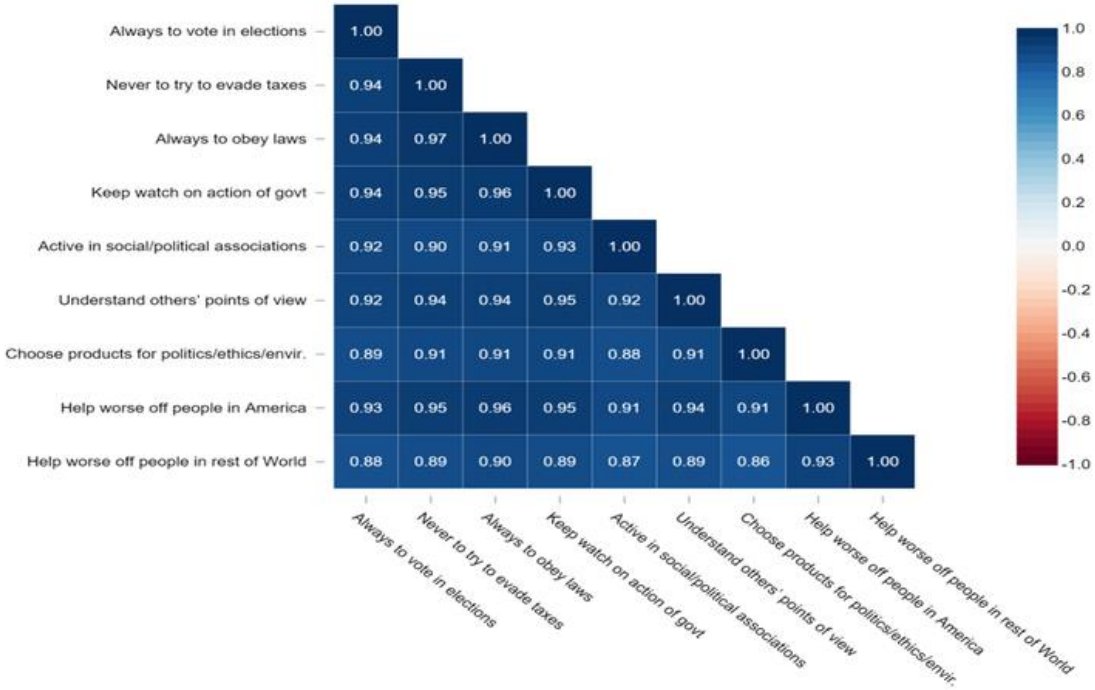
7. **Violin Plots:**

   - Violin plots combine aspects of box plots and kernel density plots to show the distribution of a variable for different categories.



8. **Correlation Matrix:**

- A correlation matrix visually displays the correlation coefficients between pairs of variables, providing insights into relationships.



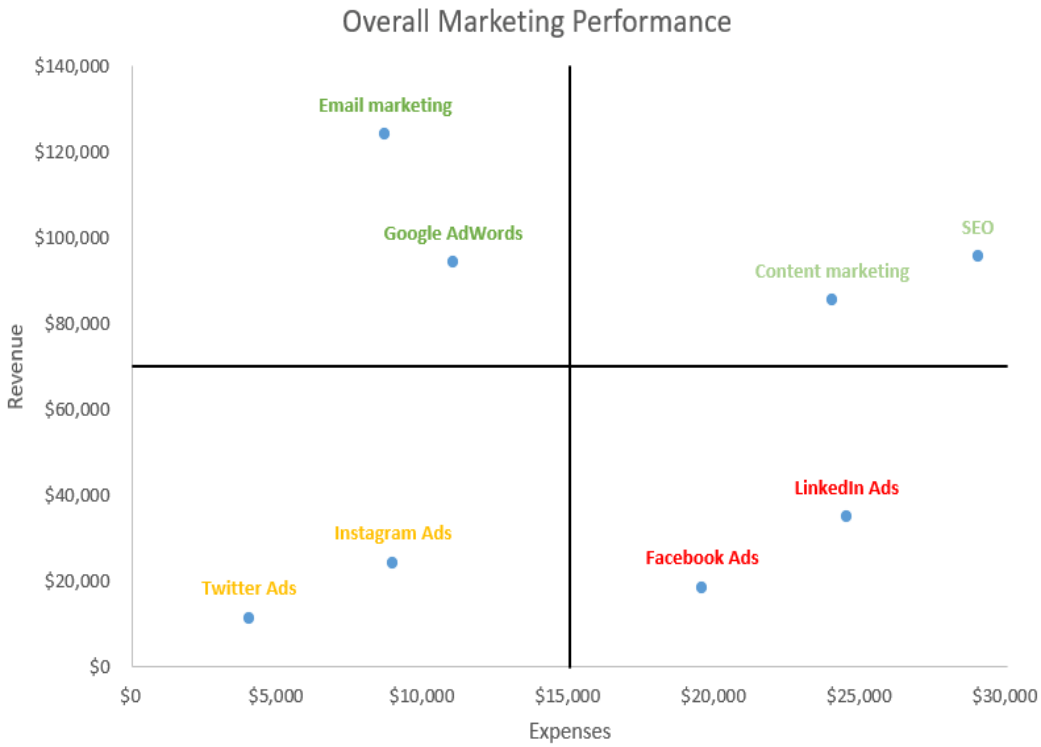
9. **Area Charts:**

- Area charts can be used to represent the cumulative effect of two variables over time.

- The shaded area between the lines represents the combined impact.

10. **Quadrant Charts:**

- Quadrant charts divide a two-dimensional space into four quadrants, helping classify data points based on two variables.



Data visualization on Multi dimensional data

Introduction

One can view numbers and play with numbers if they are statisticians to get some valuable insights from the raw data, but in the real world do we know whether our stakeholders are that much into statistics? Majorly answers will be straight **“no”**, to give your series of knowledge and insights a **storytelling** nature, we need to provide it in the form of **Data Visualization** hence converting your numeric data into insightful **graphs/plots/charts**.

Importing all the Necessary Packages

Starting with importing the relevant libraries like **NumPy** (handling mathematical calculations), **pandas** (DataFrame manipulations), **matplotlib** (The OG visualization library closest to python interpreter and **“C”** development), and last but not least- **seaborn** (built on top of matplotlib it give way more options and better look and feels comparative).

Inference: We will be using two datasets for this article, one will be the **diabetes patients dataset,** and another one is the height and weight of the persons. In the above output, we can see a glimpse of the first dataset using the **head**() method.

Python Code:

Inference: Removing the **missing or junk** values from the dataset is always the priority step. Here we are first replacing the **0** value with **NaN** as we had 0 as the bad data in our **features** column.

```
df.info()
```

Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   763 non-null    float64
 2   BloodPressure             733 non-null    float64
 3   SkinThickness             541 non-null    float64
 4   Insulin                   394 non-null    float64
 5   BMI                       757 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    float64
 8   Outcome                   768 non-null    int64
dtypes: float64(7), int64(2)
memory usage: 54.1 KB
```

Inference: Now we can see that in some of the columns, there are a few Null values like **Skin thickness, Insulin, BMI**, etc.

Viewing the Data

So, surprisingly no one it's useful to view the data. Straight up by using head, we can see that this dataset is utilizing **0** to represent no value – unless some poor unfortunate soul has a skin thickness of 0.

If we want to do more than expect the data, we can use the **described** function we talked about in the previous section.

```
df.describe()
```

Output:

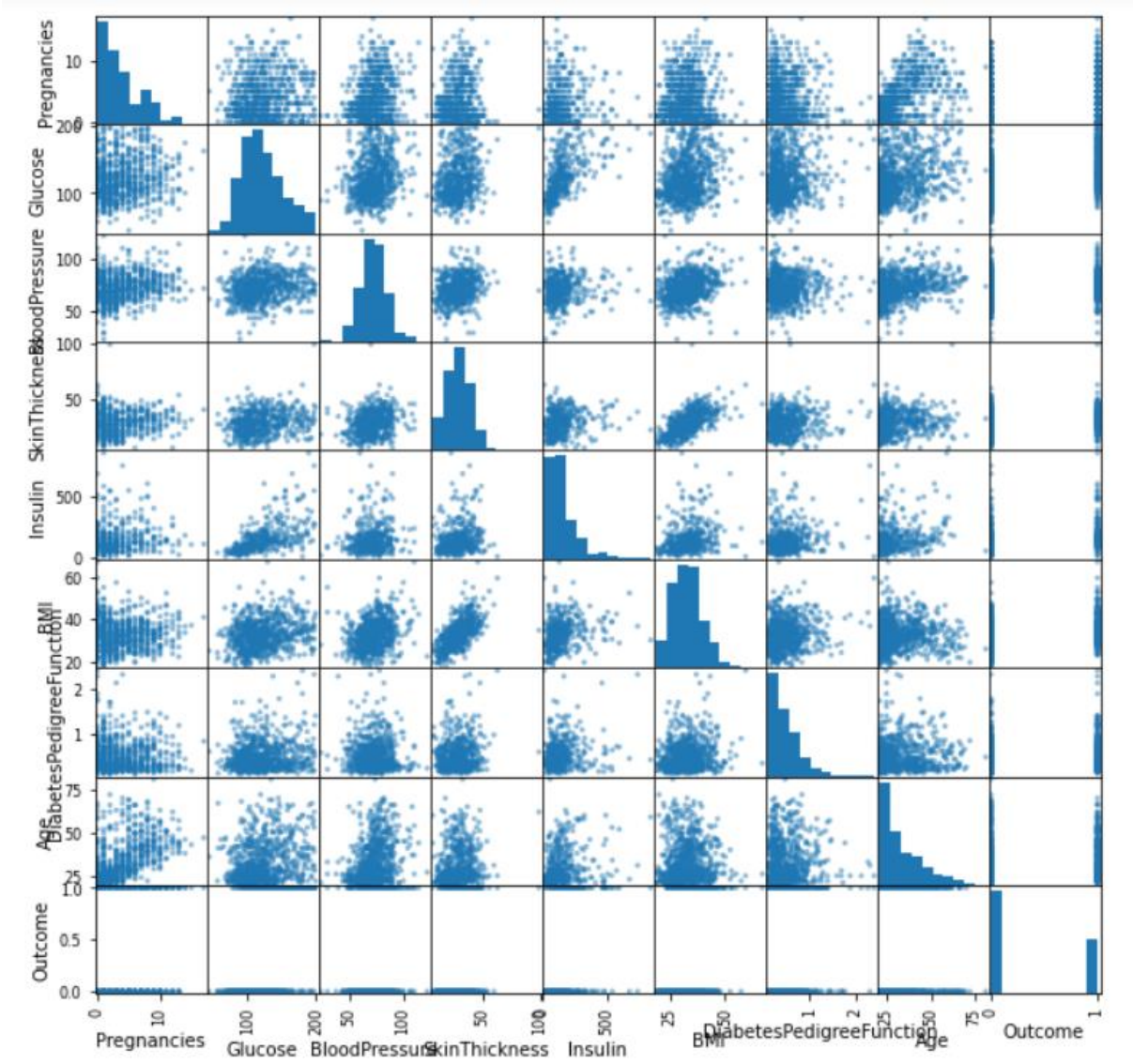
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	763.000000	733.000000	541.000000	394.000000	757.000000	768.000000	768.000000	768.000000
mean	3.845052	121.686763	72.405184	29.153420	155.548223	32.457464	0.471876	33.240885	0.348958
std	3.369578	30.535641	12.382158	10.476982	118.775855	6.924988	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	64.000000	22.000000	76.250000	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	29.000000	125.000000	32.300000	0.372500	29.000000	0.000000
75%	6.000000	141.000000	80.000000	36.000000	190.000000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Scatter Matrix

Scatter Matrix is one of the best plots to determine the relationship (**linear** mostly) between the **multiple variables** of all the datasets; when you will see the linear graph between two or more variables that indicates the **high correlation** between those features, it can be either **positive** or **negative** correlation.

