Data visualization

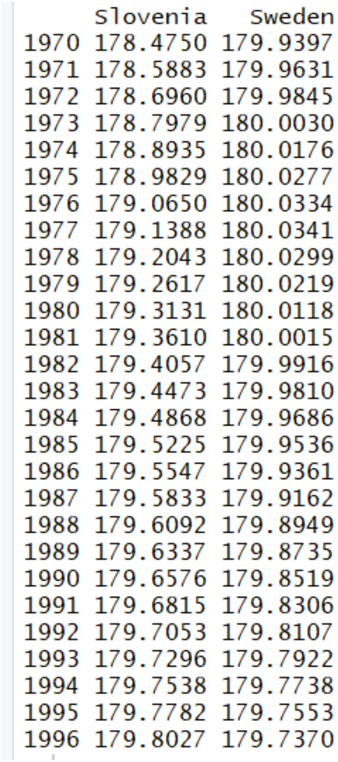
The science and art of communicating information more efficiently and effectively by representing it visually.

The wooden cabin stood alone on the edge of a snow-covered field against the backdrop of a lush pine forest. The rising mist obscured the sun, but could not hide the towering mountains in the background.

This is what it looked like.

Average height of 18-year old men

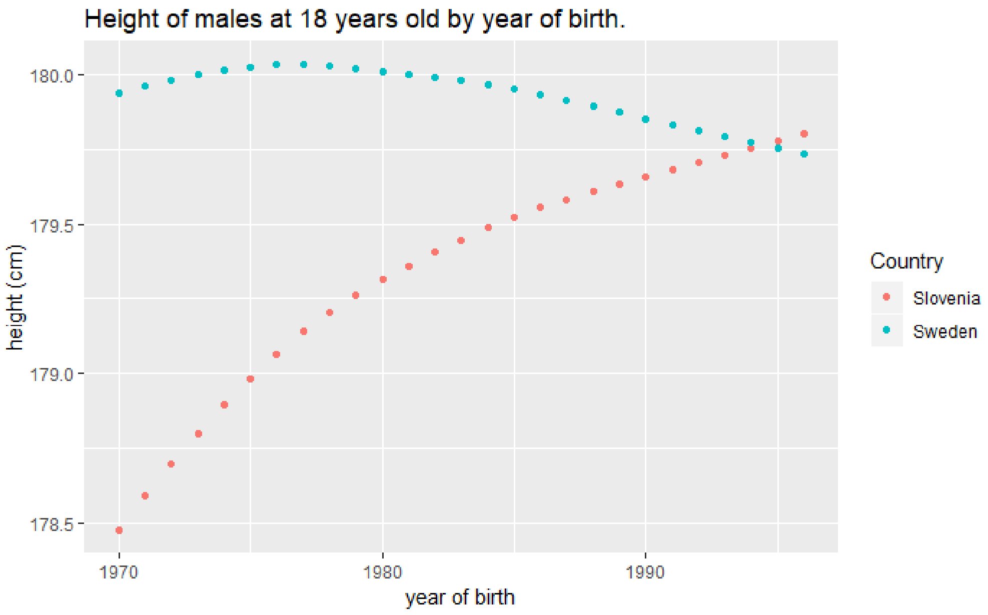
How does Slovenia compare with Sweden over time?



```
     Slovenia   Sweden
1970 178.4750 179.9397
1971 178.5883 179.9631
1972 178.6960 179.9845
1973 178.7979 180.0030
1974 178.8935 180.0176
1975 178.9829 180.0277
1976 179.0650 180.0334
1977 179.1388 180.0341
1978 179.2043 180.0299
1979 179.2617 180.0219
1980 179.3131 180.0118
1981 179.3610 180.0015
1982 179.4057 179.9916
1983 179.4473 179.9810
1984 179.4868 179.9686
1985 179.5225 179.9536
1986 179.5547 179.9361
1987 179.5833 179.9162
1988 179.6092 179.8949
1989 179.6337 179.8735
1990 179.6576 179.8519
1991 179.6815 179.8306
1992 179.7053 179.8107
1993 179.7296 179.7922
1994 179.7538 179.7738
1995 179.7782 179.7553
1996 179.8027 179.7370
```

Average height of 18-year old men

How does Slovenia compare with Sweden over time?



