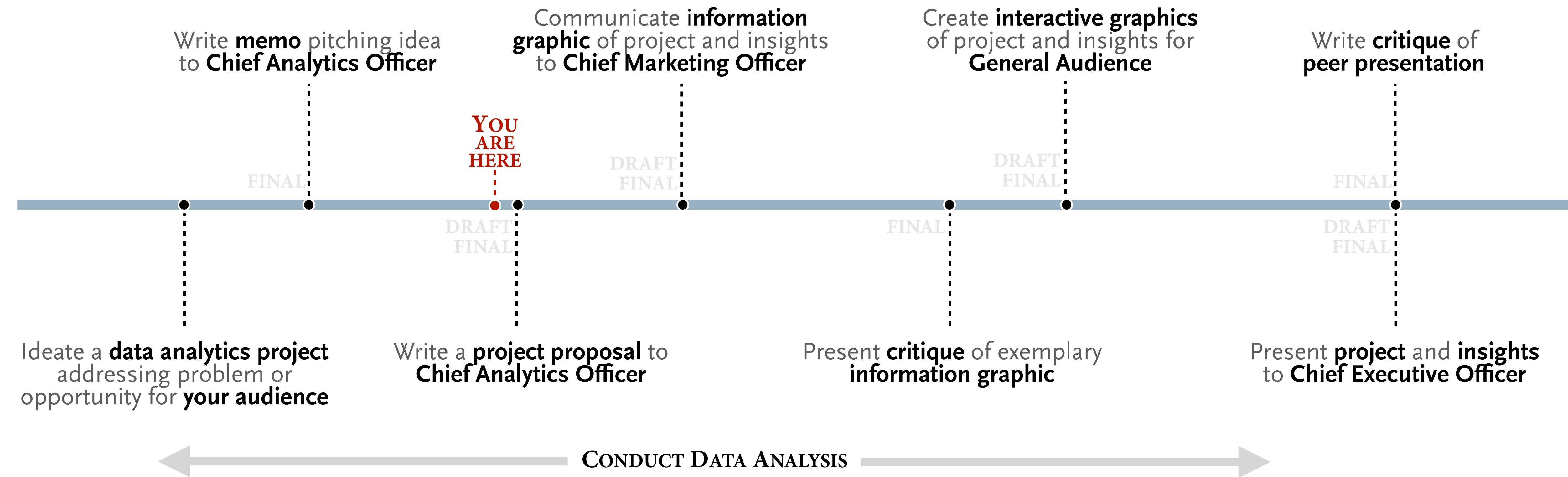


# Storytelling with data

**06 | the grammar of graphics; Doumont's three laws applied to data encoding; color; typologies**

# course overview | main course deliverables



**the grammar of graphics**

# *the grammar of graphics, statistical graphic specifications are expressed in six statements*

**DATA** : a set of data operations that create variables from datasets

**TRANSFORMATIONS** : variable transformations (*e.g., rank, log, square root*)

**SCALES** : scale transformations (*e.g., linear, log, square root*)

**COORDINATES** : a coordinate system (*e.g., cartesian, polar*)

**ELEMENTS** : graphs (*e.g., points, lines, areas*) and their aesthetic attributes (*e.g., position, size, hue, saturation, luminance, opacity, orientation, shape*)

**GUIDES** : one or more guides (*axes, legends, etc.*)

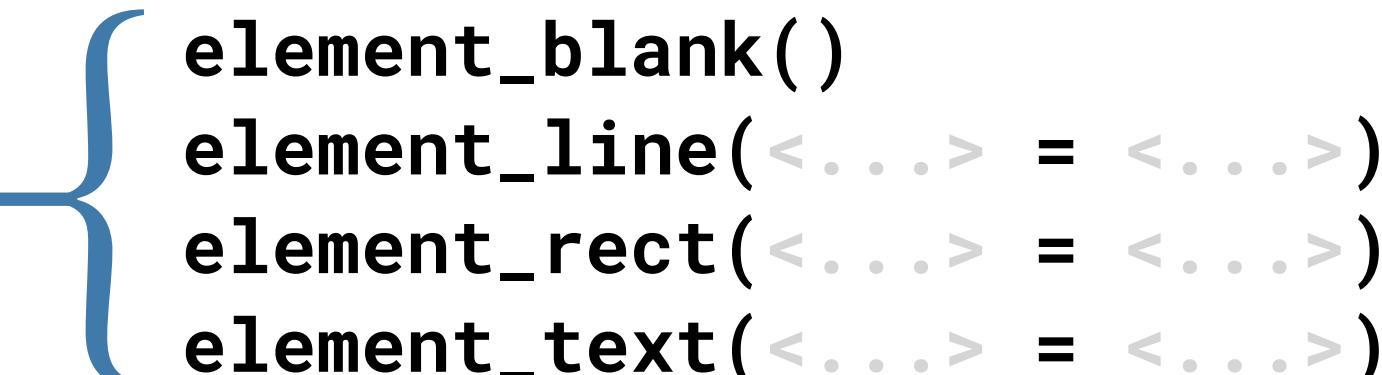
# the grammar of graphics, implementation example — ggplot2 (grammar of graphics)

```
# load grammar of graphics
library(ggplot2)

p <-

# functions for data ink

DATA      ggplot(data = <data>,
                mapping = aes(<aesthetic> = <variable>,
                            <aesthetic> = <variable>,
                            <...> = <...>) +
TRANSFORMATIONS
ELEMENTS   geom_<type>(<...>) +
SCALES & GUIDES    scale_<mapping>_<type>(<...>) +
COORDINATES   coord_<type>(<...>) +
               facet_<type>(<...>) +
               <...> +
GUIDES      # functions for non-data ink
               labs(<...>) +
               theme(<...> = <...>) +
               annotate(<...>) +
               <...>
```



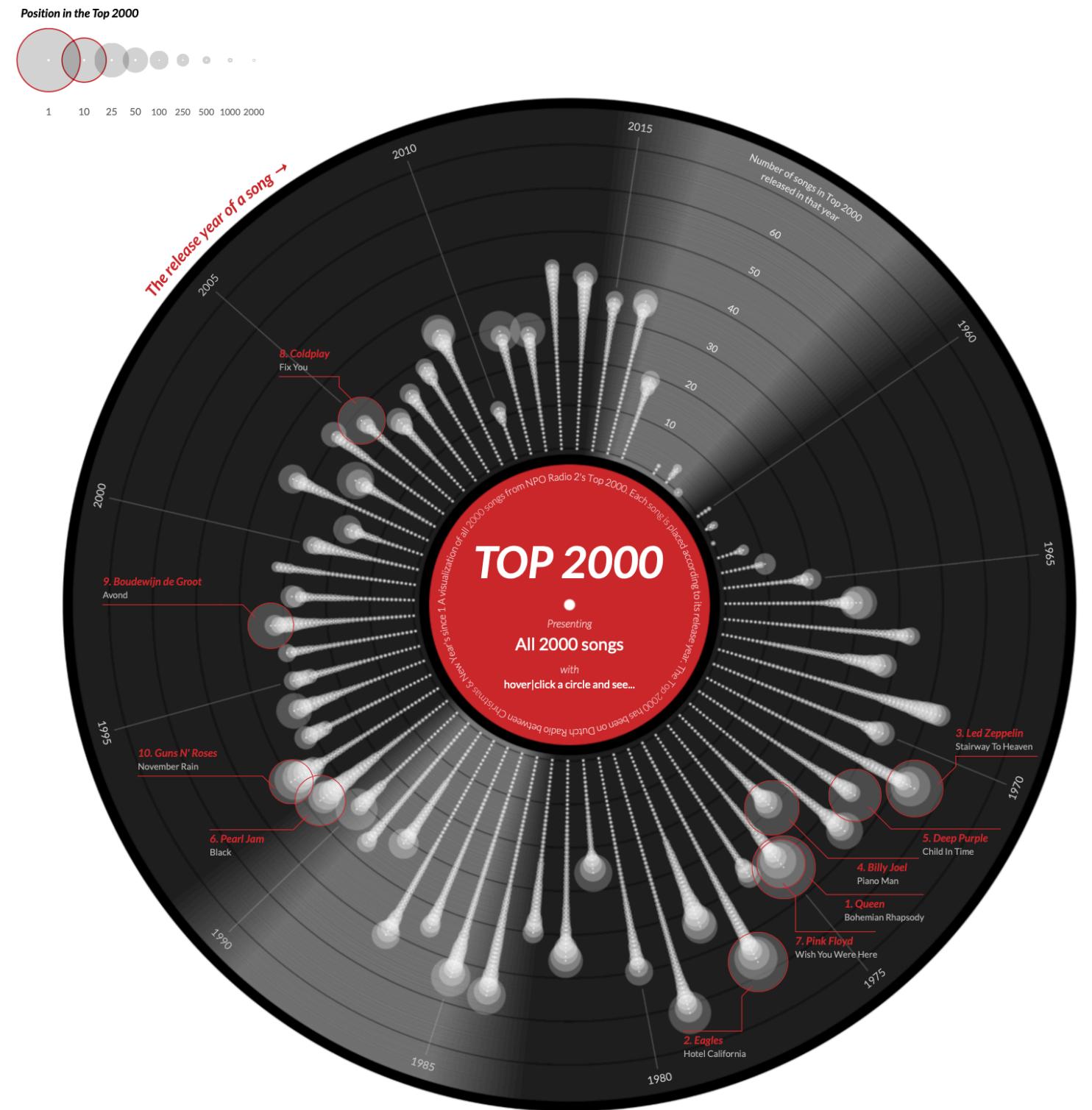
The diagram shows a brace grouping four functions: element\_blank(), element\_line(), element\_rect(), and element\_text(). A blue arrow points from the 'GUIDES' section of the code to this group of functions.

- element\_blank()
- element\_line(<...> = <...>)
- element\_rect(<...> = <...>)
- element\_text(<...> = <...>)

# *the grammar of graphics, elements in layers — example*

## Let The Music Play

All songs from the **TOP 2000** of 2017 according to their release years



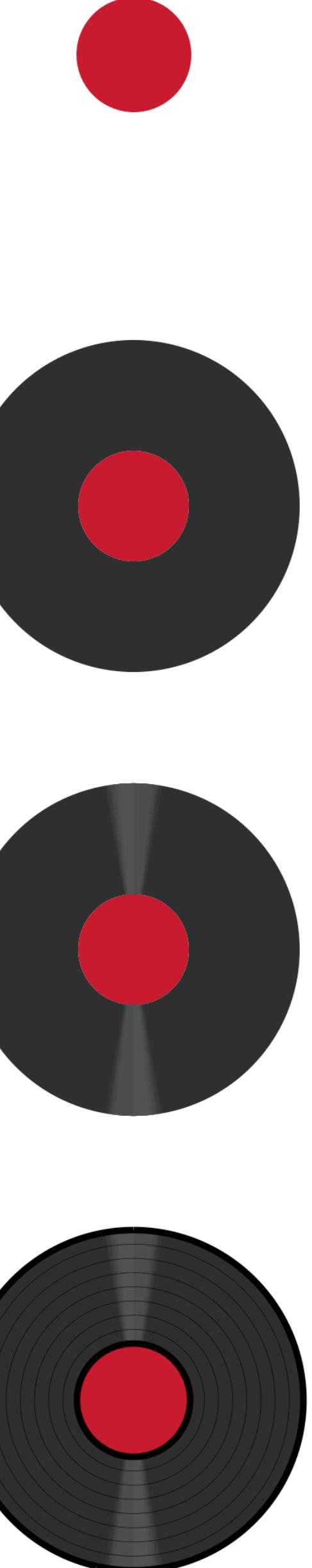
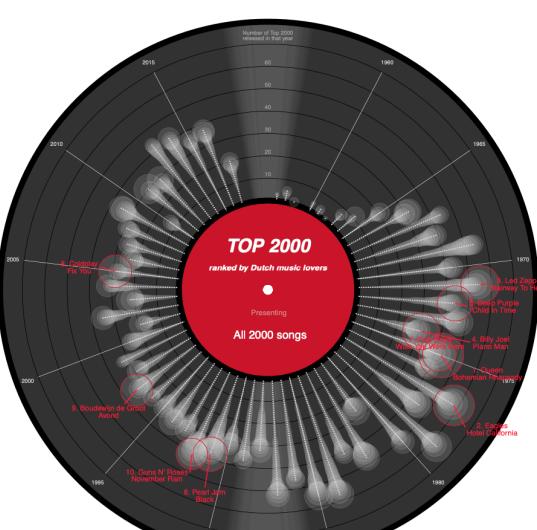
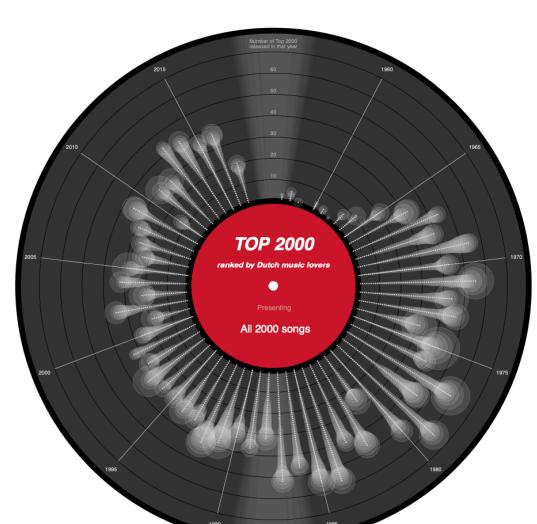
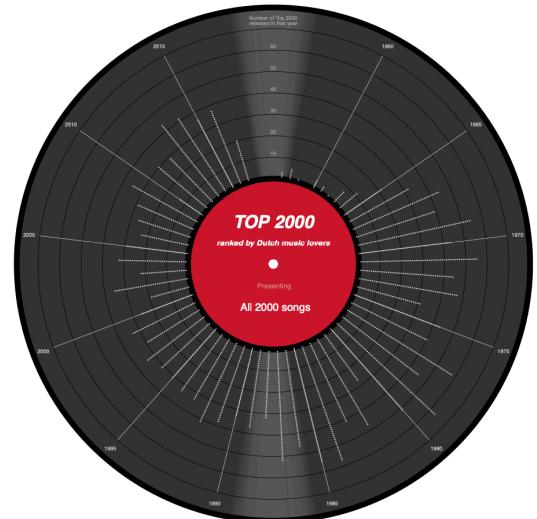
When hovering (or clicking on mobile) a circle all the songs of that artist are **highlighted**  
and connected by a **line**. When the song happens to be a collaboration between two  
artists, you'll see multiple lines, one for each artist.