```
pd.plotting.scatter_matrix(df, figsize=(10, 10));
```

Output:



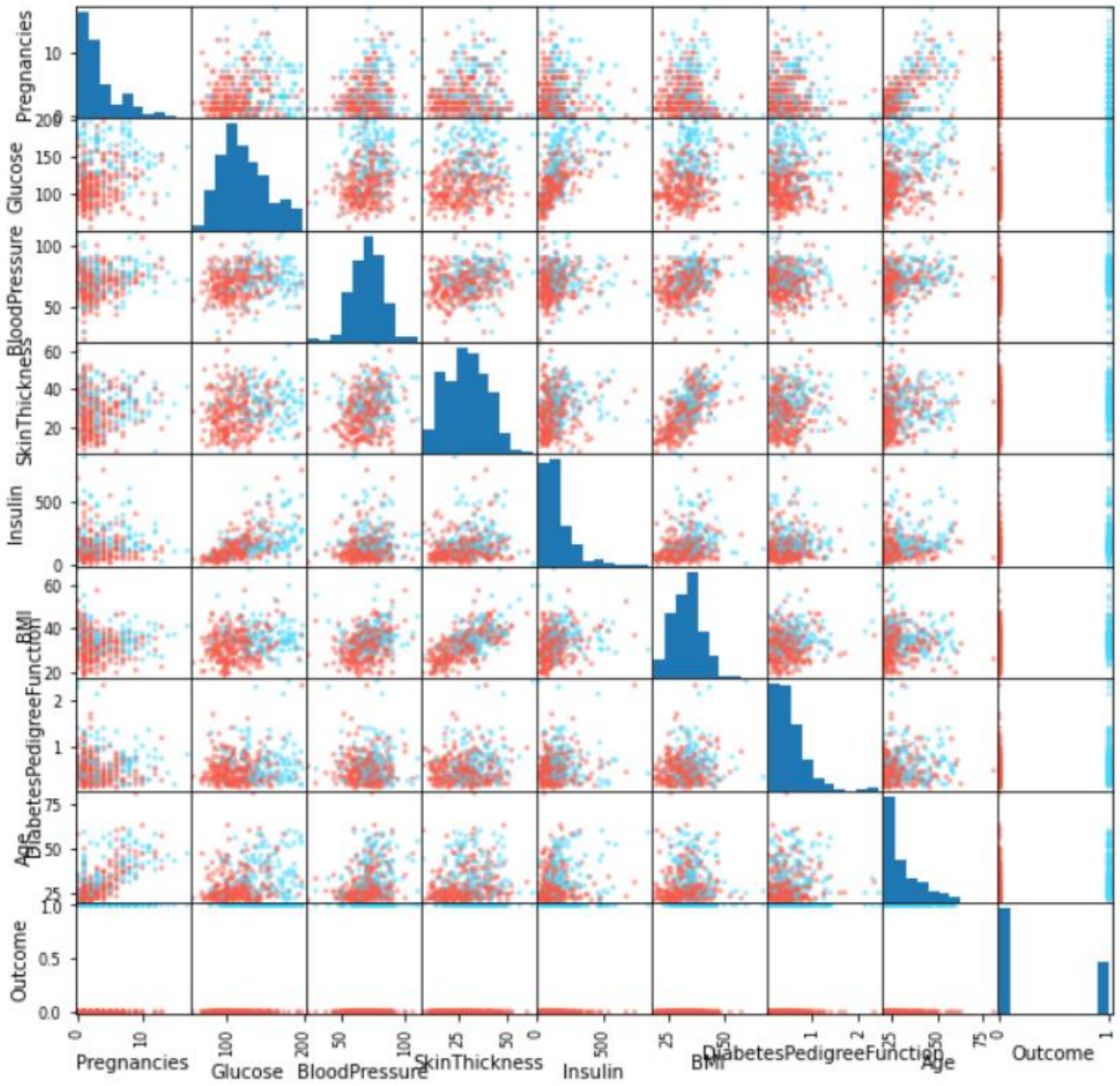
Inference: From the above plot, we can say that this plot alone is quite **descriptive** as it is showing us the linear relationship between all the variables in the dataset. For example, we can see **that skin Thickness** and **BMI** is sharing linear tendencies.

Note: Due to the big names of columns, we are facing a bit issue while reading the same though that can be improved (out of the scope of the article).

```
df2 = df.dropna()
colors = df2["Outcome"].map(lambda x: "#44d9ff" if x else "#f95b4a")
```

```
pd.plotting.scatter_matrix(df2, figsize=(10,10), color=colors);
```

Output:



Inference: The scatter plot gives us both the histograms for the distributions **along the diagonal** and also a lot of 2D scatter plots **off-diagonal**. Not that this is a symmetric matrix, so I just look at the diagonal and below it normally. We can see that some variables have a lot of scattering, and some are correlated (ie, there is a direction in their scatter). This leads us to another type of plot i.e., **correlation plot**.

Correlation Plots

Before going into a deep discussion with the correlation plot, we first need to understand the correlation and for that reason, we are using the **pandas' corr()** method that will return the **Pearson's correlation coefficient** between two data inputs. In a nutshell, these plots easily quantify which variables or attributes are correlated with each other.

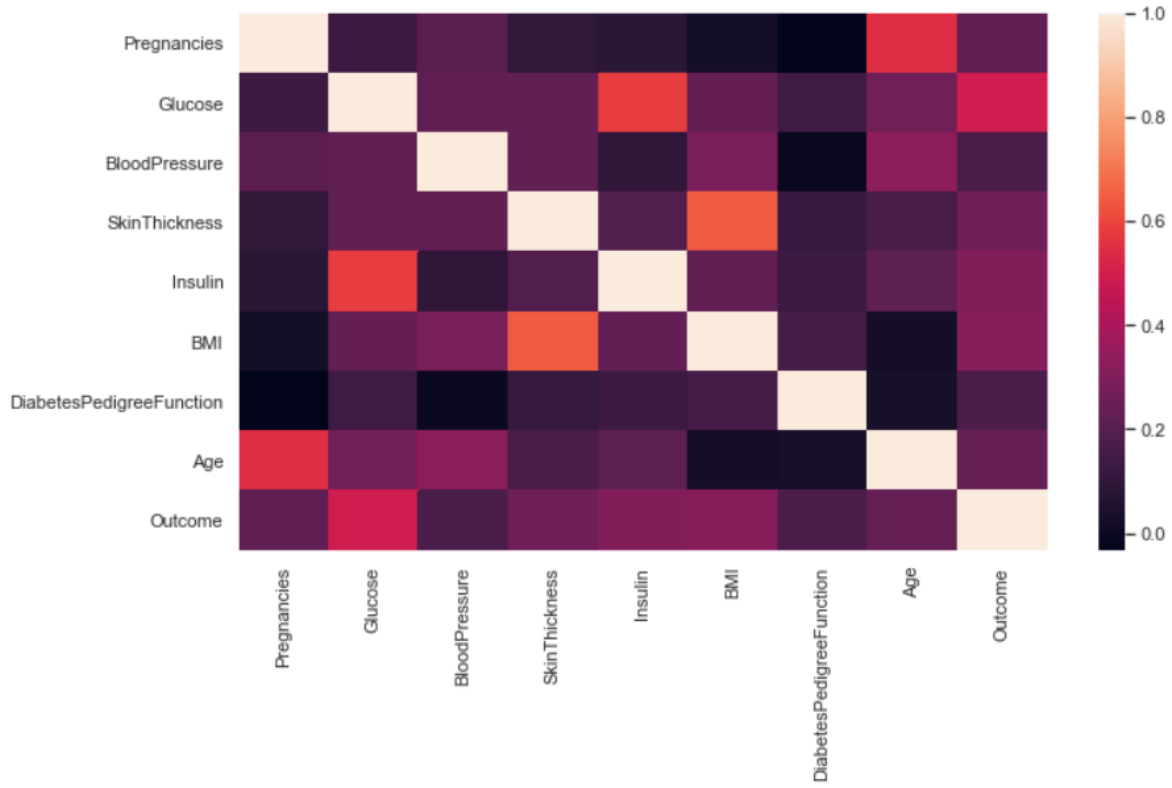
```
df.corr()
```

Output:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.128135	0.214178	0.100239	0.082171	0.021719	-0.033523	0.544341	0.221898
Glucose	0.128135	1.000000	0.223192	0.228043	0.581186	0.232771	0.137246	0.267136	0.494650
BloodPressure	0.214178	0.223192	1.000000	0.226839	0.098272	0.289230	-0.002805	0.330107	0.170589
SkinThickness	0.100239	0.228043	0.226839	1.000000	0.184888	0.648214	0.115016	0.166816	0.259491
Insulin	0.082171	0.581186	0.098272	0.184888	1.000000	0.228050	0.130395	0.220261	0.303454
BMI	0.021719	0.232771	0.289230	0.648214	0.228050	1.000000	0.155382	0.025841	0.313680
DiabetesPedigreeFunction	-0.033523	0.137246	-0.002805	0.115016	0.130395	0.155382	1.000000	0.033561	0.173844
Age	0.544341	0.267136	0.330107	0.166816	0.220261	0.025841	0.033561	1.000000	0.238356
Outcome	0.221898	0.494650	0.170589	0.259491	0.303454	0.313680	0.173844	0.238356	1.000000

