Rotman

INTRO TO R - VISUALIZATION

R Workshop

March 6, 2025 Prepared by Jay Cao / TDMDAL

Website: https://tdmdal.github.io/r-visualization-2025-winter/

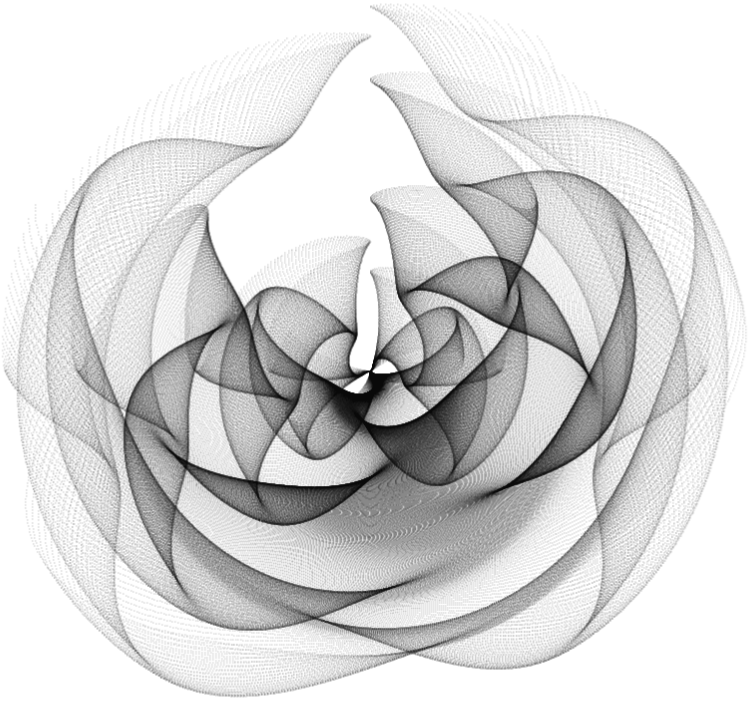
Rotman School of Management
UNIVERSITY OF TORONTO

R Graphics - Overview

- Base plot

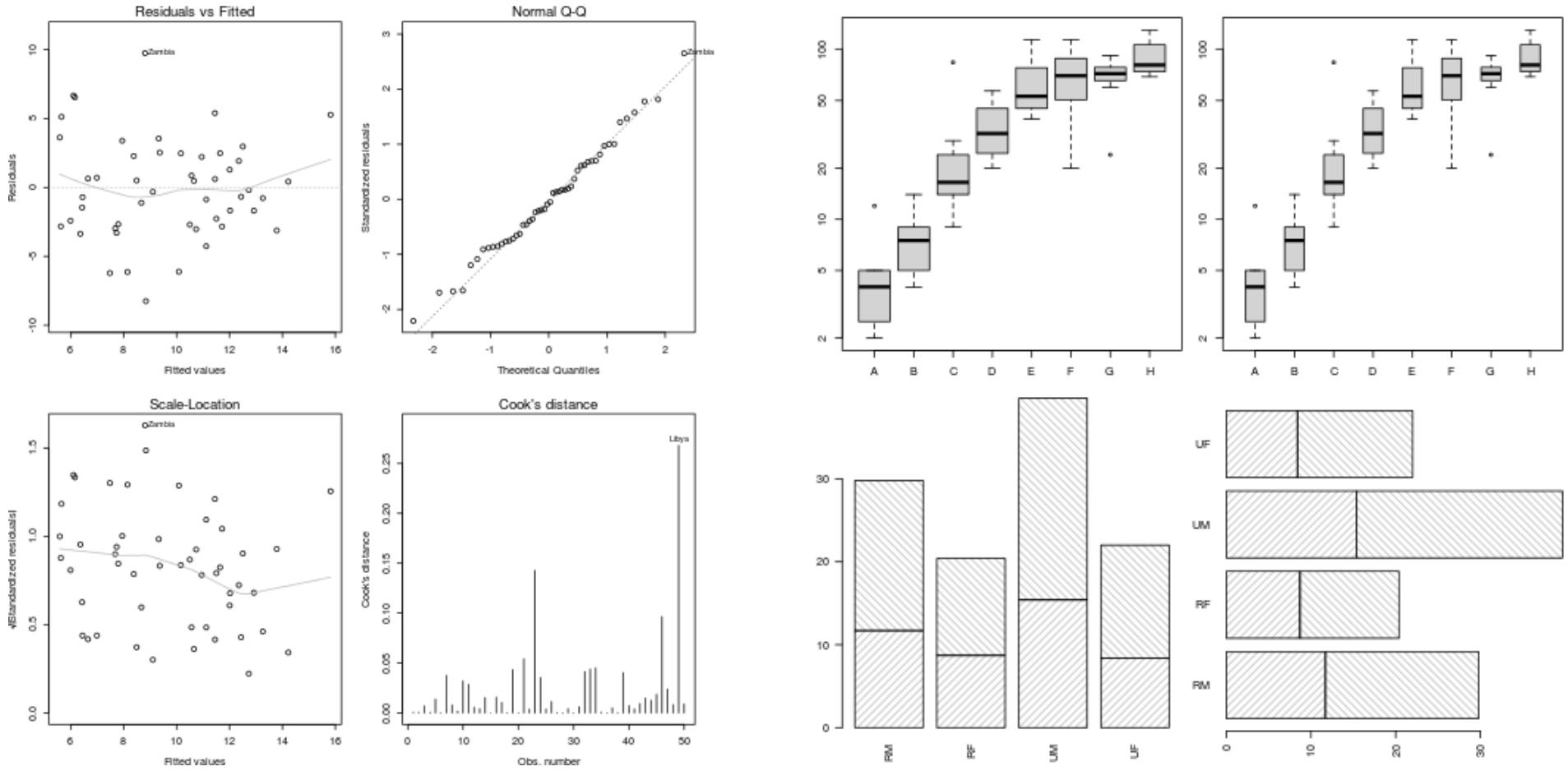
- Two main plotting systems
  - lattice
  - **ggplot2 (our focus today)**

