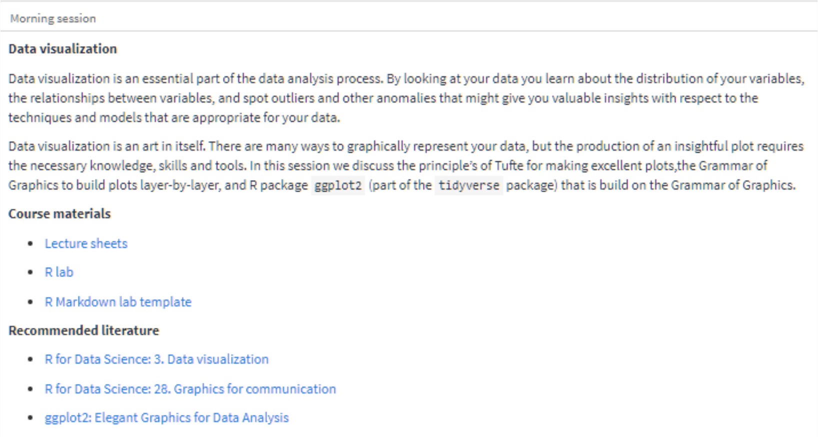
S31: Data Analysis

Data Visualization

Maarten Cruyff

Program



Morning session

Data visualization

Data visualization is an essential part of the data analysis process. By looking at your data you learn about the distribution of your variables, the relationships between variables, and spot outliers and other anomalies that might give you valuable insights with respect to the techniques and models that are appropriate for your data.

Data visualization is an art in itself. There are many ways to graphically represent your data, but the production of an insightful plot requires the necessary knowledge, skills and tools. In this session we discuss the principle's of Tufte for making excellent plots,the Grammar of Graphics to build plots layer-by-layer, and R package `ggplot2` (part of the `tidyverse` package) that is build on the Grammar of Graphics.

Course materials

- Lecture sheets
- R lab
- R Markdown lab template

Recommended literature

- R for Data Science: 3. Data visualization
- R for Data Science: 28. Graphics for communication
- ggplot2: Elegant Graphics for Data Analysis

Content

Data visualization

1. Exploratory data analysis
2. Tufte's Principles of Graphical Excellence
3. Grammar of Graphics
4. Some examples
5. Lab preview

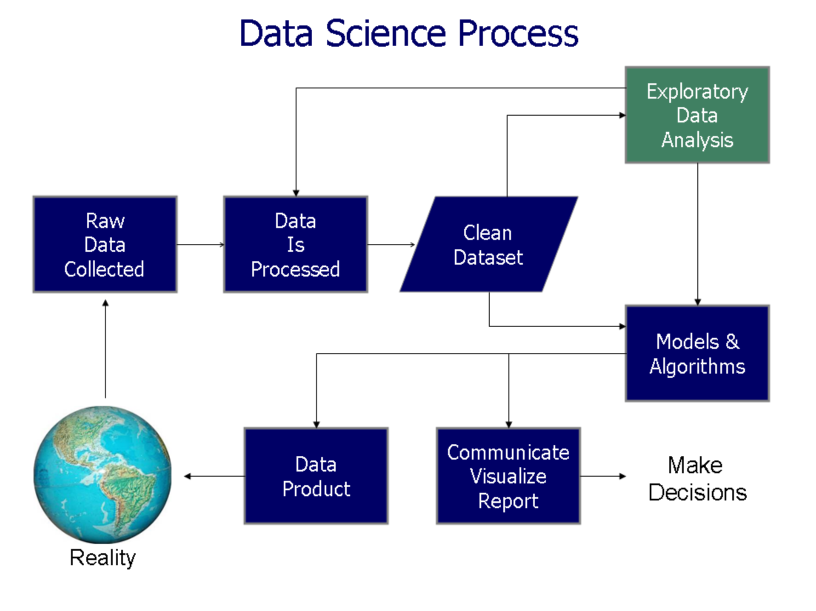
What’s data visualization?