```
sb.set(rc={'figure.figsize':(11,6)})
sb.heatmap(df.corr());
```

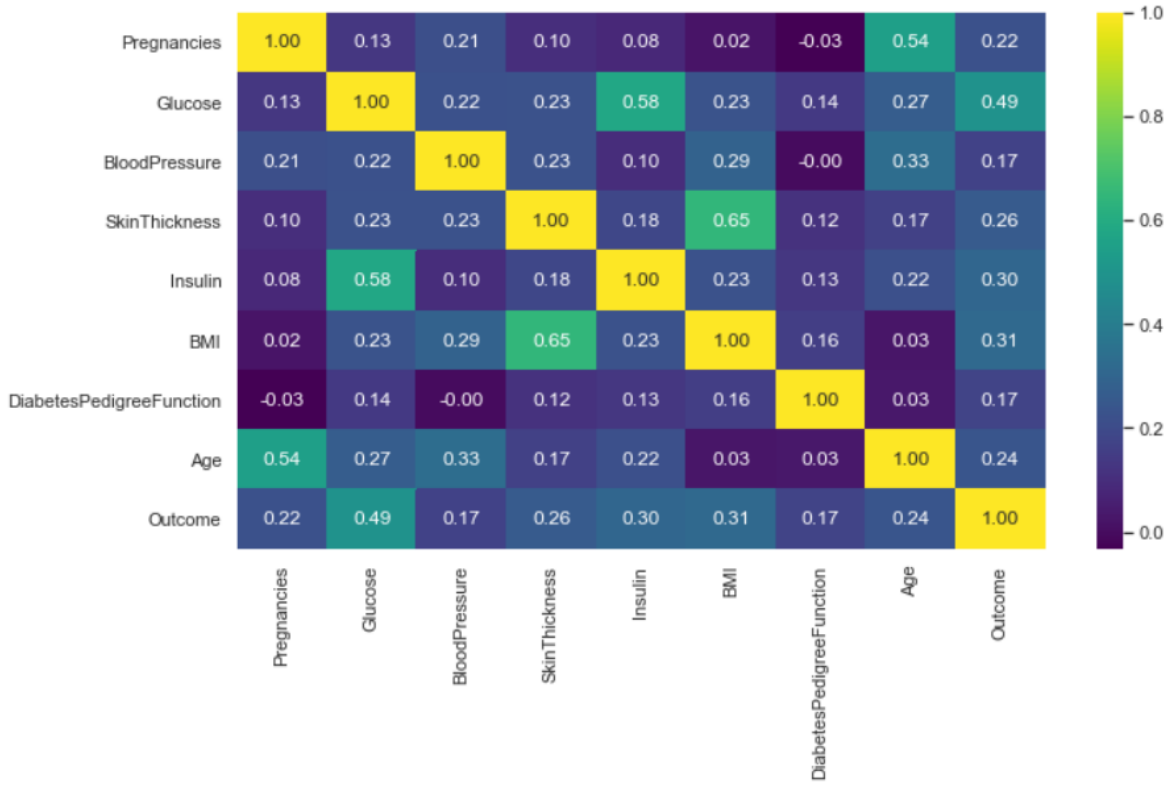
Output:



Inference: In a seaborn or matplotlib supported **correlation plot,** we can compare the higher and lower correlation between the variables using its color palette and scale. In the above graph, **the lighter the color, the more the correlation and vice versa**. There are some drawbacks in this plot which we will get rid of in the very next graph.

```
sb.heatmap(df.corr(), annot=True, cmap="viridis", fmt="0.2f");
```

Output:



Inference: Now one can see this is a **symmetric matrix** too. But it immediately allows us to point out the most **correlated** and **anti-correlated attributes**. Some might just be common sense – Pregnancies v Age for example – but some might give us a real insight into the data.

Here, we have also used some parameters like **annot= True** so that we can see correlated values and some formatting as well.

2D Histograms

2D Histograms are mainly used for **image processing,** showing the **intensities of pixels** at a certain position of the image. Similarly, we can also use it for other problem statements, where we need to analyze two or more variables as **two-dimensional** or **three-dimensional** histograms, which provide multi-dimensional Data.

For the rest of this section, we're going to use a different dataset with more data.

Note: 2-D histograms are very useful when you have a **lot** of data. See here for the API.

```
df2 = pd.read_csv("height_weight.csv")
df2.info()
df2.describe()
```

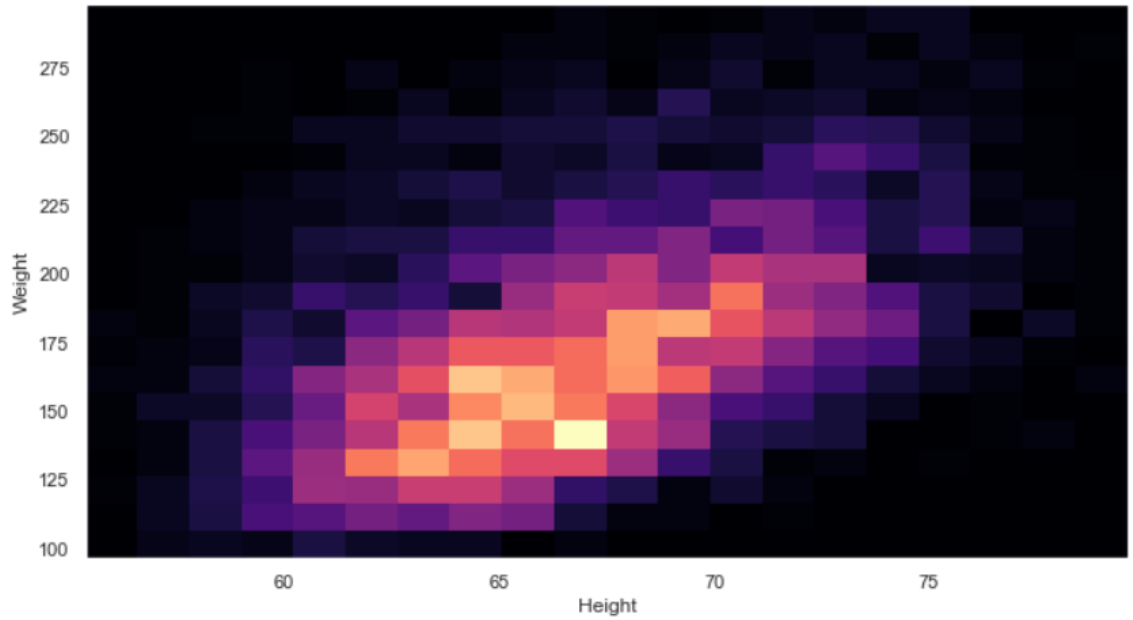
Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4231 entries, 0 to 4230
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   sex     4231 non-null   int64  
 1   height  4231 non-null   float64
 2   weight  4231 non-null   float64
dtypes: float64(2), int64(1)
memory usage: 99.3 KB
```

	sex	height	weight
count	4231.000000	4231.000000	4231.000000
mean	1.540061	66.903607	174.095122
std	0.498451	4.313004	38.896171
min	1.000000	55.400000	96.590000
25%	1.000000	63.730000	144.315000
50%	2.000000	66.630000	170.100000
75%	2.000000	69.970000	198.660000
max	2.000000	79.610000	298.440000

```
plt.hist2d(df2["height"], df2["weight"], bins=20, cmap="magma")
plt.xlabel("Height")
plt.ylabel("Weight");
```

Output:



Inference: We have also worked with one-dimensional Histograms for multi-dimensional Data, but that is for **univariate analysis** now if we want to get the data distribution of more than one feature then we have to shift our focus to **2-D Histograms**. In the above 2-D graph height and weight is plotted against each other, keeping the C-MAP as **magma**.