Created by Nadieh Bremer | Visual  
Cinnamon

Data from [NPO Radio 2](#)

Ga naar de [Nederlandse](#) versie

Check out my Top 2000 visuals from [2015](#) &  
[2016](#)



**Doumont's *three laws of communication* applied to data encoding**

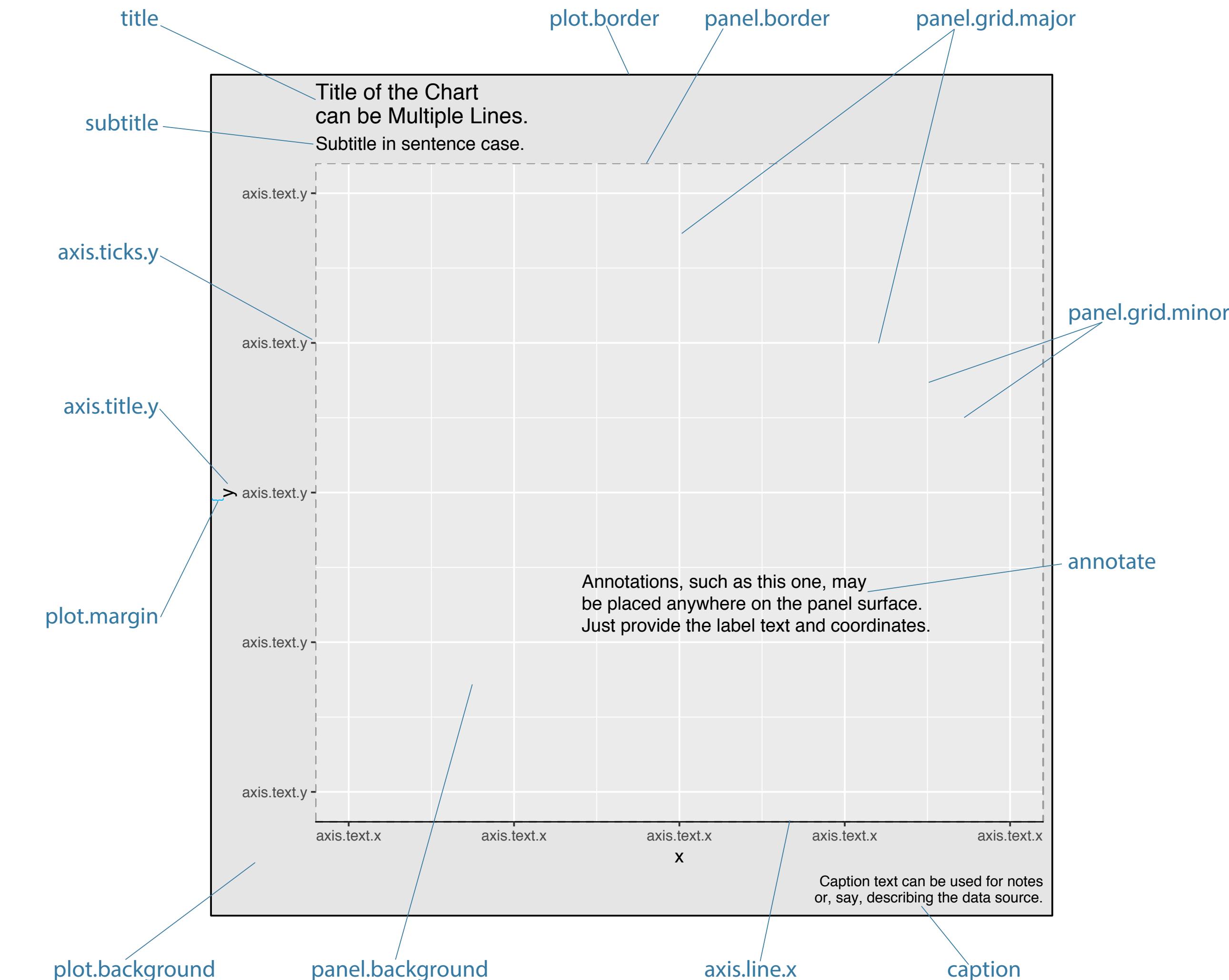
# Doumont applied to data encoding, **Doumont's three laws of communication**

**Adapt to your audience**

**Maximize the signal-to-noise ratio**

**Use effective redundancy**

# Doumont applied to data encoding, non-“data ink”



# Doumont applied to data encoding, non-“data ink” — example functions to draw non-data ink in ggplot2

```
# load grammar of graphics
library(ggplot2)

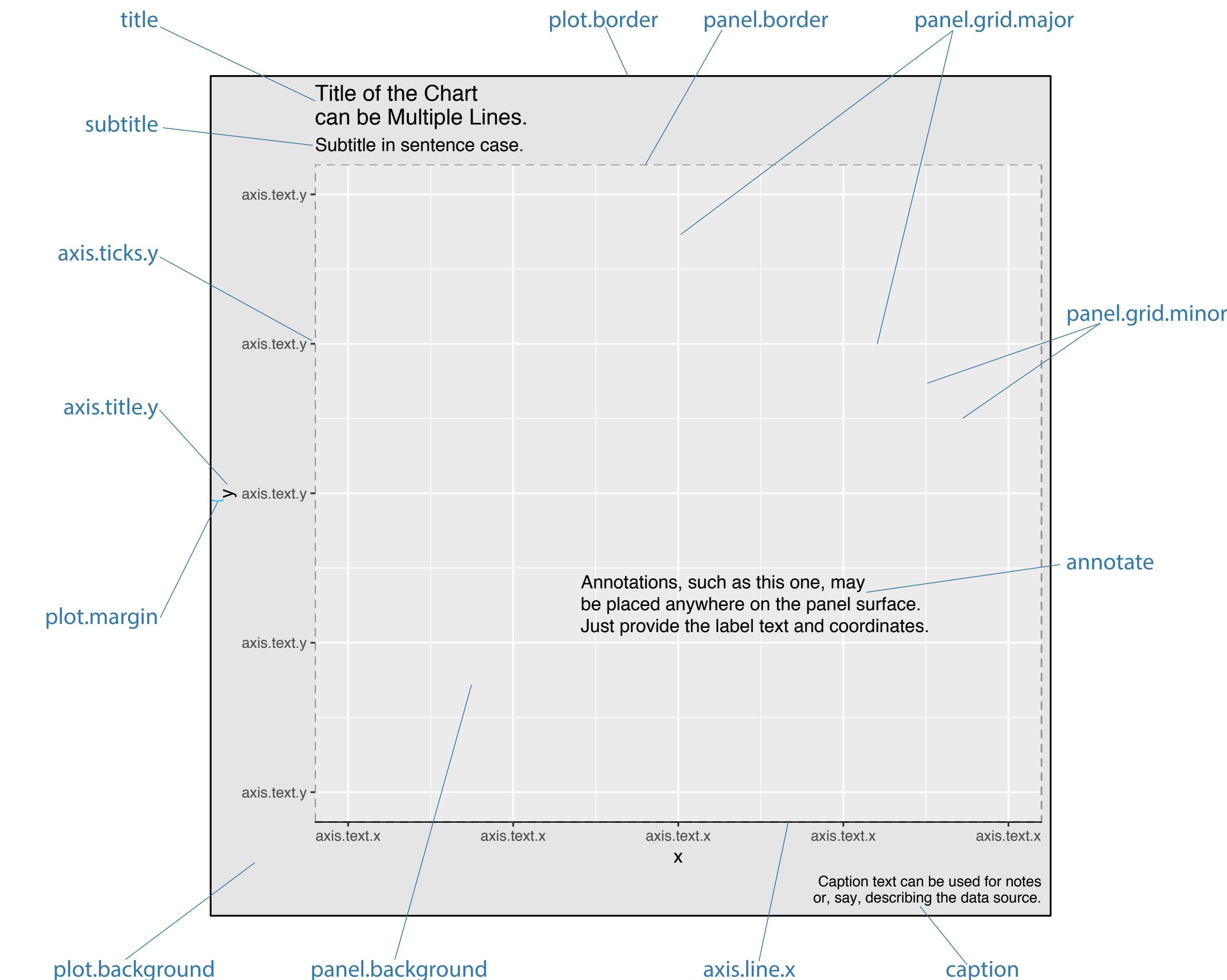
p <-

# functions for data ink

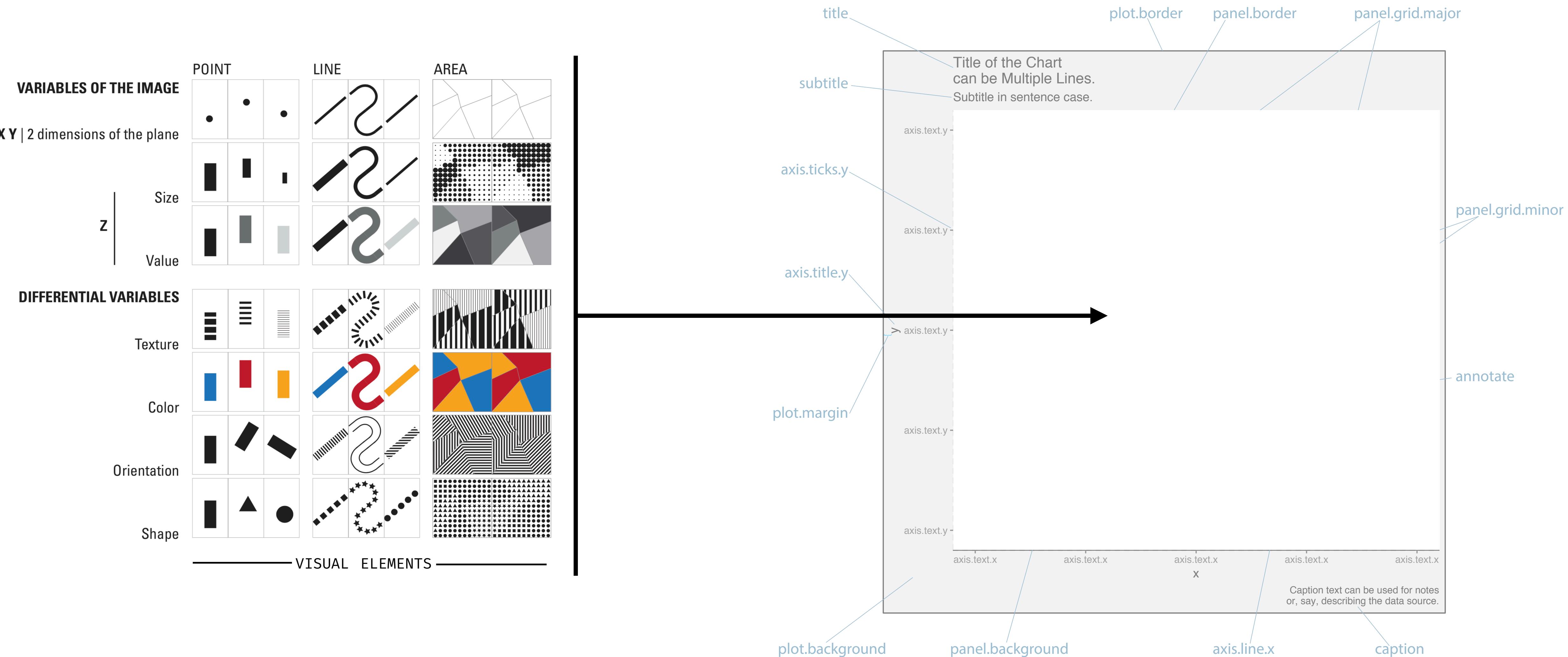
ggplot(data = <data>,
       mapping = aes(<aesthetic> = <variable>,
                     <aesthetic> = <variable>,
                     <...> = <...>) +
  geom_<type>(<...>) +
  scale_<mapping>_<type>(<...>) +
  coord_<type>(<...>) +
  facet_<type>(<...>) +
  <...> +

# functions for non-data ink
```

element\_blank()  
element\_line(<...> = <...>)  
element\_rect(<...> = <...>)  
element\_text(<...> = <...>)