Data visualization

- General principles
  (grammar of graphics, common types of plots, best practices)

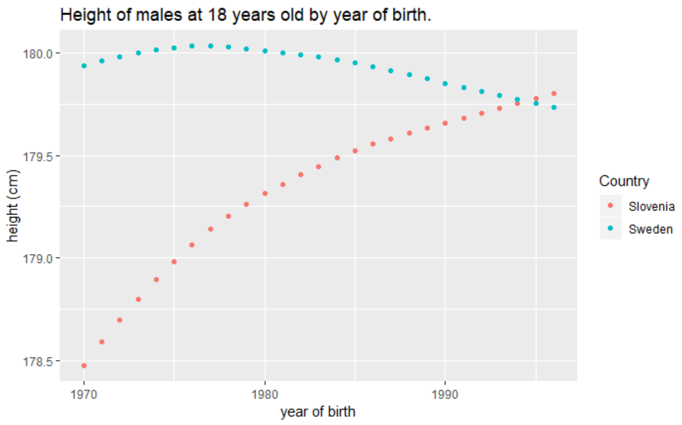
- Tools
  (R + ggplot2)

Grammar of graphics

= Systematically breaking down statistical graphics into (independent) components that can be used to describe plots in a concise and flexible way:

- Data & Mapping to plot aesthetics
- Geometric object
  (point, line, bar…)
- Statistical transformation
  (boxplot, bin, density)
- Position adjustment
- Scales
- Coordinate system
- Grouping (faceting)

a layer

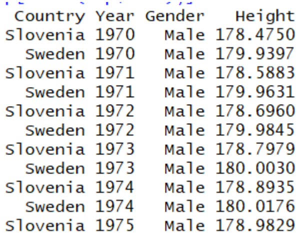


Height of males at 18 years old by year of birth.

Wickham, H. (2010). A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, *19*(1), 3-28.

Grammar of graphics

data (data.frame)



```
  Country Year Gender   Height
 Slovenia 1970   Male 178.4750
   Sweden 1970   Male 179.9397
 Slovenia 1971   Male 178.5883
   Sweden 1971   Male 179.9631
 Slovenia 1972   Male 178.6960
   Sweden 1972   Male 179.9845
 Slovenia 1973   Male 178.7979
   Sweden 1973   Male 180.0030
 Slovenia 1974   Male 178.8935
   Sweden 1974   Male 180.0176
 Slovenia 1975   Male 178.9829
```