Contour plots

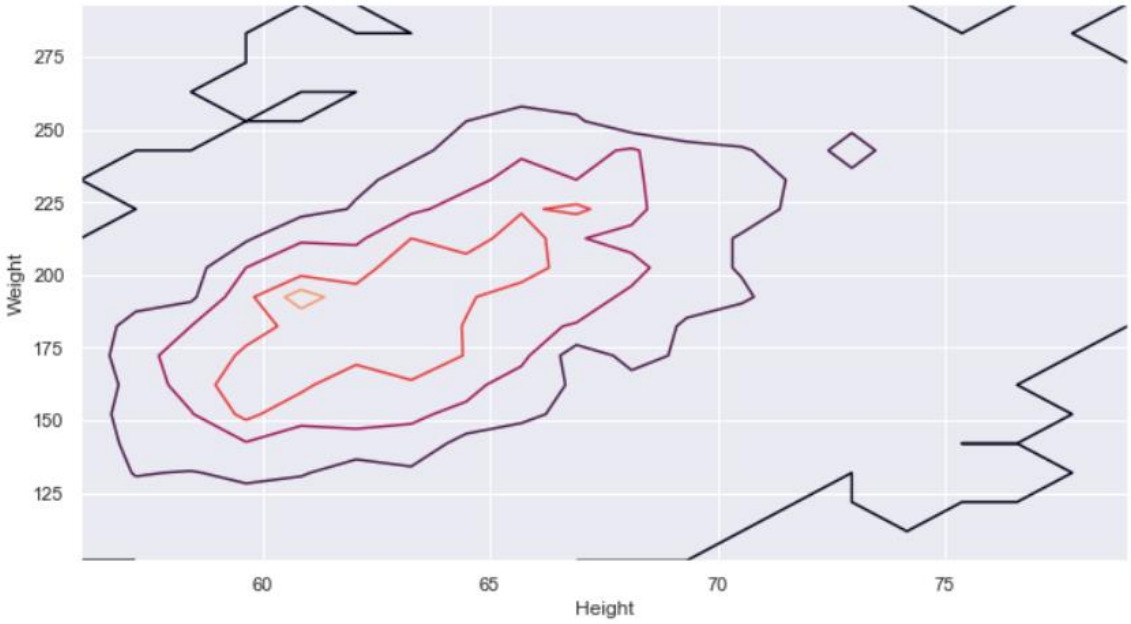
Bit hard to get information from the 2D histogram, isn't it? Too much noise in the image. What if we try and contour diagram? We'll have to bin the data ourselves.

Every alternative comes into the picture when the original one has some drawbacks, Similarly, in the case with **2-D histograms** it becomes a bit hard to get the information from it as there is soo much noise in the graph. Hence now, we will go with **a contour plot**

Here is the resource that can help you deep dive, into this plot. The contour API is here.

```
hist, x_edge, y_edge = np.histogram2d(df2["height"], df2["weight"], bins=20)
x_center = 0.5 * (x_edge[1:] + x_edge[:-1])
y_center = 0.5 * (y_edge[1:] + y_edge[:-1])
plt.contour(x_center, y_center, hist, levels=4)
plt.xlabel("Height")
plt.ylabel("Weight");
```

Output:



Inference: Now we can see that this contour plot which is way better than a complex and noisy 2-D Histogram as it shows the clear distribution between **height and weight** simultaneously. There is still room for improvement. If we will use the **KDE plot from seaborn,** then the same contours will be smoothened and more clearly informative.

Conclusion

From the very beginning of the article, we are primarily focussing on data visualization for the multi-dimensional data, and in this journey, we got through all the important graphs/plots that could derive business-related insights from the

numeric data from multiple features all at once. In the last section, we will cover all these graphs in a nutshell.

1. Firstly we got introduced to **a scatter matrix** that shows us the relationship of every variable with the other one. Then using seaborn **heat map** is used to get a better approach to **multivariable analysis**.
2. Then came the **2-D histograms,** where we can go with binary variable analysis, i.e., 2 variables can be simultaneously seen, and we can get insights from them.
3. At last, we got to know about **the Contour plot,** which helped us to get a better version of 2-D histograms as it **removes the noise** from the image and has a more clear interpretation.

Text and Document Visualization

last updated 11/11/19

Screen Mode

Introduction

Here we consider visualizing the text within a document, and collections of documents which are likely related (corpus).

Difficulty in analysis includes the loose structure, varied vocabulary, and optional metadata such as author(s), date, modification dates, comments, keywords, catalog codes, citations.

Levels of text to be represented:

- Lexical level -- Simple grouping of characters into "tokens" which are typically words, but word stems, phrases, word n-grams and character n-grams may be beneficial
- Syntactic level --Parsing purpose of token, grammatical category, tense, plurality, in the context of the phrase, sentence and paragraph
- Semantic level -- Extract meaning of the syntactic structure with the tokens using fuller analysis of the context.

Vector Space Model

Analysis of the words in a document and determine their value in contribution and significance to the document.

Removal of noise words ("a", "an", "the", "that") and punctuation, and stemming (collecting roots of words) are typical of preprocessing.

Simple frequency counts of significant words ordered by decreasing frequency is a simple vector.

Here is a web site to generate this vector: http://www.wordcounter.com/ It has options to remove noise and do stemming.

Plugging the above Introduction text into this web site, we get the following vector:

Word	Frequency
level	4
word	3
token	3
syntactic	2
character	2
analysis	2
n-gram	2
text	2
context	2
phrase	2
structure	2

Term Frequency--Inverse Document Frequency

Computing weights can associate more significance to words.

Tf(word) = *frequency or number of times the word appears in the current document*

Df(word) = *frequency or number of documents the word appears in the corpus*

N = *number of documents*

TfIdf(word) = *Tf(word) * log (N / Df(word))*

This scales the frequency of a word by reducing its significance if it occurs in fewer documents. For example if it occurs in all documents, log (n/n) = log(1) which is zero. If it occurs in one of 10 documents then log(10) is 1. The algorithm is on page 295.

id	men	entered	bank	charlotte	missiles	masks	aryan	guns	witnesses	reported	silver	suv	august
seg1.txt	0.239441	0	0.153457	0.195243	0	0.237029	0	0.195243	0.237029	0.140004	0.195243	0.237029	0
seg13.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg14.txt	0	0.192197	0	0	0	0	0	0	0	0	0	0	0.172681
seg15.txt	0	0	0	0	0	0	0	0	0	0	0	0	0.149652
seg16.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg17.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg18.txt	0	0.158432	0	0	0	0	0	0	0	0	0	0	0
seg19.txt	0	0	0	0.197255	0	0	0	0	0	0.141447	0	0	0.155038
seg2.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg20.txt	0	0.234323	0	0	0	0	0	0	0	0	0	0	0
seg21.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg22.txt	0	0	0	0	0.139629	0	0.127389	0	0	0	0	0	0
seg23.txt	0	0	0	0	0	0	0	0	0	0.180656	0	0	0
seg24.txt	0	0	0	0	0	0	0.117966	0	0	0.117966	0	0	0
seg25.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg26.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg27.txt	0	0	0.235418	0	0	0	0.214781	0	0	0	0	0	0
seg28.txt	0	0	0	0	0.151753	0	0	0	0	0	0	0	0
seg29.txt	0	0	0	0	0	0	0.129852	0	0	0	0	0	0.142329
seg3.txt	0	0	0	0	0.18432	0	0	0	0	0	0	0	0
seg30.txt	0.078262	0	0	0	0	0	0	0	0	0	0	0	0
seg31.txt	0	0	0.213409	0	0	0	0.194701	0	0	0	0	0	0
seg32.txt	0	0	0	0	0	0	0	0	0	0	0	0	0

Mapping vector space models to the document.

This example highlights named entities.

Source:
Date: Nov 16, 2004

Color Legend:

person
place
organization
date
money
time

Alderwood to probe voting machines

Story by: Ellie Olmsen

Republicans in Alderwood joined Democrats yesterday in criticizing the performance of the city's costly new high-tech voting system, saying that it may have disenfranchised voters in the Nov. 4 election.

The Republican commission scolded the city board of elections for minimizing problems with the touch-screen machines that the city purchased this year for $1.5 million and asked Mayor Rex Luthor to investigate what went wrong before the machines are pressed into service again.

Alderwood's touch-screen voting machines, which resemble laptop computers without keyboards, were supposed to simplify voting and tabulating results. But in a debut that mirrored many of the problems experienced last year in areas across the country, some voters found the machines confusing, and the reporting of vote tallies was delayed almost a day.

Luthor responded that he would try to address the board's concerns. He said he has called for a public meeting of the three-member board of elections to go over the requests at 5 p.m. today.

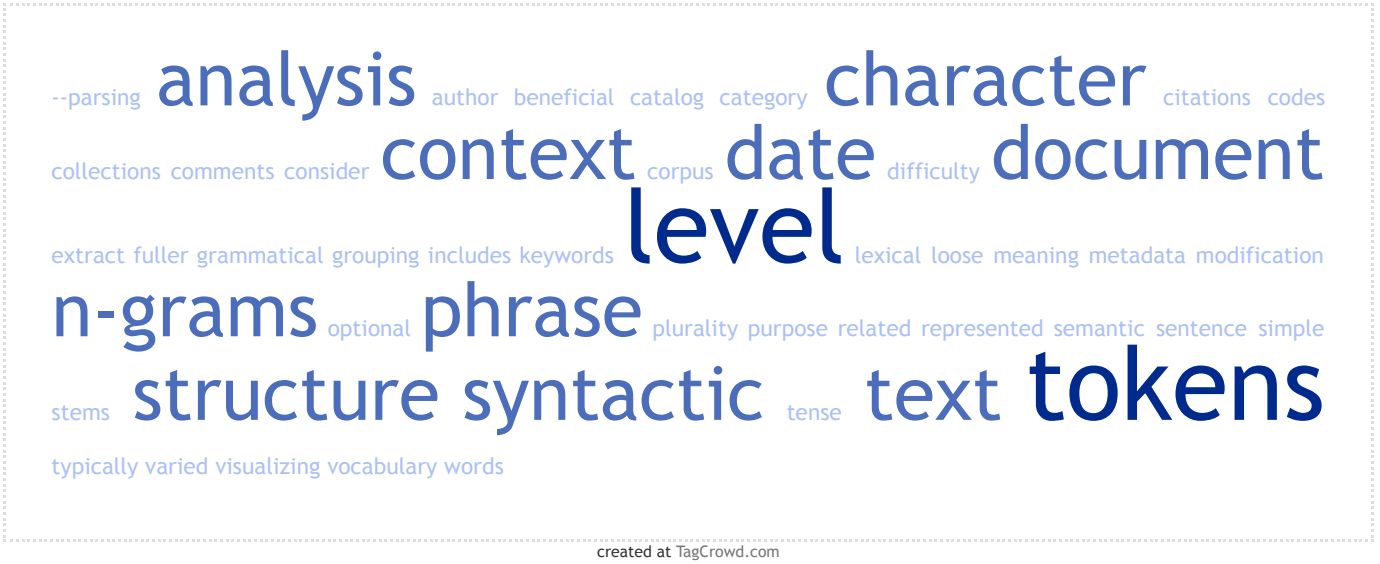
"I pledge that I will answer every question as soon as I possibly can in the proper fashion," he said.

