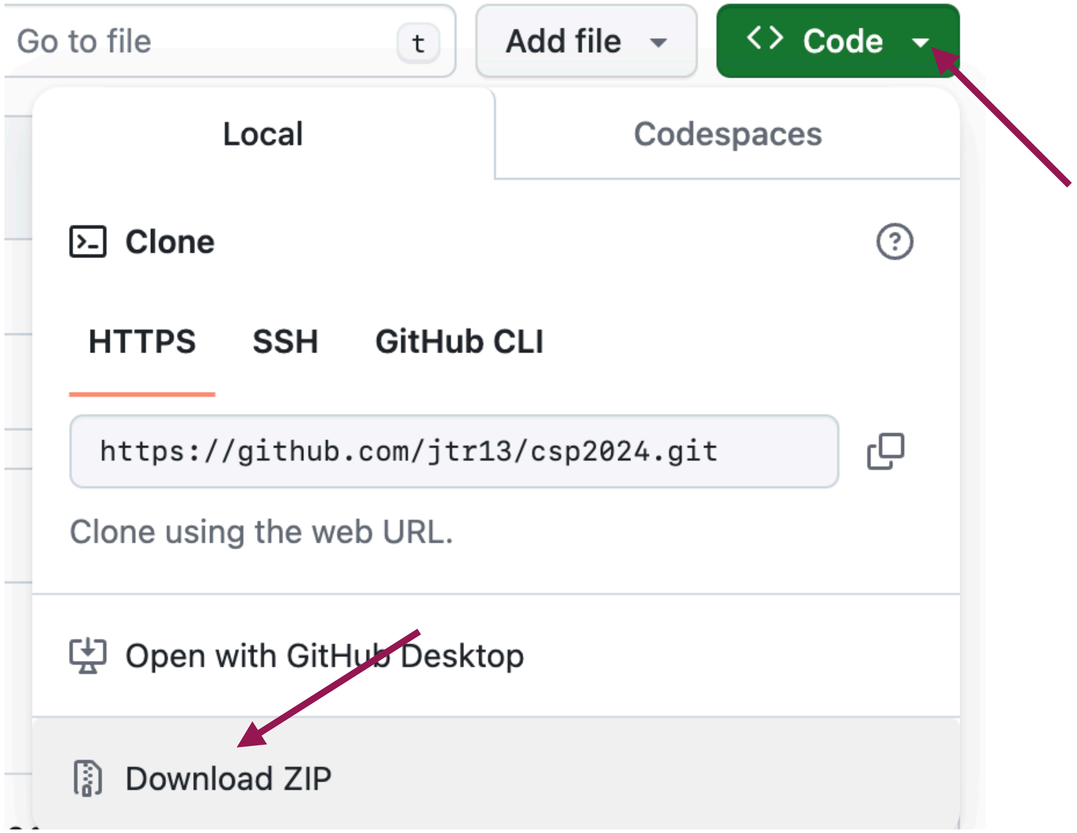
Effective Graphs with `ggplot2`

Full day short course, CSP 2024 (New Orleans)
Feburary 27, 2024

Joyce Robbins
Dept. of Statistics, Columbia University
jtr13@columbia.edu

Slides and code

www.github.com/jtr13/csp2024



Introduction

slides/01_introduction.pdf

Morning schedule

github.com/jtr13/csp2024/

8:00 - 9:00	Introduction, grammar of graphics, data layers with one mapping: histograms and density curves	`slides/01_introduction.pdf` `slides/02_datalayer1.pdf`
9:00 - 9:30	LAB: histograms and density curves	`labs/histogram.Rmd` `labs/density.Rmd`
9:30 - 10:00	data layers with two mappings: boxplots and scatterplots	`slides/03_datalayer2.pdf`
10:00 - 10:15	BREAK ☕	
10:15 - 10:30	LAB: boxplots and scatterplots	`labs/boxplots.Rmd` `labs/scatterplots.Rmd`
10:30 - 11:15	*Grammar of graphics* / **`ggplot2:`** x, y, color scales	`slides/04_scales.pdf`
11:15 - 11:30	LAB: scales	`labs/scales.Rmd`
11:30 - 12:00	Transform data with **`tidyr::pivot_longer()`**	`slides/05_pivot_longer.pdf` `labs/pivot_longer.Rmd`