- Specialized plots
  - Plot functions bundled with specific R packages



https://twitter.com/aschinchon/status/1405136386034970630

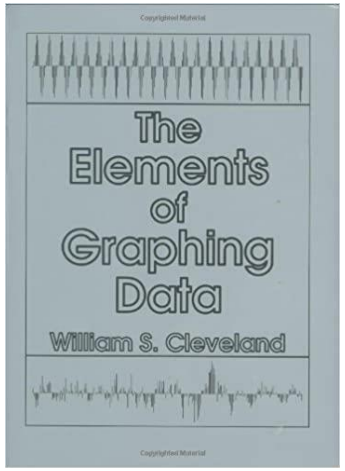
R Graphics – Base plots (examples)

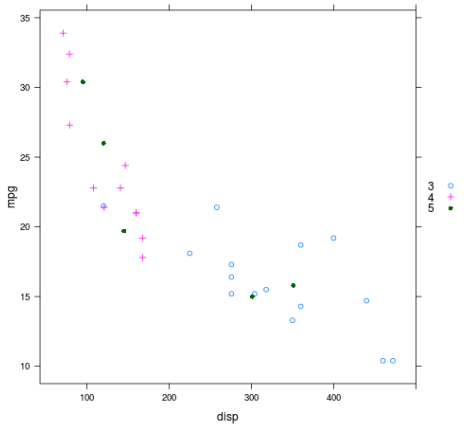


https://www.stat.auckland.ac.nz/~paul/RG3e/chapter2.html

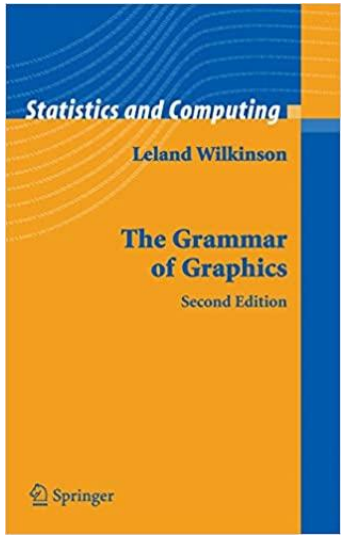
R Graphics – Two Main Plotting Systems

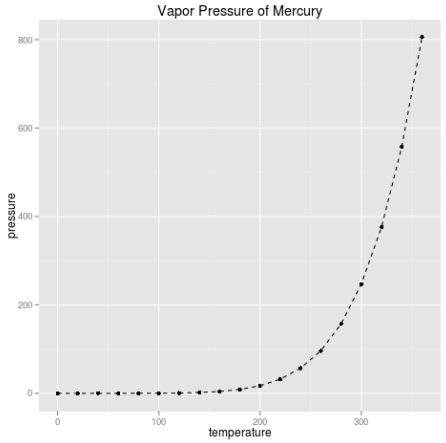
- R package: lattice
  - implements Trellis system by William Cleveland





- R package: ggplot2
  - implements "A Grammar of Graphics" by Leland Wilkinson
  - **Our focus today**

