Introduction to ggplot2

Yu Cheng Hsu

2025-08-06

Before Class Exercise

Based on the data we collected in the first lecture:

- What kind of message can be expressed through visualization?
- What kind of graph will you use?
- Why will you choose this graph to express such an idea?

Introduction to ggplot2

In this chapter, we will introduce **ggplot2**, a powerful plotting library in R for creating elegant and complex visualizations. We will guide you through the basics of ggplot2, including its grammar of graphics approach, and demonstrate how to create various types of plots.

What is ggplot2?

ggplot2 is part of the tidyverse collection of R packages developed by Hadley Wickham in his work (Wickham 2010). He received the COPSS Presidents' Award for his contributions to the tidyverse collections. It is based on the grammar of graphics (Wilkinson 2011). As part of the tidyverse collection, it shares a similar framework that allows users to build plots layer by layer. This approach makes it highly flexible and intuitive once you understand its core concepts.

Basic Concepts of the Grammar of Graphics

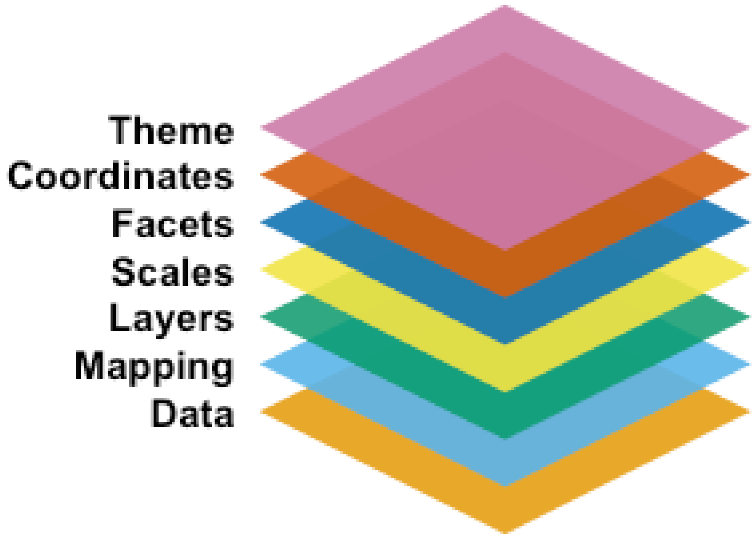

Figure 1: Layers of a graph from Wickham (2016b)

The concept of the grammar of graphics was first proposed in Wilkinson (2011) (I cite the second edition, but it was actually described in the first edition in 2005). It describes the seven basic elements of a statistical graph:

1. **Data**:
    - The information to visualize.
2. **Mapping**:

- How data variables connect to aesthetic attributes.
- Displayed as x, y, color, shape, etc.

3. **Layer**:

    - Combines **geometric elements** (geoms: points, lines, polygons) and **statistical transformations** (stats: e.g., binning for histograms, fitting models).
    - Represents what is visually displayed in the plot.

4. **Scales**:

    - Map data values to aesthetic values (e.g., color, size).
    - Generate legends and axes for reading original data values.

5. **Coordinate System (Coord)**:

    - Defines how data is mapped to the plot plane.
    - Provides axes and gridlines (e.g., Cartesian, polar, map projections).

6. **Facet**:

    - Breaks data into subsets for small multiple plots (also called conditioning or trellising).

7. **Theme**:

    - Adjusts visual elements like fonts and colors.
    - Default settings in ggplot2 are carefully chosen, but customization may require references like Tufte (1990, 1997, 2001).