Communication of data by encoding it as visual objects, i.e.

- dots, lines, bars, etc.

to make data more accessible, understandable and usable.

Integral part of data science process



Exploratory data analysis

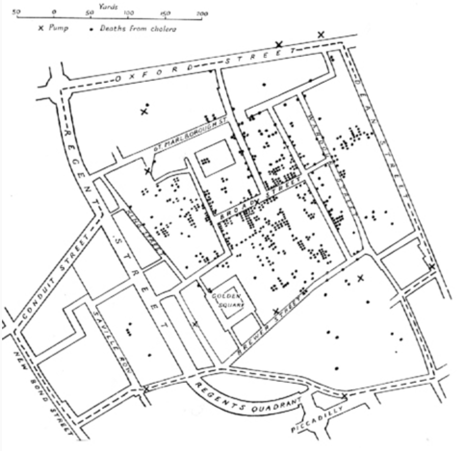
Visualize variation and covariation in your data

- discover unexpected patterns in your data
- detect outliers and other anomalies
- understand your data

Get new ideas about your data!

Early example

The 1854 Soho cholera outbreak was not due to ‘miasma’ (John Snow)



Tufte's Principles of Graphical Excellence

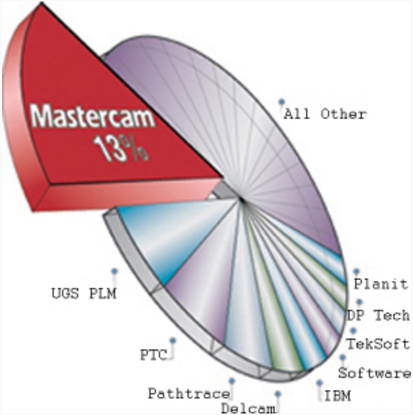
Guidelines for representing visual information

“Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.” **Edward R. Tufte, The Visual Display of Quantitative Information**

- Proportionality principle
- Maximize data-to-ink ratio
- Omit chart junk

Proportionality principle

- proportions should correspond to area/surface



Early example

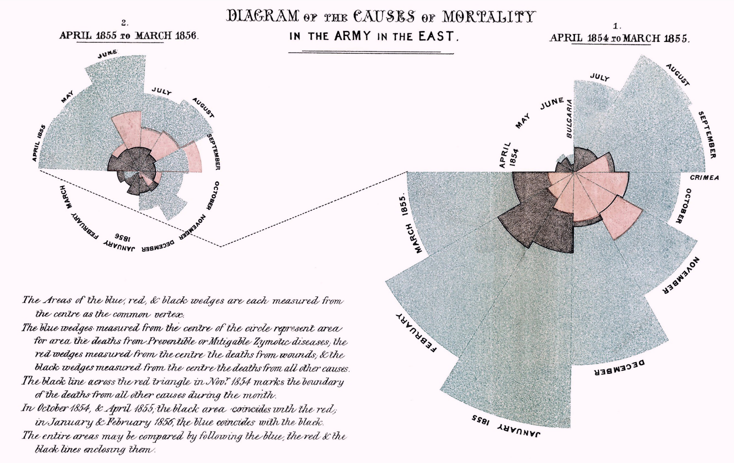


DIAGRAM of the CAUSES of MORTALITY

IN THE ARMY IN THE EAST.

The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex.

The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Mitigable Zymotic diseases; the red wedges measured from the centre the deaths from wounds; & the black wedges measured from the centre the deaths from all other causes.

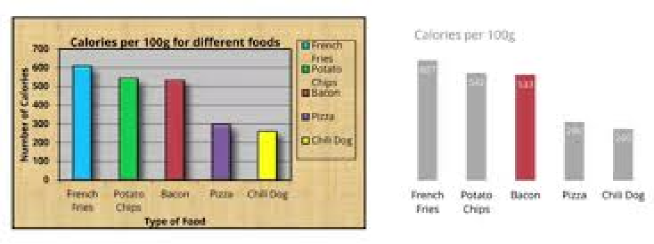
The black line across the red triangle in Nov.r 1854 marks the boundary of the deaths from all other causes during the month.