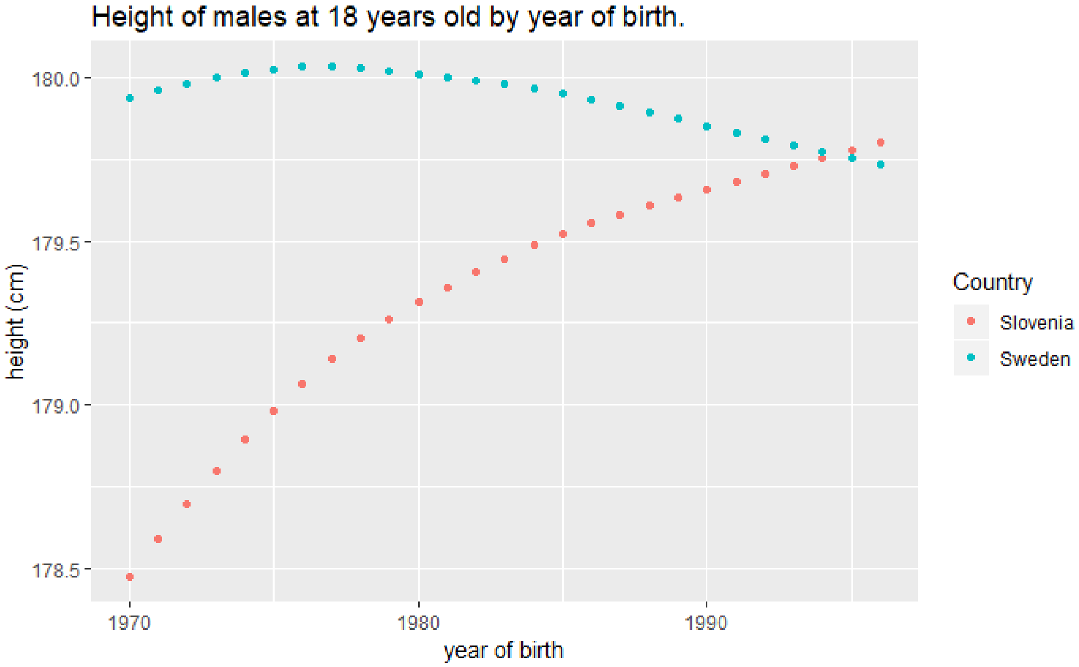
aesthetics mapping

Year -> x-axis,
Height -> y-axis,
Country -> colour

(there are others, such as shape, size, fill…)

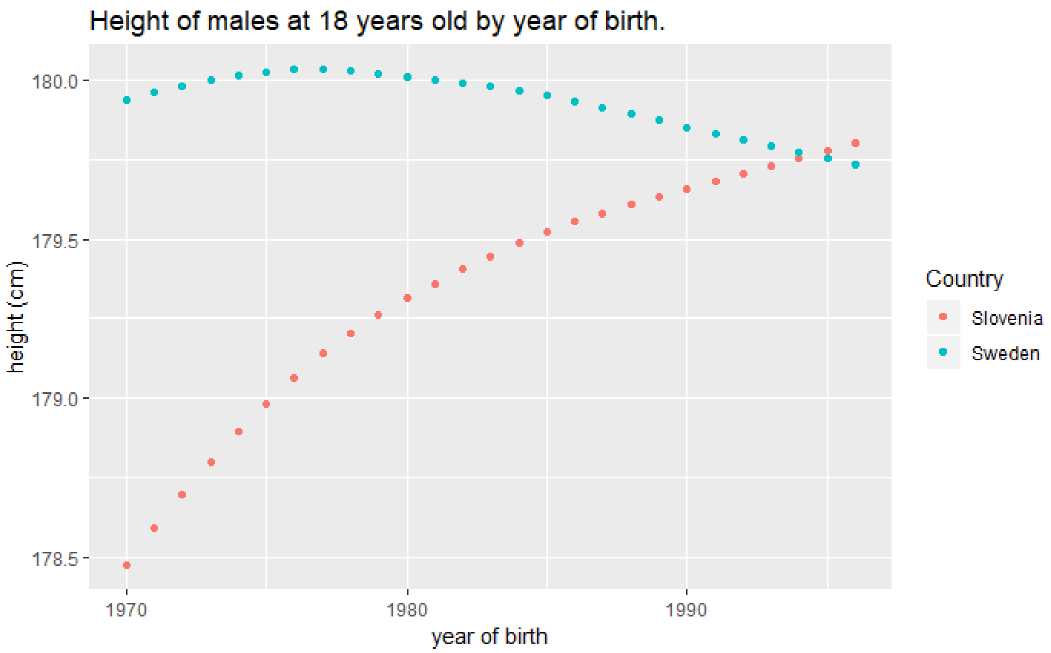
geometric object (geom)

point



ggplot2

= An implementation of
the grammar of graphics.