Single Document Visualization

Tag Clouds

Tag crowd (tagcrowd.com)

visualizes the words by size based on frequency. Again this is the opening Intro section.

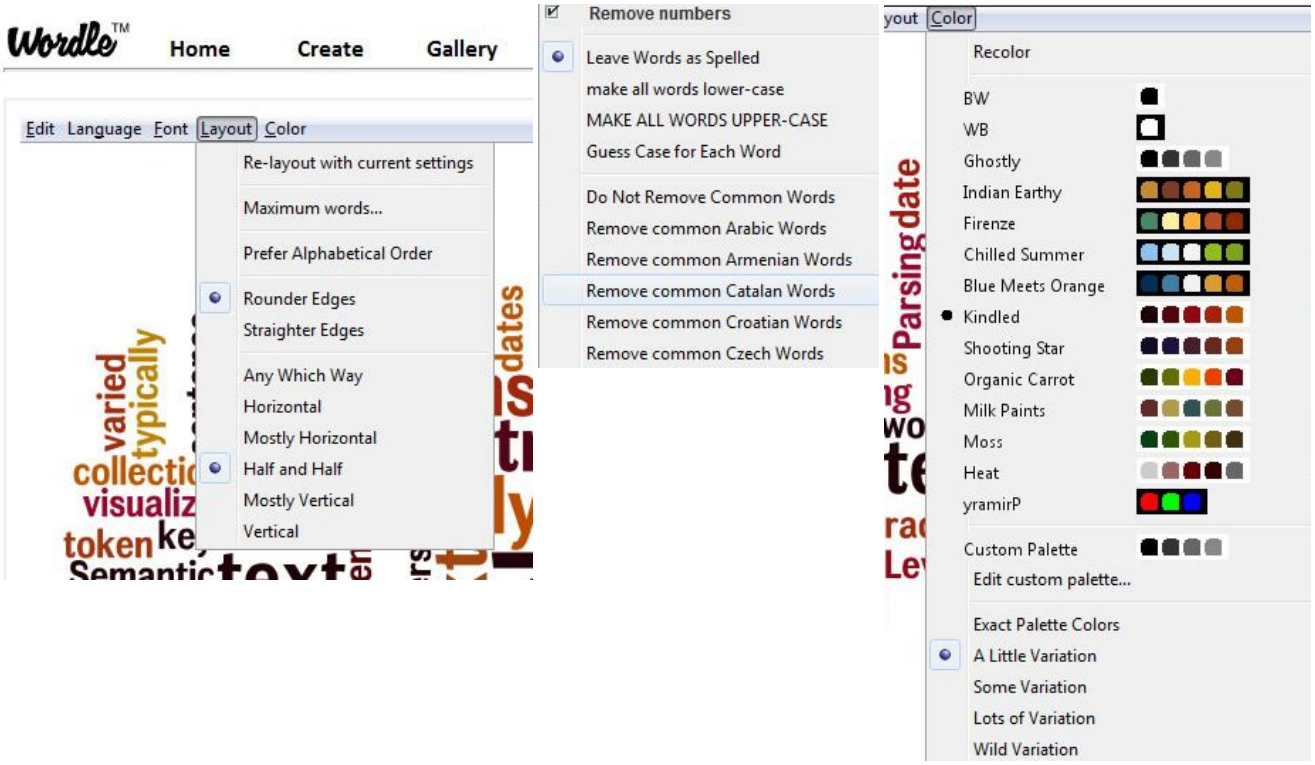


created at TagCrowd.com

Wordle

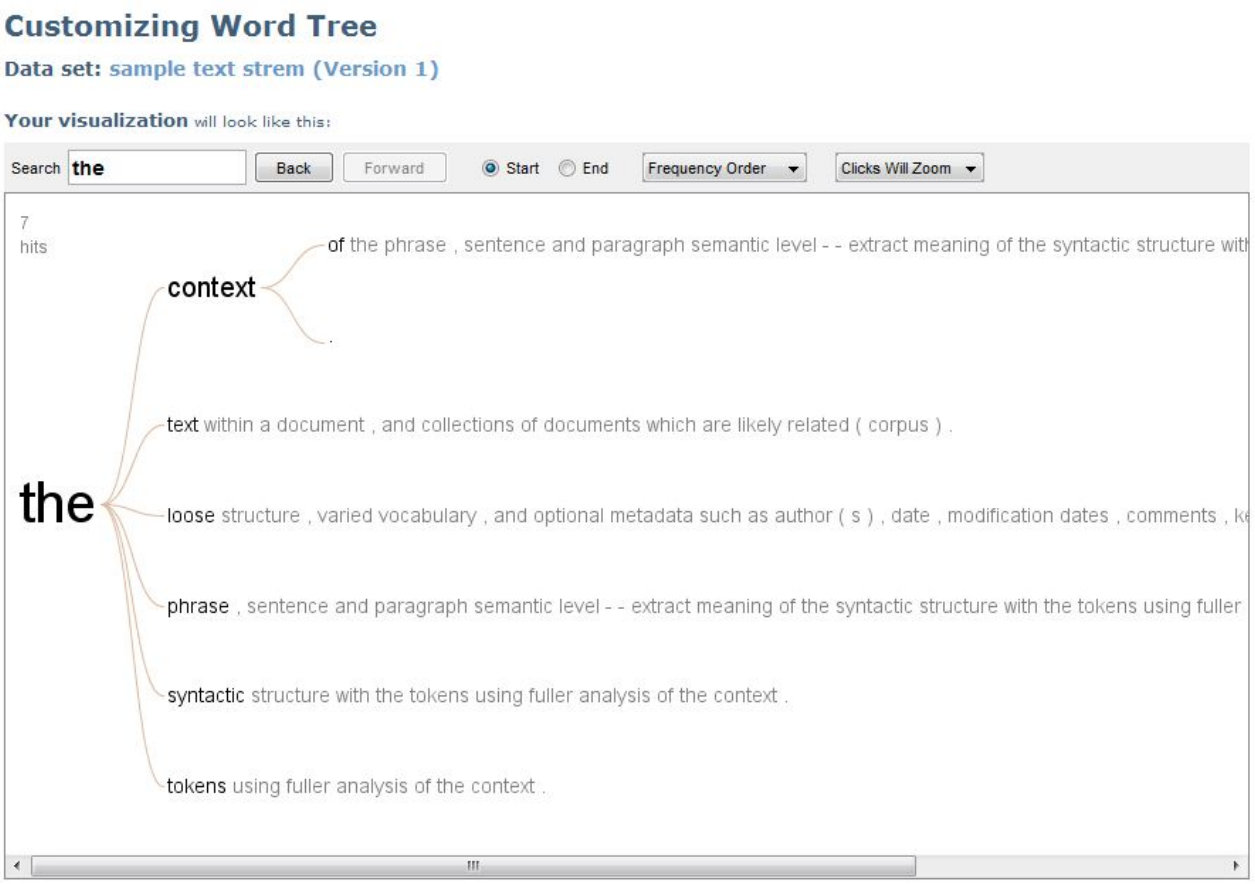
Below is a visualization by http://wordle.net with size based on frequency. You have to download an app for Windows or MacOS from the web site.

Some layout, language and color options available in Wordle



Word Tree

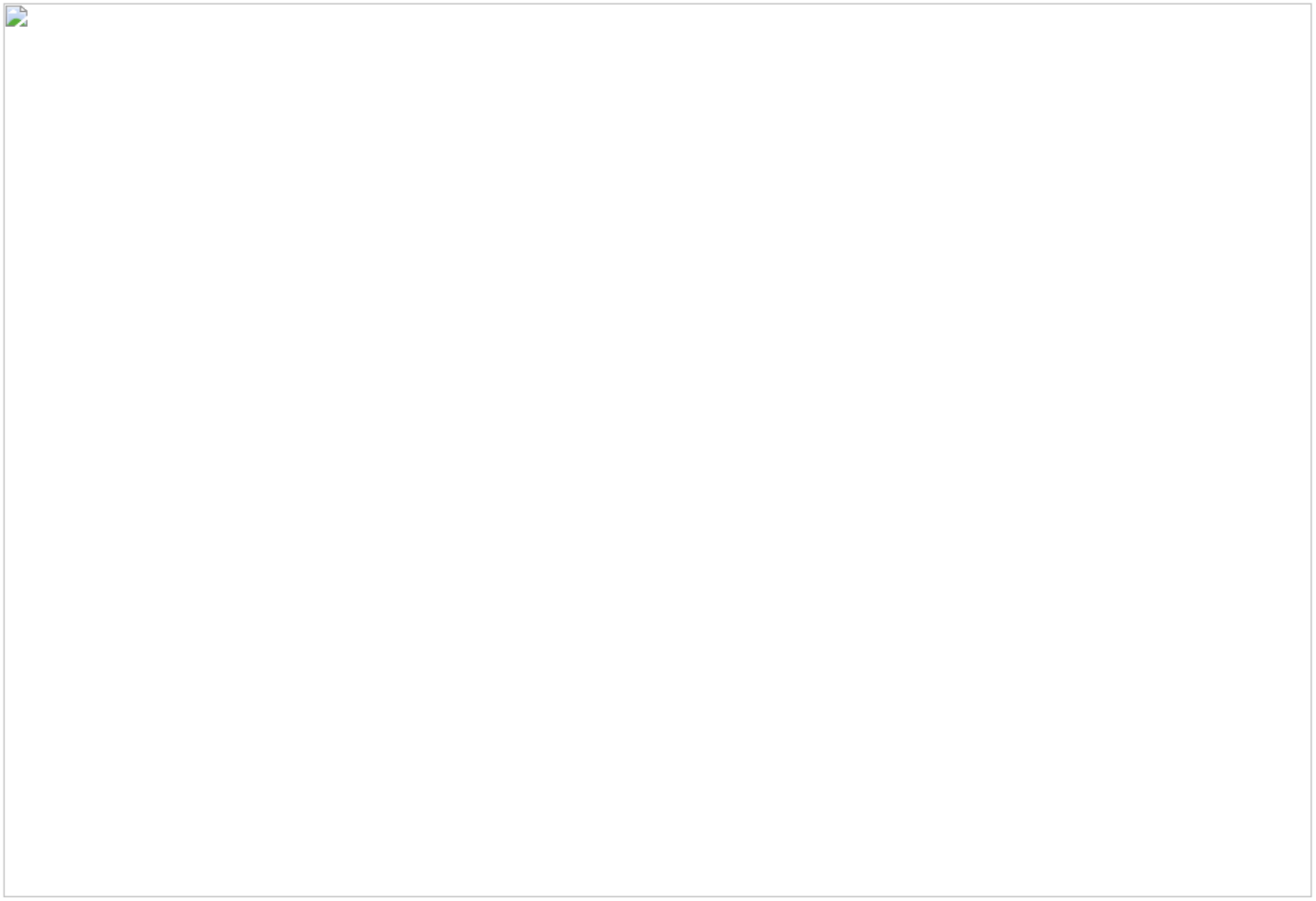
available through https://www.jasondavies.com/wordtree/. Enter a word and then interactively choose next words