In October 1854, & April 1855, the black area coincides with the red; in January & February 1856, the blue coincides with the black.

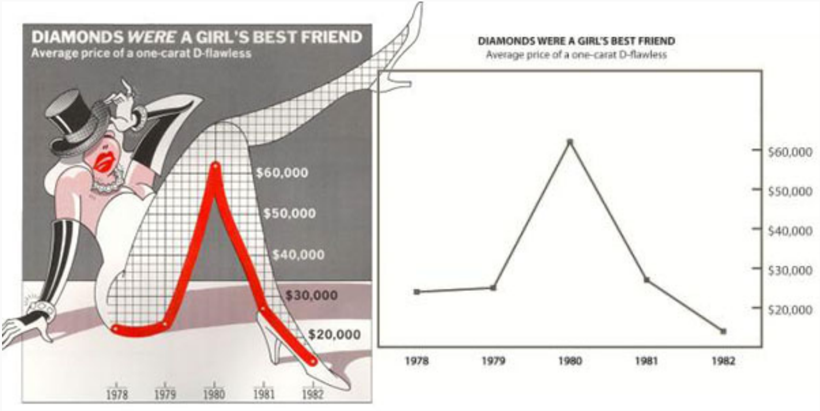
The entire areas may be compared by following the blue, the red & the black lines enclosing them.