Afternoon schedule

github.com/jtr13/csp2024/

1:30 - 2:00	Categorical data: bar charts and Cleveland dot plots	`slides/06_categorical.pdf`
		`labs/barchart.Rmd`
		`labs/dotplot.Rmd`

2:00 - 3:00	Controling your data: counting (binning) data, factor levels, ordering categories with **`forcats`**	`slides/07_forcats.pdf` `labs/forcats.Rmd`
3:00 - 3:15	BREAK ☕	
3:15 - 3:45	Faceting: why and how	`slides/08_faceting.pdf`
3:45 - 4:15	LAB: putting it all together	`labs/everything.Rmd`
4:15 - 5:00	Presentation ready charts: themes, colors, ...	`slides/09_present.pdf`
5:00 - 5:30	Q & A	

Why effective graphs with ggplot2?

- Goal: discover and communicate trends in data

- "There is little evidence that the quality of the best graphics has improved over the last 100 years. I wonder if technology serves primarily as a **quantity**-multiplier, rather than a **quality**-multiplier." -Hadley Wickham

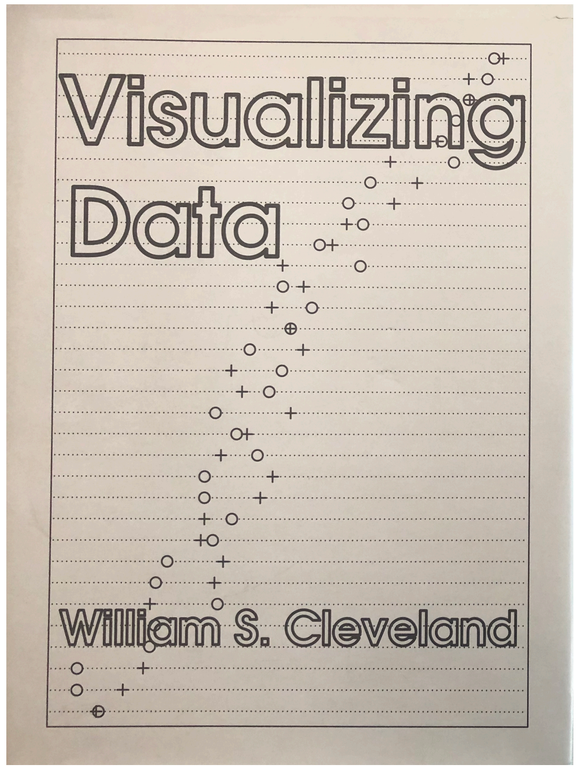
- Best practices are based on research on human perception

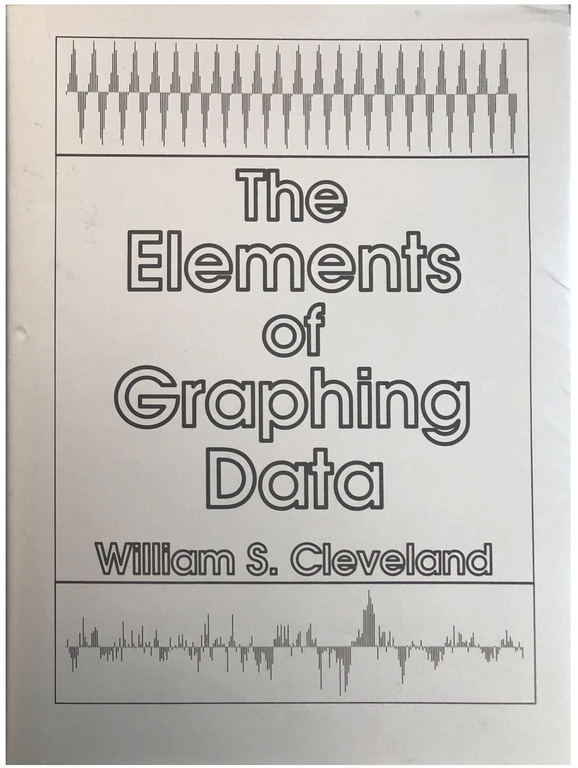
- Tight link between the development of (precursor to R) and these studies

Why effective graphs with ggplot2?