TextArc

relates words by connecting them in a graph

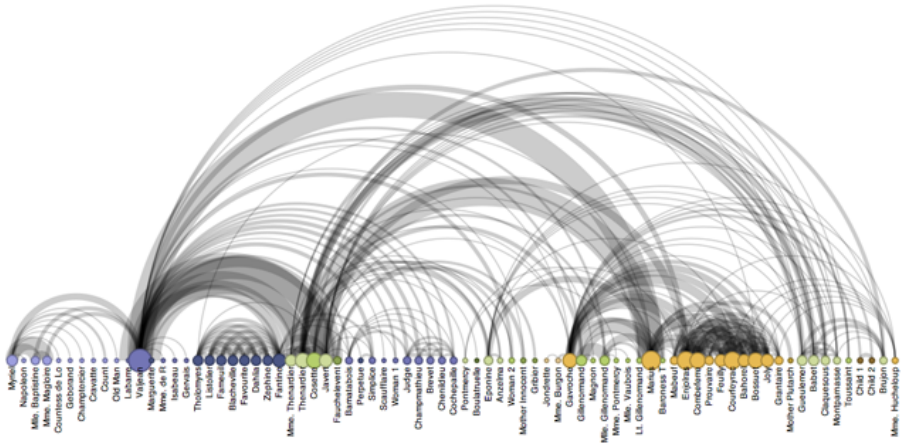
http://textarc.org/Stills.html shows some static example of their classic display of *Alice's Adventures in Wonderland*



Arc Diagrams

Visualization of repeated text sequences or even music sequences.

This arc diagram shows the character interactions in Les Miserables.



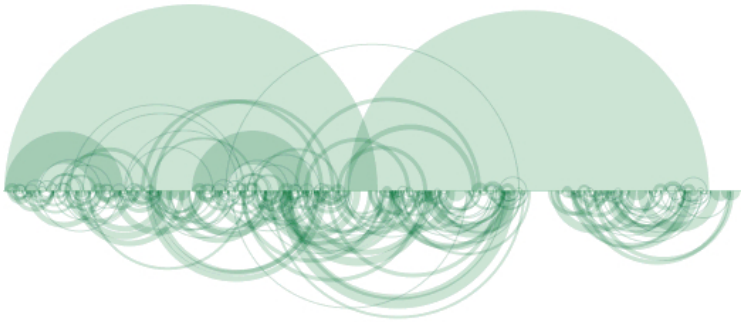
http://mbostock.github.io/protovis/ex/arc.html is the website for this diagram through Protovis/D3.

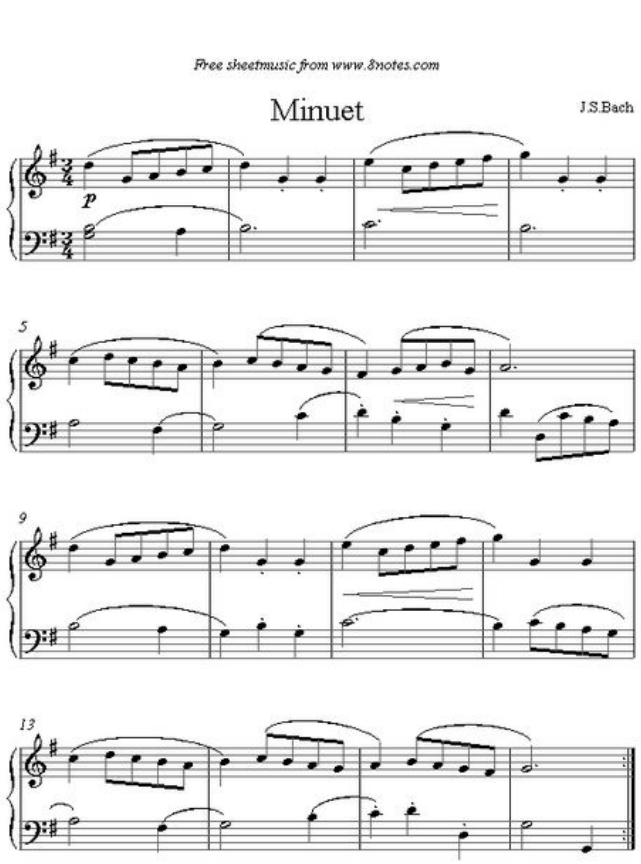
The input data is in http://mbostock.github.io/protovis/ex/miserables.js

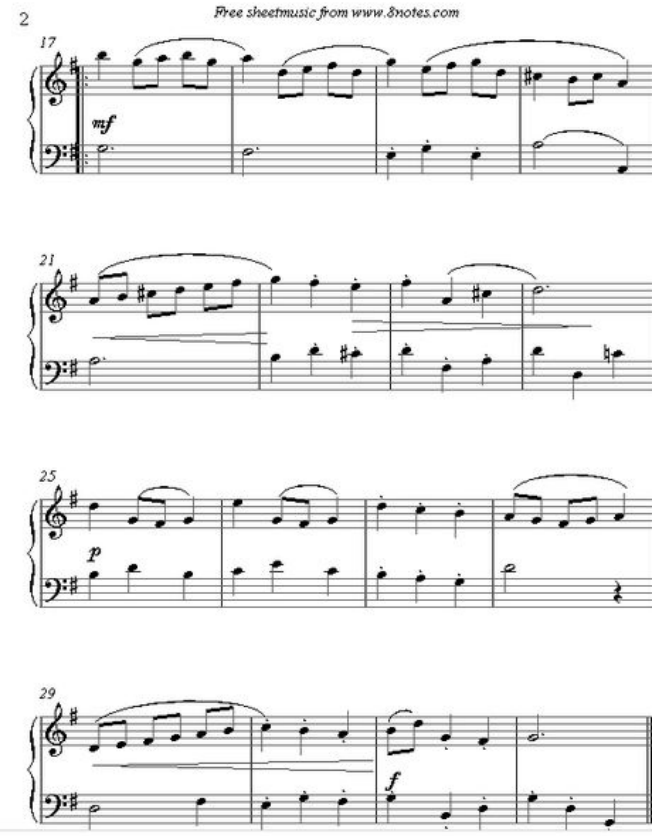
---

Music theme visualization example

Repeating sequences are connected via arcs:



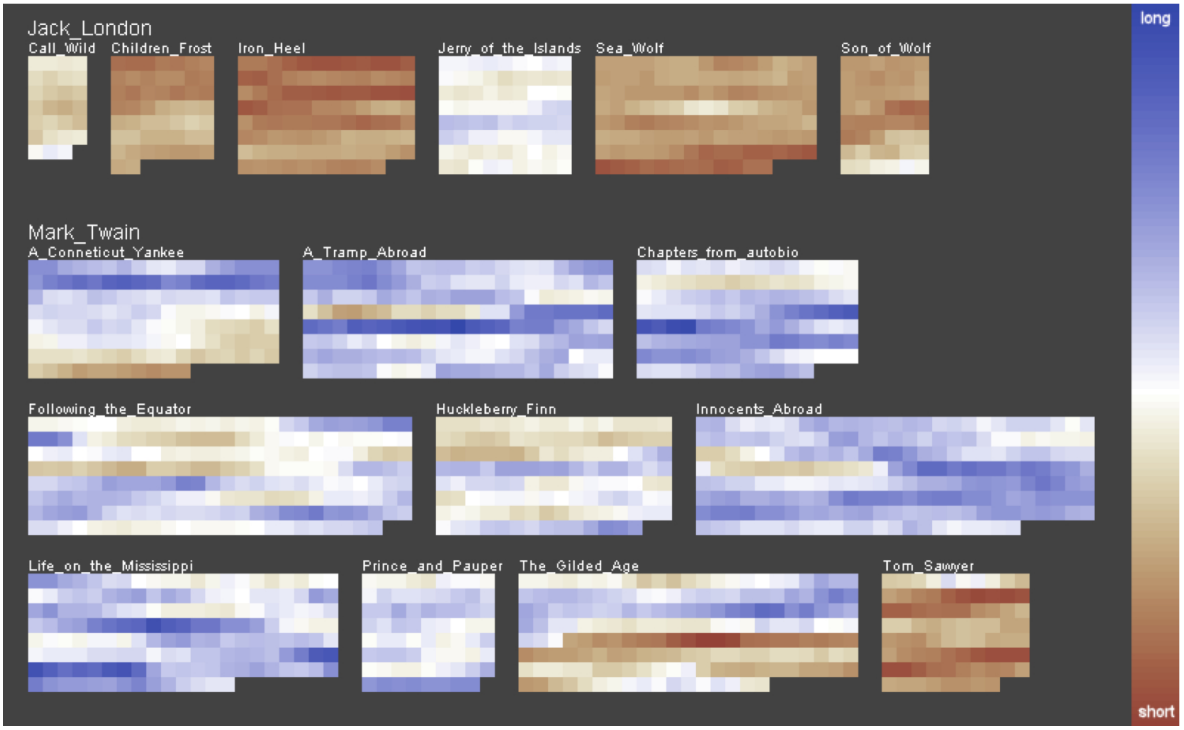




Literature fingerprinting

Here we look at n-word-grams to match patterns of the author.