Maximize data-to-ink ratio

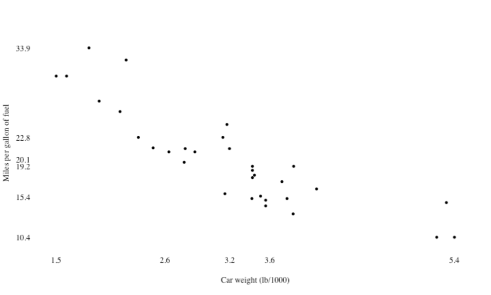
- use of colors and text sparsely



Omit chart junk



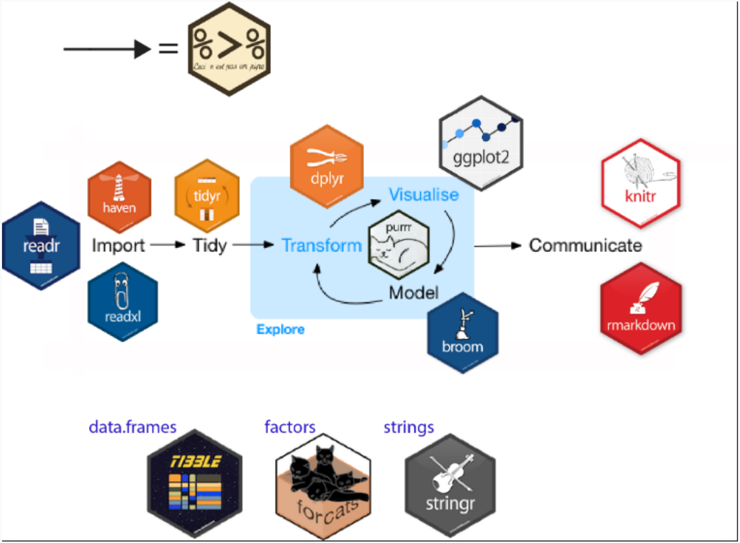
Tufte in practice



Grammar of Graphics

R data science packages

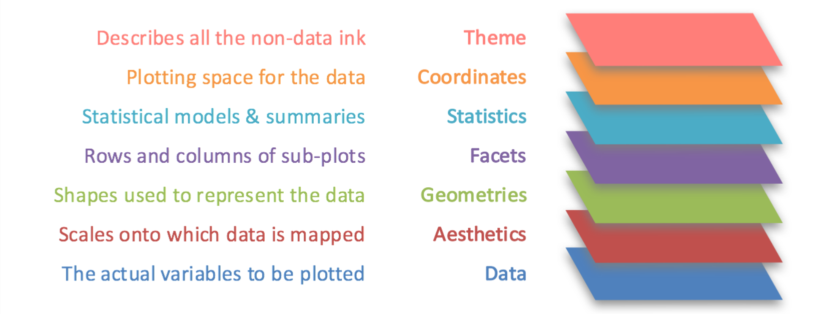
The tidyverse package



The ggplot2 package

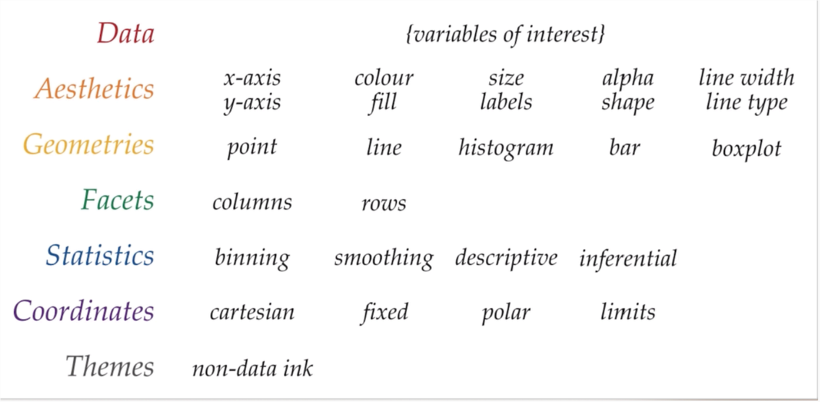
Grammar of Graphics

- Build plots layer-by-layer



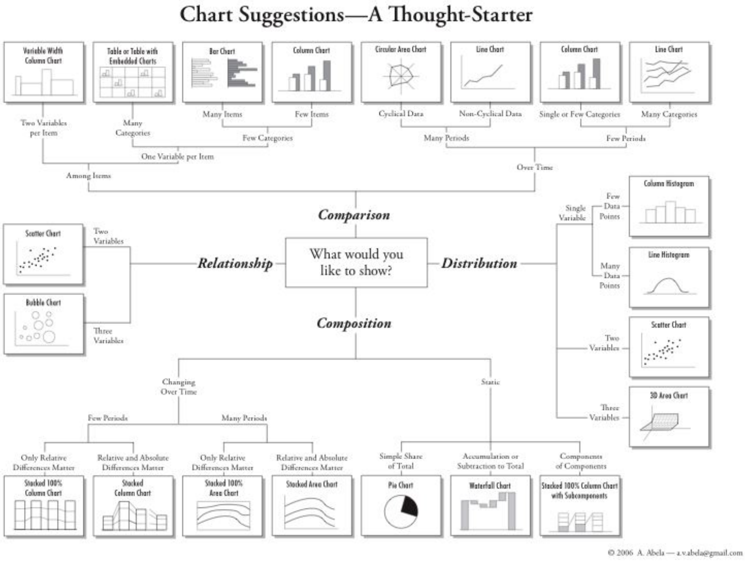
Describes all the non-data ink	Theme
Plotting space for the data	Coordinates
Statistical models & summaries	Statistics
Rows and columns of sub-plots	Facets
Shapes used to represent the data	Geometries
Scales onto which data is mapped	Aesthetics
The actual variables to be plotted	Data

Content of building blocks



Data	{variables of interest}				
Aesthetics	x-axis y-axis	colour fill	size labels	alpha shape	line width line type
Geometries	point	line	histogram	bar	boxplot
Facets	columns	rows			
Statistics	binning	smoothing	descriptive	inferential	
Coordinates	cartesian	fixed	polar	limits	
Themes	non-data ink				

Translation to aesthetics, geoms, etc.