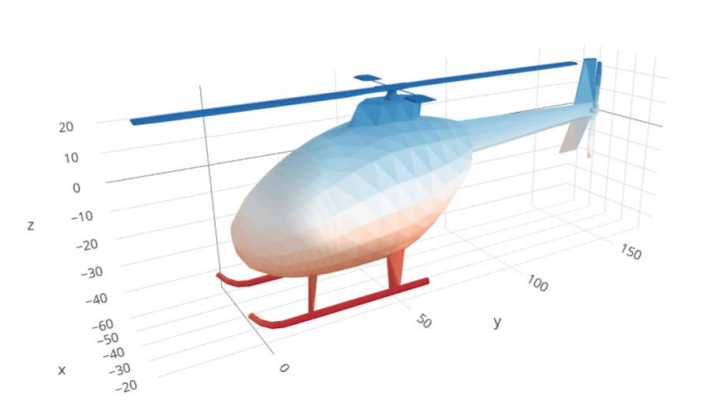




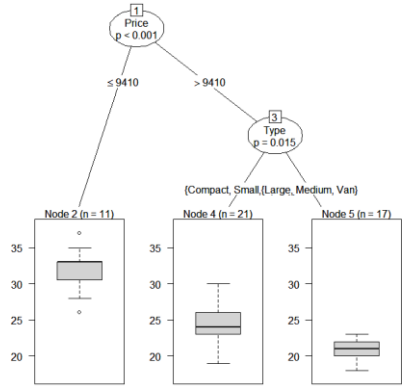
https://www.stat.auckland.ac.nz/~paul/RG3e/chapter4.html
https://www.stat.auckland.ac.nz/~paul/RG3e/chapter5.html

Other - Specialized Plots

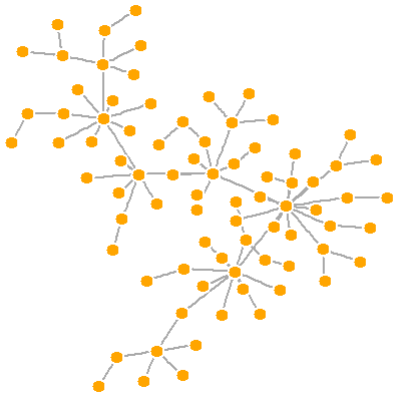
- Graphic functions provided by specialized packages
  - Based on R primitive graphical engines like grid (eg. plot() in party, igraph)
  - Following a plotting system (eg. ggmap, tmap, gganimate, plotly, etc.)
  - Wrapper of plotting tools in another languages (ex. leaflet, grViz() in DiagrammeR, dygraphs)



3D tri-surface interactive plot using the plotly package https://plot.ly/r/trisurf/



Decision tree plot using party package https://www.datacamp.com/doc/r/cart



Network plot using igraph package http://kateto.net/networks-r-igraph

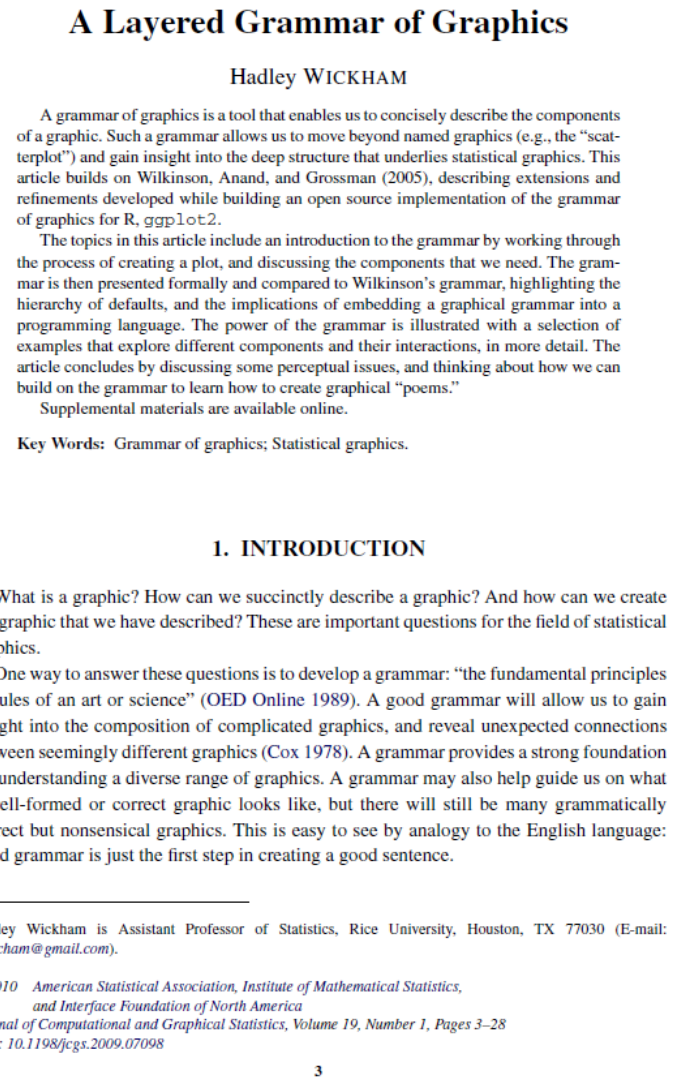
ggplot2

- Based on the Grammar of Graphics

- Basic idea: you can build any graph from the same components
  - **Data**
  - **Coordinate system**
  - **Geoms** – visual marks that represent data points

- A **layer-by-layer** approach

Ref. 1) http://amzn.to/2ef1eWp
2) https://vita.had.co.nz/papers/layered-grammar.html



A Layered Grammar of Graphics

Hadley WICKHAM

A grammar of graphics is a tool that enables us to concisely describe the components of a graphic. Such a grammar allows us to move beyond named graphics (e.g., the "scatterplot") and gain insight into the deep structure that underlies statistical graphics. This article builds on Wilkinson, Anand, and Grossman (2005), describing extensions and refinements developed while building an open source implementation of the grammar of graphics for R, `ggplot2`.

