Exploring Uncommon - Advanced Visualization Charts: Techniques, Advantages, and Applications

Created by – Sachin Hawaldar –
Follow me - https://www.linkedin.com/in/contact-sachin-hawaldar/

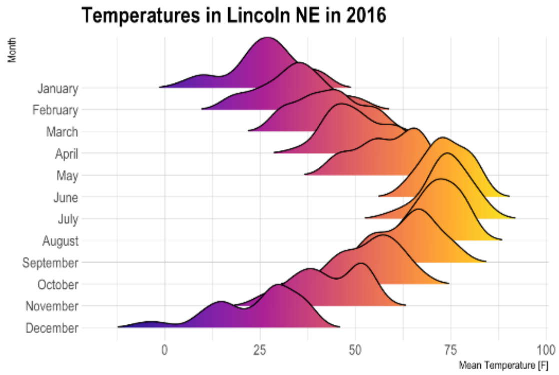
Data visualization serves as a powerful tool in transforming raw data into actionable insights, enabling users to comprehend complex patterns and relationships. While traditional charts like bar graphs and scatter plots are widely used, advanced visualization charts offer deeper insights into intricate datasets. In this article, we'll delve into ten advanced visualization charts, exploring when to use them, their advantages and disadvantages, and how to implement them using R and Python.

1) Ridge Line Plots:

When to Use It:

- Ridge line plots are effective for visualizing the distribution of a continuous variable across multiple categories.
- Useful for comparing distributions while preserving the overall shape and trend.

- Advantages: Facilitates comparison of distributions, especially when overlapping.
- Disadvantages: May obscure small differences between distributions.



How to Use in R and Python:
- R: Utilize the `ggridges` package to create ridge line plots.
- Python: Employ libraries like `seaborn` to generate ridge plots.

Example Scenario:

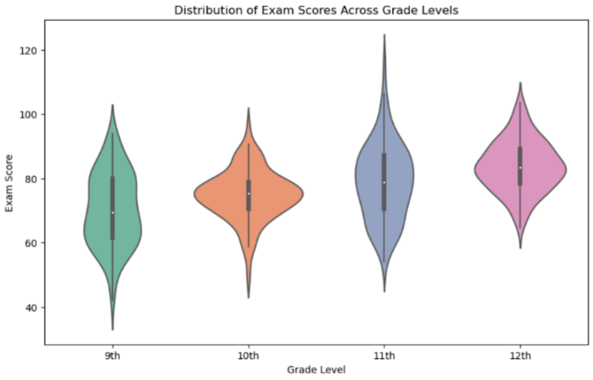
- Use ridge line plots to visualize the distribution of temperature across different months, allowing comparison of temperature trends over time.

2) Violin Plots:

When to Use It:

- Violin plots are suitable for displaying the distribution of a continuous variable. It is combination of Box Plot and Density plots. – Box plot does not provide density whereas density plots do not provide Min, Max and quartiles.
- Effective in comparing distributions and detecting outliers.
- **Advantages:** Provides insights into distribution shape and spread, in addition to summary statistics.
- Disadvantages: May not be ideal for large datasets or precise comparison of multiple groups.

How to Use in R and Python:
- R: Use the `vioplot` function or libraries like `ggplot2` to create violin plots.
- Python: Utilize the `seaborn` library to generate violin plots.



Example Scenario:

- Use violin plots to visualize the distribution of exam scores levels across grade levels, aiding in identifying discrepancies in exam performance.

In this example, we use violin plots to show the distribution of exam scores across grade levels (9th, 10th, 11th, and 12th). Each grade level is represented by a violin plot, which allows us to compare the distribution of exam scores among students across grades. This visualization can assist uncover discrepancies in exam performance and provide insight into the overall distribution of exam scores within each grade level.

3) Heat Plots:

When to Use It:

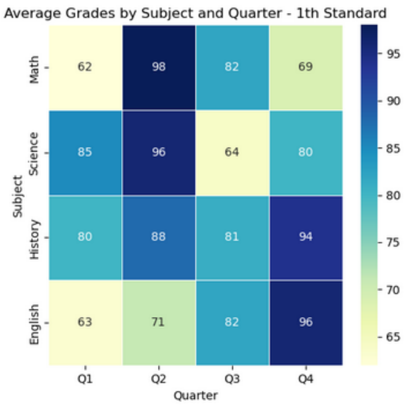
- Heat plots are effective for visualizing the correlation or relationship between multiple variables.
- Useful for identifying patterns or clusters within large datasets.