Although this approach can identify individual elements of a statistical graph, it has several critiques:

1. What graph should I use
2. This framework does not work well in the programming language setting, and later Wickham (2010) implicitly modified these layers
3. It does not describe an interactive graph

Getting Started

To use ggplot2, you first need to install and load the package in R:

```r
# Install ggplot2 and patchwork if not already installed
# 'requireNamespace' checks if the packages are available without loading them
if (!requireNamespace(c("ggplot2","patchwork"), quietly = TRUE)) {
  install.packages("ggplot2")  # Core package for advanced plotting
  install.packages("patchwork")  # Package for combining multiple plots
}

# Load the libraries into the current R session
library(ggplot2)  # For creating visualizations using grammar of graphics
library(patchwork)  # For arranging multiple ggplot objects
```

Basic Plot Example

Data and mapping layer

Let's create a simple scatter plot using the `mtcars` dataset, which is built into R:

```r
library(ggplot2)  # Ensure ggplot2 is loaded for plotting

# Prepare the dataset by copying mtcars and converting 'am' to a factor
mtcars2 <- mtcars
mtcars2$am <- factor(
  mtcars$am, labels = c('automatic', 'manual')  # Label transmission types
)

# Initialize a ggplot object with data and aesthetic mappings
p <- ggplot(data=mtcars2, mapping = aes(x = mpg, y = cyl, colour = am))
p  # Display the base plot (no layers added yet)

# Alternative syntax (commented out) using positional arguments
# p <- ggplot(mtcars2, aes(mpg, cyl, color = am))
# p
```

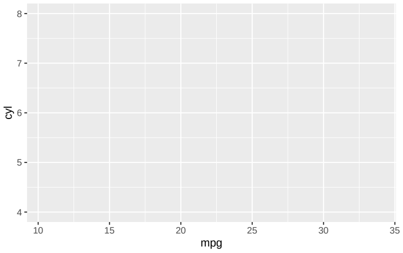

Figure 2: Scatter plot of MPG vs cylinders, colored by transmission type.

From the code, you can figure out that, data and mapping was encode in the first line of ggplot function

$$\text{ggplot(data=}\underbrace{\text{mtcars2}}_{\text{data}}\text{, mapping=}\underbrace{\text{aes(x = mpg, y = cyl, colour=am)}}_{\text{mapping}}\text{)}$$

Occasionally (actually, very frequently), you will see people ignoring everything on the left-hand side (LHS) of the equal sign for `data`, `mapping`, `x`, and `y` as they are standard arguments for ggplot2.

There are also some other mapping options other than color

- Size
- Line
  - linetype
  - lineend
  - linejoin
- Dot
  - Shape

Layers

This series of function are named in the format *geom_XXXX*

One variable

```
# Create a histogram to show the distribution of miles per gallon (mpg)
ggplot(mtcars2) +
  geom_histogram(mapping = aes(x = mpg), binwidth=5)  #                             histogram bi

# Note: This is equivalent to using a pre-defined plot c
# p + geom_histogram(mapping = aes(x = mpg), binwidth=5)
```

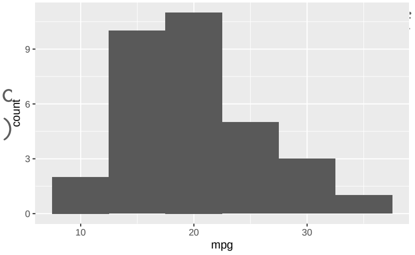

Figure 3: Histogram of miles per gallon (MPG) distribution.

```
# Create a boxplot to summarize the distribution of mile
ggplot(mtcars2) +
  geom_boxplot(mapping = aes(y = mpg))  # Boxplot on y-a
```

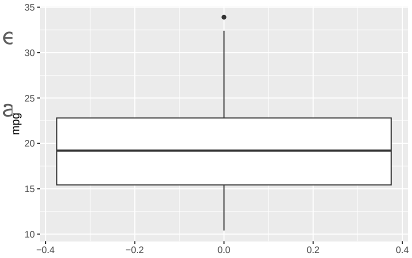

Figure 4: Boxplot of miles per gallon (MPG) distribution.

