# Designing Data Visualizations

Sean Hellingman ©

Data Visualization and Manipulation through Scripting (ADSC1010)

*shellingman@tru.ca*

Fall 2024

THOMPSON RIVERS UNIVERSITY

# Topics


2. Introduction
3. Purpose
4. Layouts
5. Encodings
6. Expressive Data Displays
7. Enhancing Aesthetics
8. Exercises and References

# Introduction

- Data visualization allows you to reveal patterns in your data and communicate insights.
- We will be expanding on our current knowledge of creating visualizations in R.
- Conceptual skills needed to create *effective* and *expressive* visual representations of data.

# Visualization Steps

1. Understanding the *purpose of visualization*.
2. Selecting a *visual layout* based on your question & data type.
3. Choosing the best *graphical encodings* for your variables.
4. Identifying visualizations that are able to *express* your data.
5. Improving the *aesthetics* (readability).

# Purpose of Visualizations

- **The purpose of visualization is to gain insights, not creating pretty pictures.**
- Visualizing your data is an important step in any data science project.
- Creating appropriate visualizations can help to expose previously unseen patterns in data.

# Intro to Example 1

- Anscombe's Quartet is a dataset designed to test your ability to identify differences between pairs of variables.

- Summary:

| Set | Mean X | S.D. X | Mean Y | S.D. Y | Correlation | Linear Fit |
|---|---|---|---|---|---|---|
| 1 | 9.00 | 3.32 | 7.50 | 2.03 | 0.82 | $y = 3 + 0.5x$ |
| 2 | 9.00 | 3.32 | 7.50 | 2.03 | 0.82 | $y = 3 + 0.5x$ |
| 3 | 9.00 | 3.32 | 7.50 | 2.03 | 0.82 | $y = 3 + 0.5x$ |
| 4 | 9.00 | 3.32 | 7.50 | 2.03 | 0.82 | $y = 3 + 0.5x$ |

# Example 1

- Load the *anscombe* dataset into R.
- Use `par(mfrow=c(2, 2))` to simultaneously generate four plots.
- Create a scatter plot for each of the pairs of anscombe variables.
- What do you notice?

# Selecting Visual Layouts

- Selecting a visual layout may be thought of as an optimization problem.
  - We want to select the best possible visualization subject to a set of constraints.

- Constraints:
  1. The specific *question of interest* you are attempting to answer (within a specific domain).
  2. **The *type of data* you have available to answer your question.**
  3. The limitations of the human *visual processing system*.
  4. The *spatial limitations* in the medium you are using (screen size or available pixels).

- We will discuss selecting visual layouts based on the available data (*nominal*, *ordinal*, or *continuous*).

# Visualizing a Single Variable

- Before we start examining how variables interact with each other, we should understand how each variable is distributed.

- The specific layout will depend on if your variable is **categorical** or **continuous**.

- Continuous variables:
  - Histograms
  - Boxplots
  - Violin plots

- Categorical:
  - Bar charts
  - Proportional representations

# Proportional Representations

- We may be interested in showing each value relative to the total of the variable.
- *Which proportions of outcomes are attributable to each category?*
- Visualizations:
  - Stacked bar chart
  - Pie chart
  - Treemap (hierarchical data)

# Example 2

- Load the *Employment.csv* dataset into R.
  - *This dataset comes from Statistics Canada and gives estimated counts on the number of people employed in each sector (2018 - 2022).*
- Use the provided code to generate the following:
  - Bar chart of employees by sector (Year = 2022)
  - Stacked bar chart of employees by sector (Year = 2022)
  - Pie chart of employees by sector (Year = 2022)

# Visualizing Multiple Variables

- After exploring each variable individually, we can assess possible relationships between variables.
- The specific layout will again depend on if your variables are **categorical** or **continuous**.
- Two continuous variables:
  - Scatterplot
  - Scatterplot matrix (all continuous variables)
- One categorical and one continuous variable:
  - Faceting (show distributions of each category)
  - Boxplot/violin plot for each category
- Two categorical variables:
  - Cross-tabulation
  - Heatmap

# Example 3

- Load the *Football22.csv* dataset into R.
- Use the example code to generate the following:
  - Boxplots for each league generated for the *Goals_For* variable.
  - Histograms for each league generated for the *Goals_For* variable.
- Run the example code to see what a cross-tabulation and what a basic heatmap look like and how they are made in R.