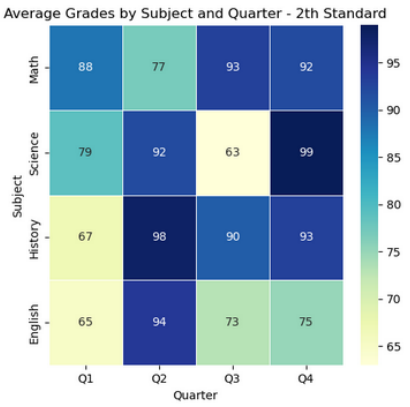
- Heatmaps are used to illustrate the relationships between two variables, one on each axis. Observing how cell colors change across each axis allows you to see whether there are any patterns in value for one or both variables.
The variables shown on each axis can be of any form, including categories labels and numerical values. In the latter situation, the numeric value must be binned, as in a histogram, to create grid cells for plotting colors associated with the main variable of interest.

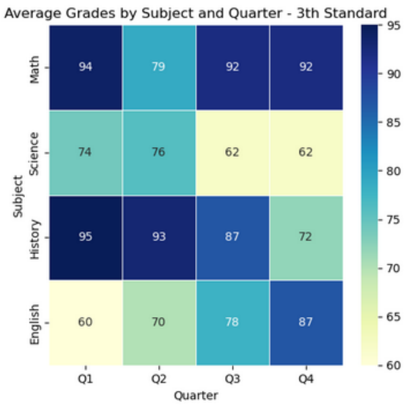
- **Advantages**: Clearly visualizes patterns in large datasets, facilitating interpretation.
- **Disadvantages**: May become cluttered or less effective with highly correlated variables.
How to Use in R and Python:
- R: Utilize functions like `heatmap` from the `stats` package or `geom_tile` from `ggplot2`.
- Python: Use libraries such as `seaborn` and `matplotlib` to create heatmaps.

Example Scenario:

- Create a heat map to show performance of students across various subjects throughout a school year for 3 different classes. We want to create a heat map to visualize the average grades obtained by students in each subject over different quarters of the year.

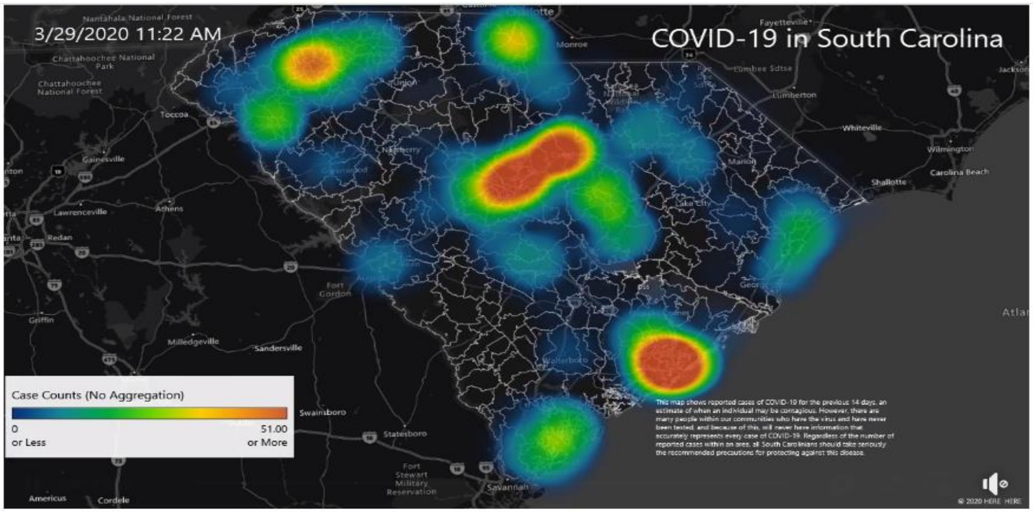






Density Heat map –

Heat map to visually represent the spread of the Covid, now this is taken in time snap, but we can do rolling Heat map like using Timeline.



This map shows reported cases of COVID-19 for the previous 14 days, an estimate of when an individual may be contagious. However, there are many people within our communities who have the virus and have never been tested, and because of this, will never have information that accurately represents every case of COVID-19. Regardless of the number of reported cases within an area, all South Carolinians should take seriously the recommended precautions for protecting against this disease.

COVID-19: Heat Map (April 2, 2020)

4) Sparkline Plots:

This is easy to use visualization you all might have done this in excel.

When to Use It:

- Sparkline plots are useful for visualizing trends or changes in data over time, especially within constrained spaces.
- Ideal for displaying trends or patterns within a single data series.

- Advantages: Condenses data into a compact format suitable for dashboards or reports.