- Developed in the 1970s at Bell Labs as a system "for organizing, visualizing, and analyzing data"
- Main goal: to create an interactive environment for statisticians using the most advanced analytical tools
- Influenced by John Tukey's work on exploratory data analysis
- Importance of statistical perspective / graphics research is still a defining feature of R today

William Cleveland





R help example

pie {graphics} R Documentation

Pie Charts

Description

Draw a pie chart.

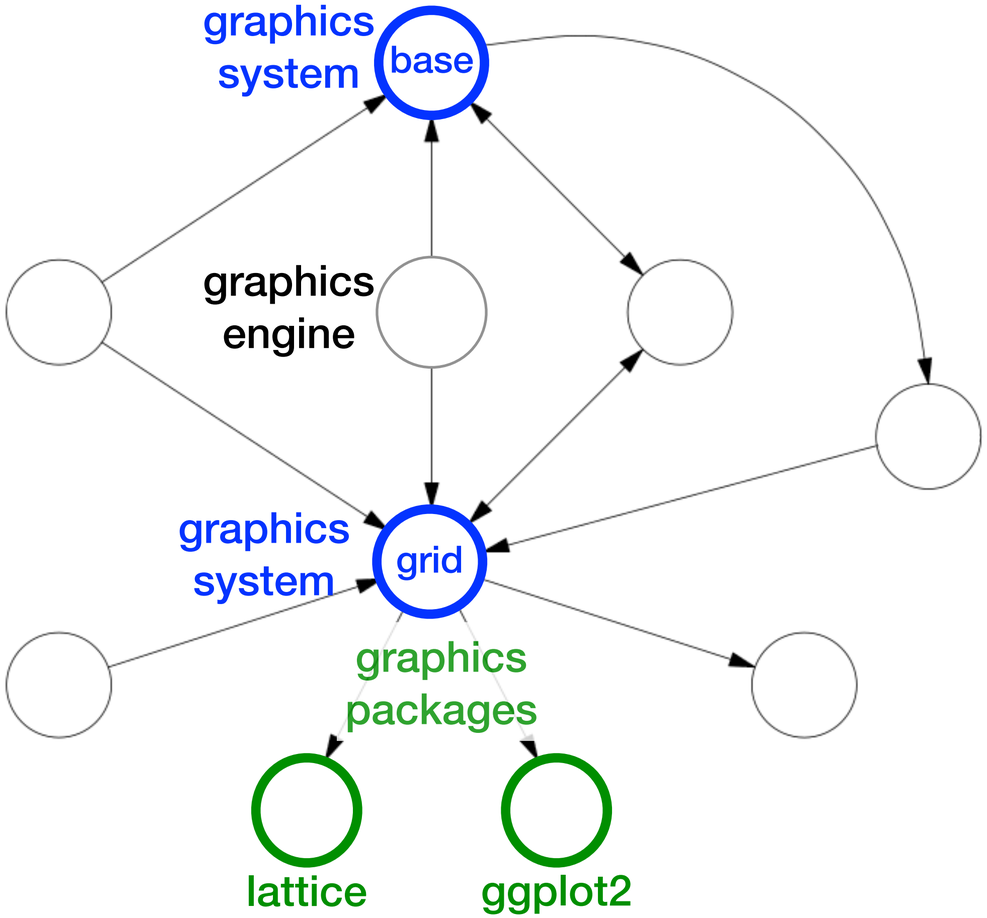
Usage

```
pie(x, labels = names(x), edges = 200, radius = 0.8,
    clockwise = FALSE, init.angle = if(clockwise) 90 else 0,
    density = NULL, angle = 45, col = NULL, border = NULL,
    lty = NULL, main = NULL, ...)
```

Arguments

`x`	a vector of non-negative numerical quantities. The values in `x` are displayed as the areas of pie slices.
`labels`	one or more expressions or character strings giving names for the slices. Other objects are coerced by `as.graphicsAnnot`. For empty or `NA` (after coercion to character) labels, no label nor pointing line is drawn.
`edges`	the circular outline of the pie is approximated by a polygon with this many edges.
`radius`	the pie is drawn centered in a square box whose sides range from *-1* to *1*. If the character strings labeling the slices are long it may be necessary to use a smaller radius.
`clockwise`	logical indicating if slices are drawn clockwise or counter clockwise (i.e., mathematically positive direction), the latter is default.
`init.angle`	number specifying the *starting angle* (in degrees) for the slices. Defaults to 0 (i.e., '3 o'clock') unless `clockwise` is true where `init.angle` defaults to 90 (degrees), (i.e., '12 o'clock').
`density`	the density of shading lines, in lines per inch. The default value of `NULL` means that no shading lines are drawn