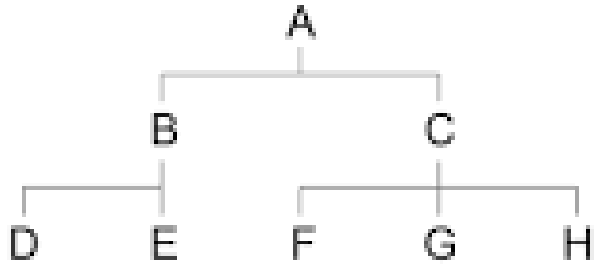
# Visualizing Hierarchical Data

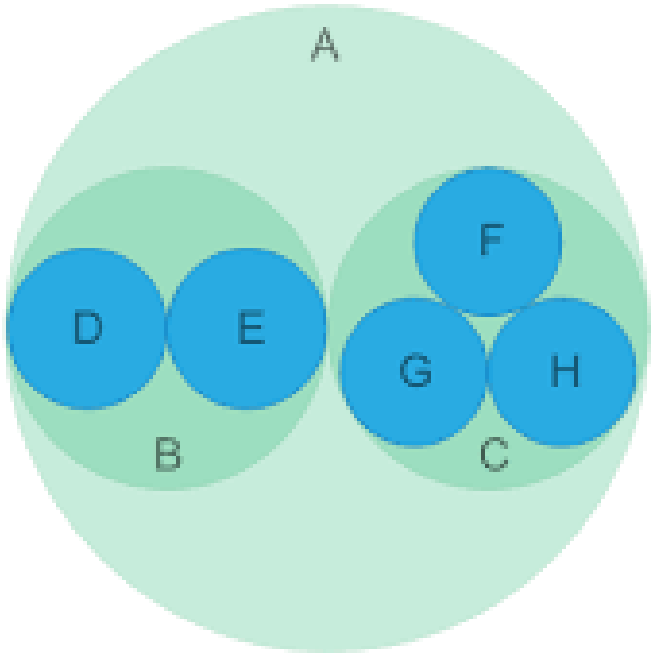
- It may be difficult to show that a hierarchy exists in your data.
- If the data contains a natural **nested structure** expressing this hierarchy can be important to your analysis.
- Example: Regions within regions.
- Visualizations:
  - Treemaps
  - Circle packing
  - Sunburst diagrams