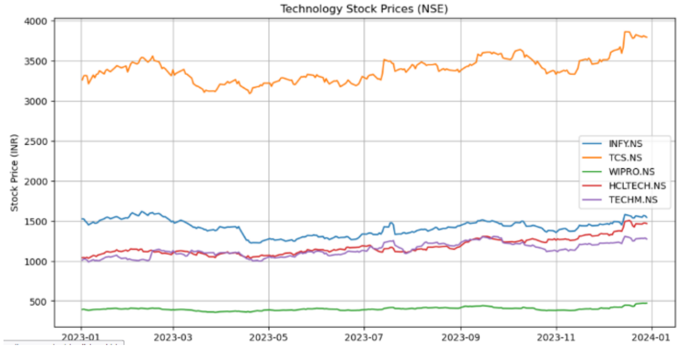
- Disadvantages: May lack context without accompanying information.

How to Use in R and Python:

- R: Utilize the `sparkline` package to create sparkline plots.
- Python: Implement sparkline plots using libraries like `pysparklines`.
- You can also directly create this in Excel, PowerBI and Tableau

Example Scenario:

- Sparkline plots to show performance of tech companies from NSE.



5) Rainbow Plots:

When to Use It:

- Rainbow plots are effective for visualizing multivariate data with sequential or ordinal variables.
- Useful for identifying patterns or trends across multiple variables simultaneously.

- Advantages: Provides a visually appealing representation of multivariate data.

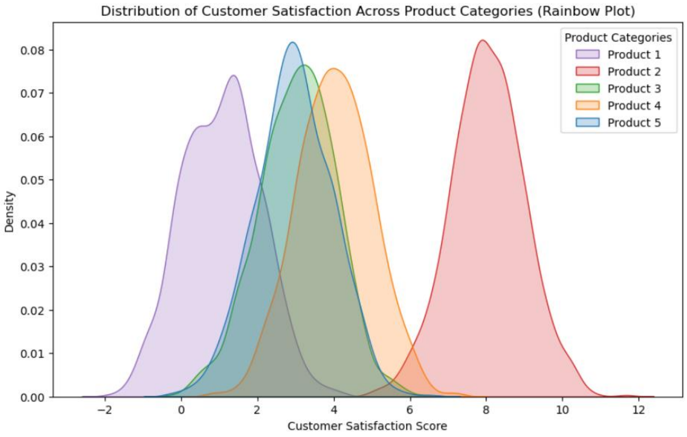
- Disadvantages: May not be suitable for datasets with too many variables or categories.

How to Use in R and Python:

- R: Implement rainbow plots using functions from the `rainbow` package.
- Python: Customize existing plotting libraries to create rainbow-like effects.

Example Scenario:

- Utilize rainbow plots to visualize the distribution of customer satisfaction scores across different product categories, aiding in identifying areas for improvement.



6) Table Plots:

When to Use It:

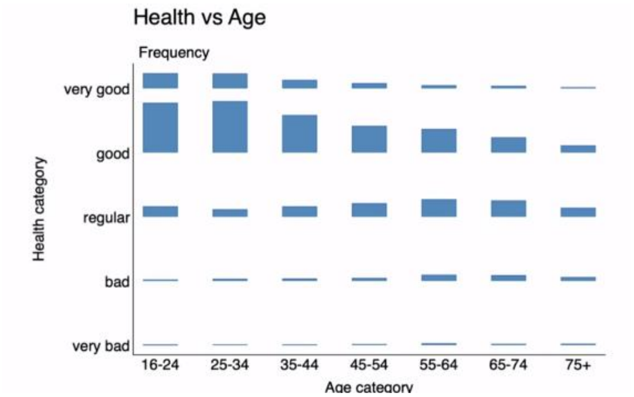
- Table plots are useful for displaying categorical data in tabular form, with additional visual elements.
- Effective for summarizing data and facilitating comparison between categories.

Great for presentations or reports with too many large tables.

- **Advantages:** Presents data in a structured format, allowing for easy comparison.
- **Disadvantages:** May not be suitable for conveying trends or patterns as effectively as other chart types.

How to Use in R and Python:

- R: Utilize functions like `grid.table` from the `gridExtra` package or `tableGrob` from `grid`.
- Python: Utilize the `pandas` library to create table-like visualizations.



Example Scenario:

- Use table plots to summarize sales performance by region, providing a comprehensive overview of revenue and profit margins.

7) Mosaic Plots:

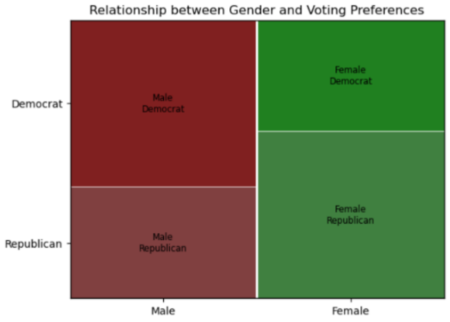
When to Use It:

- Mosaic plots are effective for visualizing the relationship between two categorical variables.
- Useful for identifying patterns or associations within contingency tables.