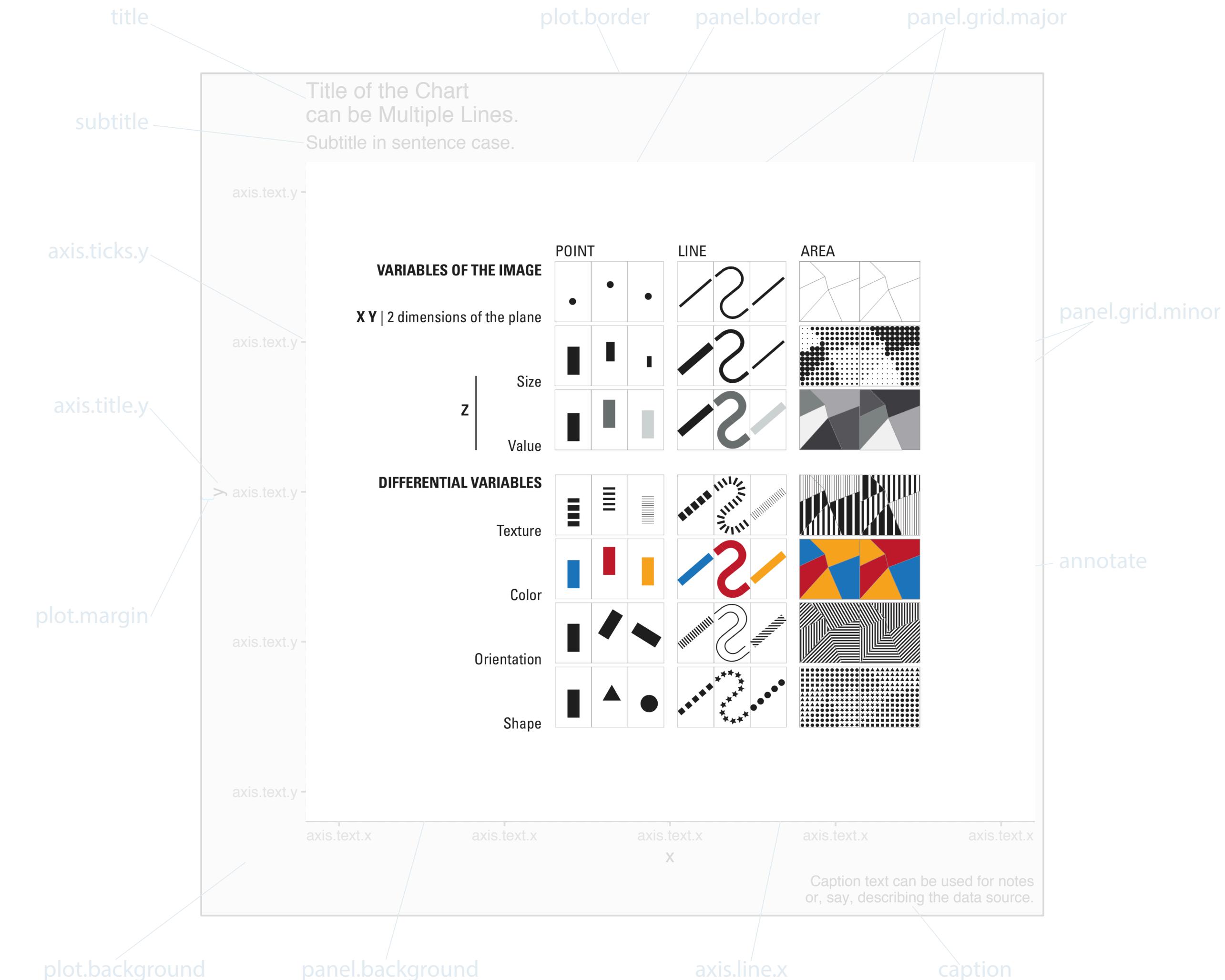
# Douumont applied to data encoding, “data ink” – Jacques Bertin’s visual channels for encoding data



# Doumont applied to data encoding, the data ink – example functions to draw encoded data in ggplot2

```
# load grammar of graphics
library(ggplot2)

p <-
  # functions for data ink
  ggplot(data = <data>,
         mapping = aes(<aesthetic> = <variable>,
                       <aesthetic> = <variable>,
                       <...> = <...>) +
    geom_<type>(<...>) +
    scale_<mapping>_<type>(<...>) +
    coord_<type>(<...>) +
    facet_<type>(<...>) +
    <...> +
  # functions for non-data ink
  labs(<...>) +
  theme(<...> = <...>) +
  annotate(<...>) +
  <...>
```



# Doumont applied to data encoding, Tufte — data-ink maximization, *within reason*

$$\text{data-ink ratio} = \frac{\text{data-ink}}{\text{total ink used to print the graphic}}$$

= proportion of a graphic's ink devoted to the non-redundant display of data-information

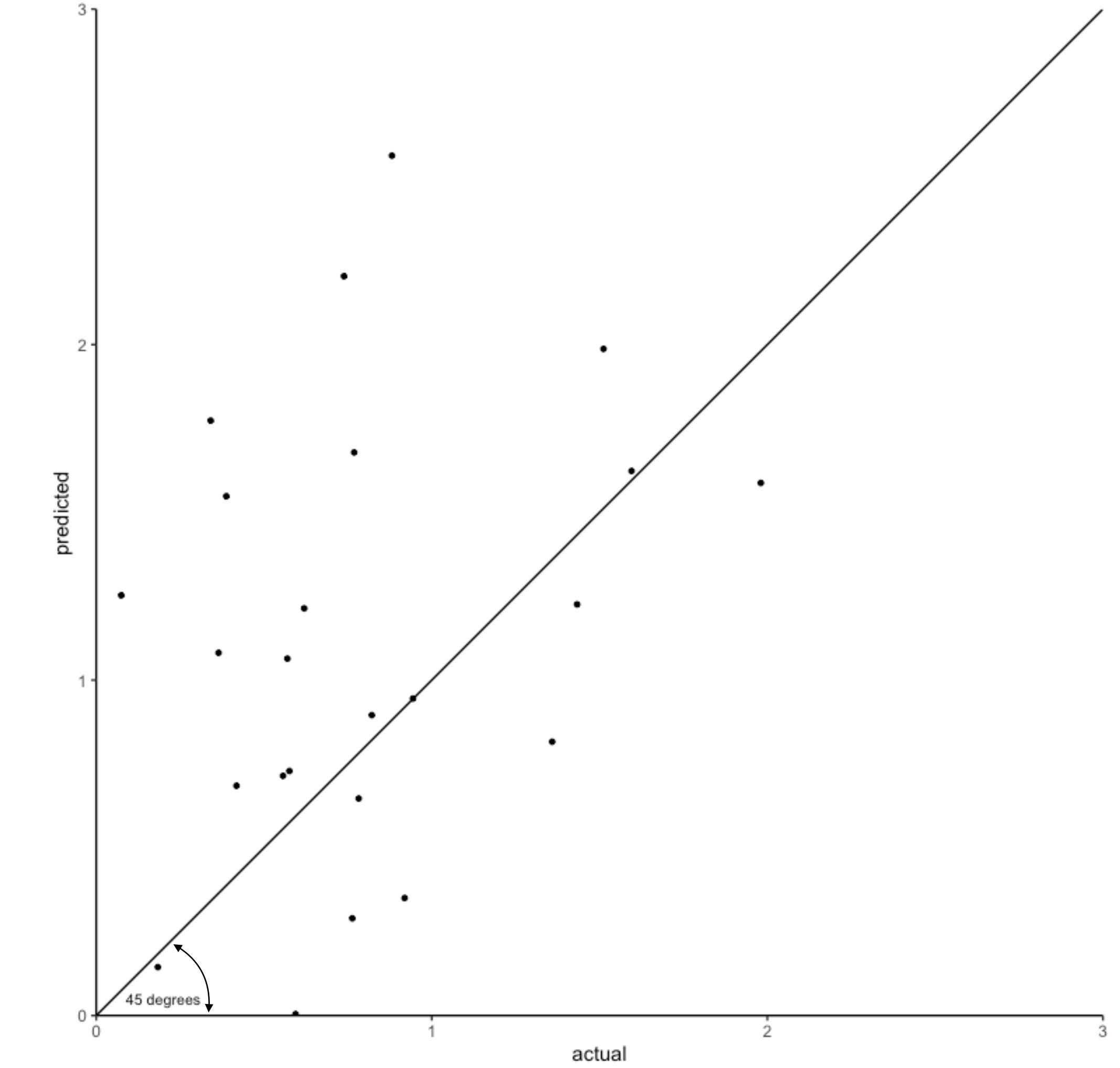
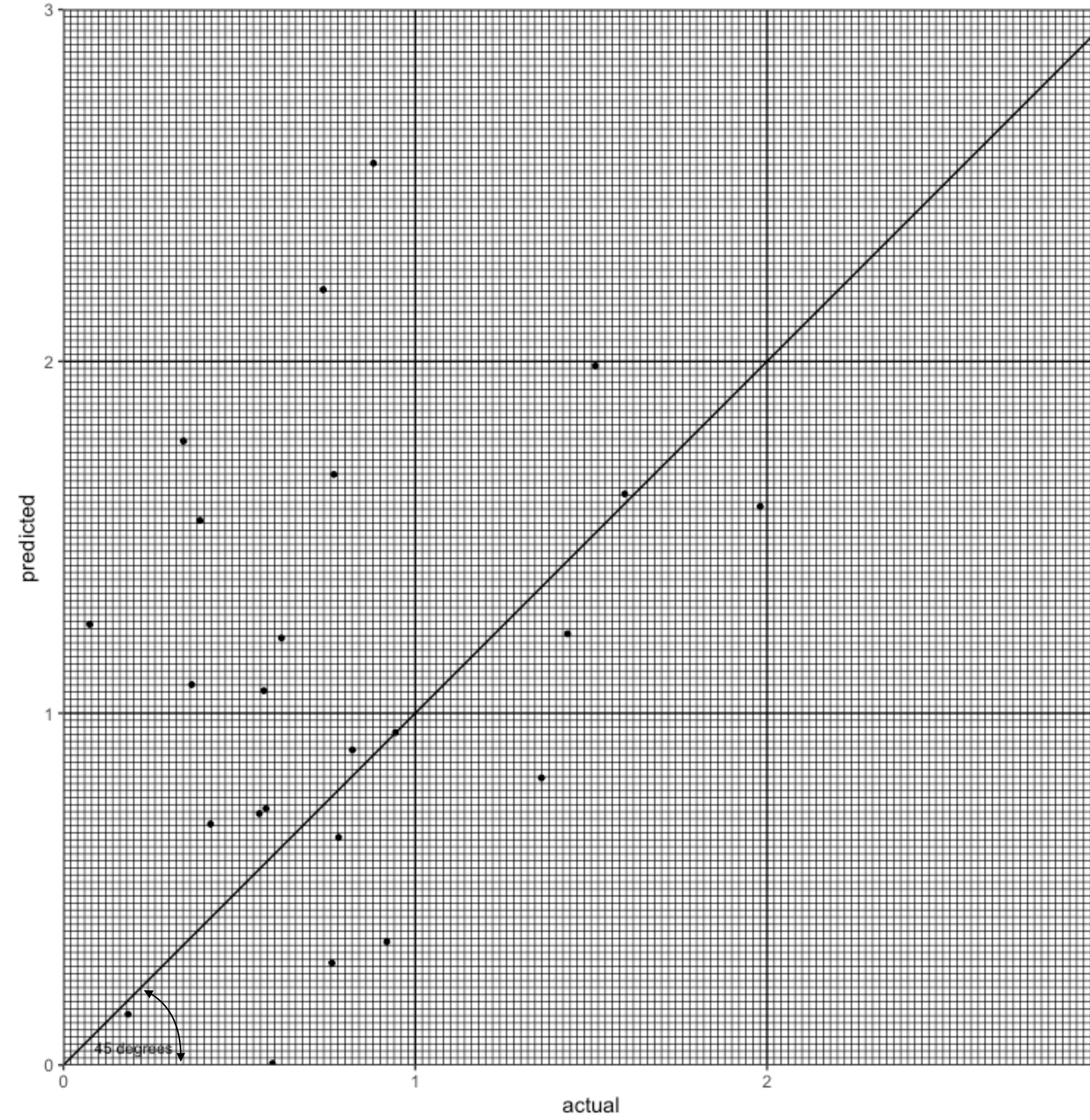
=  $1.0 -$  proportion of a graphic that can be erased without loss of data-information

“The principle ***helps conduct experiments*** in graphical design.”