# Example 4

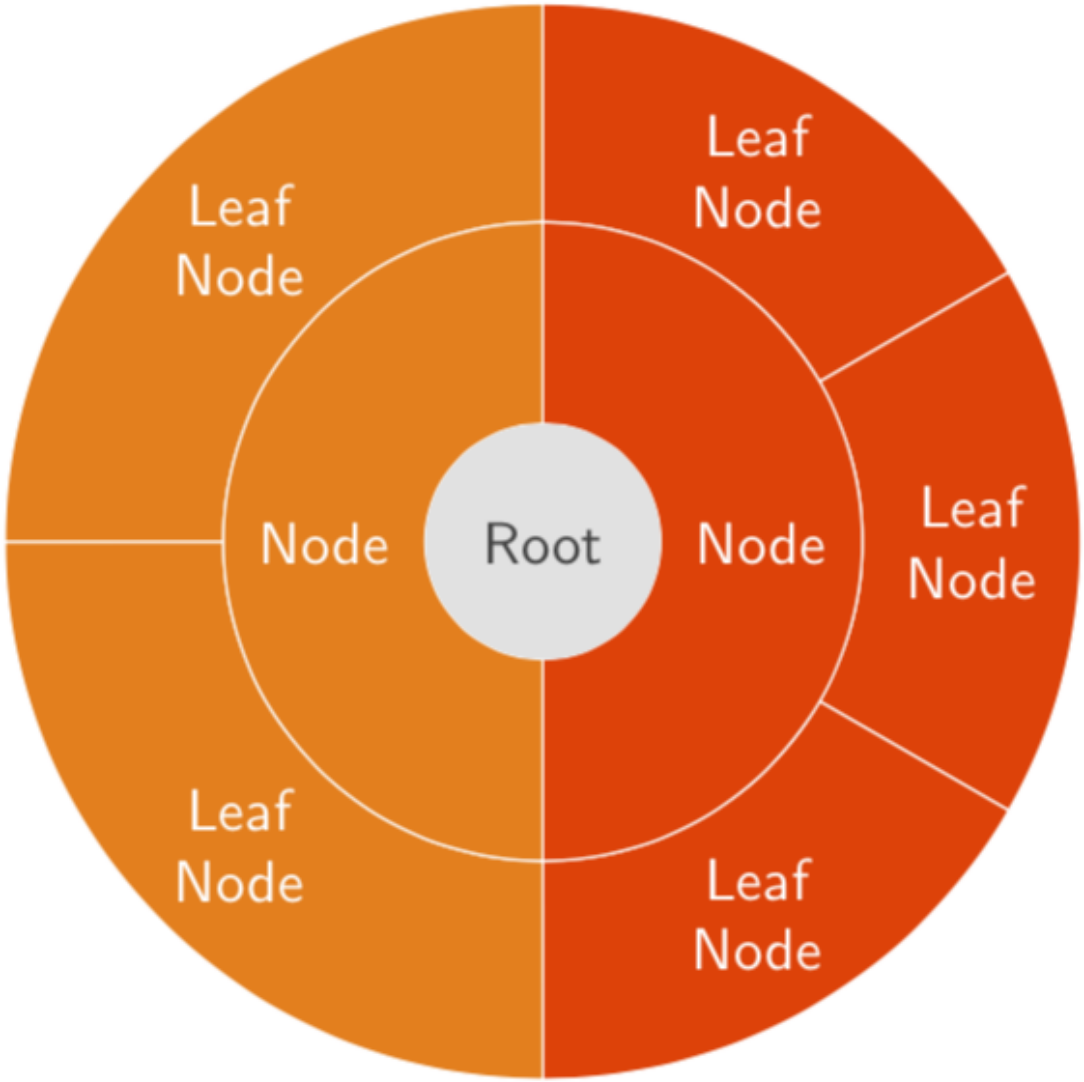
- Load the *Population.csv* dataset into R.
  - *This dataset gives quarterly population estimates for each Canadian province and Territory. It also contains information about the Region of Canada that the province is located in.*
- Use the example code to generate the following:
  - A treemap of the estimated populations from Q1_2023.
- What information can you get from a graphic like this?

# Circle Packing



# Sunburst Diagram



# Effective Graphical Encodings I

- There are often multiple ways to represent the same set of data.
- Representing data in another visual format is called **encoding** that data.
- This should be done in a way such that the representations are easily understood (*decoded*) by users.

# Effective Graphical Encodings II

- You should choose the **graphical encodings** that are most accurately decoded by your audience.

- The accuracy of perceptions is called the **effectiveness** of graphical encoding.

- Common set of possible encodings listed from most effective to least effective:
  1. **Position:** The horizontal or vertical position of an element along a common scale.
  2. **Length:** The length of a segment (stacked bar chart).
  3. **Area:** The area of an element (circle or rectangle) typically used in a bubble chart (scatter plot with differently sized markers) or a tree map.
  4. **Angle:** The rotational angle of each marker (circular layouts like pie charts).
  5. **Colour:** The colour of each marker, usually along a continuous colour scale.
  6. **Volume:** The volume of a three-dimensional shape (3D bar chart).

# Example 5

- Use the *Population.csv* dataset in R to examine four different kinds of encodings.

- **You should always start by encoding the most important data features with the most accurately decoded visual features.**

# Colours

- One possible conceptualization of colour spaces is the **Hue-Saturation-Lightness** (HSL) model.
- **Hue:** How we think of describing a colour (green or blue).
- **Saturation:** Intensity of a colour; describes how *rich* the colour is on a linear scale between grey (0%) to full display of the hue (100%).
- **Lightness:** Describes how *bright* the colour is on a linear scale from black (0%) to white (100%).

