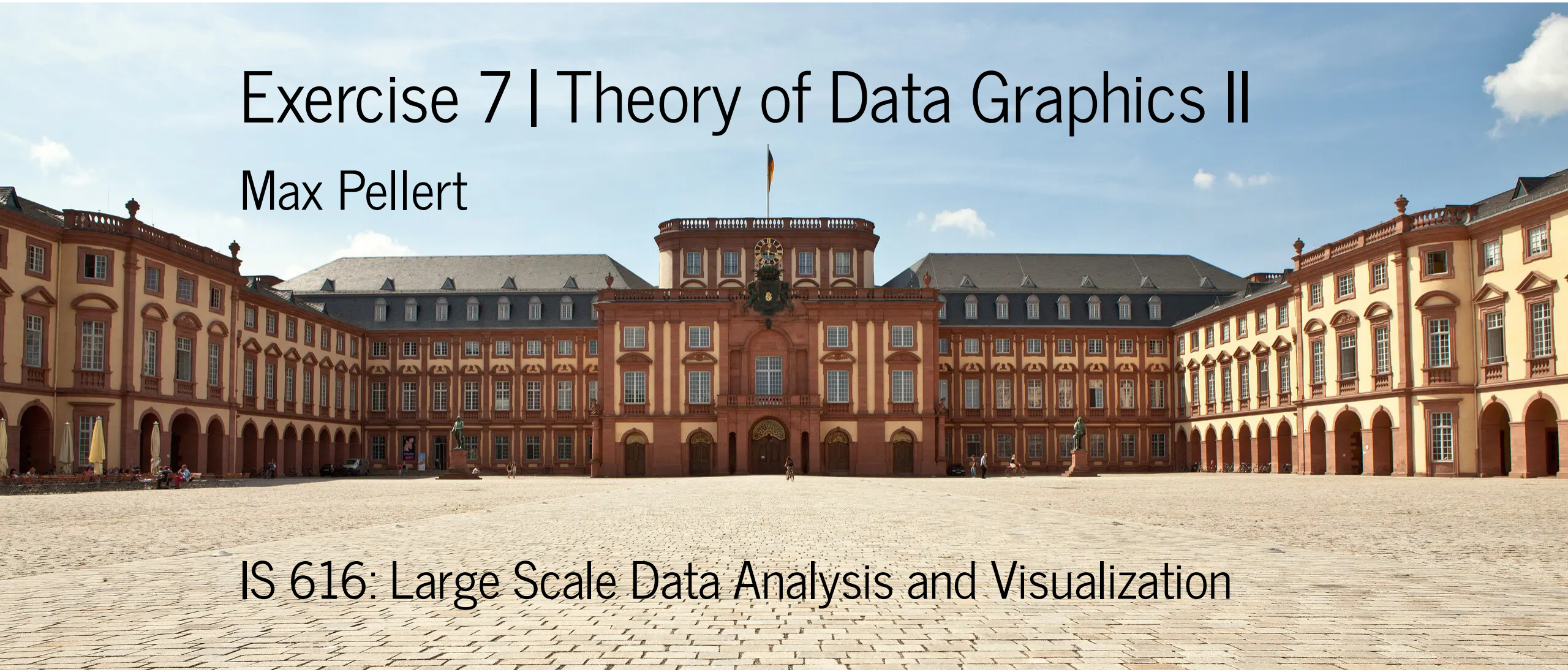
Exercise 7 | Theory of Data Graphics II

Max Pellert



IS 616: Large Scale Data Analysis and Visualization

Accessible Complexity: The Friendly Data Graphic

An occasional data graphic displays such care in design that it is particularly accessible and open to the eye, as if the designer had the viewer in mind at every turn while constructing the graphic. This is the *friendly data graphic*.

Friendly

Words are spelled out, mysterious and elaborate encoding avoided

Words run from left to right, the usual direction for reading occidental languages

Little messages help explain data

Elaborately encoded shadings, crosshatching, and colors are avoided; instead, labels are placed on the graphic itself; no legend is required

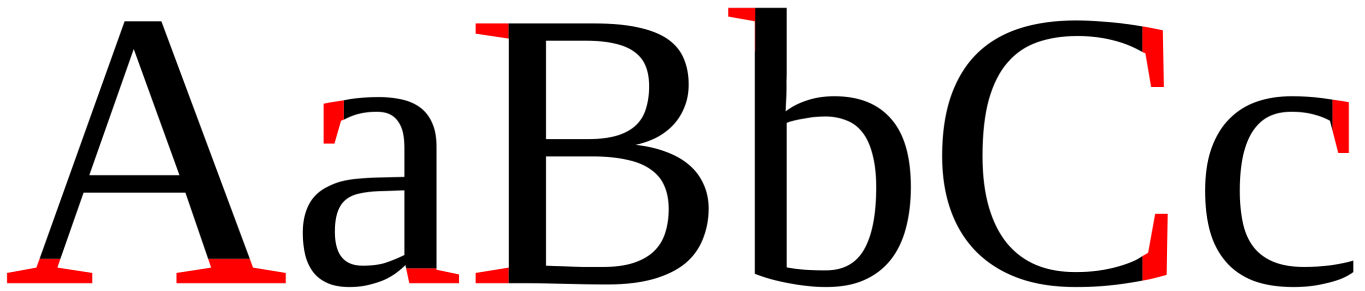
Graphic attracts viewer, provokes curiosity

Friendly

Colors, if used, are chosen so that the color-deficient and color-blind (5 to 10% of viewers) can make sense of the graphic (blue can be distinguished from other colors by most color-deficient people)

Type is clear, precise, modest; lettering may be done by hand

Type is upper-and-lower case, with serifs



Unfriendly

Abbreviations abound, requiring the viewer to sort through text to decode abbreviations

Words run vertically, particularly along the Y-axis; words run in several different directions

Graphic is cryptic, requires repeated references to scattered text

Obscure codings require going back and forth between legend and graphic

Unfriendly

Graphic is repellent, filled with chartjunk

Design insensitive to color-deficient viewers; red and green used for essential contrasts

Type is clotted, overbearing

Type is all capitals, sans serif

AaBbCc

Aspect Ratios

When we are working with data graphics, we are usually quite free to choose the dimensions of our graphic (height and width)

In the case of vector graphics, we can choose arbitrary (also very large) dimensions without loss of quality

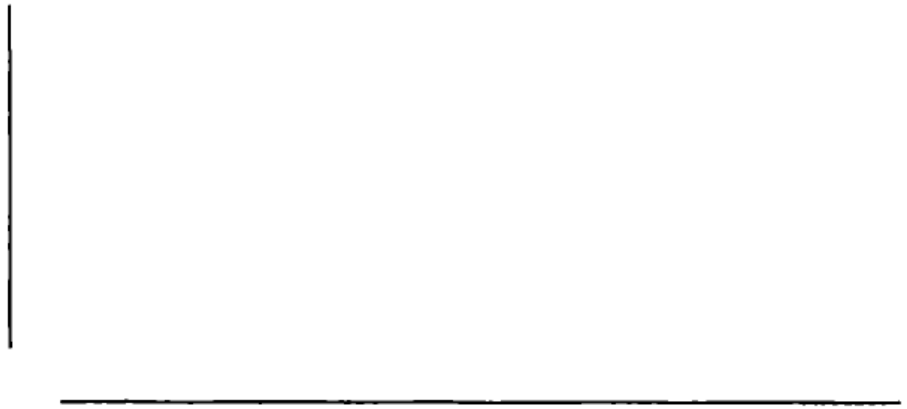
For bitmap graphics, there are limits after which quality noticeable decreases and we don't want that

An important question concerns the ratio of width to height, the **aspect ratio**

Which aspect ratio to choose for one specific data graphic?