— Tufte, Edward, *The Visual Display of Quantitative Information*

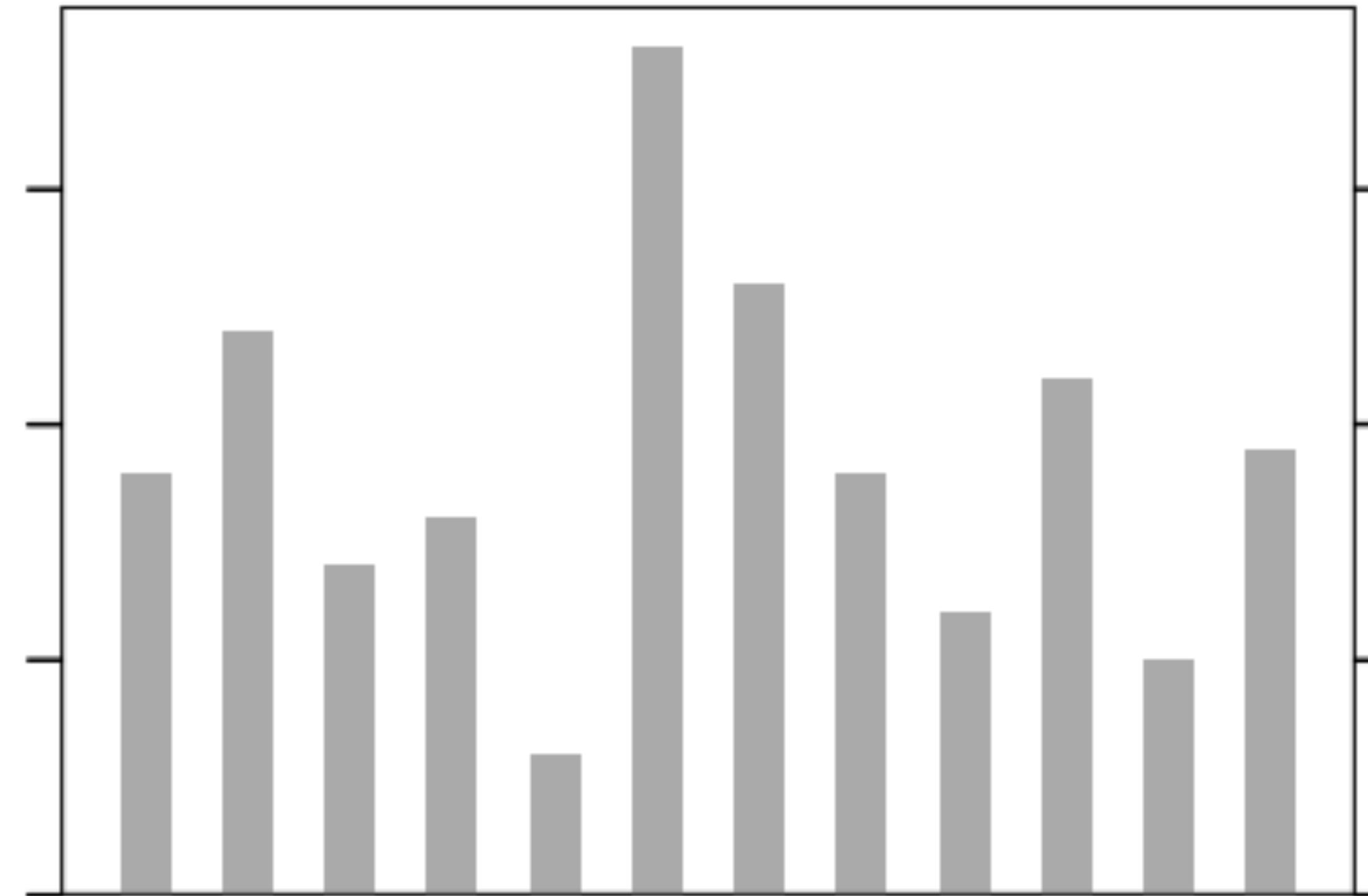
**Doumont — “maximize the signal-to-noise ratio”**

# Doumont applied to data encoding, example — removing gridlines increases the data-ink ratio

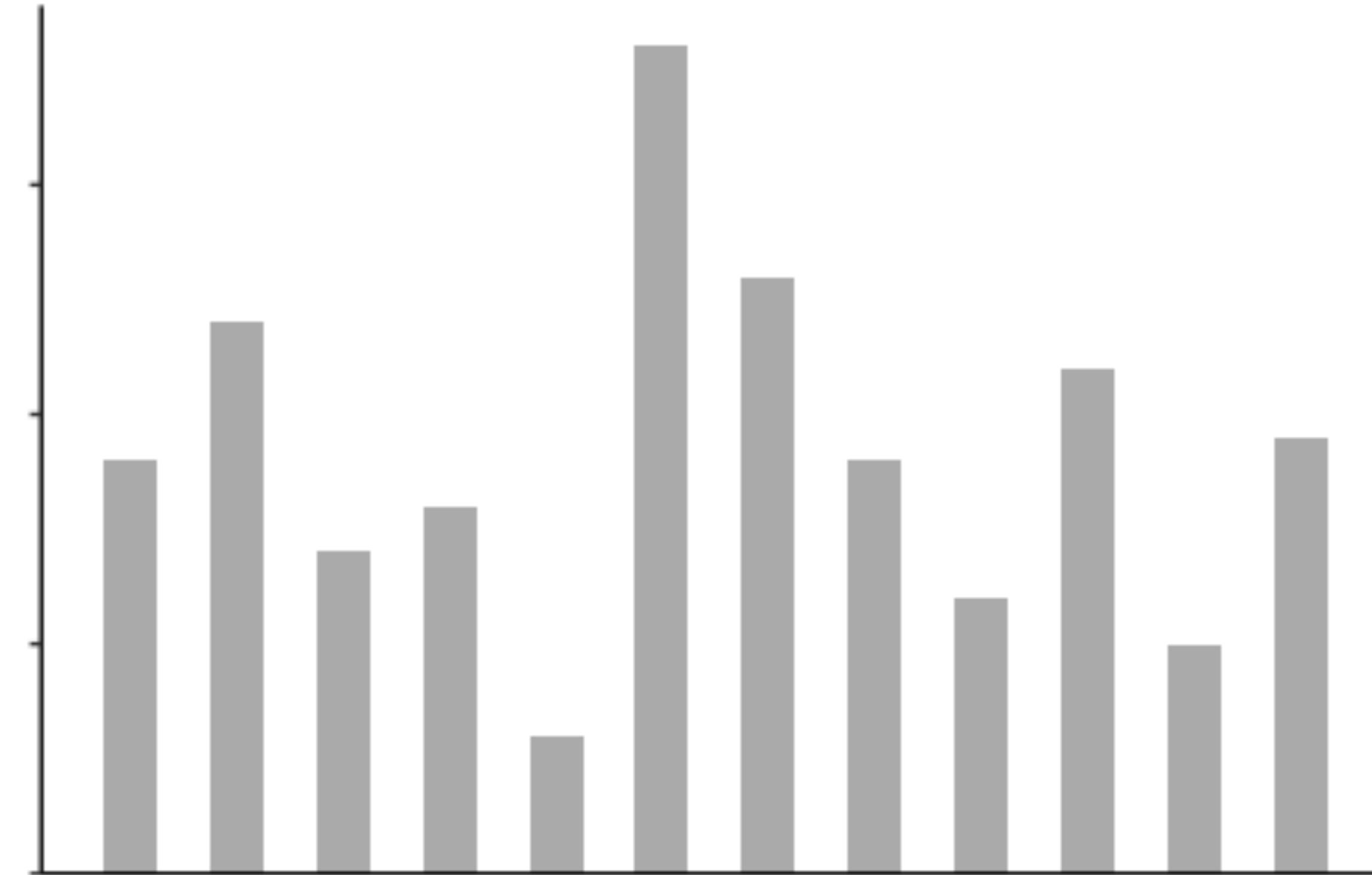


**Doumont — “maximize the signal-to-noise ratio”**

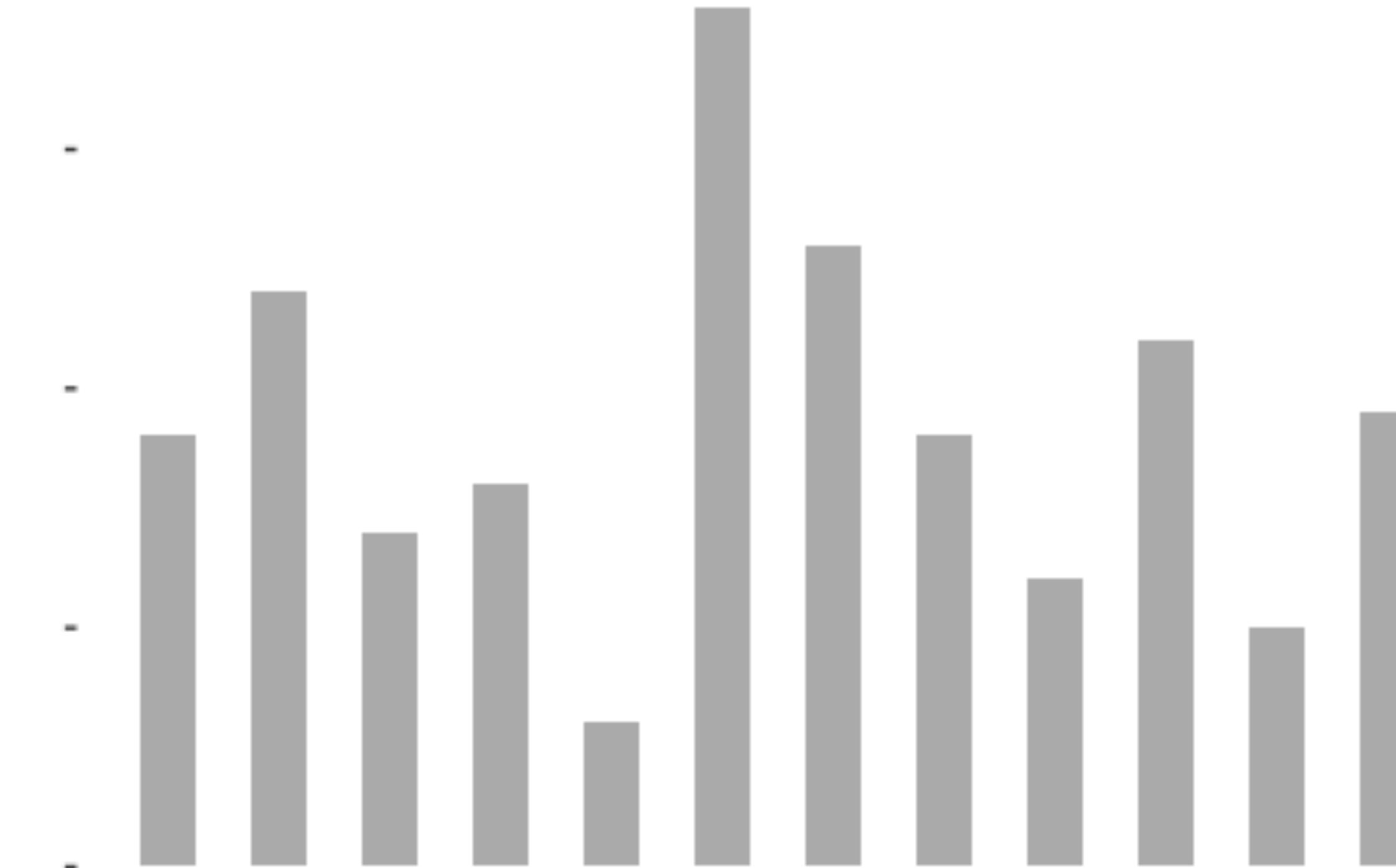
# Doumont applied to data encoding, experimentation — redesigning a bar chart



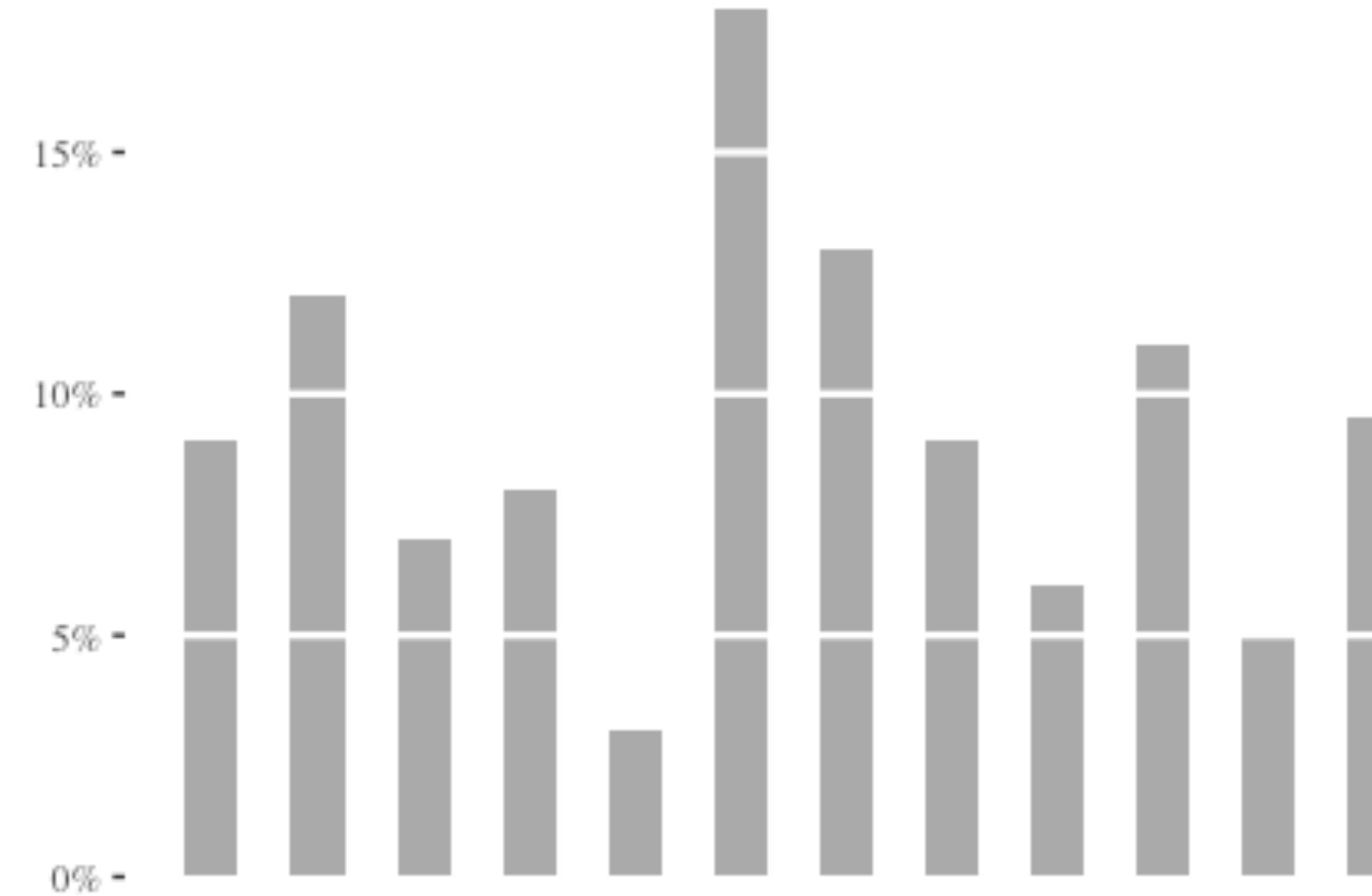
# Doumont applied to data encoding, experimentation — redesigning a bar chart