Color is sentence length.

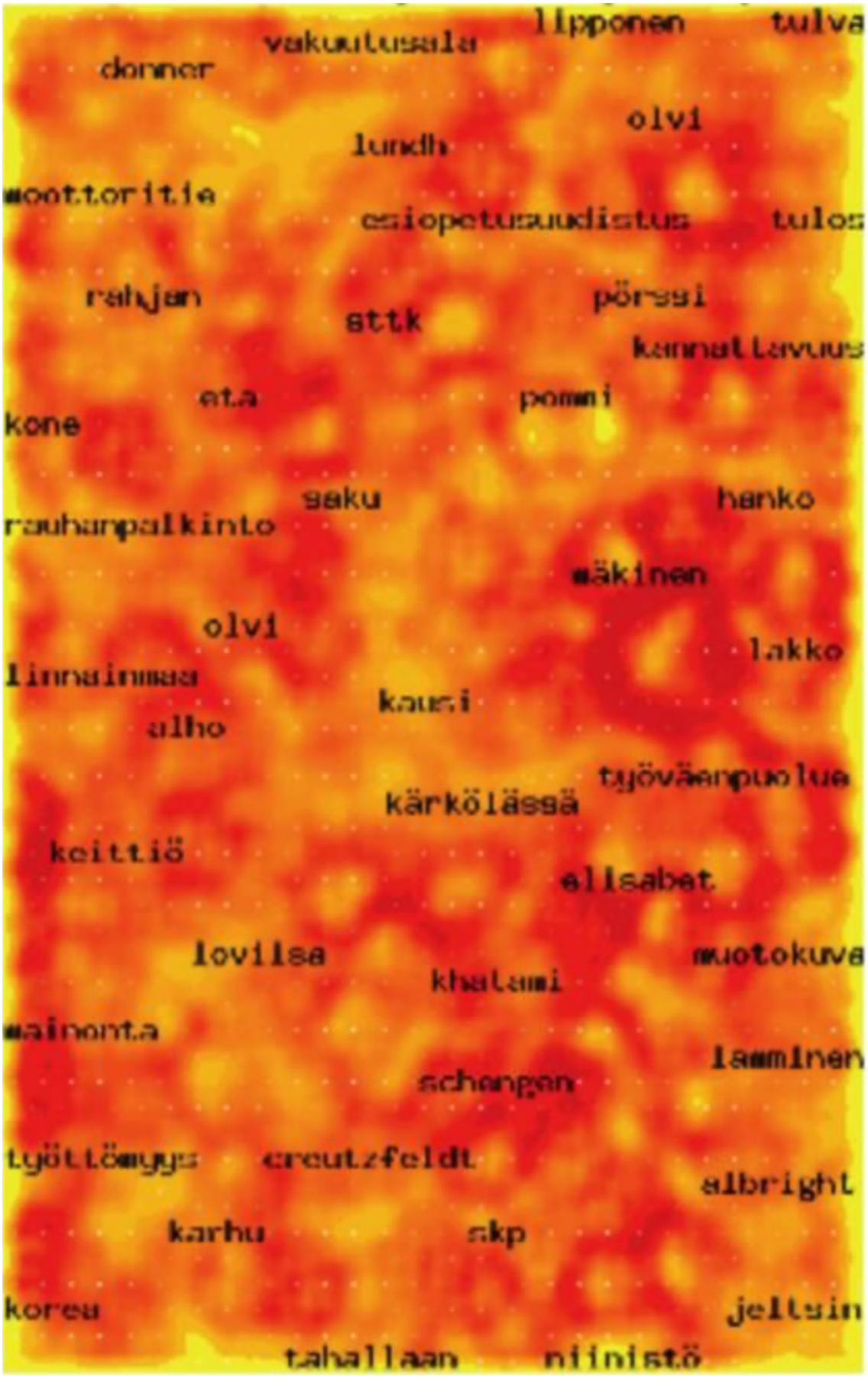


Document Collection Visualizations

Goal is to place similar documents close together.

- graph spring layouts,
- multi-dimensional scaling
- clustering (K-means, hierarchical)
- self-organizing maps

Self-organizing maps -- use the vectors from each document to calculate distances from each other. Higher weights draw the documents closer together. Randomly start with one document.



Themescape using height in 3D to represent frequency of themes

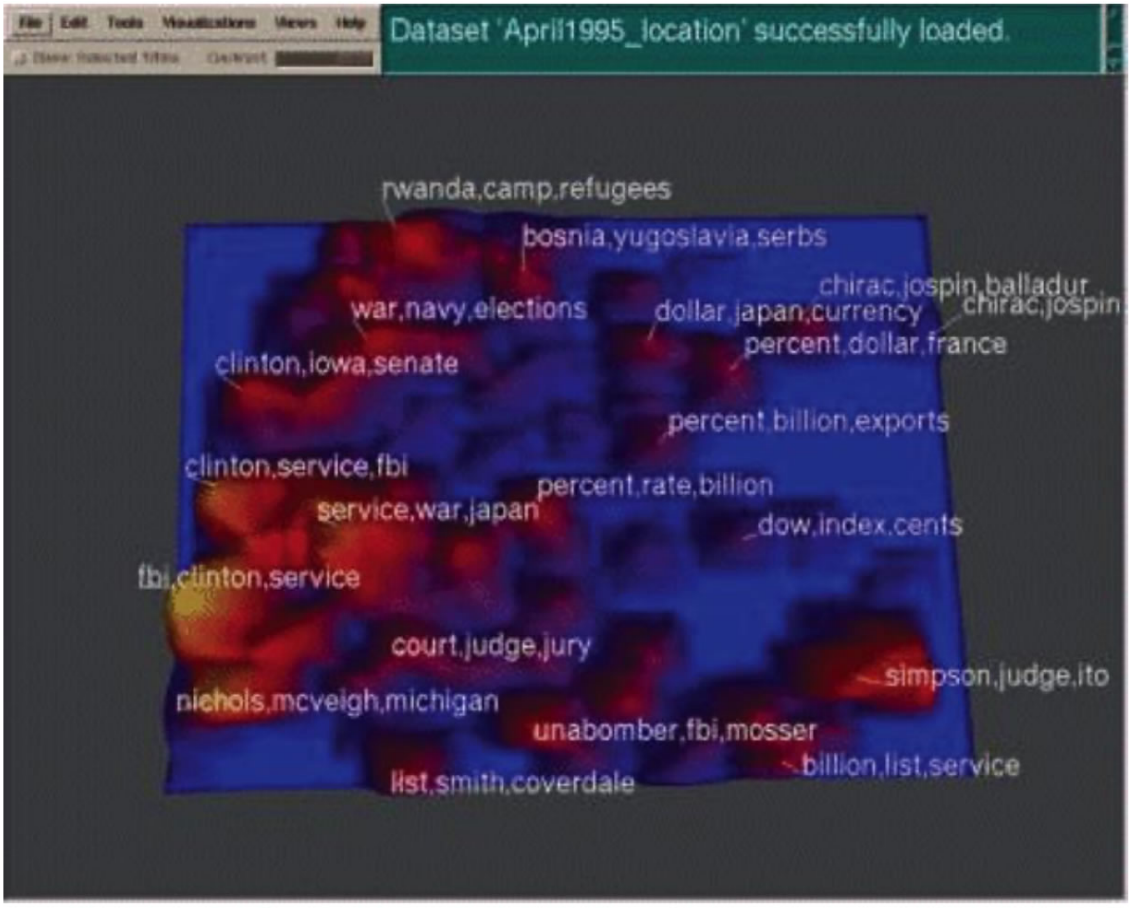
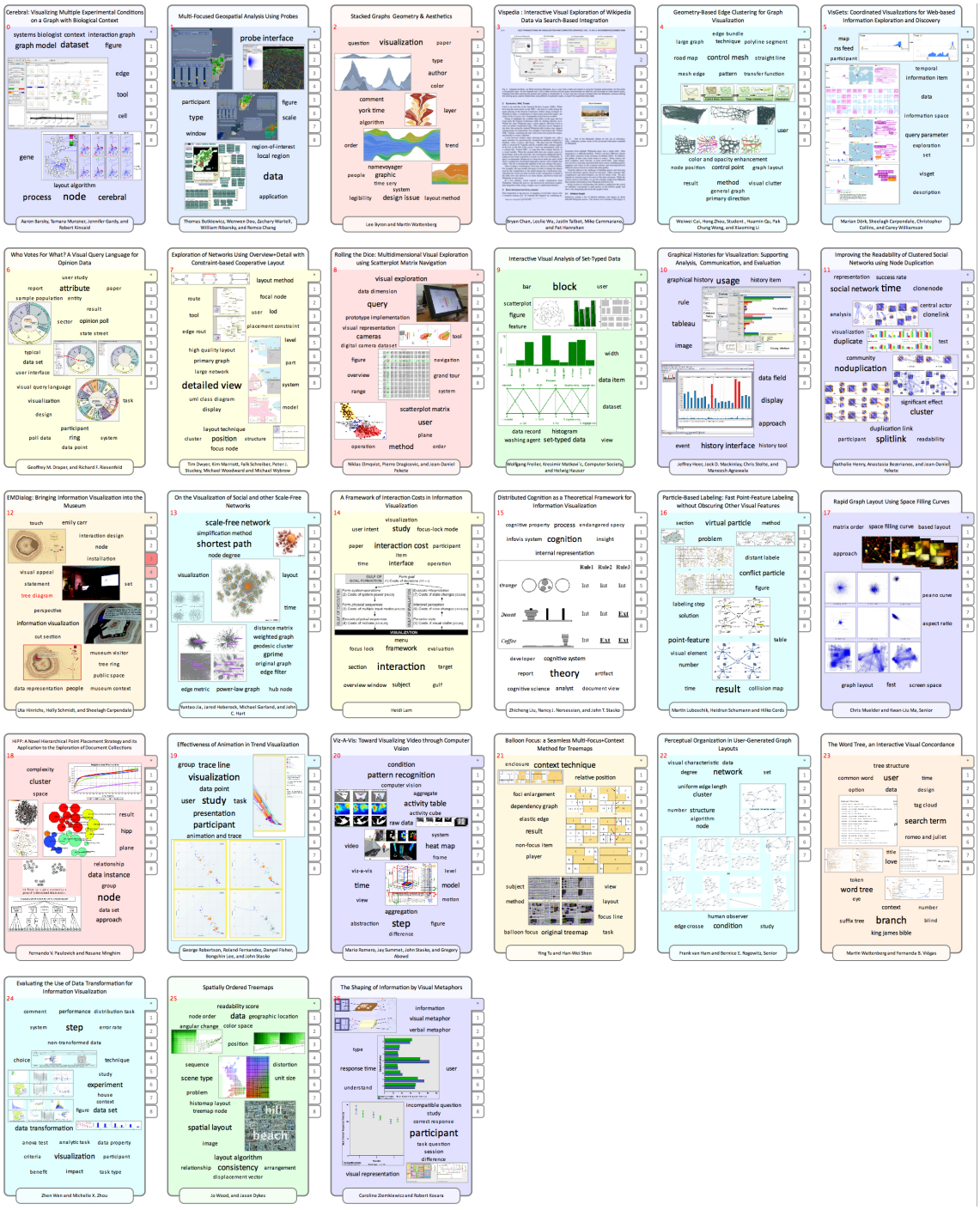