Graphics should tend toward the horizontal, greater in length than height:

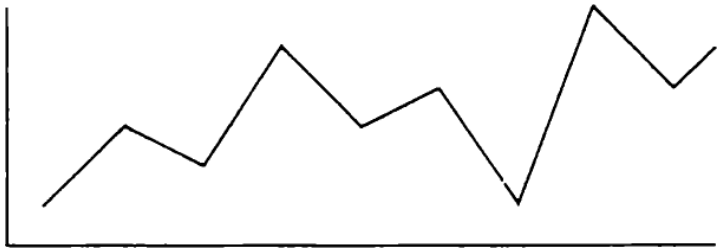
lesser height



greater length

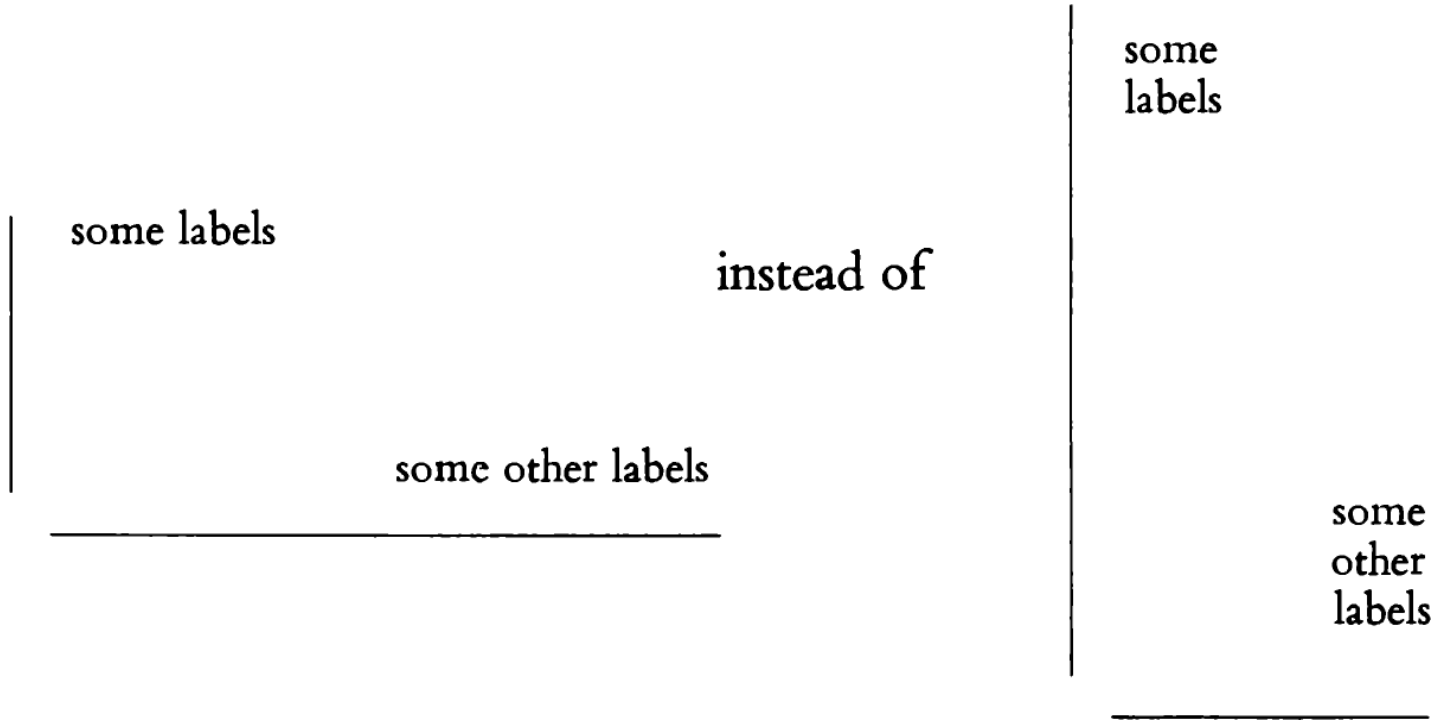
Several lines of reasoning favor horizontal over vertical displays.

First, analogy to the horizon. Our eye is naturally practiced in detecting deviations from the horizon, and graphic design should take advantage of this fact. Horizontally stretched time-series are more accessible to the eye:

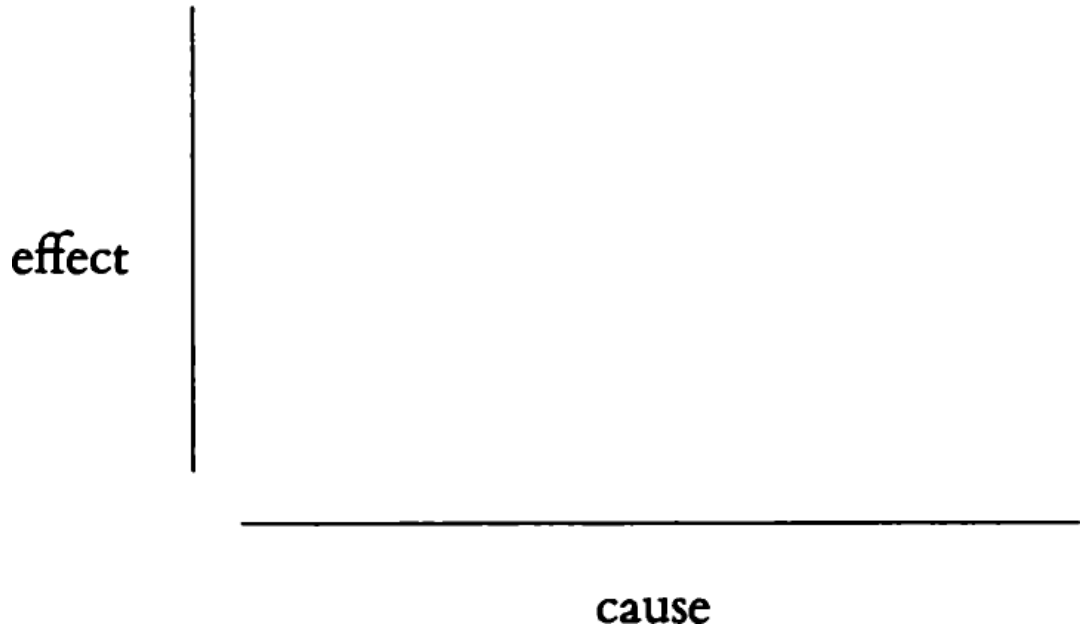


The analogy to the horizon also suggests that a shaded, high contrast display might occasionally be better than the floating snake.

Second, ease of labeling. It is easier to write and to read words that read from left to right on a horizontally stretched plotting-field:

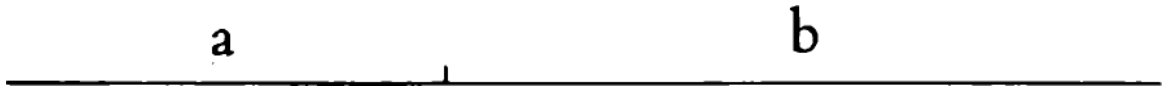


Third, emphasis on causal influence. Many graphics plot, in essence,



and a longer horizontal helps to elaborate the workings of the causal variable in more detail.

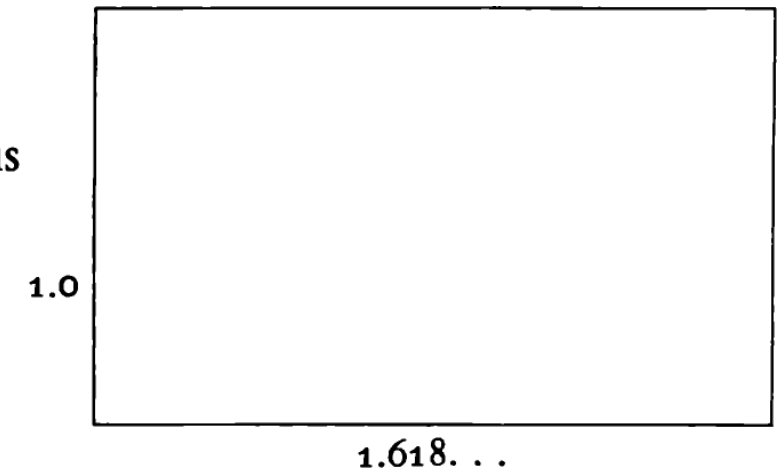
If graphics should tend toward the horizontal rather than the vertical, then how much so? A venerable (fifth-century B.C.) but dubious rule of aesthetic proportion is the Golden Section, a "divine division" of a line.[8] A length is divided such that the smaller is to the greater part as the greater is to the whole:

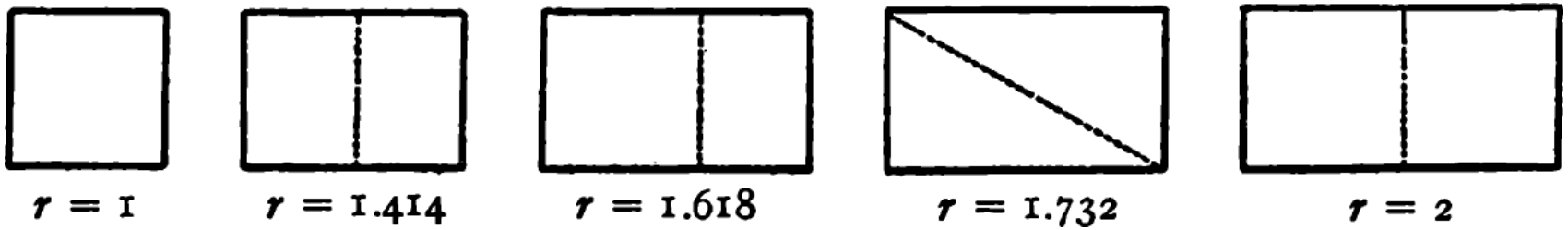


$$\frac{a}{b} = \frac{b}{a + b}$$

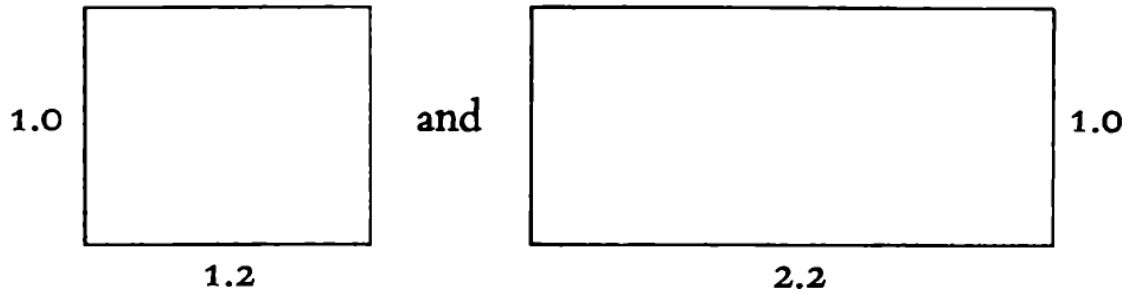
Solving the quadratic when $a = 1$ yields $b = \frac{\sqrt{5} + 1}{2} = 1.618\ldots.$

In turn the Golden Rectangle is

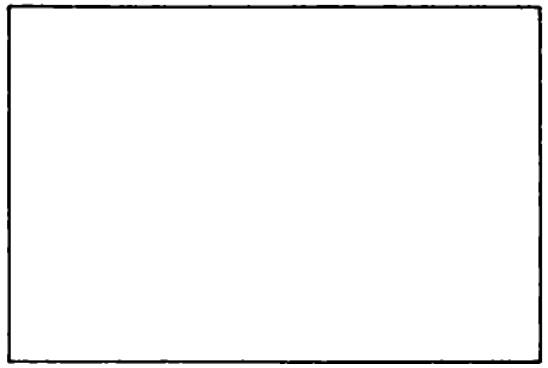




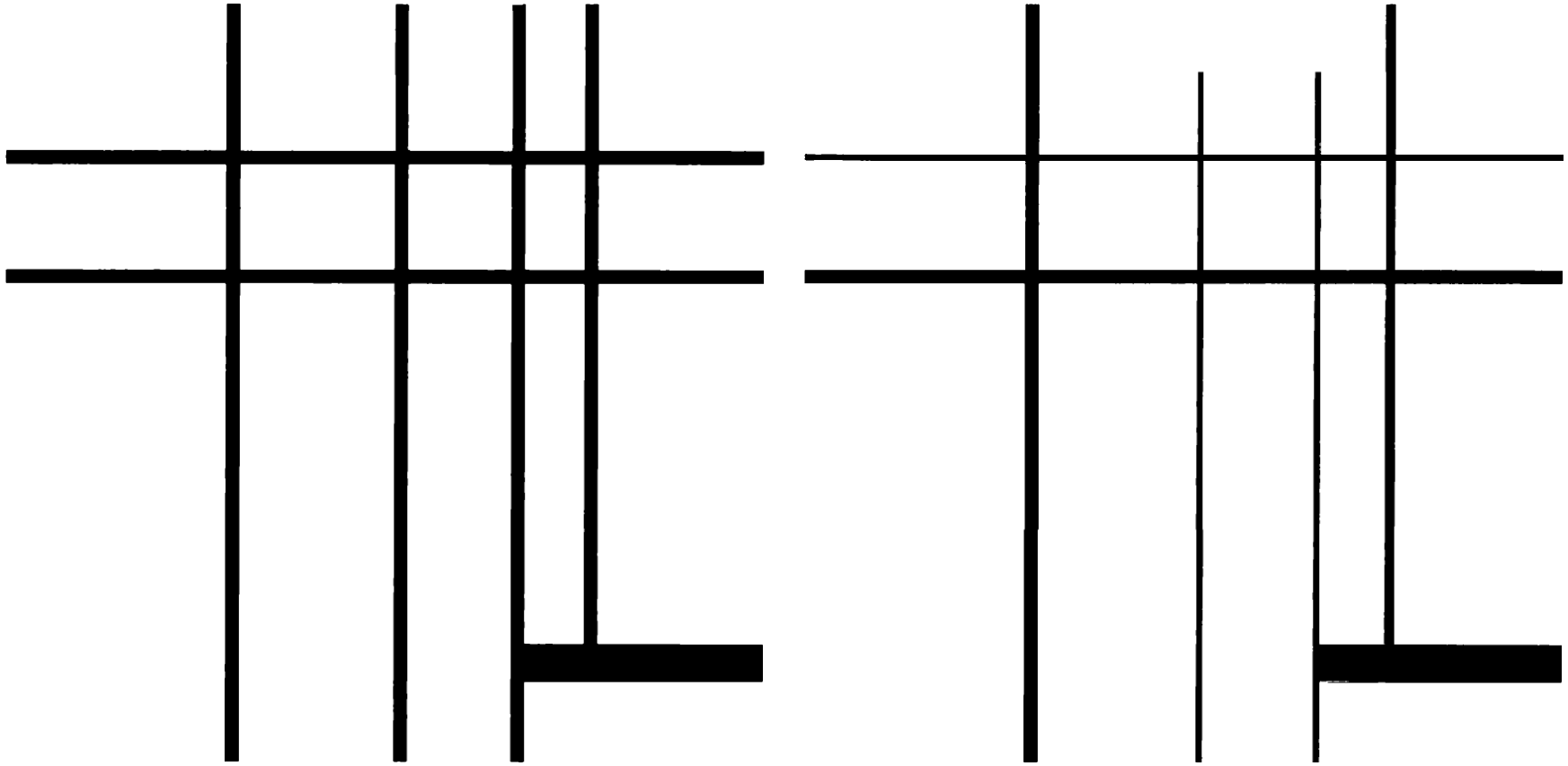
A mild preference for proportions near
to the Golden Rectangle is found among those taking part in the experiments, but the preferred height/length ratios also vary a great deal, ranging between



- If the nature of the data suggests the shape of the graphic, follow that suggestion.
- Otherwise, move toward horizontal graphics about 50 percent wider than tall:



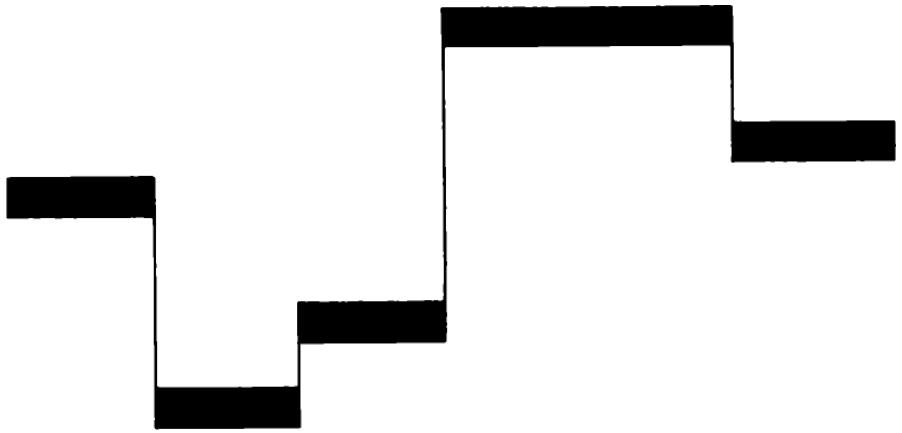
An effective aesthetic device is the orthogonal intersection of lines of different weights:



Line Width

Heavier lines should be a data measure

As an example consider a time series plot:

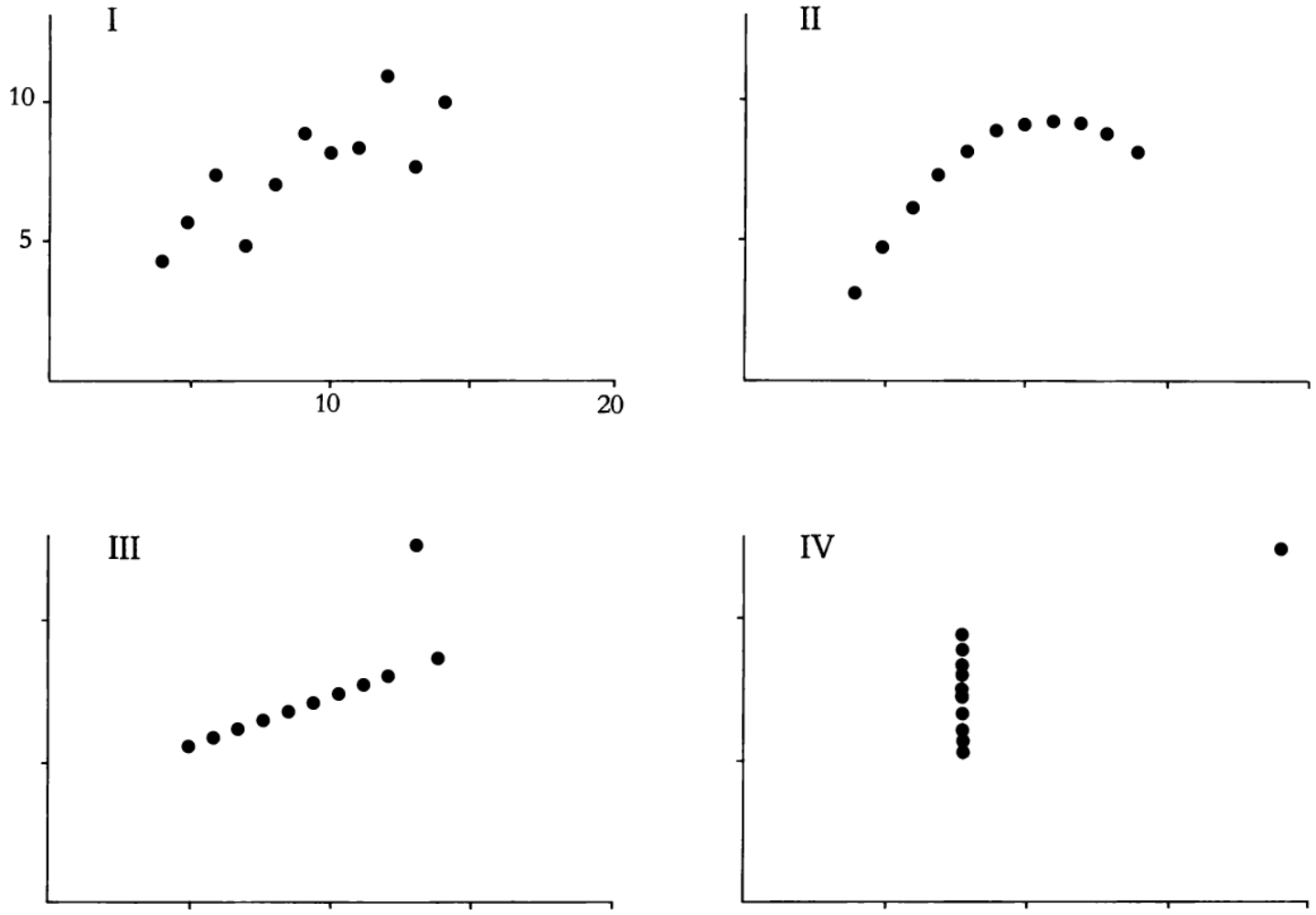


The contrast in line weight represents contrast in meaning. The greater meaning is given to the greater line weight; thus the data line should receive greater weight than the connecting verticals. The logic here is a restatement, in different language, of the principle of data-ink maximization.

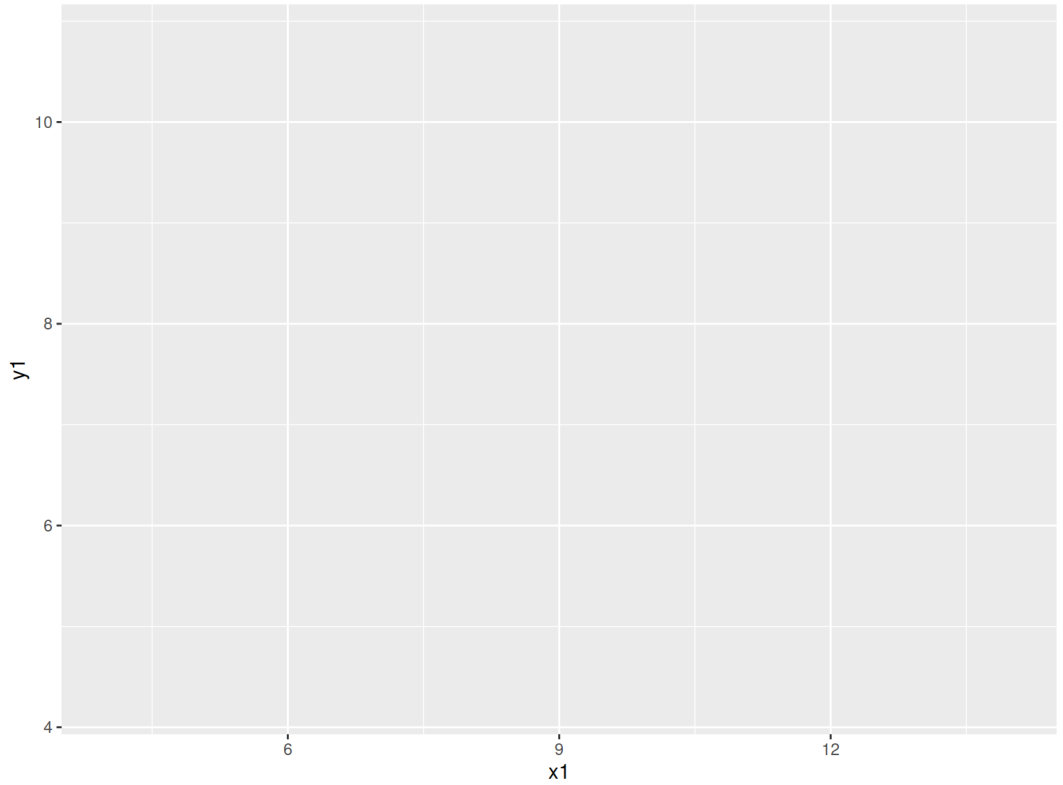
Coding

Increase the ink-data ratio of Anscombes Quartett

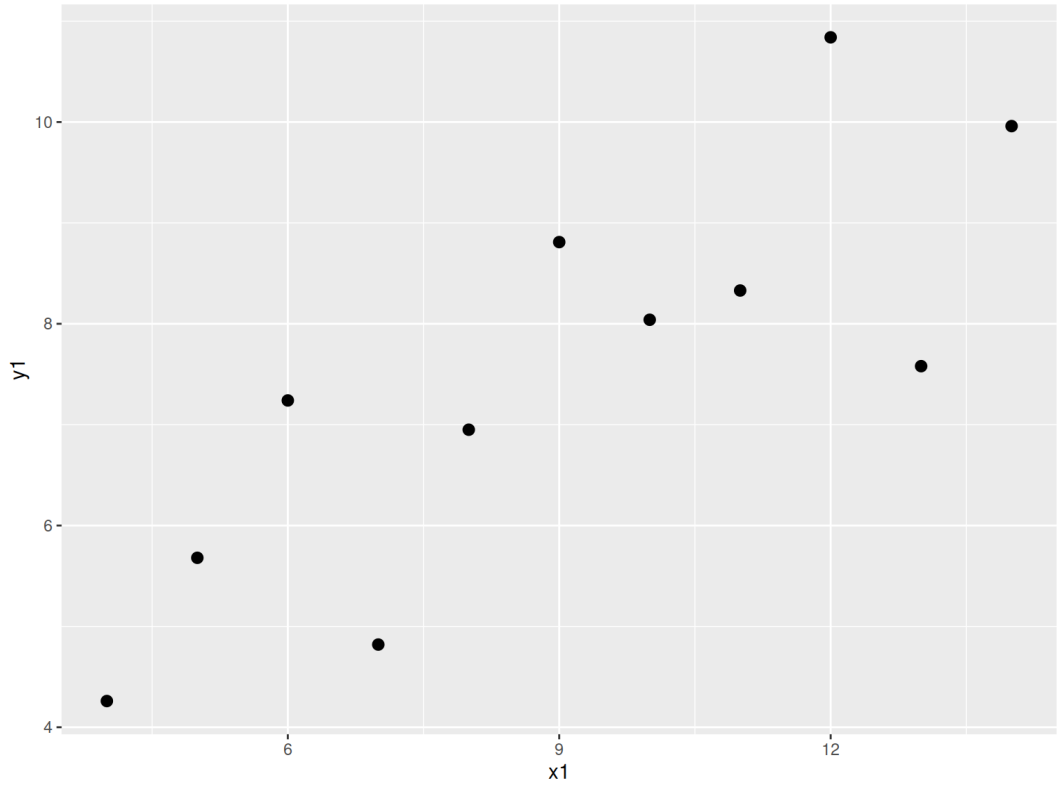
```
anscombe <- datasets::anscombe

# earlier function
# create_plot <- function(dataset_x,dataset_y,size_points=4,size_text=21,
#   ggplot(anscombe,
#          aes({{ dataset_x }},{{ dataset_y }})) +
#   geom_point(
#     size = size_points) +
#   geom_smooth(method="lm", se=F, fullrange = TRUE,
#               color="darkgrey") +
#   scale_x_continuous(
#     breaks = seq(0,20,2)) +
#   scale_y_continuous(
#     breaks = seq(0,14,2)) +
#   expand_limits(x = c(0,20), y = c(0,14)) +
#   labs(x = deparse(substitute(dataset_x)),
#        y = deparse(substitute(dataset_y))) +
```