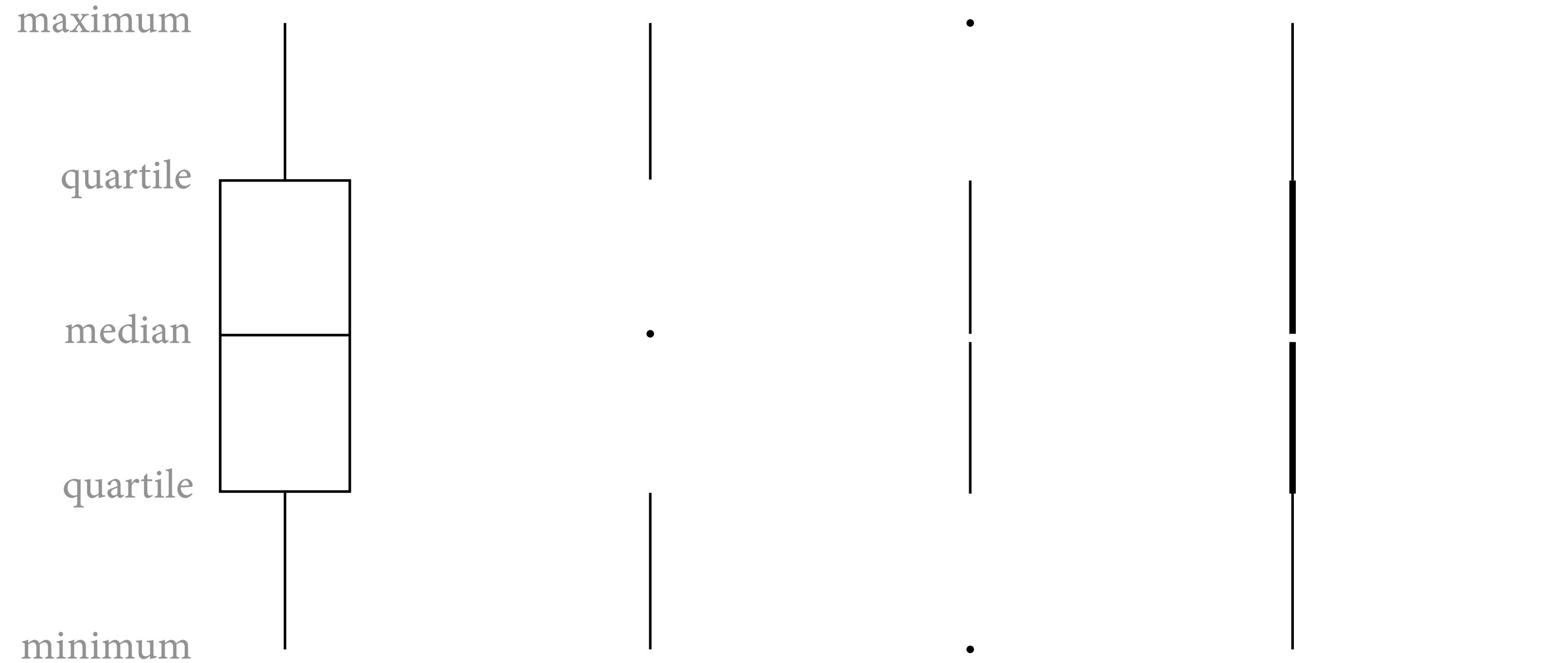
# Doumont applied to data encoding, experimentation — redesigning a bar chart



# Doumont applied to data encoding, experimentation — redesigning a bar chart



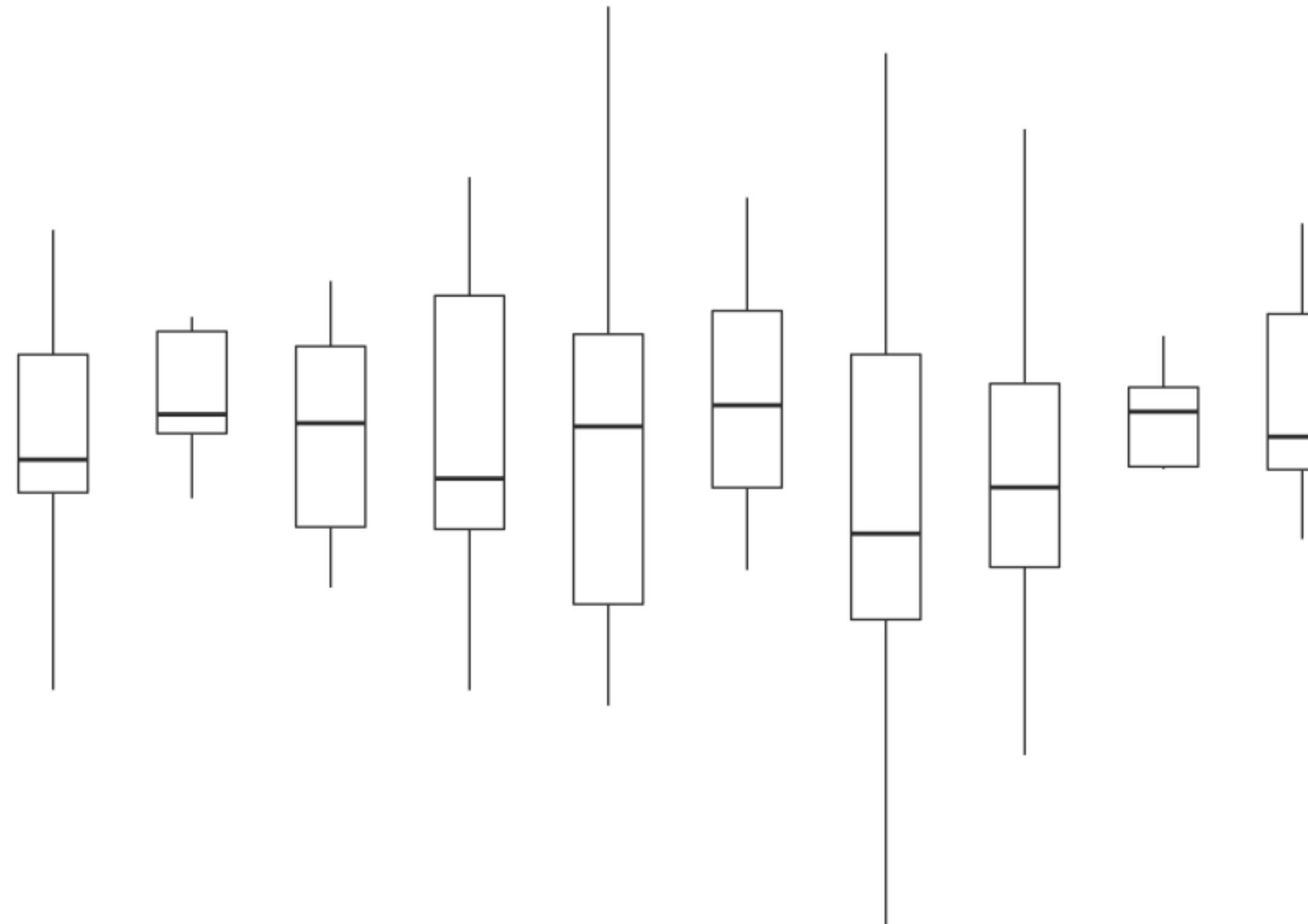
# Doumont applied to data encoding, experimentation — redesigning John Tukey's box plot



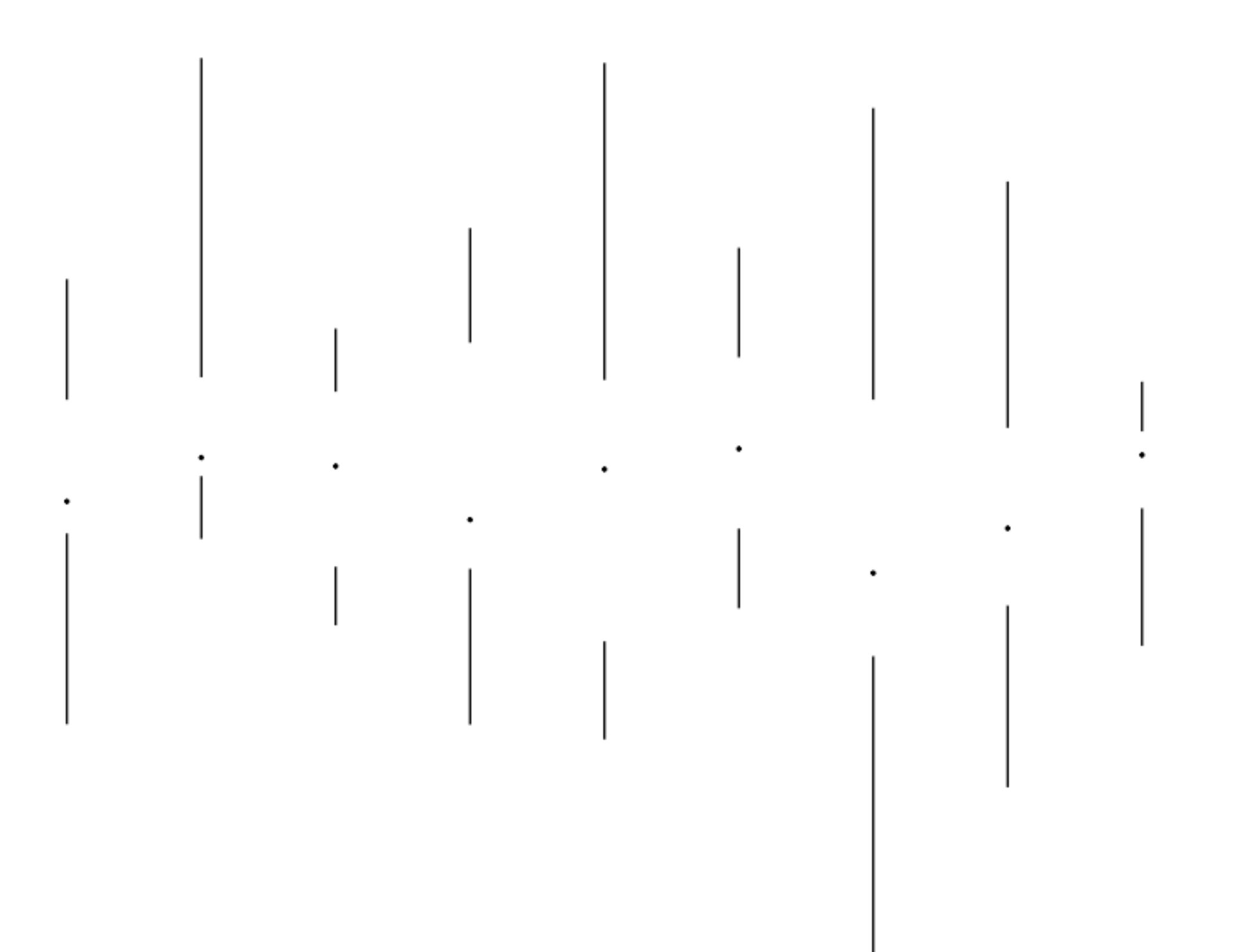
“In these revisions of the box plot, . . . the best overall arrangement naturally also rests on statistical and aesthetic criteria — in other words, the procedure is one of *reasonable* data-ink maximizing.”

— Tufte, Edward, *The Visual Display of Quantitative Information*

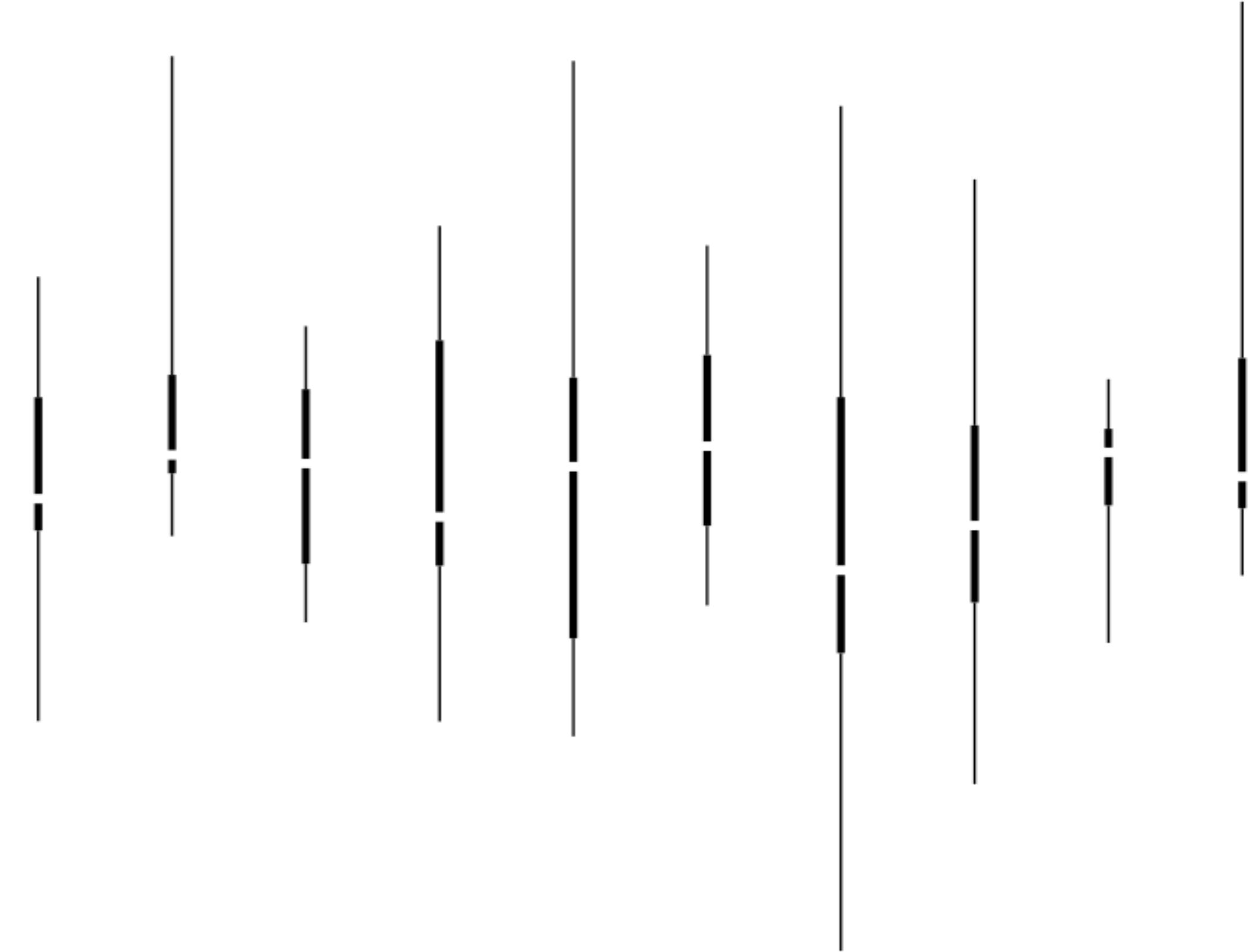
# Doumont applied to data encoding, experimentation — redesigning John Tukey's box plot