Using `ggplot()`

Data and aesthetics

The function `ggplot()` has two main arguments:

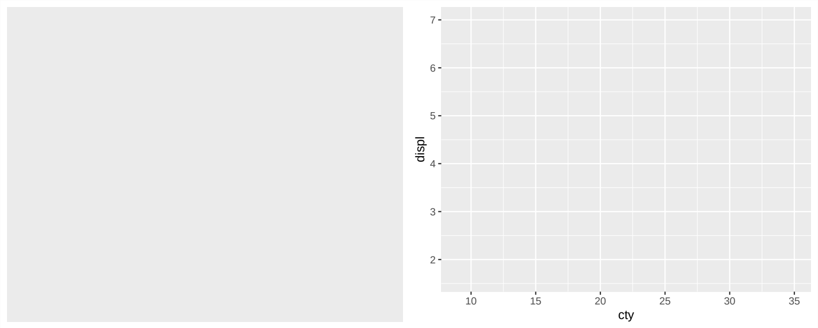
```
ggplot(data = <data>, mapping = aes(x = <var>, y = <var>, ...))
```

- `<data>` name of the data set
- `mapping` maps variables to aesthetics (axis, color, group, etc.)

Example

- `ggplot(mpg)` creates an empty plot array for data set `mpg`
- `aes(x = cty, y = displ)` maps variable values to axes

```
grid.arrange(
  ggplot(mpg),
  ggplot(mpg, aes(x = cty, y = displ)),
nrow = 1)
```



Geometrics

Geoms define shapes for representation (lines, points, bars, etc.)

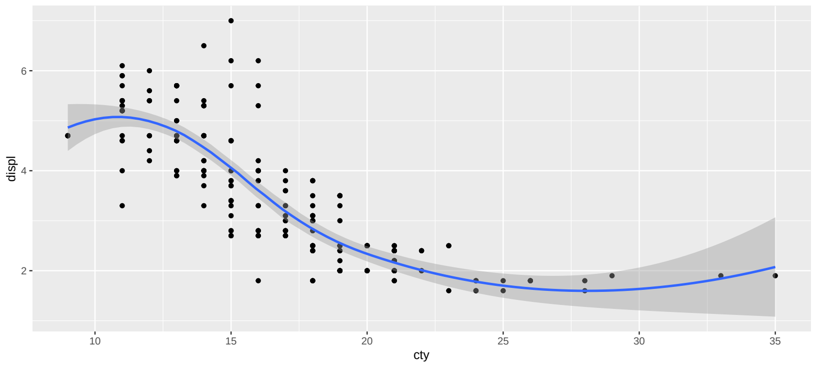
- geoms are added with + sign
- multiple geoms can be added to same plot (e.g points and lines)

```
ggplot(data = <data>, mapping = aes(x = <var>, y = <var>)) +
  geom_point() +
  geom_line()
```

Example

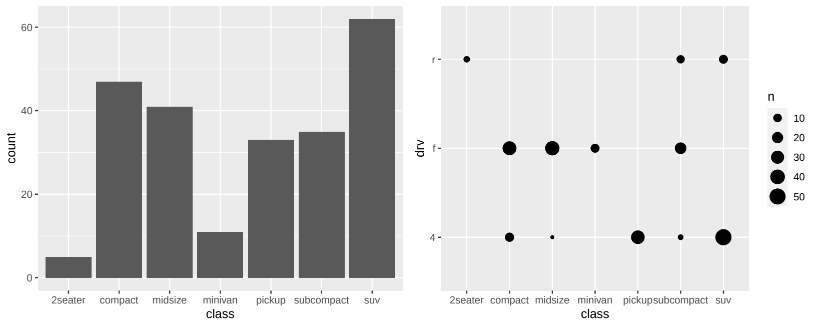
Scatter plot plus regression line

```
ggplot(mpg, aes(x = cty, y = displ)) +
  geom_point() +
  geom_smooth()
```