I
10
5
10
20
II
III
IV

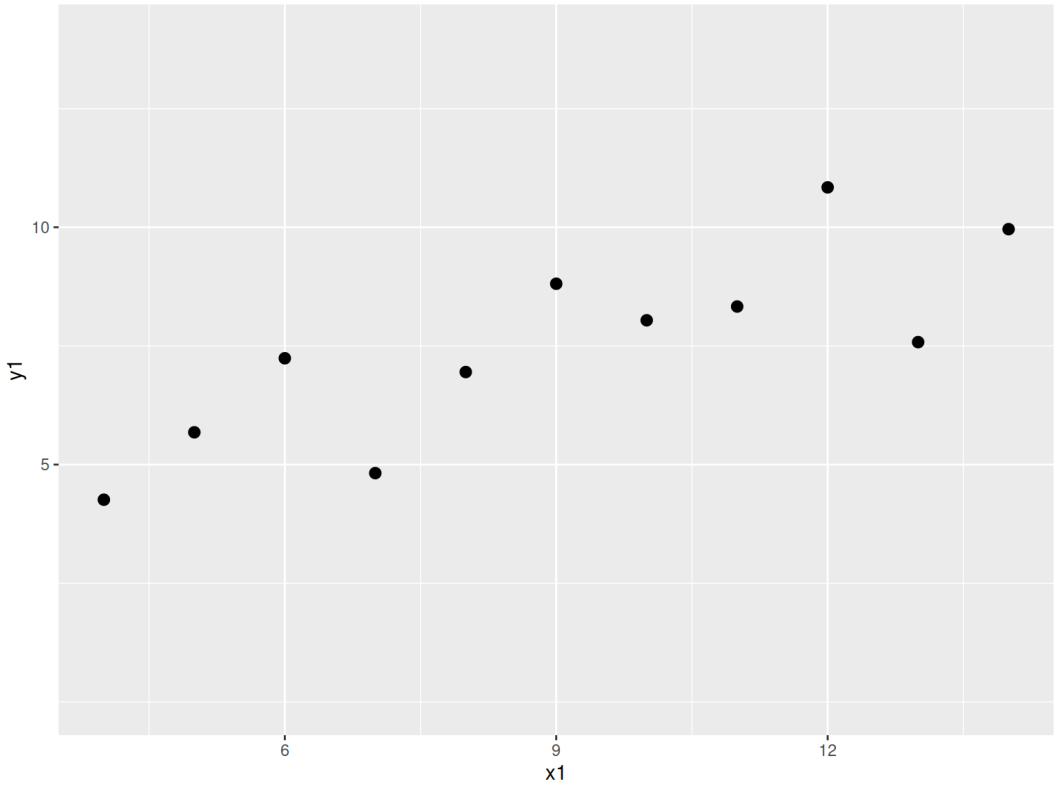
```
p0 <- ggplot(anscombe, aes(x1,y1))

p0
```



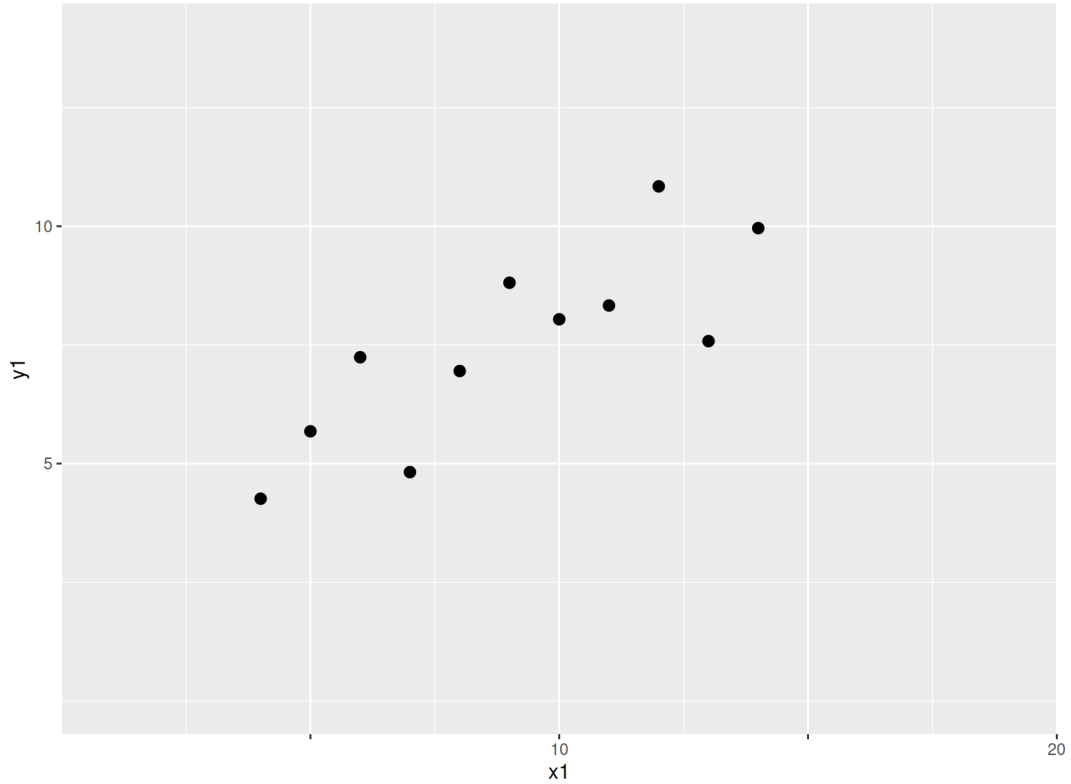
```
p01 <- p0 + geom_point(size = 2.5)

p01
```



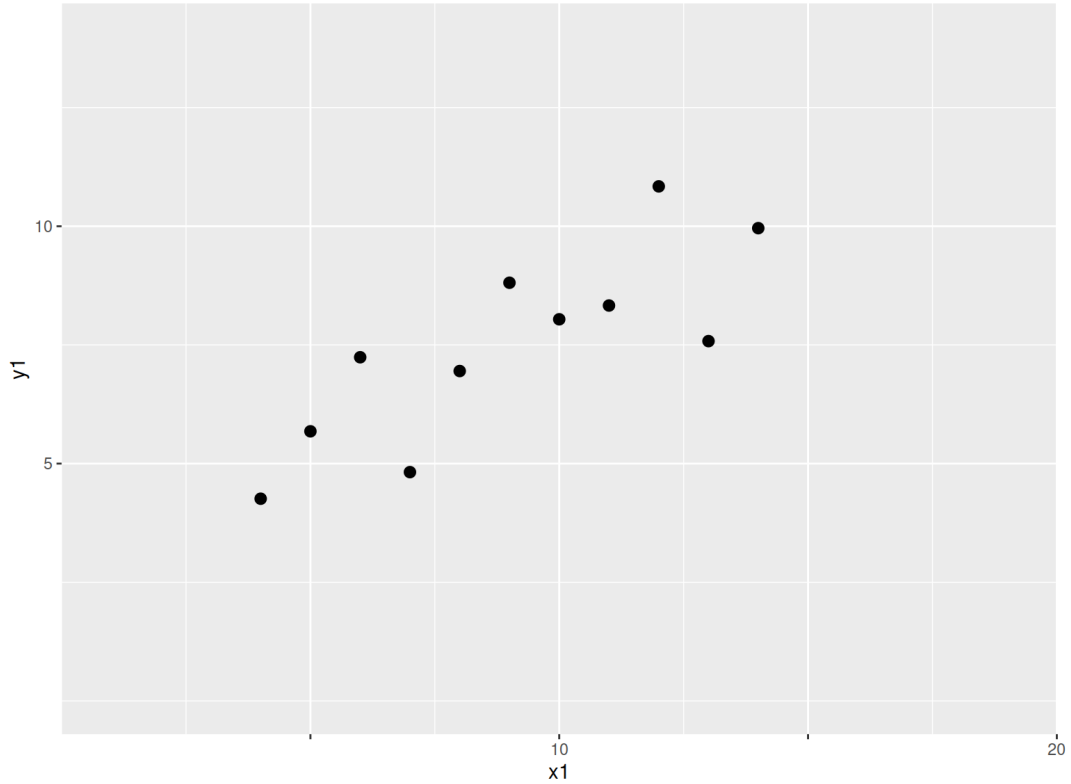
```r
p02 <- p01 + scale_y_continuous(breaks = c(5,10)) +
  expand_limits(y = c(0,14))

p02
```