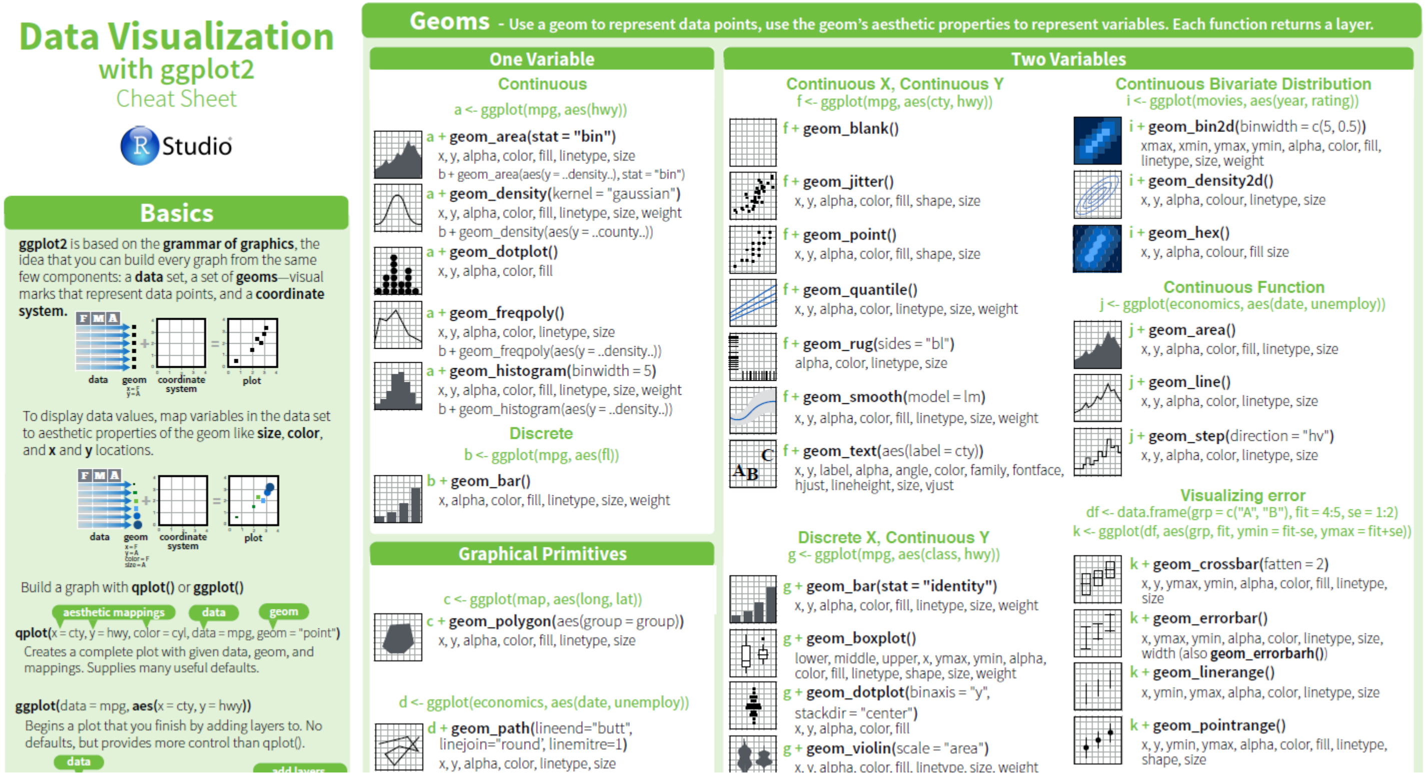
An explicit use of the grammar:

```
ggplot() +
layer(data = tmp, geom = "point", mapping = aes(x = Year, y = Height, colour = Country), stat = "identity", position = "identity") +
ggtitle("Height of males at 18 years old by year of birth.") + ylab("height (cm)") + xlab("year of birth")
```

ggplot2 implements “shorthand” instructions for common plots:

```
ggplot(tmp, aes(x = Year, y = Height, colour = Country)) + geom_point() + ggtitle("Height of males at
18 years old by year of birth.") + ylab("height (cm)") + xlab("year of birth")
```

ggplot2 cheat sheet



Data Visualization
with ggplot2
Cheat Sheet

RStudio

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms**—visual marks that represent data points, and a **coordinate system.**

To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.