Figure 10.11. A themescape from PNNL that uses height to represent the frequency of themes in news articles.

Document cards represent the document's key semantics as a mixture of images and key terms. This works for academic documents where this metadata is available.

Example Figure 10.12



Software visualization

Figure 10.13. The SeeSoft software visualization. Rectangles represent source code files. The sizes of the rectangles in each column correspond to the length of the source code file, and the color of each line represents parameters related to modification.

TileBars search result

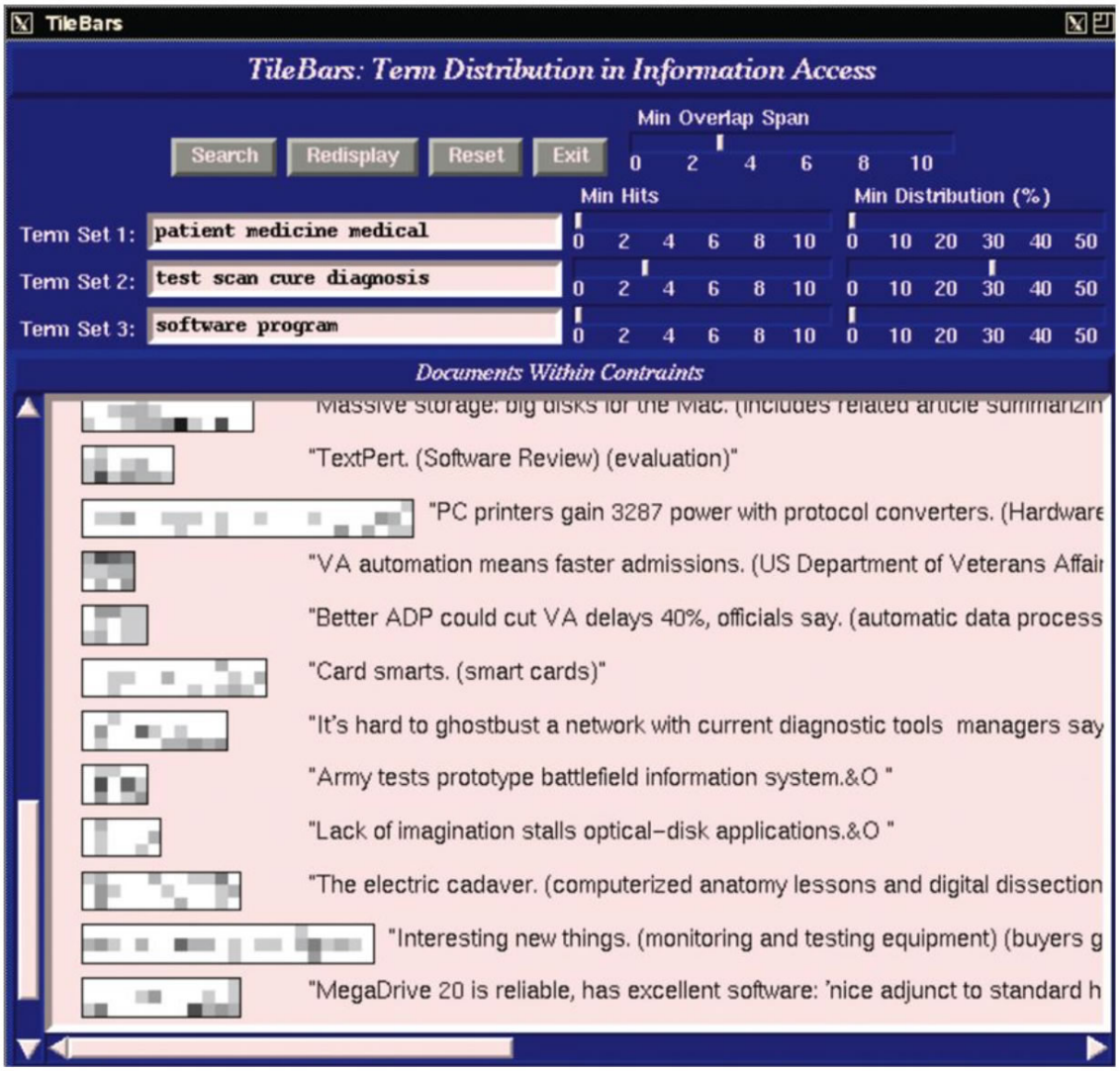


Figure 10.14. The TileBars query result visualization. Each large rectangle indicates a document, and each square within the document represents a text segment. The darker the tile, the more frequent the query term set.

Stream graph for theme visualizations

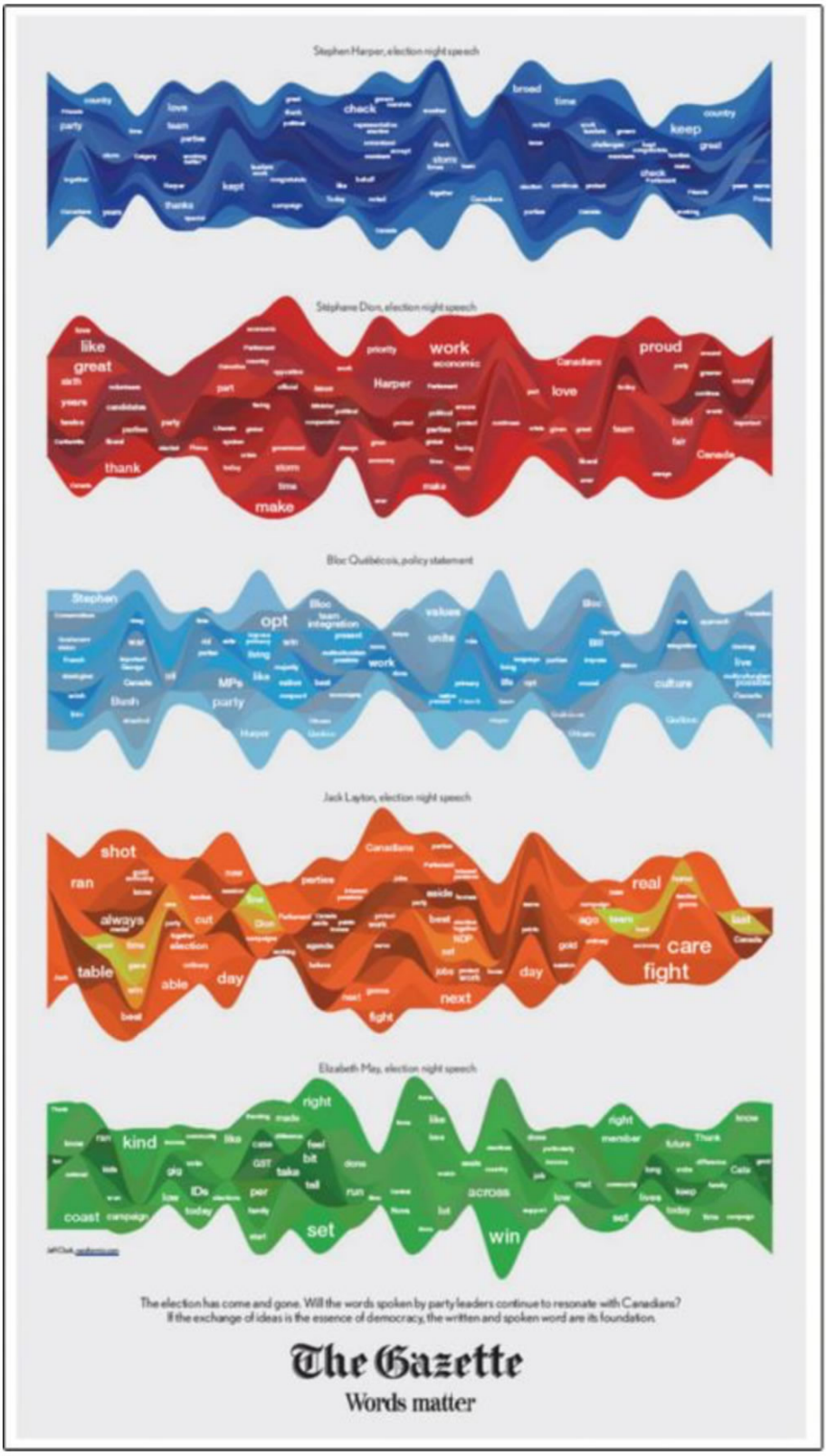


Figure 10.15. A stream graph (ThemeRiver), depicting the election night speeches of several different candidates for a Canadian election.