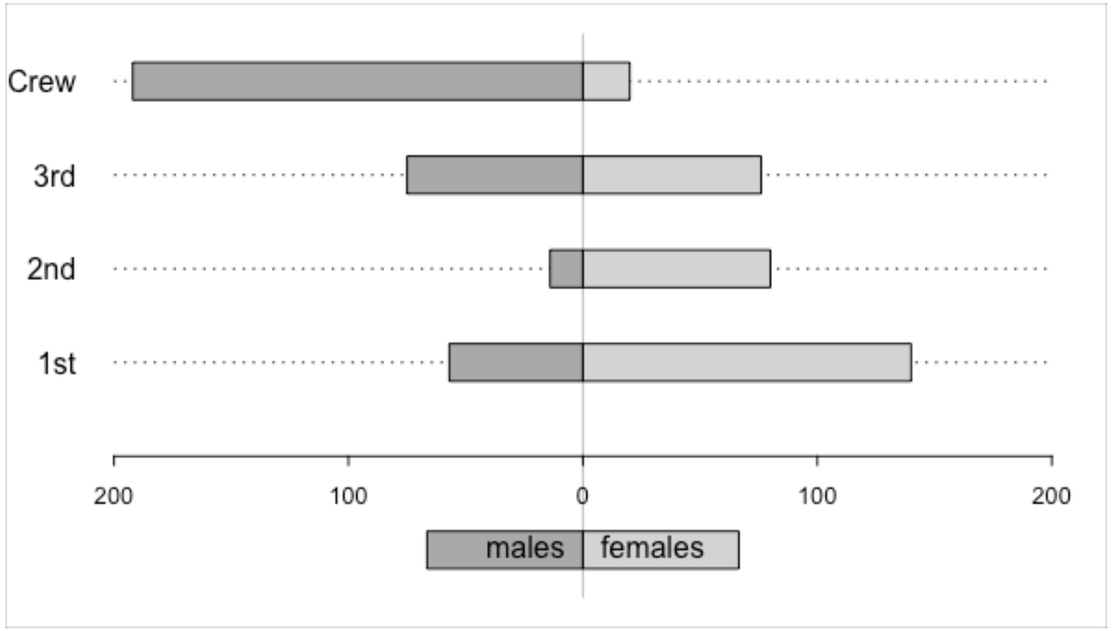
R graphics



Based on https://www.stat.auckland.ac.nz/~paul/RG3e/organisation-graphicslevels.png