The topics in this article include an introduction to the grammar by working through the process of creating a plot, and discussing the components that we need. The grammar is then presented formally and compared to Wilkinson's grammar, highlighting the hierarchy of defaults, and the implications of embedding a graphical grammar into a programming language. The power of the grammar is illustrated with a selection of examples that explore different components and their interactions, in more detail. The article concludes by discussing some perceptual issues, and thinking about how we can build on the grammar to learn how to create graphical "poems."

Supplemental materials are available online.

Key Words: Grammar of graphics; Statistical graphics.

1. INTRODUCTION

What is a graphic? How can we succinctly describe a graphic? And how can we create the graphic that we have described? These are important questions for the field of statistical graphics.

One way to answer these questions is to develop a grammar: "the fundamental principles or rules of an art or science" (OED Online 1989). A good grammar will allow us to gain insight into the composition of complicated graphics, and reveal unexpected connections between seemingly different graphics (Cox 1978). A grammar provides a strong foundation for understanding a diverse range of graphics. A grammar may also help guide us on what a well-formed or correct graphic looks like, but there will still be many grammatically correct but nonsensical graphics. This is easy to see by analogy to the English language: good grammar is just the first step in creating a good sentence.

Hadley Wickham is Assistant Professor of Statistics, Rice University, Houston, TX 77030 (E-mail: *h.wickham@gmail.com*).

Journal of Computational and Graphical Statistics, Volume 19, Number 1, Pages 3–28
DOI: 10.1198/jcgs.2009.07098

ggplot() – “base layer”

data

```
p <- ggplot(df, aes(x, y, other_aesthetics))
```

https://ggplot2.tidyverse.org/reference/ggplot.html

ggplot() – “base layer”

mapping: linking variables in the data to aesthetic elements in the plot

data

```
p <- ggplot(df, aes(x, y, other_aesthetics))
```

https://ggplot2.tidyverse.org/reference/ggplot.html

ggplot() – “base layer”

mapping: linking variables in the data to aesthetic elements in the plot

data

```
p <- ggplot(df, aes(x, y, other_aesthetics))
```

(x, y) coordinates mapping

color-, size- mapping, etc.

https://ggplot2.tidyverse.org/reference/ggplot.html

Example 1 - step 1. “base layer”

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
1 Afghanistan	Asia	2007	43.8	31889923	975.
2 Albania	Europe	2007	76.4	3600523	5937.
3 Algeria	Africa	2007	72.3	33333216	6223.

```
p <- ggplot(df, aes(x = gdpPercap, y = lifeExp))
```

ggplot() – Add Other Layers

mapping: linking variables in the data to aesthetic elements in the plot

data

```
p <- ggplot(df, aes(x, y, other_aesthetics)) +
  another_layer +
  another_layer +
  …
```

what to plot : point, line, label, etc. (geom-, scale-functions...)

- If data and mapping are not specified in the base layer, they must be supplied in each layer added to the plot

https://ggplot2.tidyverse.org/reference/ggplot.html

ggplot() – geom layers (eg. geom_point )

layer specific data and mapping
If not specified, inherit from base layer

```
p + geom_point(data, mapping, stat, position, …)
```

https://ggplot2.tidyverse.org/reference/geom_point.html

ggplot() – geom layers (eg. geom_point )

layer specific data and mapping
If not specified, inherit from base layer

```
p + geom_point(data, mapping, stat, position, …)
```

statistical transformation &
position adjustment
e.g. `position = “jitter"`

https://ggplot2.tidyverse.org/reference/geom_point.html

ggplot() – geom layers (eg. geom_point )

other arguments:
e.g. `color = "red"`,
`alpha = 0.5`, etc.

layer specific data and mapping
If not specified, inherit from base layer

```
p + geom_point(data, mapping, stat, position, …)
```

statistical transformation &
position adjustment
e.g. `position = “jitter"`

https://ggplot2.tidyverse.org/reference/geom_point.html

Example 1 - step 2. geom_point layer

```
p +
  geom_point(aes(size = pop,
    color = continent),
    alpha = 0.5)
```