Another example

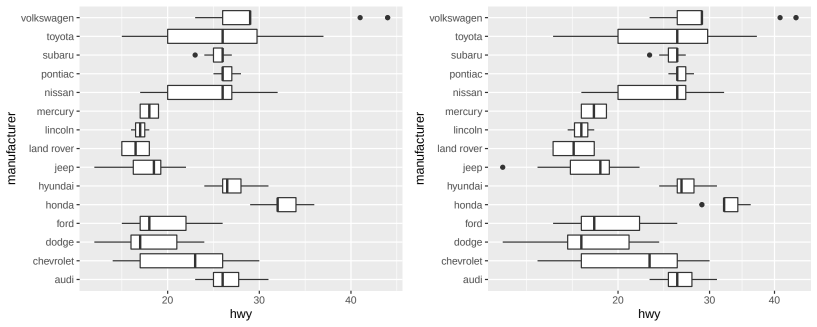
```
grid.arrange(

  ggplot(mpg) +
    geom_bar(aes(class)),

  ggplot(mpg) +
    geom_count(aes(class, drv)),

  nrow = 1)
```



Coordinates and scales

Flip the axis or transform the scales of the axes

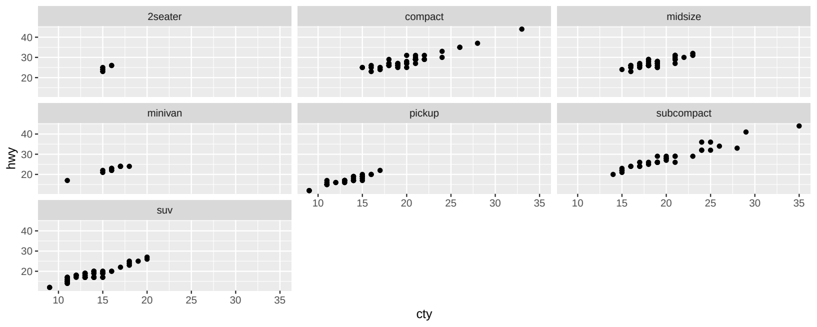
```
grid.arrange(
  ggplot(mpg) + geom_boxplot(aes(manufacturer, hwy)) + coord_flip(),
  ggplot(mpg) + geom_boxplot(aes(manufacturer, hwy)) + coord_flip() +
    scale_y_log10(),
nrow = 1)
```



Facets

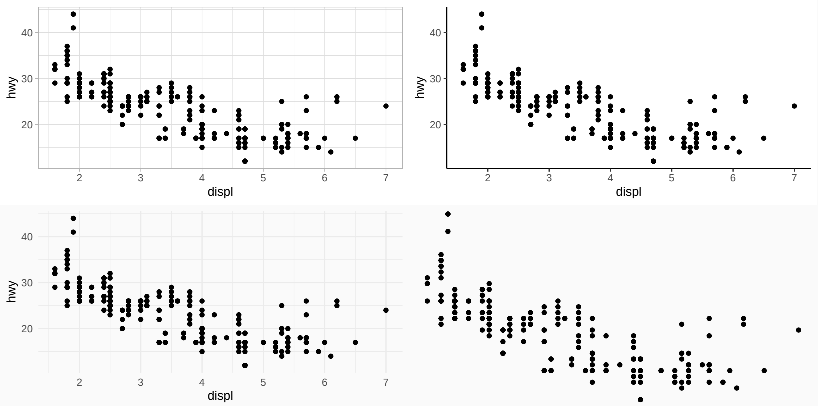
Make separate plots for the levels of a categorical variable (`factor`)

```
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point() +
  facet_wrap(vars(class))
```



Themes

```r
grid.arrange(
  ggplot(mpg) + geom_point(aes(displ, hwy)) + theme_light(),
  ggplot(mpg) + geom_point(aes(displ, hwy)) + theme_classic(),
  ggplot(mpg) + geom_point(aes(displ, hwy)) + theme_minimal(),
  ggplot(mpg) + geom_point(aes(displ, hwy)) + theme_void(),
  nrow = 2)
```



Preview lab 1A

1. Make plots with `ggplot2`
2. Combine aesthetics, geoms, facets, themes

Make the exercises in the R Markdown template

- Open template in RStudio and read the instructions
- Save the file in an appropriate folder
- Insert R code in the R chunks
- Run the chunks to test for errors
- Knit the HTML file when the code is error free