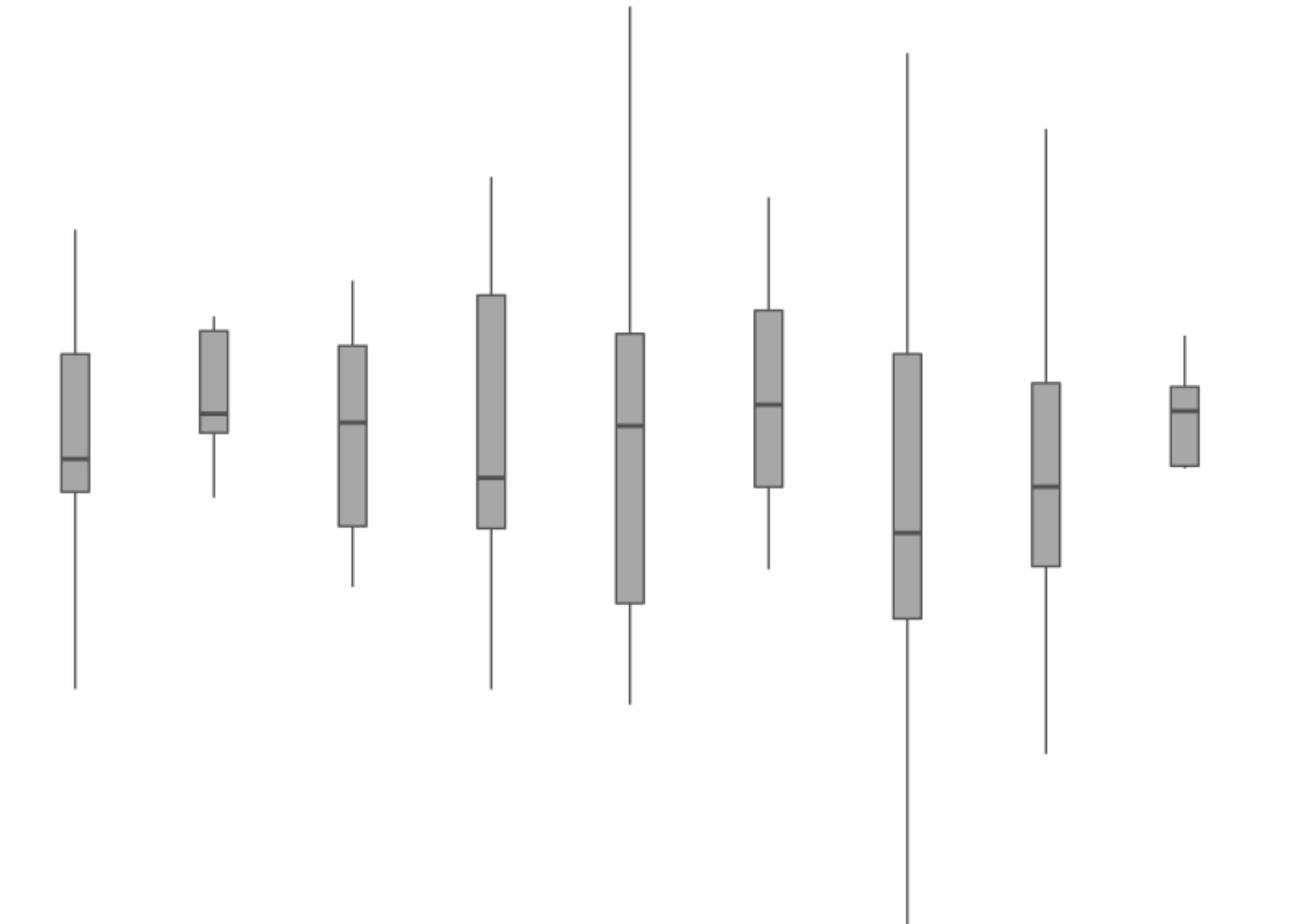
# Doumont applied to data encoding, experimentation — redesigning John Tukey's box plot



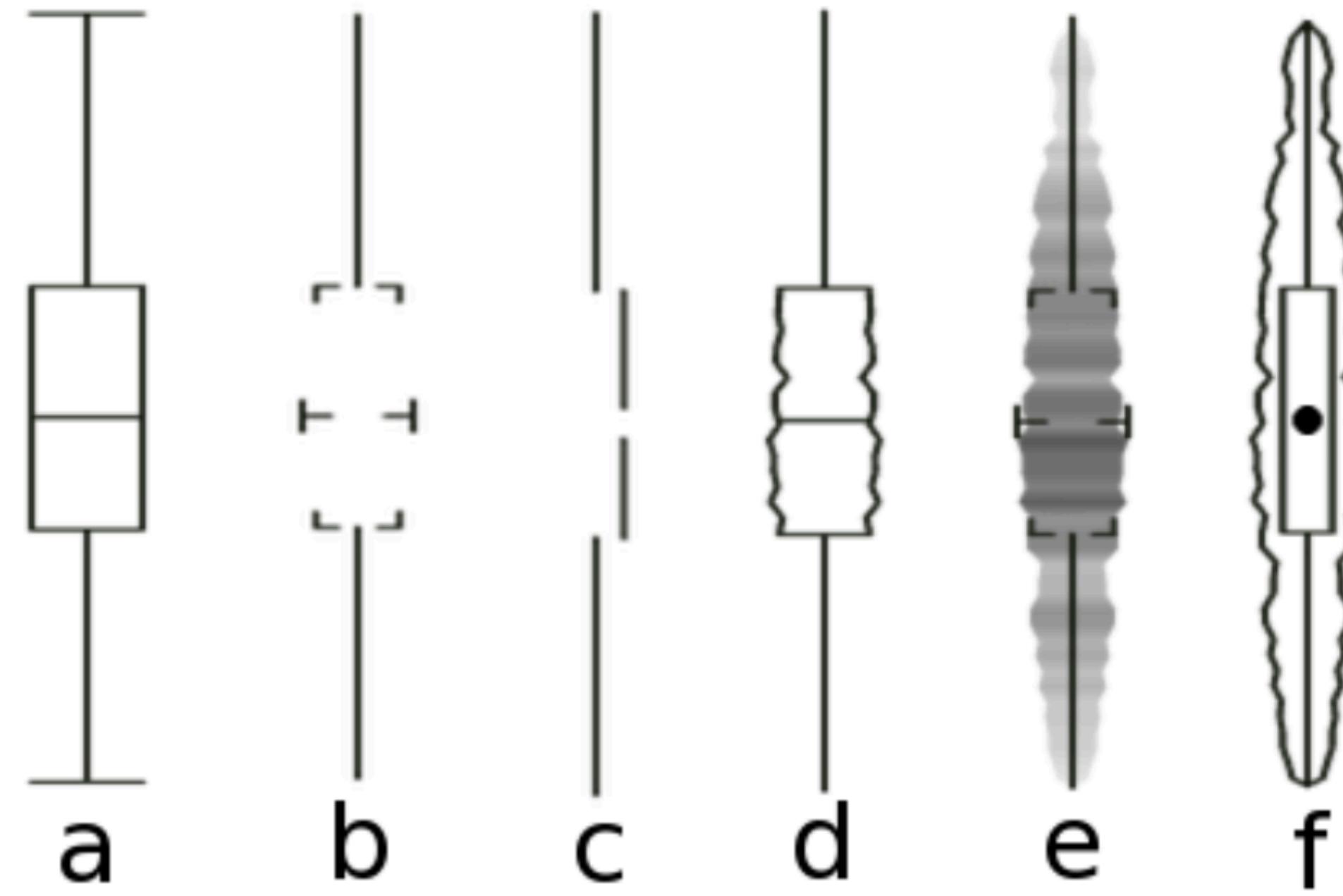
# Doumont applied to data encoding, experimentation — redesigning John Tukey's box plot



# Doumont applied to data encoding, experimentation — redesigning John Tukey's box plot



# Doumont applied to data encoding, which works best? — iterative process of *creating, questioning, testing!*



One empirical experiment, asking participants to state data characteristics, given one of these randomized versions of the boxplots shown here, suggested that one of Tufte's erasures (option C) was most cognitively difficult for viewers to interpret.

Cues like labels and gridlines, together with some strictly superfluous embellishment of data points or other design elements, may sometimes be an aid rather than an impediment to interpretation.

Adapt to your audience.

**Doumont** — “adapt to your audience”

# Doumont applied to data encoding, data-ink maximization — one of many design considerations

“Maximizing data ink (within reason) is **but a single dimension of a complex and multivariate design task.**

The principle helps conduct  
**experiments in graphical design.**

Some of those  
experiments will succeed.

There remain, however, many **other considerations** in the design of statistical graphics — not only of efficiency, but also of **complexity, structure, density, and even beauty.**”

— Tufte, Edward, *The Visual Display of Quantitative Information*

Doumont applied to data encoding, which works best? — *iterative process of creating, questioning, testing!*

**Prototypes should  
emphasize speed over polish.**

**Design is a search problem**

**Get fresh eyes frequently.  
Invite criticism.**

**Move from  
exploring to refining.**

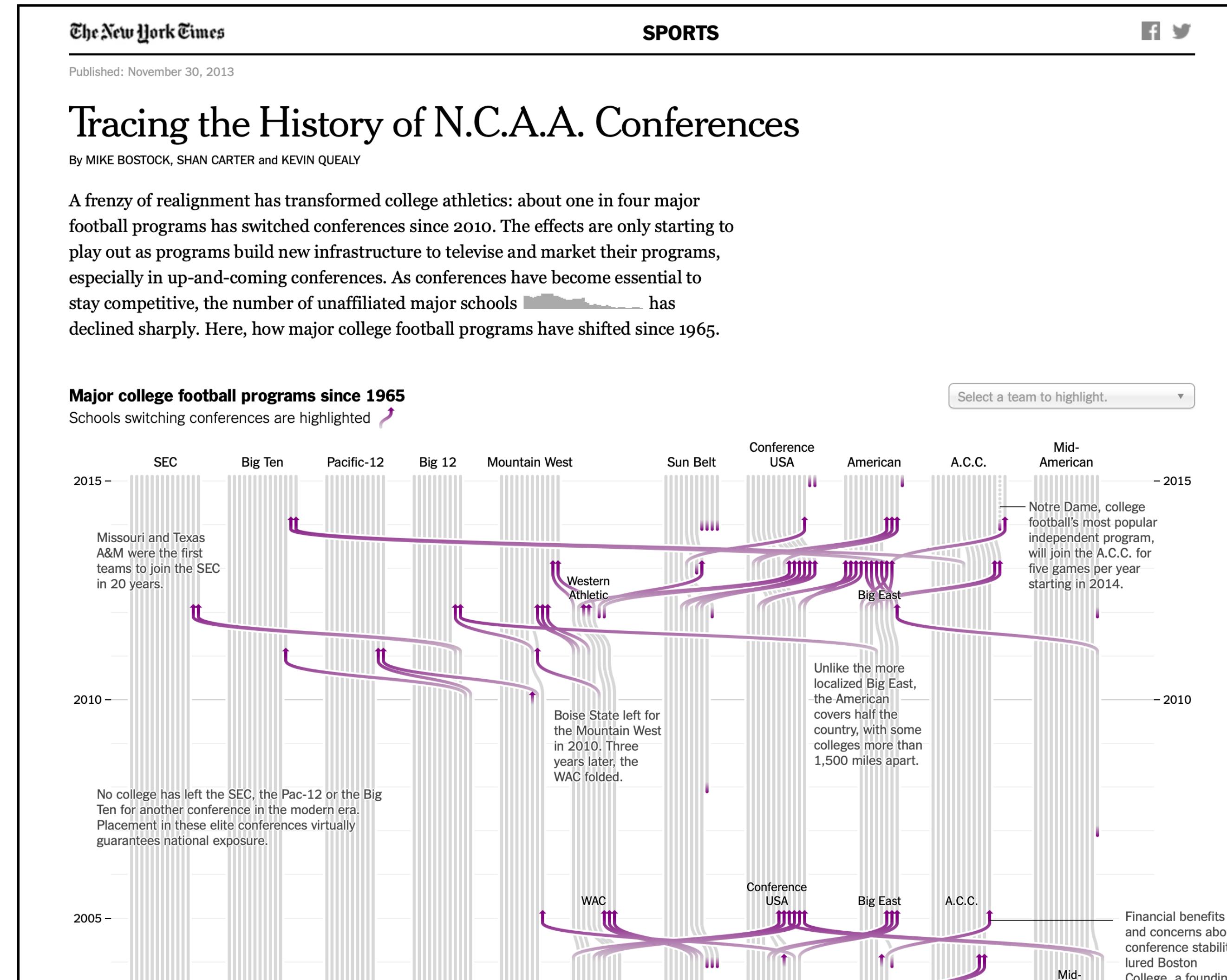
# Doumont applied to data encoding, which works best? — iterative process of *creating, questioning, testing!*