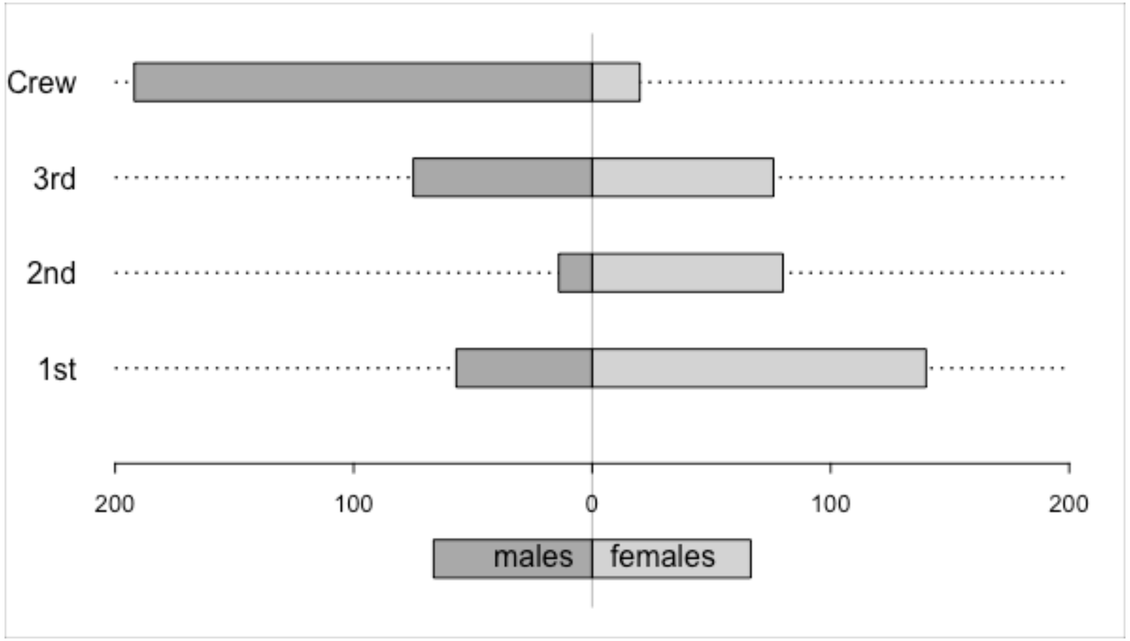
Base R graphics

```
groups <- dimnames(Titanic)[[1]]
males <- Titanic[, 1, 2, 2]
females <- Titanic[, 2, 2, 2]
par(mar=c(0.5, 4, 0.5, 1))
plot.new()
plot.window(xlim=c(-200, 200), ylim=c(-1.5, 4.5))
ticks <- seq(-200, 200, 100); y <- 1:4; h <- 0.2
lines(rep(0, 2), c(-1.5, 4.5), col="gray")
segments(-200, y, 200, y, lty="dotted")
rect(-males, y-h, 0, y+h, col="dark gray")
rect(0, y-h, females, y+h, col="light gray")
mtext(groups, at=y, adj=1, side=2, las=2)
par(cex.axis=0.8, mex=0.5)
axis(1, at=ticks, labels=abs(ticks), pos=0)
tw <- 1.5*strwidth("females")
rect(-tw, -1-h, 0, -1+h, col="dark gray")
rect(0, -1-h, tw, -1+h, col="light gray")
text(0, -1, "males", pos=2)
text(0, -1, "females", pos=4)
box("inner", col="gray")
```



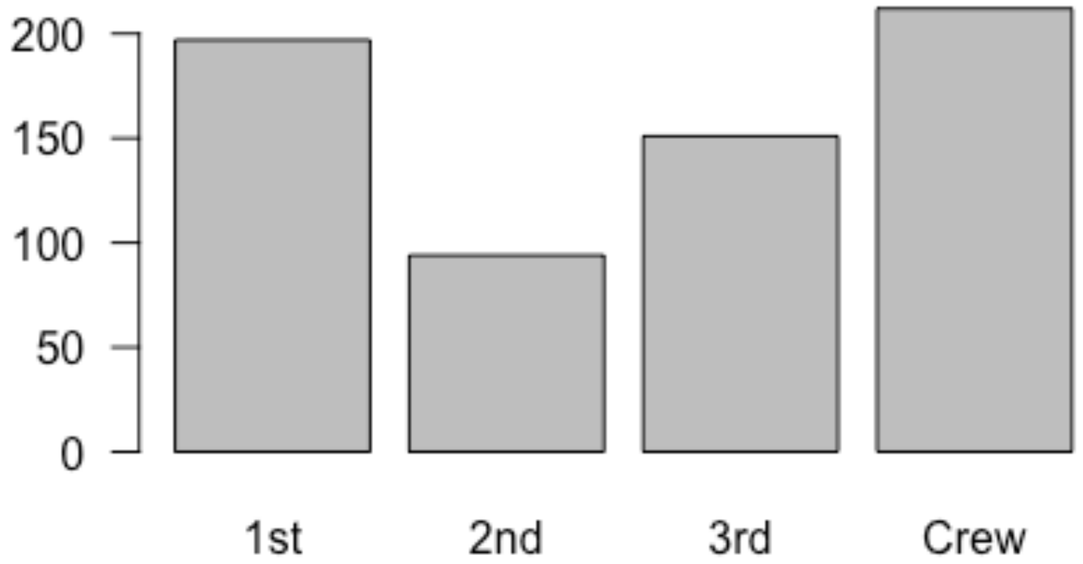
https://www.stat.auckland.ac.nz/~paul/RG3e/custombase-xmastree.R