- Advantages: Clearly illustrates associations between categorical variables.
- Disadvantages: May not be suitable for large datasets with many categories.

How to Use in R and Python:

- R: Create mosaic plots using the `vcd` package.
- Python: Implement mosaic plots using the `statsmodels.graphics.mosaicplot` module.



Example Scenario:

- Employ mosaic plots to visualize the relationship between gender and voting preferences in an election survey, helping identify demographic voting trends.

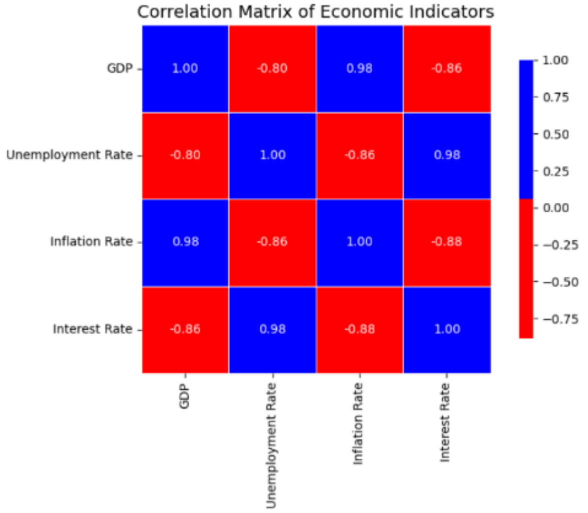
Note- Mosaic plots are different than tree maps –

Feature	Mosaic Plot	Tree map
Purpose	Visualize the relationship between two or more categorical variables.	Visualize hierarchical data structures.
Display	Rectangular plot with categories represented by rectangles.	Nested rectangles represent hierarchical relationships.
Content	Proportions of categories within one variable, broken down by levels of another variable.	Categories or nodes within a hierarchy, with sizes corresponding to quantitative values.
Focus	Association between categorical variables.	Hierarchical structure and relative sizes of categories within the hierarchy.
Example	Relationship between age groups and transportation preferences.	File directory structure on a computer.

8) Matrix Plots:

When to Use It:

- Matrix plots are useful for visualizing relationships or patterns within a matrix or grid-like structure.
- Effective for displaying pairwise relationships between variables.



- **Advantages:** Provides a comprehensive overview of relationships within a dataset.
- **Disadvantages:** May become cluttered with too many variables or categories.

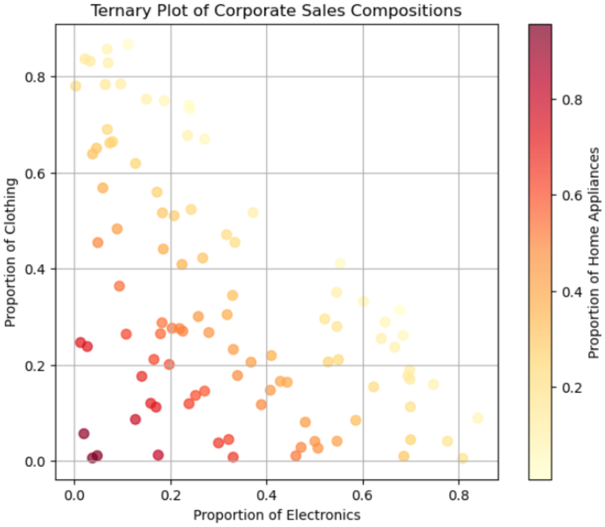
How to Use in R and Python:
- R: Utilize functions like `image` from the `graphics` package or `geom_tile` from `ggplot2`.
- Python: Both `matplotlib` and `seaborn` libraries offer support for creating matrix plots.

Example Scenario:

- Create a matrix plot to visualize the correlation between different economic indicators, aiding in understanding the interdependencies within the economy.

9) Ternary Plots:

Ternary plots, also known as ternary diagrams or Gibbs triangles, are specialized charts used to visualize compositional data consisting of three variables that sum to a constant. These plots are particularly useful for representing the relationships between three components of a whole, such as the composition of chemical compounds, mineral compositions, or proportions of different phases in a mixture.



When to Use It:

- Ternary plots are effective for visualizing compositional data with three parts

- **Advantages:** Visualizing three components simultaneously, simplicity of representation,

- **Disadvantages:** Limited to only three components, Complex interpretation for beginners,

Example –

we want to visualize the proportions of three different product categories (e.g., electronics, clothing, and home appliances) in the overall sales portfolio.