Example 2 - the Diamond Data

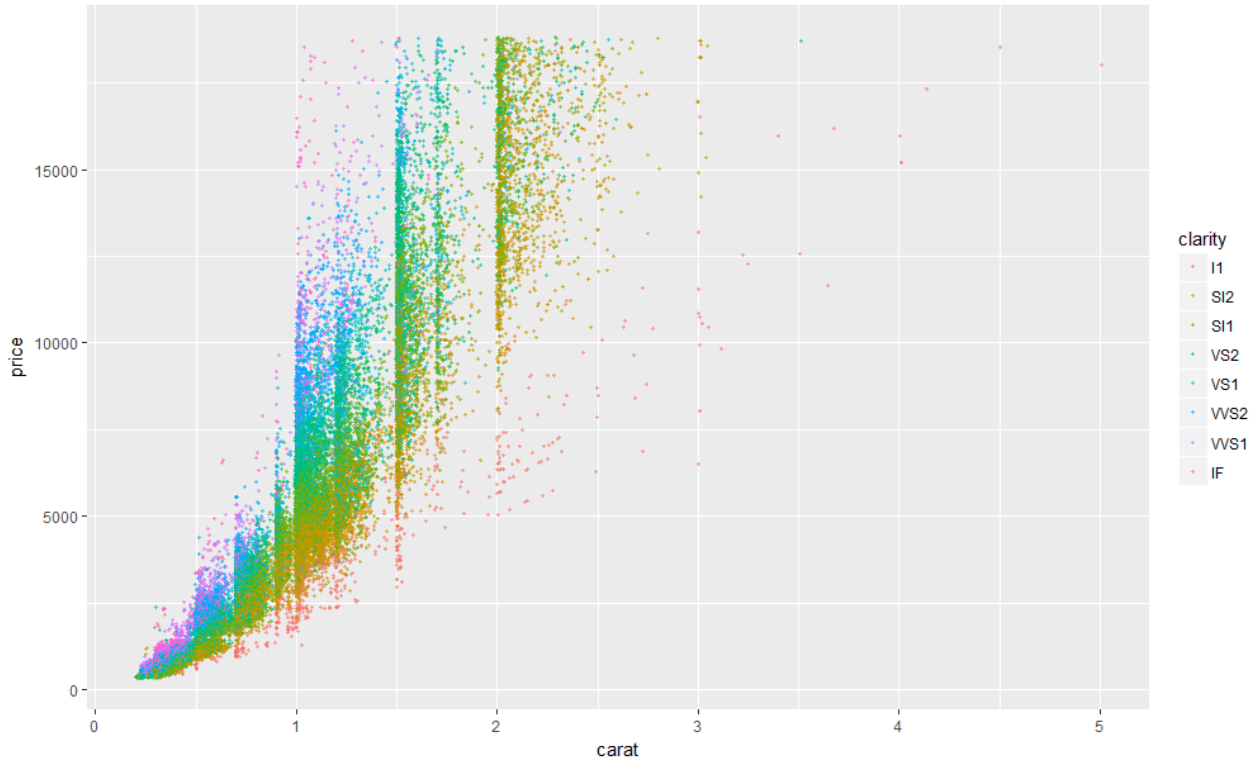
```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

Example 2 - layer 1

```
ggplot(data = diamonds, aes(carat, price)) +
```

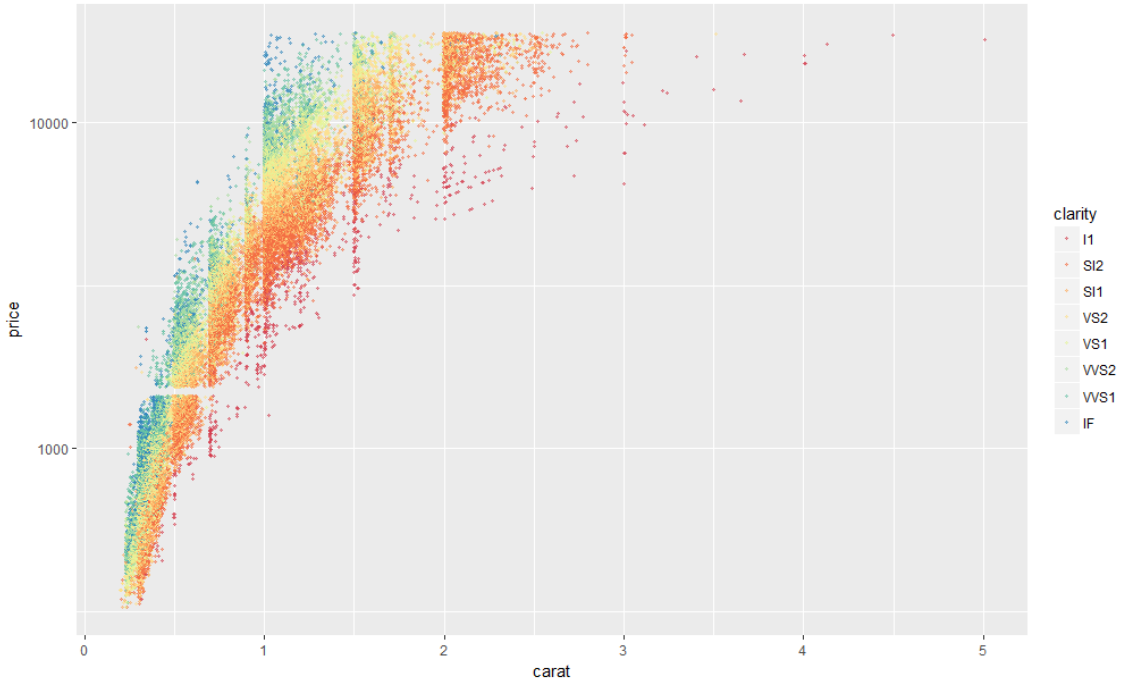
Example 2 - layer 2