# Selecting Colours

- The data type of your variable will drive your decisions.
- For **categorical** variables, use a colour encoding to distinguish between groups.
  - Select colours with different hues that are visually distinct and do not imply a rank ordering.
- For **continuous** variables, use a colour encoding that helps with estimating values.
  - Colours should be chosen using a linear interpolation between colour points (different lightness values).

# Colour Palettes

- There are colour palettes that already exist in R and we are able to use them in our own visualizations.

- Three different of **continuous colour scales**:
  1. **Sequential:** Often best for displaying continuous values along a linear scale.
  2. **Diverging:** Most appropriate when the divergence from a centre value is meaningful (midpoint is zero). Example: Population changes over time.
  3. **Multi-hue:** Allow for increased contrast between colours by providing a broader colour range (users may misinterpret the differences in hue if the scale is not carefully chosen).
  4. **Black and white:** Equivalent to sequential colour scales but the hue is grey (may be needed in publications).

- **Example:** Run the R code to examine some of the colour palettes in the *RColorBrewer* package.

# Leveraging Preattentive Attributes

- Sometimes you will want to draw attention to particular observations in your visualizations.
- To make your graphics rapidly understood you can add attributes to observations to draw attention to them.
- **Preattentive processing:** The cognitive work that your brain does without deliberately paying attention to something.

# Example 6

- Run the Example 6 R code to examine one way of drawing attention to specific observations in a plot.

- **There are other ways to draw attention to specific observations.**

# Expressive Data Displays

- You should choose layouts that allow you to *express* as much data as possible.
  - Devise visualizations that express all of (and only) the data in your dataset.

- Sometimes overlapping happens in our data points.
- Solutions:
  - We can adjust the *opacity* of each marker to reveal overlapping data (**See Expressive Example**).
  - We can break the data into different groupings or facets (only showing a subset of the data at a time).

- Sometimes there is a trade-off between the expressiveness and effectiveness of visualizations.
  - We can break the data and create multiple plots, aggregate the data (groups), and change the opacity of our symbols.

# Aesthetics

- We want to make *beautiful* graphics without adding any useless clutter to our visualizations:
  1. **Remove unnecessary encodings.** *Example: If you are creating a bar chart, the bars should only have different colours if that information is not otherwise expressed.*
  2. **Avoid visual effects.** *Any (unnecessary) 3D effects, shading, or distracting formatting should be avoided.*
  3. **Include accurate chart and axis labels.** *Provide a title for your chart, as well as meaningful labels for your axes.*
  4. **Lighten legends/labels.** *Reduce the size or opacity of axis labels. Avoid using striking colours.*

# Example 7

- I asked ChatGPT to generate *a scatterplot that has way too many encodings, visual effects, and poorly labeled axes.*
- The resulting scatterplot is found in the example code.
- Following the guidelines we have covered in this topic, take some time to improve the scatterplot.

# Exercise 1

- Take some time to look for other patterns in the *anscombe* dataset.
  - You may explore possible interactions in different pairs of variables.

# Exercise 2

- Use the ggplot2 package to generate violin plots for variables found in the *Football22.csv* data. Separate the plots by *League*.

# Exercise 3

- We will be taking a few days to learn different methods for generating visualizations in R. I encourage you to look through some of the information provided on the following R packages:
  - lattice
  - ggplot2
  - plotly
  - rbokeh
  - leaflet

# References & Resources

1. Michael Freeman, Joel Ross, *Programming Skills for Data Science: Start Writing Code to Wrangle, Analyze, and Visualize Data with R*, 2019, ISBN-13: 978-0-13-513310-1

- https://r-graph-gallery.com/circle-packing.html
- https://ggplot2.tidyverse.org/
- https://ggplot2.tidyverse.org/reference/geom_point.html
- https://cran.r-project.org/web/packages/vioplot/vioplot.pdf