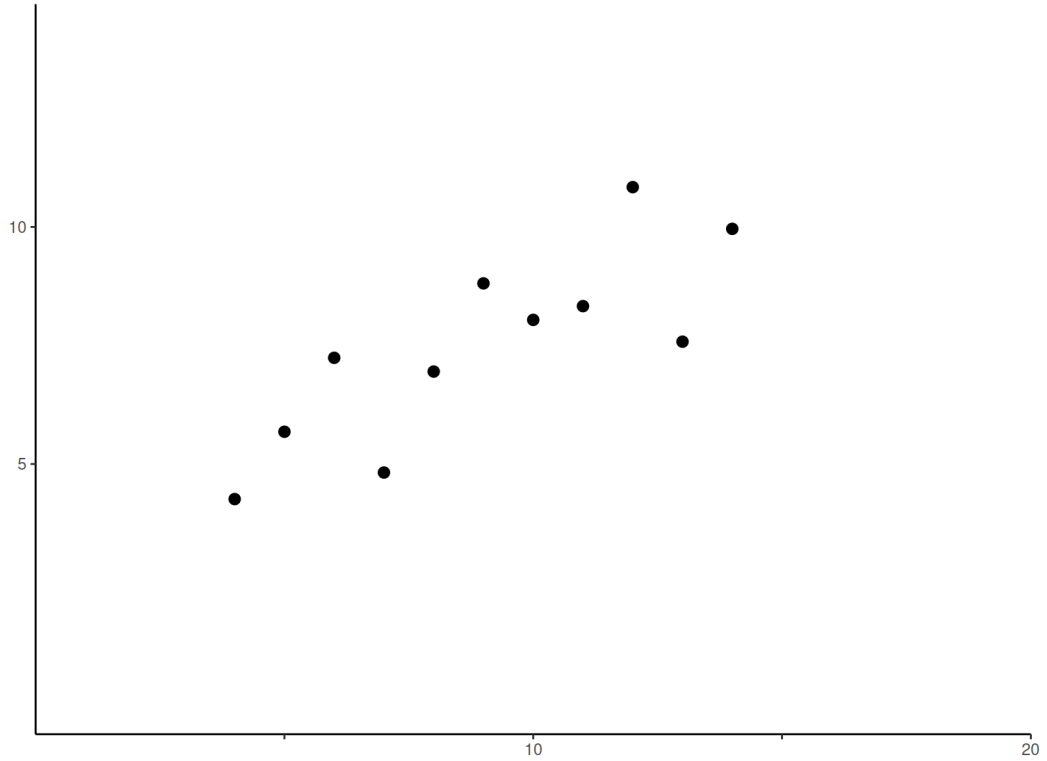
```
p03 <- p02 + scale_x_continuous(labels = c("",10,"",20),
                                breaks=c(5,10,15,20),
                                expand = c(0, 0)) +
  expand_limits(x = c(0,20))

p03
```



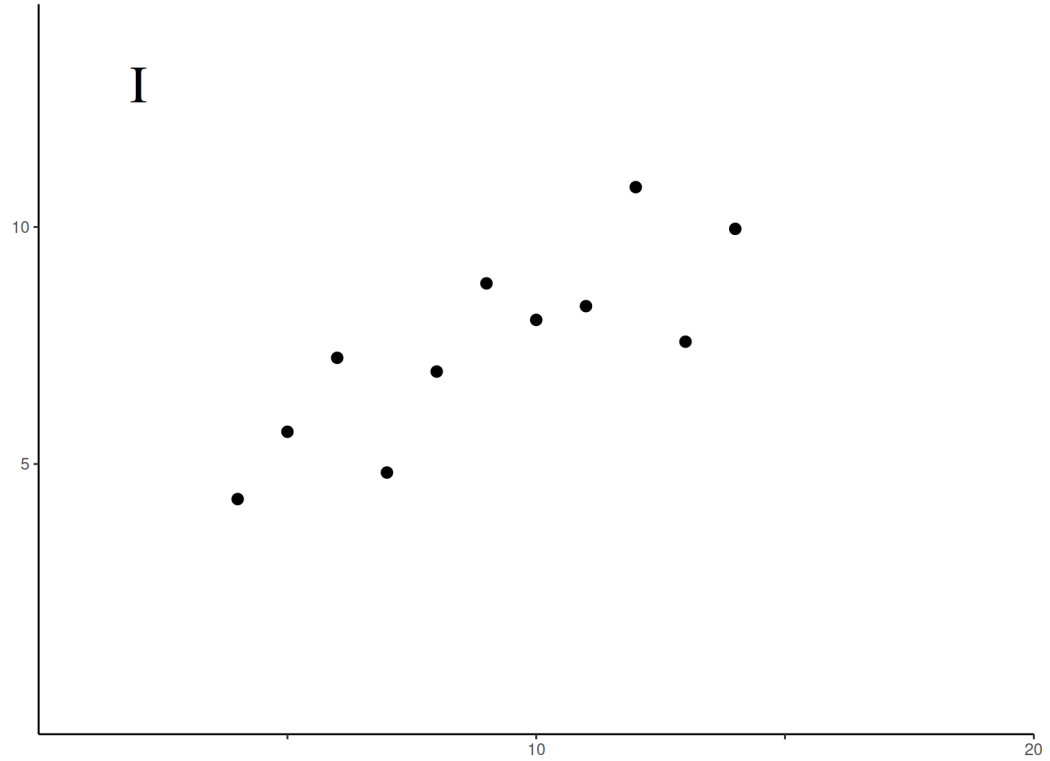
```
p03 <- p02 + scale_x_continuous(labels = c("",10,"",20),
                                breaks=c(5,10,15,20),
                                expand = c(0,  0)) +
  expand_limits(x = c(0,20))

p03
```



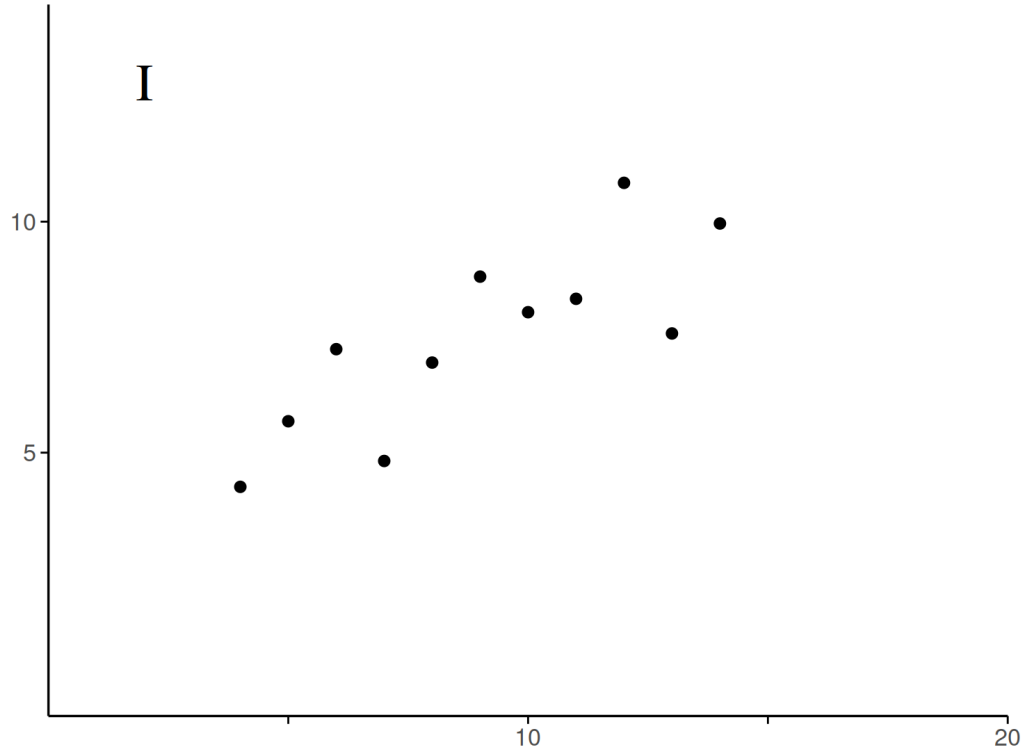
```r
p04 <- p03 + labs(x = "", y = "") + theme_classic()

p04
```



```
p05 <- p04 + annotate("text", x = 2, y = 13, size=10, family="Times", la

p05
```