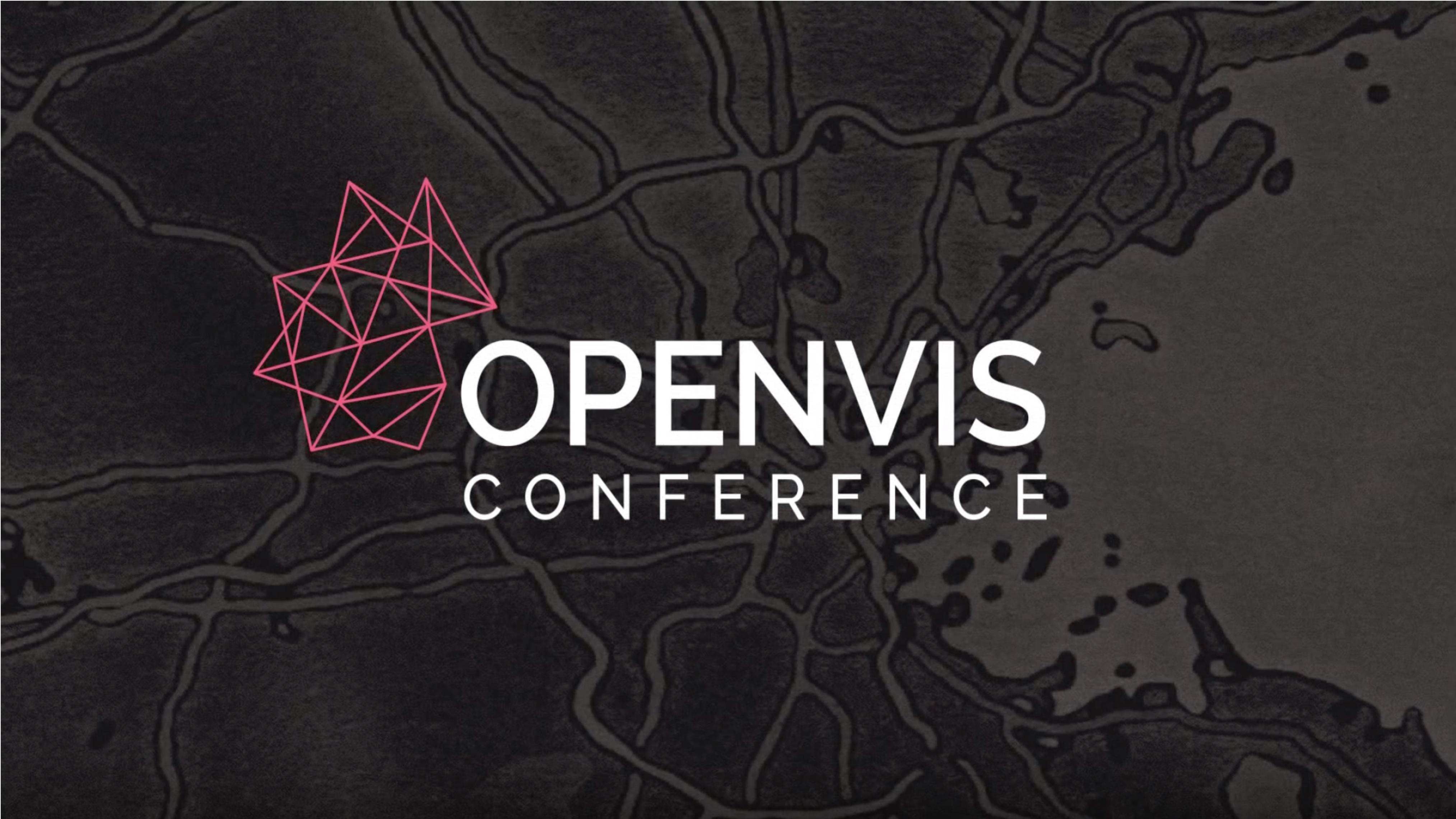
Doumont applied to data encoding, which works best? — iterative process of *creating, questioning, testing!*



# Doumont applied to data encoding, which works best? — iterative process of *creating, questioning, testing!*



Doumont applied to data encoding, which works best? — iterative process of *creating, questioning, testing!*



# Doumont applied to data encoding, which works best? — iterative process of *creating, questioning, testing!*

“

The ceramics teacher announced on opening day that he was dividing the class into two groups. All those on the left side of the studio, he said, would be graded solely on the quantity of work they produced, all those on the right solely on its quality. His procedure was simple: on the final day of class he would bring in his bathroom scales and weigh the work of the “quantity” group: fifty pounds of pots rated an “A”, forty pounds a “B”, and so on. Those being graded on “quality”, however, needed to produce only one pot —albeit a perfect one —to get an “A”.

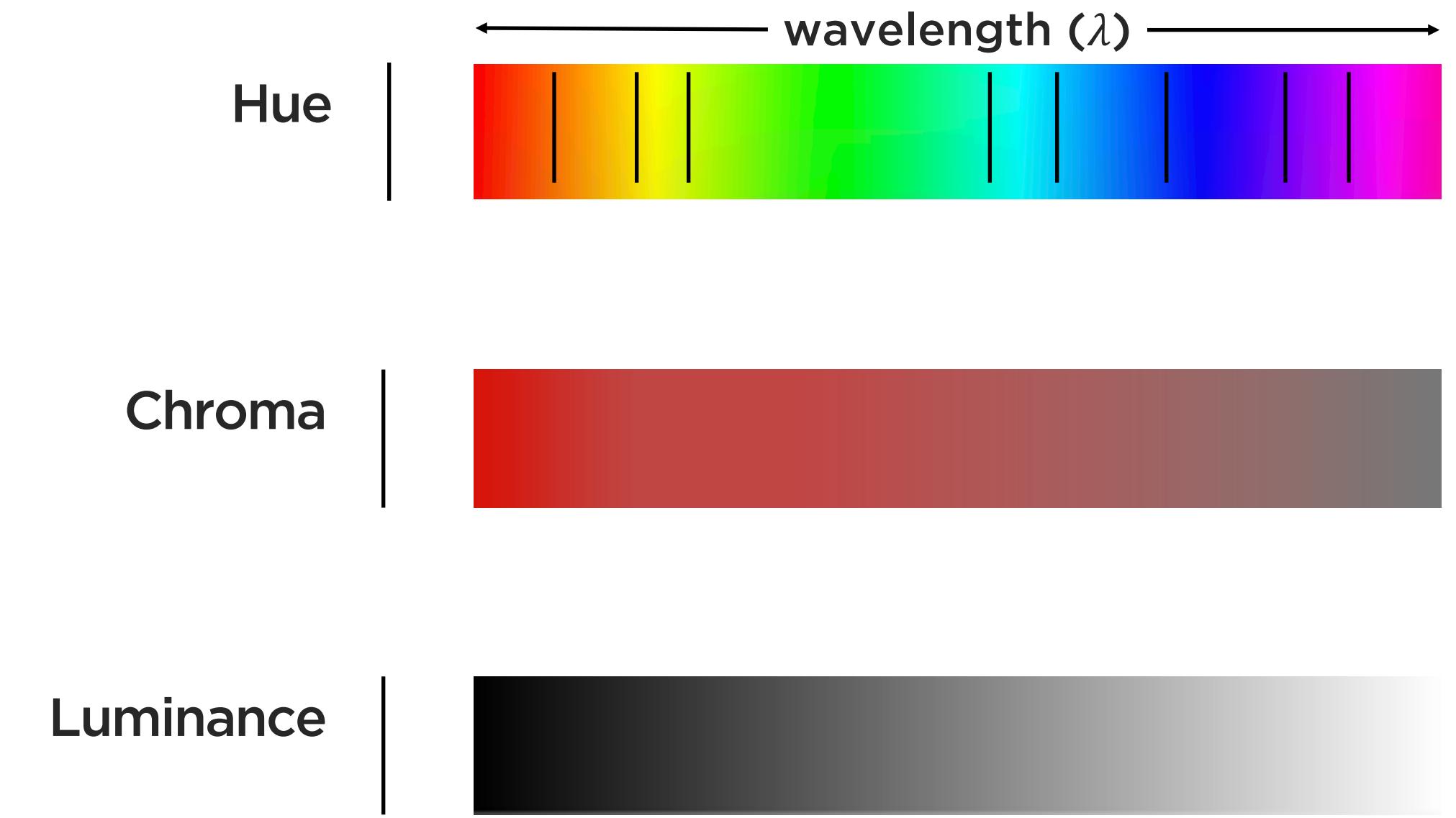
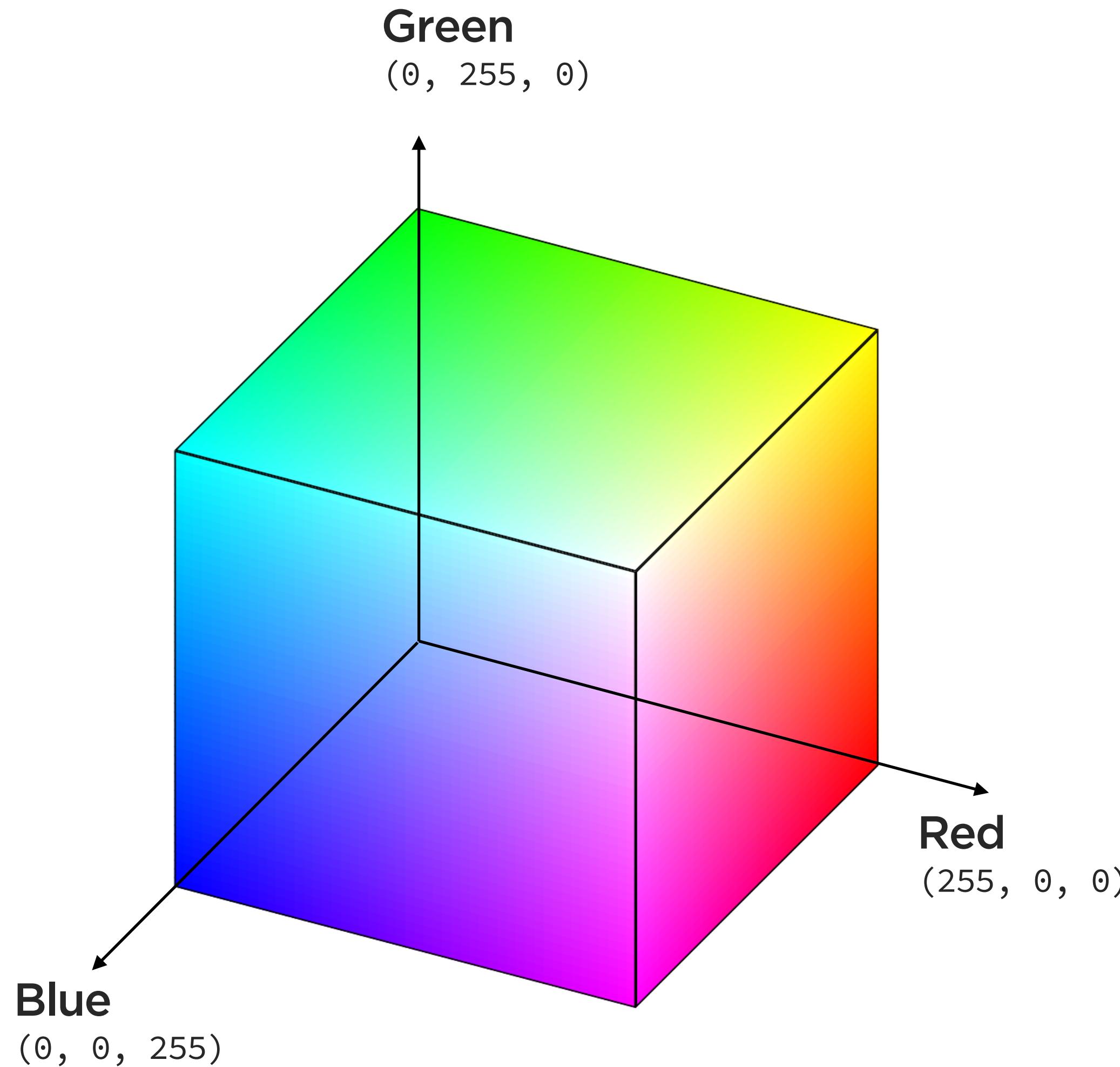
Well, came grading time and a curious fact emerged: **the works of highest quality were all produced by the group being graded for quantity.** It seems that while the “quantity” group was busily churning out piles of work—and learning from their mistakes —the “quality” group had sat theorizing about perfection, and in the end had little more to show for their efforts than grandiose theories and a pile of dead clay.

”

— Bayles and Orland, *Art & Fear. Observations on the Perils (and Rewards) of Artmaking*. The Image Continuum, 1993.

**encoding data as color**

encoding data as color, encode data using color spaces, which are mathematical models



encoding data as color, how can we map data to light, whether using its hue, chroma, or luminance?