Base R graphics



Higher level base R graphics functions

```
crew_counts <- rowSums(Titanic[,1:2,2,2])
barplot(crew_counts, las = 1)
```

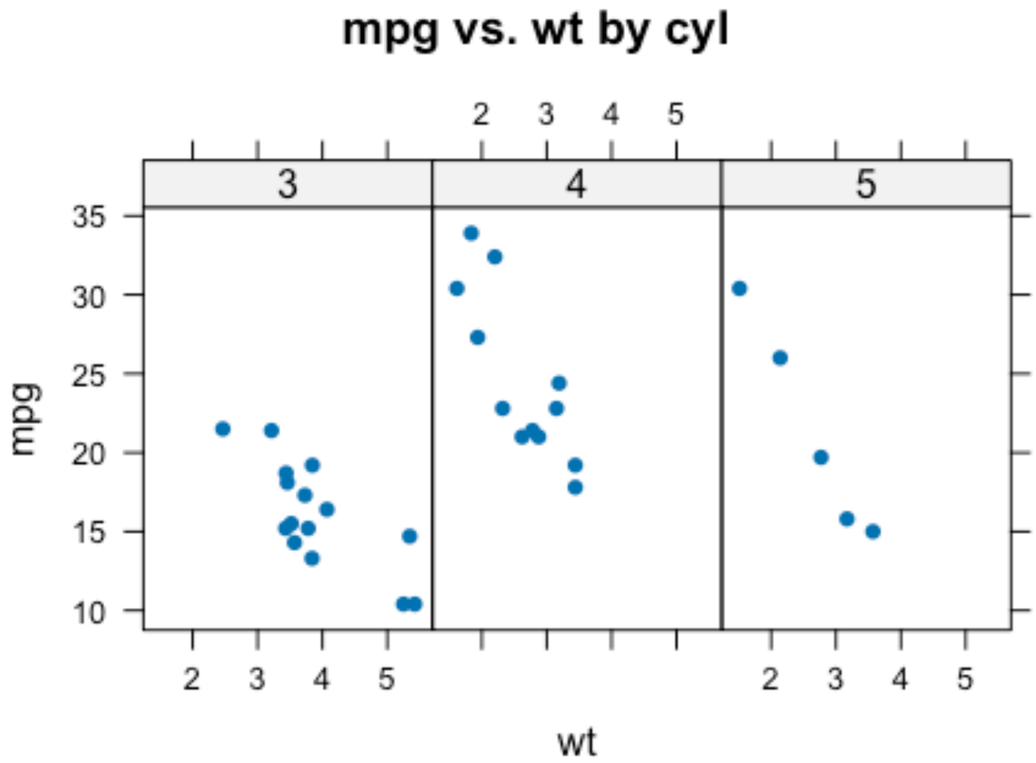


Higher level base R graphics functions

```
barplot()
boxplot()
cdplot()
contour()
coplot()
dotplot()
fourfoldplot()
hist()
matplot()
mosaicplot()
pairs()
pie()
plot()
smoothScatter()
spineplot()
stars()
stem()
stripchart()
sunflowerplot()
```

lattice package

```
library(lattice)
xyplot(mpg~wt|factor(cyl), data = mtcars,
       main="mpg vs. wt by cyl", pch = 16)
```



Higher level lattice graphing functions

```
xyplot()
splom()
cloud()
stripplot()
bwplot()
dotplot()
barchart()
histogram()
densityplot
qqmath()
qq()
contourplot()
levelplot()
parallel()
wireframe()
```

http://www.sthda.com/english/wiki/lattice-graphs

Why ggplot2?