Two variables

```
# Create a scatter plot to show relationship between mpg and cylinders, colored by transmission
ggplot(mtcars2, aes(mpg, cyl, color = am)) +
  geom_point()  # Add points layer to represent individu
```

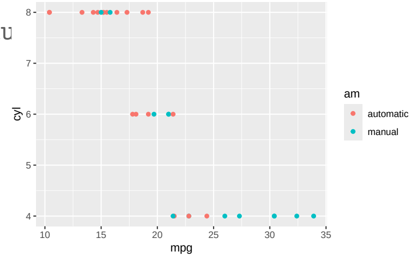

Figure 5: Scatter plot of MPG vs cylinders, colored by transmission type.

Scales

Scales are functions (processes) that transform data for the graph. This process is trivial and is done by observing the type of layer and the data, so the only thing that people frequently need to use is to modify the axis/legend display. The series of functions are named in the following format: `scale_(AES)_(datatype)`. The argument and its corresponding components are listed in the below table and figure.

Table 1: Fruit prices

Argument name	Axis	Legend
name	Label	Title
breaks	Ticks	Key
labels	Tick label	Key Label

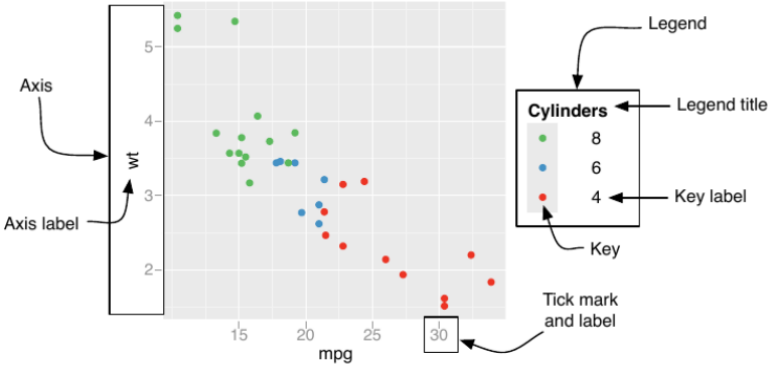

Figure 6: Common components of a figure, figure from Wickham (2016b)

```
# Create a scatter plot with customized axis and color legend labels
ggplot(mtcars2, aes(mpg, cyl, color = am)) +
  geom_point() +  # Add points layer for data representa
  scale_x_continuous(name="Miles per gallon") +  # Custo
  scale_colour_discrete(name="Automatic/Manual")  # Cust
```

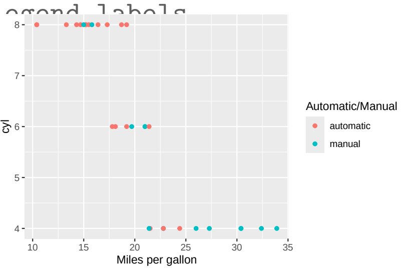

Figure 7: Scatter plot of MPG vs cylinders with customized axis and legend labels.

Coordinates and facet

Coordinates refer to the coordinate system on the graph. They can help you adjust your plot.

```
# Create a scatter plot with customized axis limits and labels
ggplot(mtcars2, aes(mpg, cyl, color = am)) +
  geom_point() +  # Add points layer for data representa
  scale_x_continuous(name="Miles per gallon") +  # Custo
  scale_colour_discrete(name="Automatic/Manual") +  # Cu
  coord_cartesian(xlim=c(10,40), ylim=c(1,8))  # Set spe
```

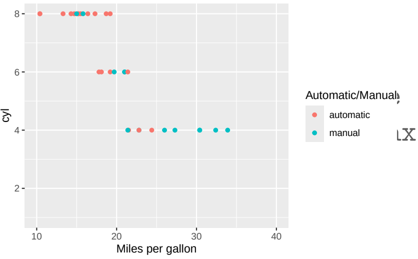

Figure 8: Scatter plot of MPG vs cylinders with customized axis limits.

Facets facilitate breaking data into several subgraphs, but this is a more complicated technique that we won't cover in class. If you are interested in the topic, you can explore it in the reference book and documentation of `ggplot2`.