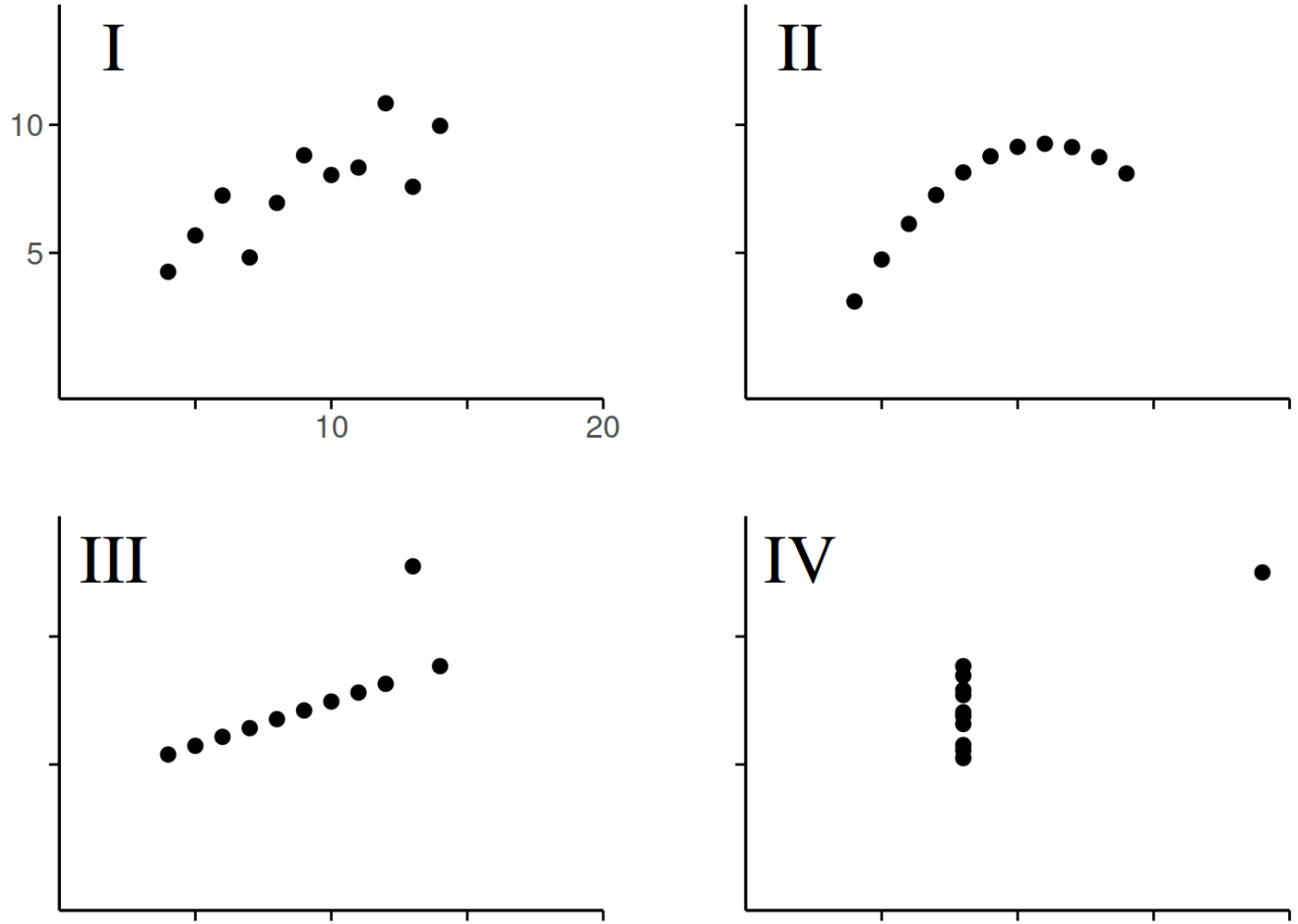
```
p06 <- p05 + labs(x = "", y = "") + theme(text=element_text(size=16),
      axis.text.x = element_text(hjust = 0.5),
      axis.line = element_line(colour = 'black', linewidth = 0.6),
      axis.ticks = element_line(colour = "black", linewidth = 0.5),
      axis.ticks.length=unit(.15, "cm"),
      plot.margin=unit(c(.2,.5,.2,.2),"cm"))

p06
```

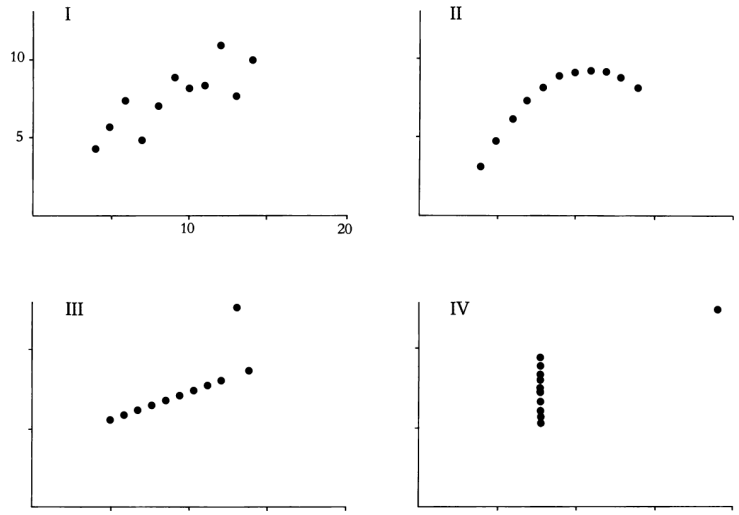


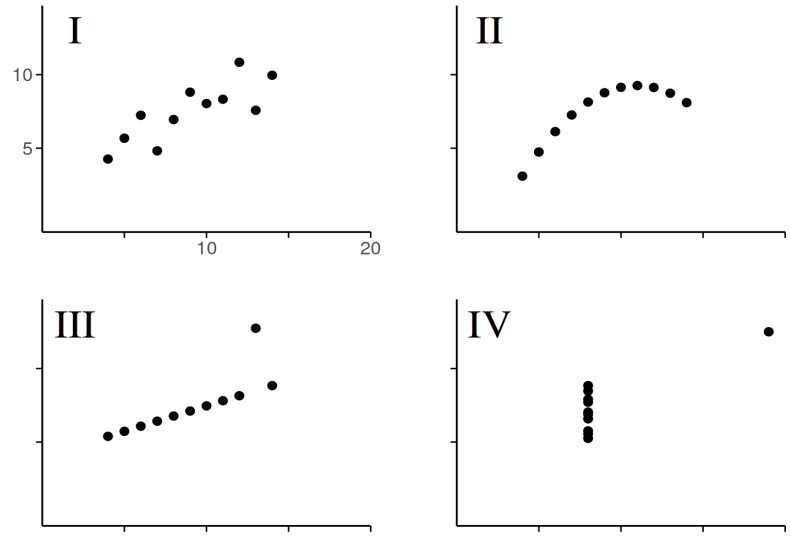
```
p1 <- ggplot(anscombe,
       aes(x1,y1)) +
geom_point(
  size = 2.5) +
# geom_smooth(method="lm", se=F, fullrange = TRUE,
#             color="darkgrey") +
annotate("text", x = 2, y = 13, size=10, family="Times", label = "I") +
scale_x_continuous(
  labels = c("",10,"",20), breaks=c(5,10,15,20), expand = c(0, 0)) +
scale_y_continuous(
  breaks = c(5,10)) +
expand_limits(x = c(0,20), y = c(0,14)) +
labs(x = "",
     y = "") +
theme_classic() +
# theme_bw() +
```

```
p2 <- ggplot(anscombe,
       aes(x2,y2)) +
geom_point(
  size = 2.5) +
annotate("text", x = 2, y = 13, size=10, family="Times", label = "II") +
scale_x_continuous(
  labels = c("","10","","20"), breaks=c(5,10,15,20), expand = c(0, 0)) +
scale_y_continuous(
  breaks = c(5,10)) +
expand_limits(x = c(0,20), y = c(0,14)) +
labs(x = "",
     y = "") +
theme_classic() +
theme(text=element_text(size=16),
      axis.line = element_line(colour = 'black', linewidth = 0.6),
      axis.ticks = element_line(colour = "black", linewidth = 0.5),
```

```
p1 + p2 + p3 + p4 + plot_layout(ncol = 2)
```




