# Chapter 3: A Grammar for Graphics

- This chapter discusses an interface for data visualization based on the *Grammar of Graphics* that was proposed in the 1990s by Leland Wilkinson (worked for SPSS and Tableau Software).
- A renowned member of the R development team, Hadley Wickham, built the R package `ggplot2`, which is based on Wilkinson's philosophy.
- Wickham also developed the `tidyverse`, which is a collection of R packages including the important packages `ggplot2` and `dplyr`.

# Different Graphics Packages in R

- There are other traditional ways to create graphs in R, using `base` R and the `lattice` package, with are often fine for basic plots and graphs.
- But `ggplot2` uses certain terms and tools that are related to the taxonomy of graphics discussed in Chapter 2.
- It also allows you to build graphs incrementally in a sensible way by adding bits of code.

# Terms in ggplot2

- A *aesthetic* connects a variable to a visual cue.
- A *glyph* (also called a *mark* or a *symbol*) is a plotting symbol that represents one observation.
- A graph can have several aesthetics that can each correspond to different visual cues.
- The foundation for the graphic is a data table:
- Each row in the data table is an observation (also called a *case*), and each case is represented by a *glyph* (mark) on the graph.

# Adding Aesthetics and Changing Scales

- In `ggplot2`, we can create (and assign a name to) a basic plot object with the `ggplot` function.
- Then we can incrementally add aesthetics to the basic plot with other functions, such as `geom_point` and `geom_text`.
- See Figures 3.1, 3.2, 3.3, 3.4 as we gradually add aesthetics (x-y position, color, area).
- The `coord_trans` directive can change the scale of the axes of a graph, such as from linear to logarithmic.
- The `scale`-type functions (such as `scale_x_continuous`, `scale_x_discrete`, `scale_x_log10`, etc.) will do similar operations.
- `scale_color`-type functions can alter the scale of the color palette for the `color` aesthetic.
- *Guides* such as legends or axis labels provide written context for visual cues to aid interpretability.

# Facets and Layers

- *Facets* are multiple graphs (side-by-side or in a grid) showing the same type of plot for several levels of a categorical variable.
- The R function `facet_wrap` creates side-by-side plots based on a single categorical variable; `facet_grid` arranges the plots in a grid.
- See Figure 3.7 and an extension of this graphic.
- Using *layers*, you can place data from two separate data tables onto the same graph (one on top of the other).
- Typically the information from the two data tables would be plotted with different glyphs — you should take care to choose the glyphs wisely to preserve readability.
- See Figures 3.8 and 3.9 for examples.

# Standard Graphical Displays for Univariate Numerical Data

- To display the distribution of one numerical variable, the *histogram* or the *density plot* are popular choices.
- In `ggplot2`, the `geom_histogram` function produces a histogram.
- The `binwidth` argument guides the number of bins in the histogram, which is a user-specified tuning parameter.
- The `bins` argument may alternatively be used to directly specify the number of bins.
- Often between 5 to 20 bins is sensible; typically larger data sets should have a larger number of bins.
- Some people recommend the number of bins should be around the square root of the number of observations.

# Another Graphical Display for Univariate Numerical Data

- The `geom_density` function gives a *kernel density estimate*, which is a smooth estimate of the underlying density function of a random variable.
- The `adjust` argument changes the *bandwidth* of the kernel — a larger bandwidth produces a smoother density plot and a smaller bandwidth a more wiggly density plot.
- See Figures 3.10 and 3.11 and the related code.

# Standard Graphical Displays for Univariate Categorical Data

- To display the distribution of one categorical variable, the *bar graph* or the *pie chart* are popular choices – the bar graph usually considered preferable (Length easier to see than Area).
- The `geom_col` and `geom_bar` functions are useful for creating bar graphs.
- See Figures 3.12 and 3.13 and related code.

# Standard Graphical Displays for Multivariate Data

- When displaying data on multiple variables, we are often interested in the relationships between variables.
- The *scatterplot* is a classical graph to see the association between two numerical variables.
- In `ggplot2`, the `geom_point` function produces a scatterplot.
- The `geom_smooth` function can add a trend line or a smooth trend curve on top of the scatterplot.
- We can add more information on the plot by faceting, layering, and/or using Color and Glyph aesthetics to distinguish cases by category.
- The `facet_wrap` function helps create a symbolic scatterplot showing levels of a categorical variable.
- Sometimes reordering levels of a categorical variable can produce a better display: The functions `reorder` and `fct_relevel` can be helpful.

# Common Specialized Plots

- A time series plot is a way to show how a numerical variable changes over time, with the variable on the y-axis and time on the x-axis.
- Smooth trend curves are often added to time series plots with `geom_smooth`.
- A single boxplot is a simple way to show the distribution of a numerical variable (not as detailed as a histogram or density plot).
- To display one numerical variable against one categorical variable, side-by-side boxplots are common.
- The `geom_boxplot` function is helpful.

# Other Common Specialized Plots

- A *mosiac plot* can show associations between two (or three) categorical variables.
- The areas in cells of a diagram correspond the relative frequencies of combinations of categories.
- The `geom_mosaic` function in the `ggmosaic` package produces mosaic plots.
- See Figure 3.19 for an example.

# Graphics for Very Specialized Data

- For geographic spatial data, a *choropleth map* displays the magnitude of a variable in various regions of the map.
- A *network diagram* shows connections between entities, which may be cases or categories.
- Sections 3.2.3 and 3.2.4 give examples of such maps.

# Historical Names Example

- Section 3.3 has an extended example about historical baby names in the U.S.
- Let's look at some example code and some variations on that example.