Build a graph with **qplot()** or **ggplot()**

qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

ggplot(data = mpg, **aes**(x = cty, y = hwy))
Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

Continuous

a <- ggplot(mpg, aes(hwy))

a + **geom_area(stat = "bin")**
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")

a + **geom_density(**kernel = "gaussian"**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))

a + **geom_dotplot()**
x, y, alpha, color, fill

a + **geom_freqpoly()**
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

a + **geom_histogram(**binwidth = 5**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

b <- ggplot(mpg, aes(fl))

b + **geom_bar()**
x, alpha, color, fill, linetype, size, weight

Graphical Primitives

c <- ggplot(map, aes(long, lat))

c + **geom_polygon(**aes(group = group)**)**
x, y, alpha, color, fill, linetype, size

d <- ggplot(economics, aes(date, unemploy))

d + **geom_path(**lineend="butt", linejoin="round', linemitre=1**)**
x, y, alpha, color, linetype, size

Two Variables

Continuous X, Continuous Y
f <- ggplot(mpg, aes(cty, hwy))

f + **geom_blank()**

f + **geom_jitter()**
x, y, alpha, color, fill, shape, size

f + **geom_point()**
x, y, alpha, color, fill, shape, size

f + **geom_quantile()**
x, y, alpha, color, linetype, size, weight

f + **geom_rug(**sides = "bl"**)**
alpha, color, linetype, size

f + **geom_smooth(**model = lm**)**
x, y, alpha, color, fill, linetype, size, weight

f + **geom_text(**aes(label = cty)**)**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y
g <- ggplot(mpg, aes(class, hwy))

g + **geom_bar(stat = "identity")**
x, y, alpha, color, fill, linetype, size, weight

g + **geom_boxplot()**
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight

g + **geom_dotplot(**binaxis = "y", stackdir = "center"**)**
x, y, alpha, color, fill

g + **geom_violin(**scale = "area"**)**
x, y, alpha, color, fill, linetype, size, weight

Continuous Bivariate Distribution
i <- ggplot(movies, aes(year, rating))

i + **geom_bin2d(**binwidth = c(5, 0.5)**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

i + **geom_density2d()**
x, y, alpha, colour, linetype, size

i + **geom_hex()**
x, y, alpha, colour, fill size

Continuous Function
j <- ggplot(economics, aes(date, unemploy))

j + **geom_area()**
x, y, alpha, color, fill, linetype, size

j + **geom_line()**
x, y, alpha, color, linetype, size

j + **geom_step(**direction = "hv"**)**
x, y, alpha, color, linetype, size

Visualizing error
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

k + **geom_crossbar(**fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, linetype, size

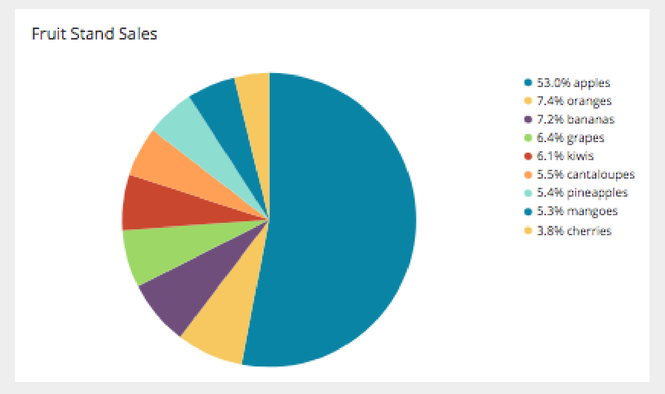
k + **geom_errorbar()**
x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)