```
ggplot(data = diamonds, aes(carat, price)) +
  geom_point(aes(colour = clarity),
    position = "jitter",
    alpha = 0.5,
    size = 0.8) +
```



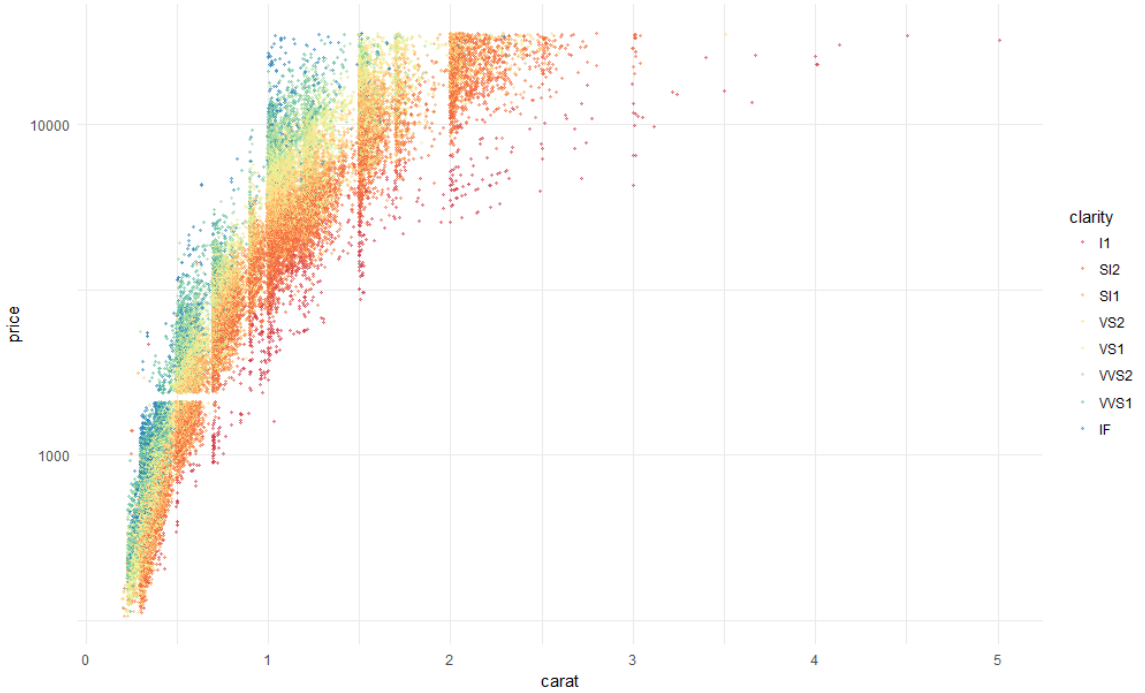
Example 2 - layer 3 & layer 4

```
ggplot(data = diamonds, aes(carat, price)) +
  geom_point(aes(colour = clarity),
    position = "jitter",
    alpha = 0.5,
    size = 0.8) +
  scale_y_continuous(transform = "log10")
  scale_color_brewer(palette = "Spectral")
```



Example 2 - layer 5

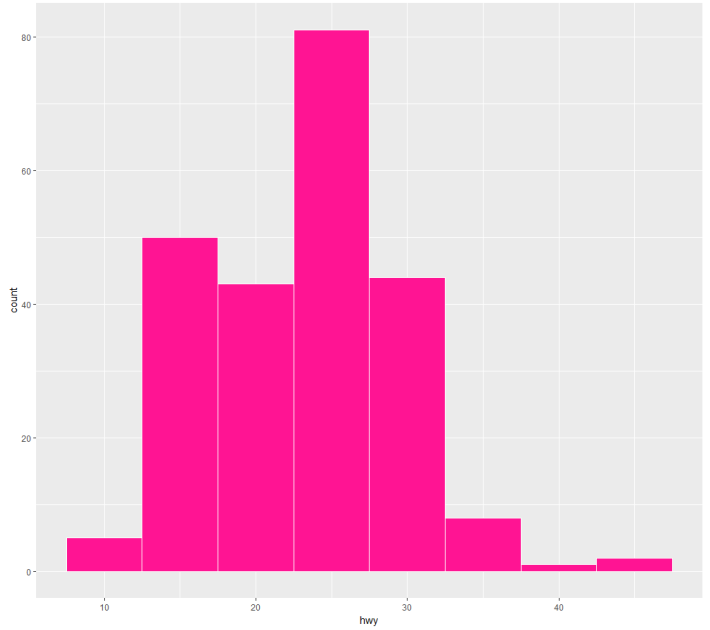
```
ggplot(data = diamonds, aes(carat, price)) +
  geom_point(aes(colour = clarity),
    position = "jitter",
    alpha=0.5,
    size = 0.8) +
  scale_y_continuous(transform = "log10") +
  scale_color_brewer(palette = "Spectral") +
  theme_minimal()
```



More Examples – geom_histogram