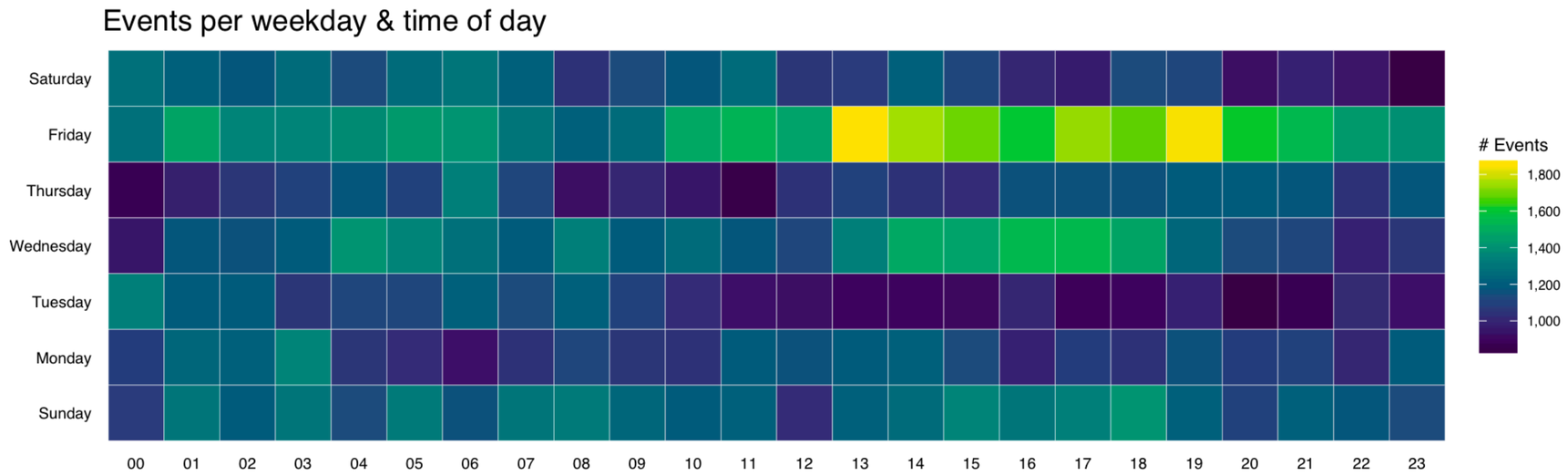
- **Many similarities to `lattice` (in constrast to base R):**
  - **automated legends and margins**
  - **easy to create panel plots***
  - **flexibility of `grid` system for manipulating graphics output**
  - **carefully chosen defaults**
- **BUT based on a grammar of graphics rather than a list of chart functions**

* also called trellis / lattice / small multiple / facet plots

Why ggplot2?

- Modular system allows low level control with ease of a relatively high level system
- Intentionally extendable -- approximately 200 packages on CRAN that begin with "gg"
- Ability to create very professional, beautiful, publication ready plots
- Large, active community of users

Building block approach

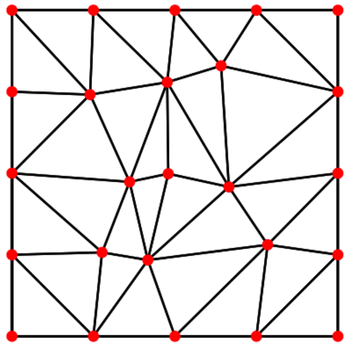


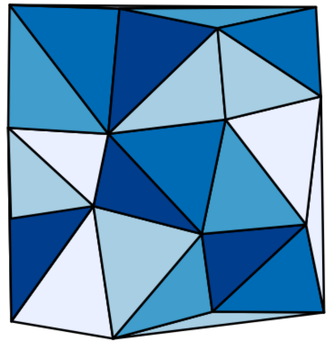
https://rud.is/b/2016/02/14/making-faceted-heatmaps-with-ggplot2/

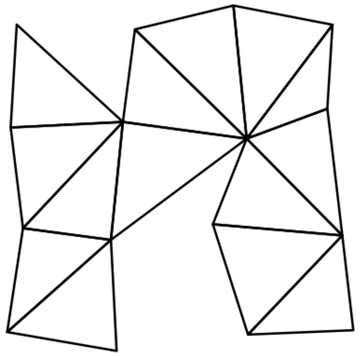
I still use base R graphics

- One dimensional graphs (***vectors***):
  `hist(x), stem(x), boxplot(x), barplot(x)`