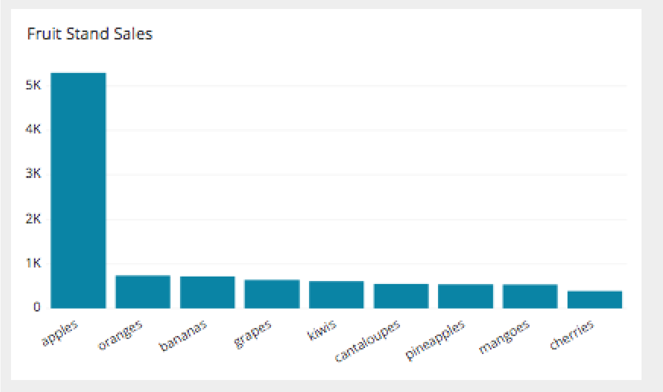
k + **geom_linerange()**
x, ymin, ymax, alpha, color, linetype, size

k + **geom_pointrange()**
x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

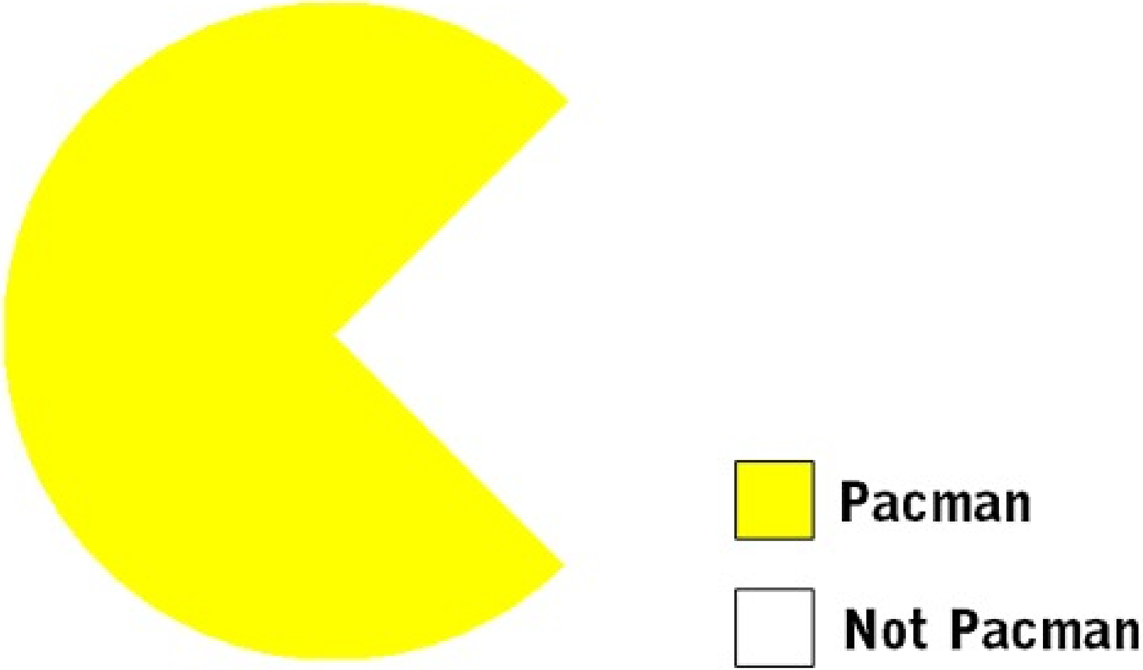
https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Don’t use pie charts!





- People are not good at judging/comparing angles and non-rectangular areas.
- A bar chart is always more appropriate.
- There is, however, one exception where a pie chart is clearly the best choice…



Pacman
Not Pacman