```
> library(ggplot2)
> mpg
# A tibble: 234 x 11
   manufacturer model      displ  year   cyl trans      drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.80  1999     4 auto(l5)   f        18    29 p     compact
 2 audi         a4           1.80  1999     4 manual(m5) f        21    29 p     compact
 3 audi         a4           2.00  2008     4 manual(m6) f        20    31 p
```

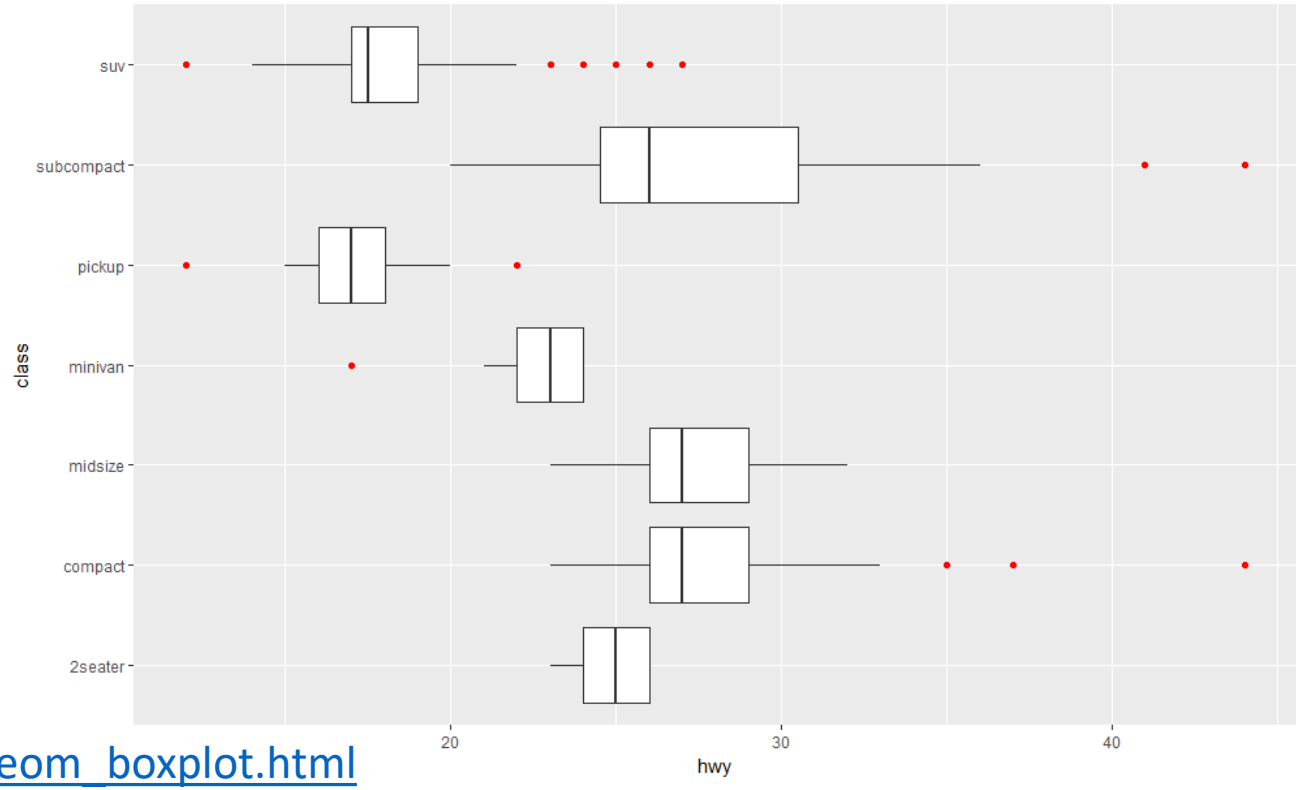
```
ggplot(mpg, aes(x = hwy)) +
  geom_histogram(binwidth=5,
    color = "white",
    fill = "deeppink")
```



https://ggplot2.tidyverse.org/reference/geom_histogram.html

More Examples – geom_boxplot

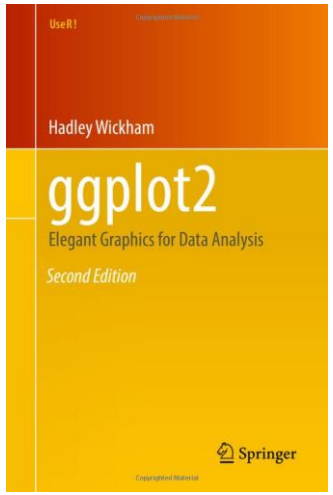
```
ggplot(mpg, aes(class, hwy)) +
  geom_boxplot(outlier.colour = "red") +