Theme

There are several available options for the theme of your plot. Meanwhile, there are also third-party packages designing different themes for plots, such as `ggtheme`.

```
# Initialize base plot with customized axis limits and labels
p <- ggplot(mtcars2, aes(mpg, cyl, color = am)) +
  geom_point() +  # Add points layer for data representation
  scale_x_continuous(name="Miles per gallon") +  # Customize x-axis label
  scale_colour_discrete(name="Automatic/Manual") +  # Customize color legend title
  coord_cartesian(xlim=c(10,40), ylim=c(1,8))  # Set specific limits for x and y axes

# Apply different themes to the base plot for comparison
p_grey <- p + theme_grey() + ggtitle("theme_grey()")  # Default grey background theme
p_bw <- p + theme_bw() + ggtitle("theme_bw()")  # Black and white theme
p_ld <- p + theme_linedraw() + ggtitle("theme_linedraw()")  # Line-drawn theme
p_l <- p + theme_light() + ggtitle("theme_light()")  # Light theme with minimal grid
p_d <- p + theme_dark() + ggtitle("theme_dark()")  # Dark background theme
p_m <- p + theme_minimal() + ggtitle("theme_minimal()")  # Minimalist theme
p_c <- p + theme_classic() + ggtitle("theme_classic()")  # Classic theme without grid
p_v <- p + theme_void() + ggtitle("theme_void()")  # Empty theme with no background or grid

# Combine all themed plots for display using patchwork
# Arrange plots in a grid layout with 2 columns
p_grey + p_bw + p_ld + p_l + p_d + p_m + p_c + p_v + plo
```

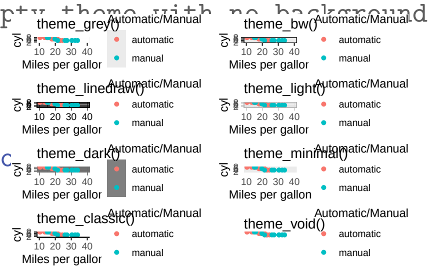

Figure 9: Comparison of different ggplot2 themes applied to a scatter plot of MPG vs cylinders.

Wrap-up

As a wrap-up, your code is usually in the following format:

$$
\begin{aligned}
&\text{ggplot()}+\\
&\underbrace{\text{geom\_XXXX(data=DATA,mapping=aes(x,y,color,...))}}_{\text{plotting data}}+\\
&\underbrace{\text{scale\_AES\_TYPE(name="TITLE",breaks="TICK LOC",labels="TICK LAB")}}_{\text{Handeling axis, legend etc}}+\\
&\underbrace{\text{coord\_cartesian(xlim=c(min,MAX), ylim=c(min,MAX))}}_{\text{Adjust coordinate systems}}+\\
&\underbrace{\text{ggtitle("CHART TITLE")}}_{\text{Plotting title}}
\end{aligned}
$$

> About making graph
>
> From the code introduced, it will be great to reflect on
>
> - How does the code construction differ from your human process of plotting code?
> - How does the 7 layer graphic language differ from the ggplot syntax?
> - If you could make `ggplot` easier to use, how would you design it?

Final remarks and acknowledgement

The materials and contents are mostly adapted from Wickham (2016a). You can get the latest edition from the book website, which also covers more advanced topics. For details on how to use the code and each function, you can find the documentation of the `ggplot2` library through `??ggplot2`.

Bibilography

Wickham, Hadley. 2010. "A Layered Grammar of Graphics." *Journal of Computational and Graphical Statistics* 19 (1): 3–28.

———. 2016a. "Getting Started with Ggplot2." In *Ggplot2: Elegant Graphics for Data Analysis*, 11–31. Springer.

———. 2016b. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wilkinson, Leland. 2011. "The Grammar of Graphics." In *Handbook of Computational Statistics: Concepts and Methods*, 375–414. Springer.