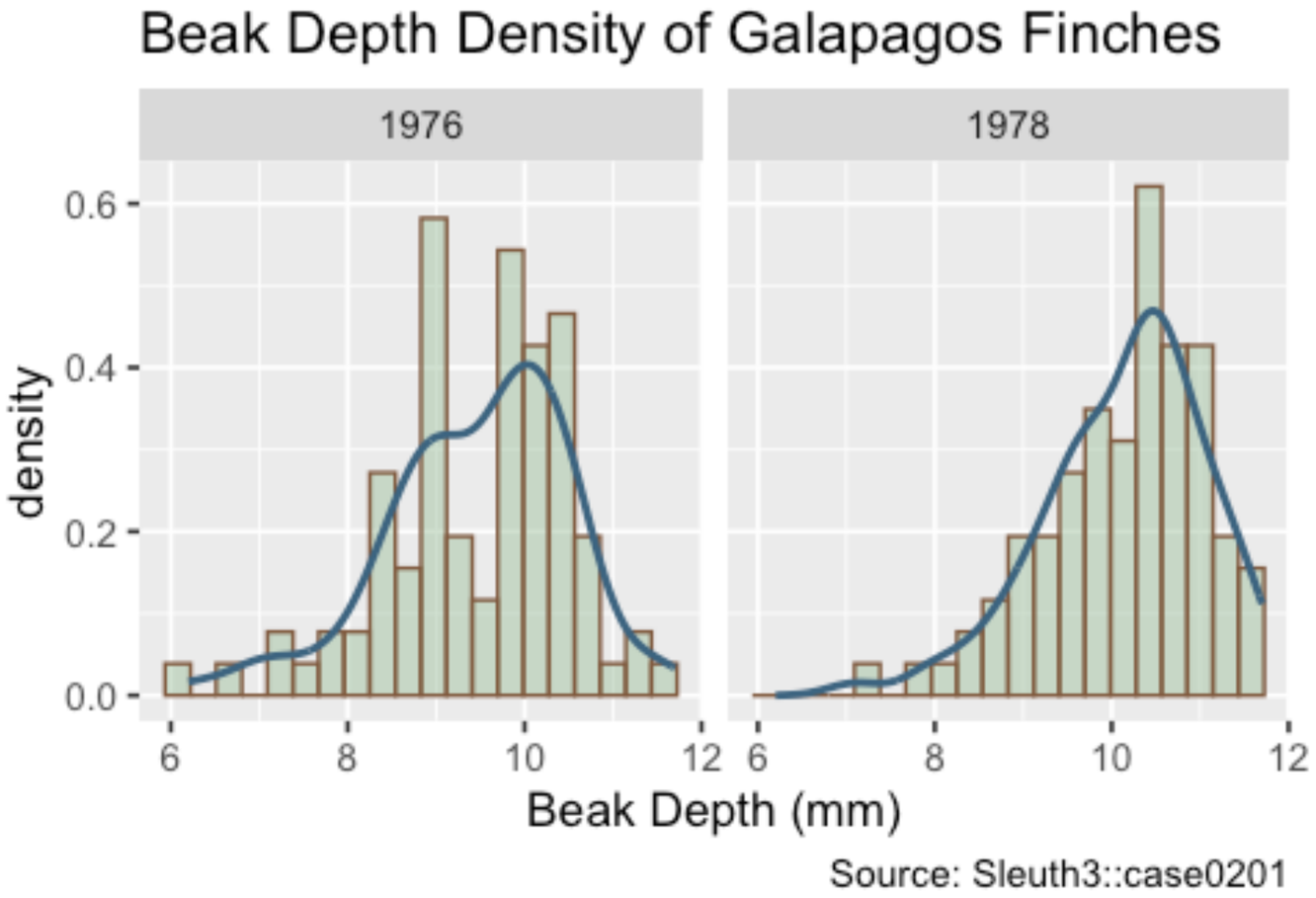
- Graphics without real data







ggplot2 example



ggplot2 example code

```
library(ggplot2)
finches <- Sleuth3::case0201
ggplot(finches, aes(x = Depth, y = after_stat(density))) +
  geom_histogram(bins = 20, color = "#80593D",
                 fill = "#9FC29F", alpha = .5) +
  geom_density(color = "#3D6480", lwd = 1) +
  facet_wrap(~Year) +
  labs(title = "Beak Depth Density of Galapagos Finches",
       x = "Beak Depth (mm)",
       caption = "Source: Sleuth3::case0201") +
  theme_grey(13)
```

Tidyverse

R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

tidyverse.org