  coord_flip()
```

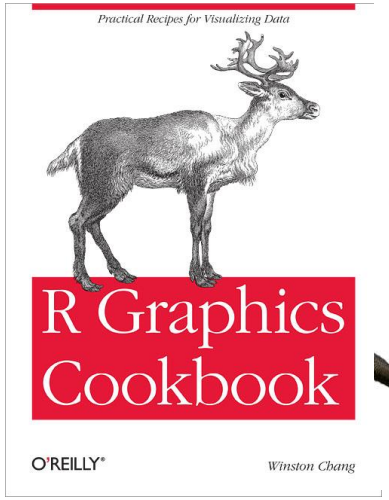


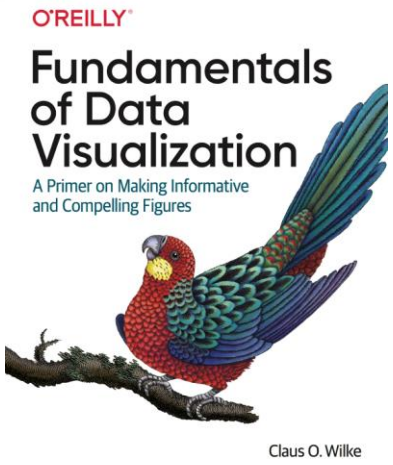
https://ggplot2.tidyverse.org/reference/geom_boxplot.html

Learning Resources (free)

- ggplot2: Elegant Graphics for Data Analysis (3rd ed.; work in progress)
- R Graphics Cookbook (2nd ed.)
- Fundamentals of Data Visualization
  - Code here

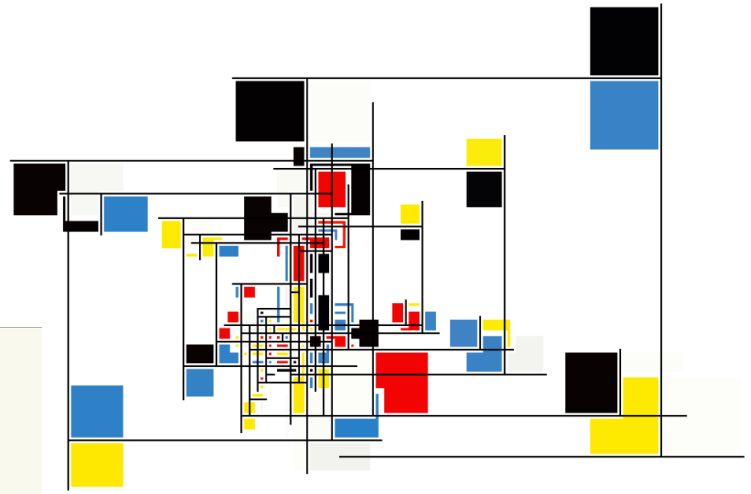




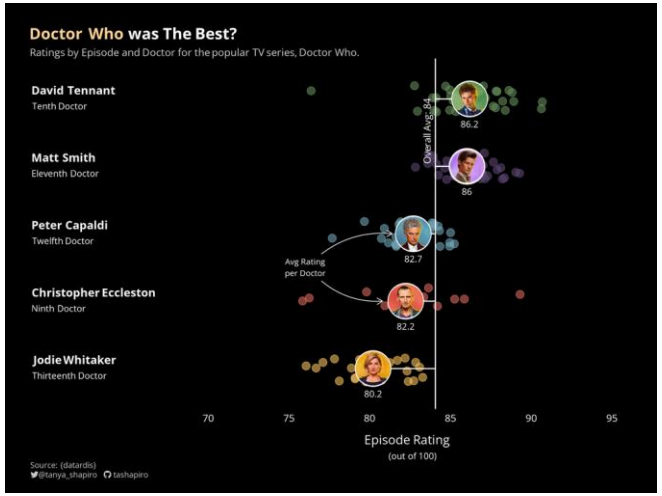


Get Inspiration from Experts

- Many data visualization experts using ggplot for their work
- Learn from the experts, for example,
  - https://github.com/tashapiro
  - https://fronkonstin.com/



https://fronkonstin.com/



https://www.tanyashapiro.com/gallery; https://github.com/tashapiro