I
II
III
IV
10
5
10
20


I
II
III
IV
10
5
10
20

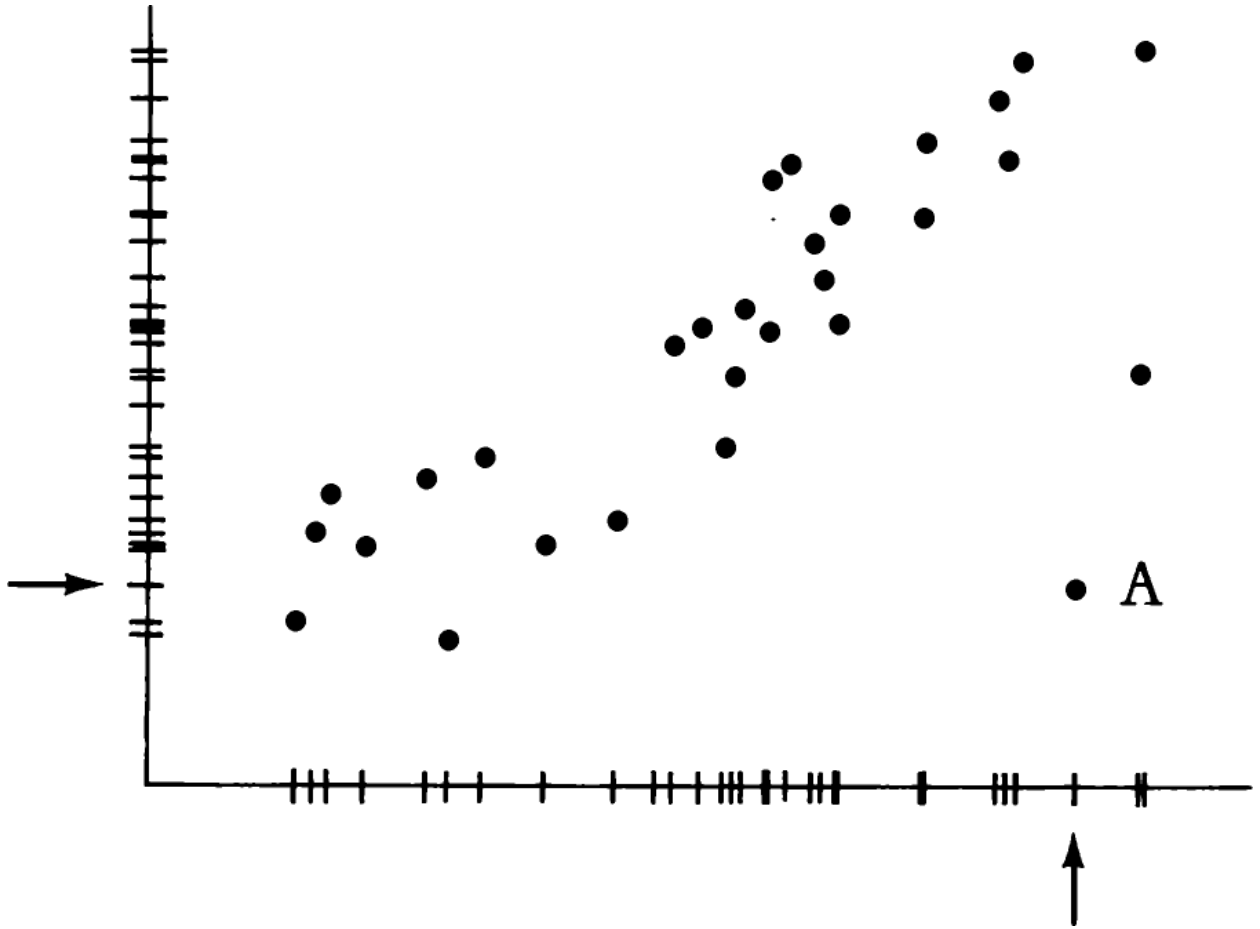
patchwork

The goal of `patchwork` is to make it ridiculously simple to combine separate ggplots into the same graphic. As such it tries to solve the same problem as `gridExtra::grid.arrange()` and `cowplot::plot_grid` but using an API that incites exploration and iteration, and scales to arbitrarily complex layouts.

https://patchwork.data-imaginist.com/

Bonus

Add rugplots to the axes…



…to detect outliers and label outliers

Small multiples

Could you provide an alternative solution not using patchwork or a similar package?

Think about rearranging the data (and of using `facets` in ggplot2 for example)

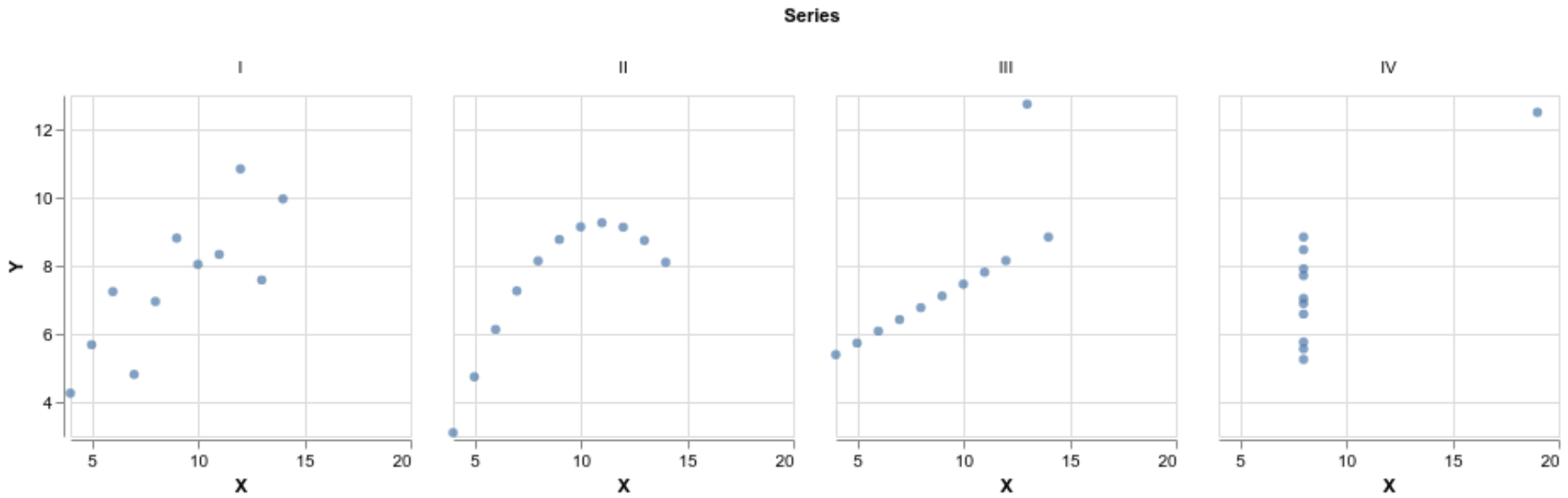
For another, Python solution: Vega-Altaire

https://iliatimofeev.github.io/altair-viz.github.io/gallery/anscombe_plot.html

```
import altair as alt
from vega_datasets import data

anscombe = data.anscombe()

anscombe
```

```
##     Series   X      Y
## 0        I  10   8.04
## 1        I   8   6.95
## 2        I  13   7.58
## 3        I   9   8.81
## 4        I  11   8.33
## 5        I  14   9.96
## 6        I   6   7.24
## 7        I   4   4.26
## 8        I  12  10.84
## 9        I   7   4.81
## 10       I   5   5.68
## 11      II  10   9.14
## 12      II   8   8.14
## 13      II  13   8.74
## 14      II   9   8.77
## 15      II  11   9.26
```

```
alt.Chart(anscombe).mark_circle().encode(
    alt.X('X', scale=alt.Scale(zero=False)),
    alt.Y('Y', scale=alt.Scale(zero=False)),
    column='Series'
).properties(
    width=200,
    height=200
)
```



Until next week…

Berkeley's 1973 Graduate Admissions Dataset

The "Berkeley Dataset" contains all 12,763 applicants to UC-Berkeley's graduate programs in Fall 1973. This dataset was published by UC-Berkeley researchers in an analysis to understand the possible gender bias in admissions and has now become a classic example of Simpson's Paradox.

- **Dataset Format**: Well-formatted CSV with column headers as the first row
- **Dataset Size**: 12,763 rows × 4 columns
- **CSV File Location**: https://waf.cs.illinois.edu/discovery/berkeley.csv
- **Dataset Variables**:
  - `Year` : number ➜ The application year (this data is always `1973`)
  - `Major` : string ➜: An anonymized major code (either `A`, `B`, `C`, `D`, `E`, `F`, or `Other`). The specific majors are unknown except that `A`-`F` are the six majors with the most applicants in Fall 1973
  - `Gender` : string ➜ Applicant self-reported gender (either `M` or `F`)
  - `Admission`: string ➜ Admission decision (either `Rejected` or `Accepted`)
- **Research Paper**: *Sex Bias in Graduate Admissions: Data from Berkeley* by P. J. Bickel, E. A. Hammel, and J. W. O'Connell (1975)

Get the data…

https://waf.cs.illinois.edu/discovery/berkeley.csv

https://discovery.cs.illinois.edu/dataset/berkeley/

Until next time

Visualize the Berkeley data as an informative graphic (and a table and possible a combination of both), investigating admission rates by gender (following the design principles discussed in the course so far)

What do you find? Think about possible reasons for your findings

Until next time

Now, create small multiples split up by the study program applicants applied to (Variable “Major”) and take a look at admission rates by gender again for each of the majors

What do you find now? What could be the reasons for your earlier findings?

What other informative aspects of the data are there to be uncovered? Visualize them with techniques that you deem adequate