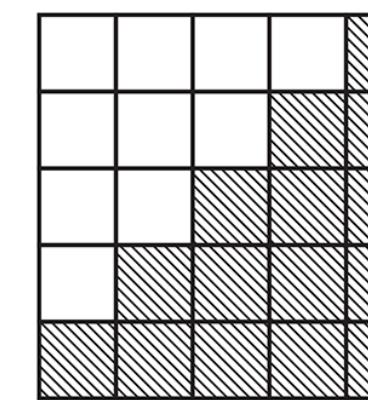


# encoding data as color, perceived brightness is nonlinear function of luminance

**LUMINANCE** : the *measured* amount of light coming from some region of space.

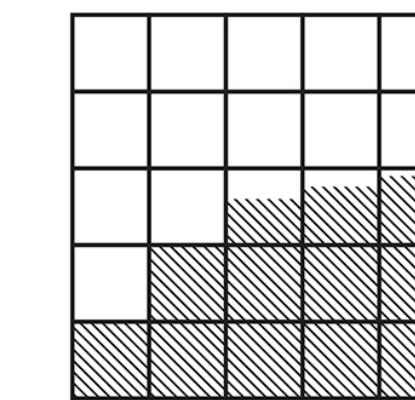
**BRIGHTNESS** : the *perceived* amount of light coming from that region of space.

# encoding data as color, visual perception of arithmetical progression depends on physical geometric progression

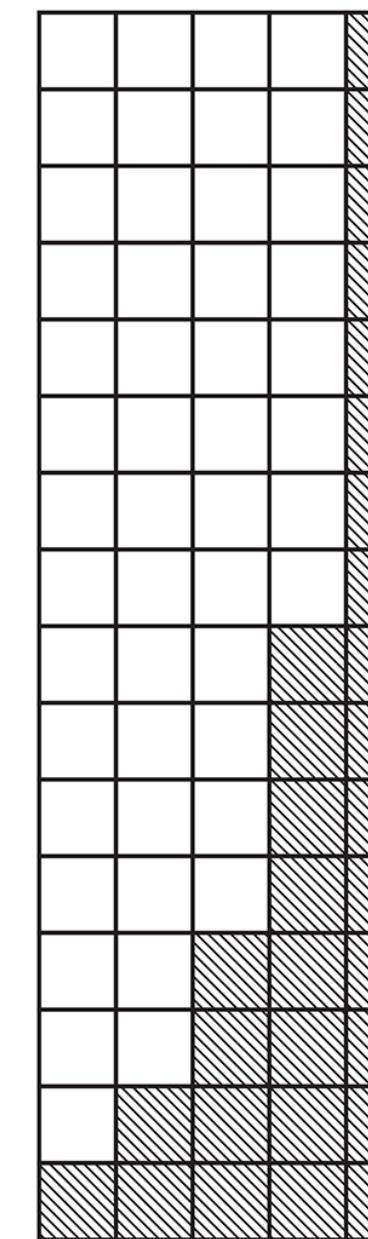


THIS PHYSICAL FACT

REDUCES TO

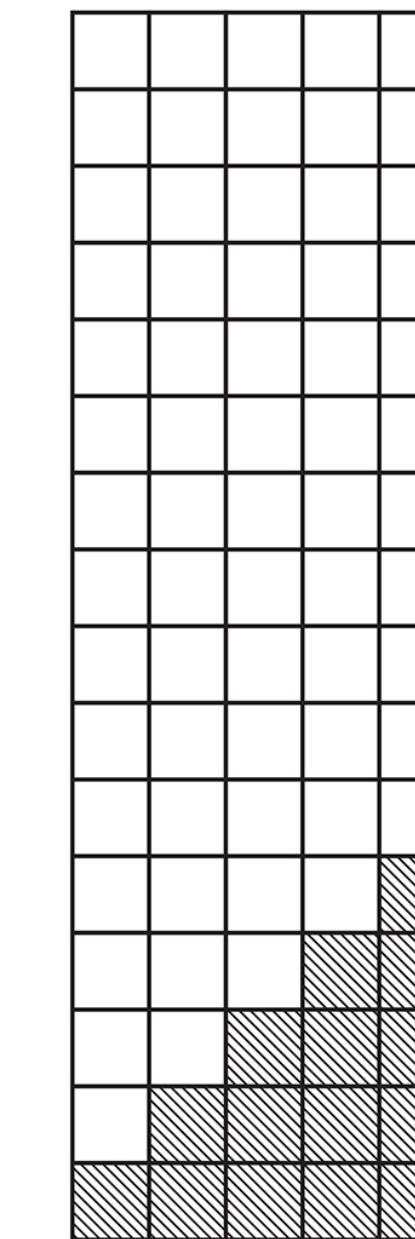


THIS PSYCHOLOGICAL EFFECT



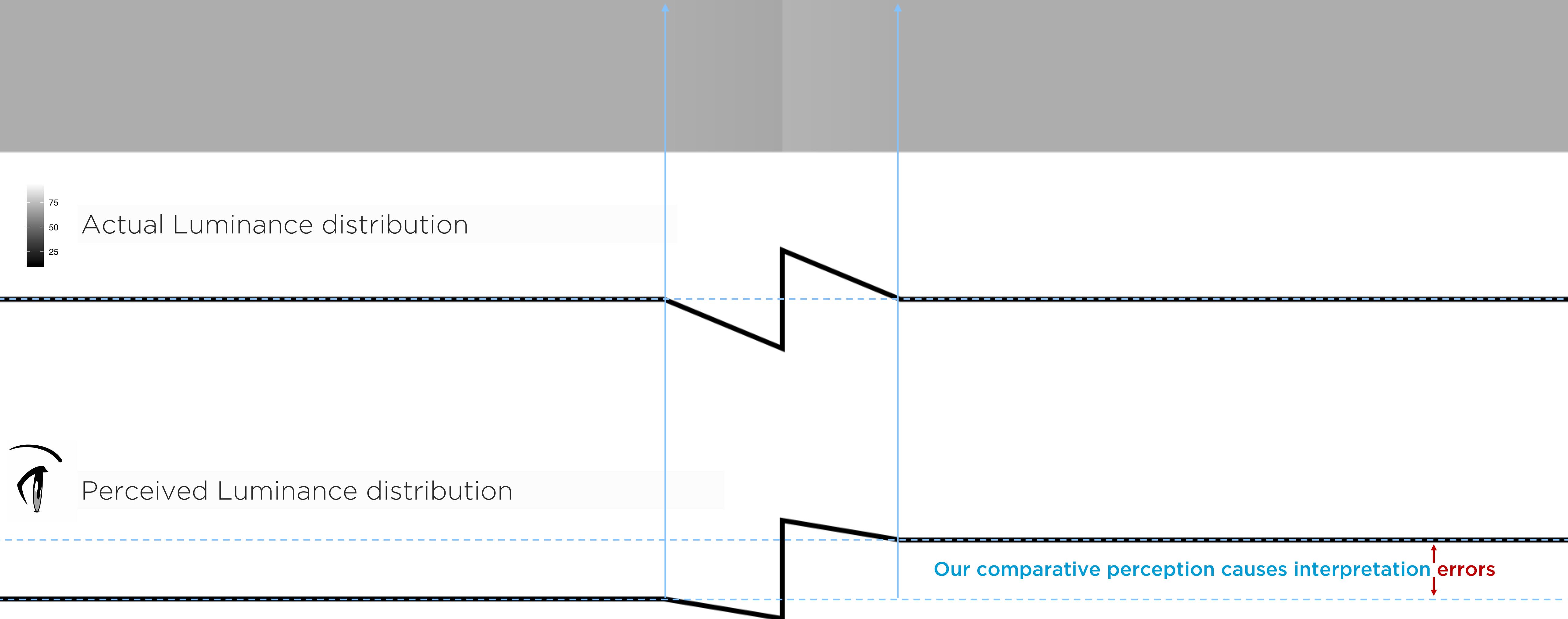
THIS PHYSICAL FACT

PRODUCES

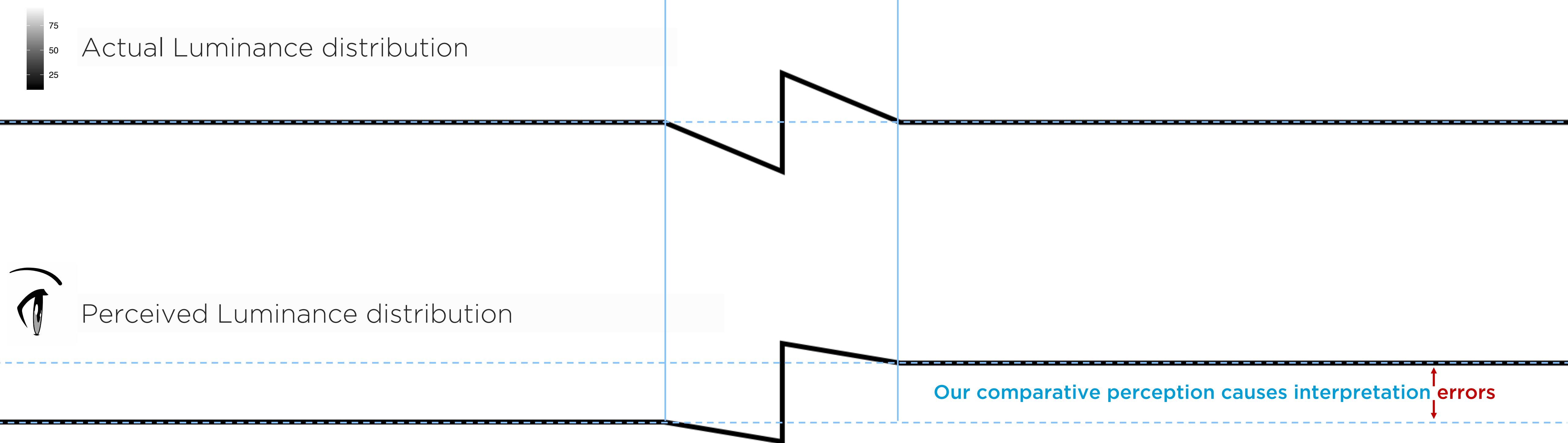


THIS PSYCHOLOGICAL EFFECT

color, humans have evolved to see edge contrasts. We see comparative — not absolute — luminance value.



color, humans have evolved to see edge contrasts. We see comparative — not absolute — luminance value.



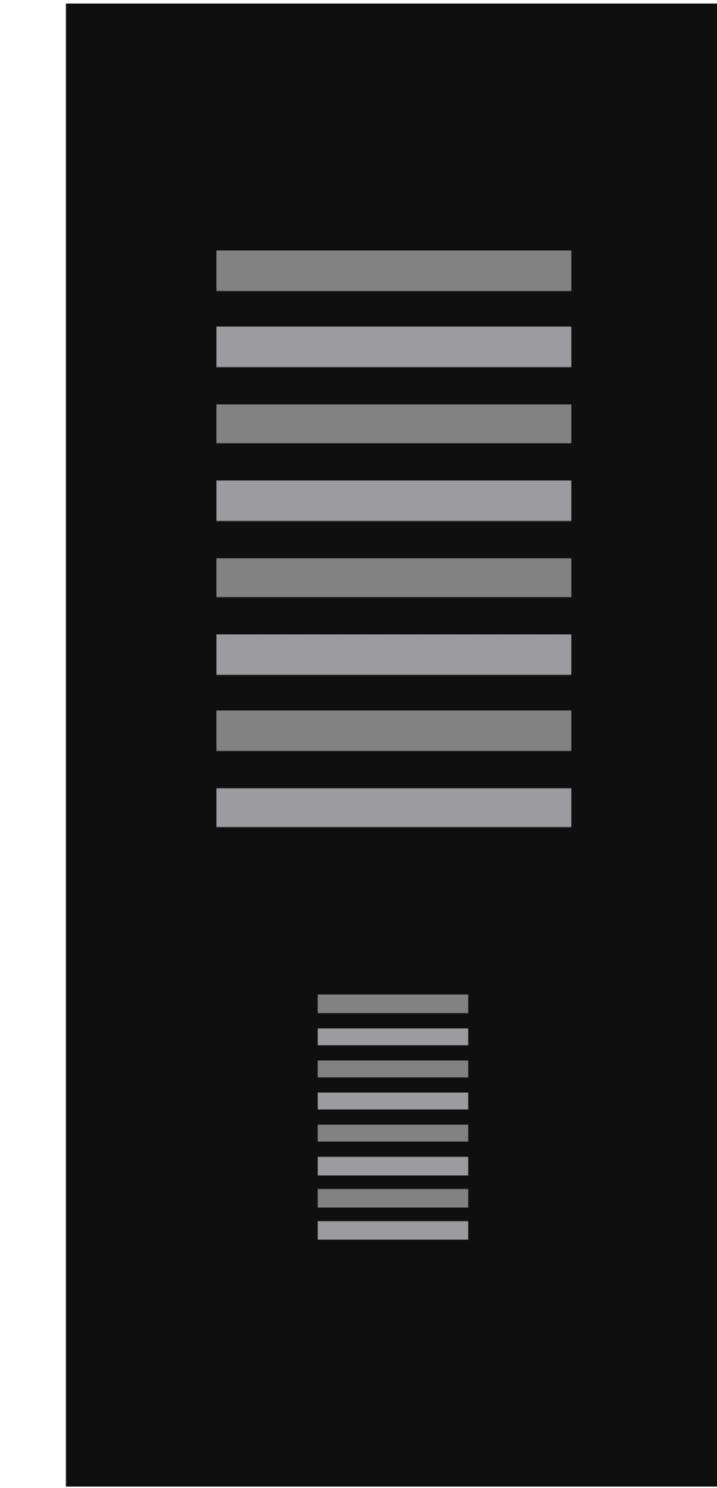
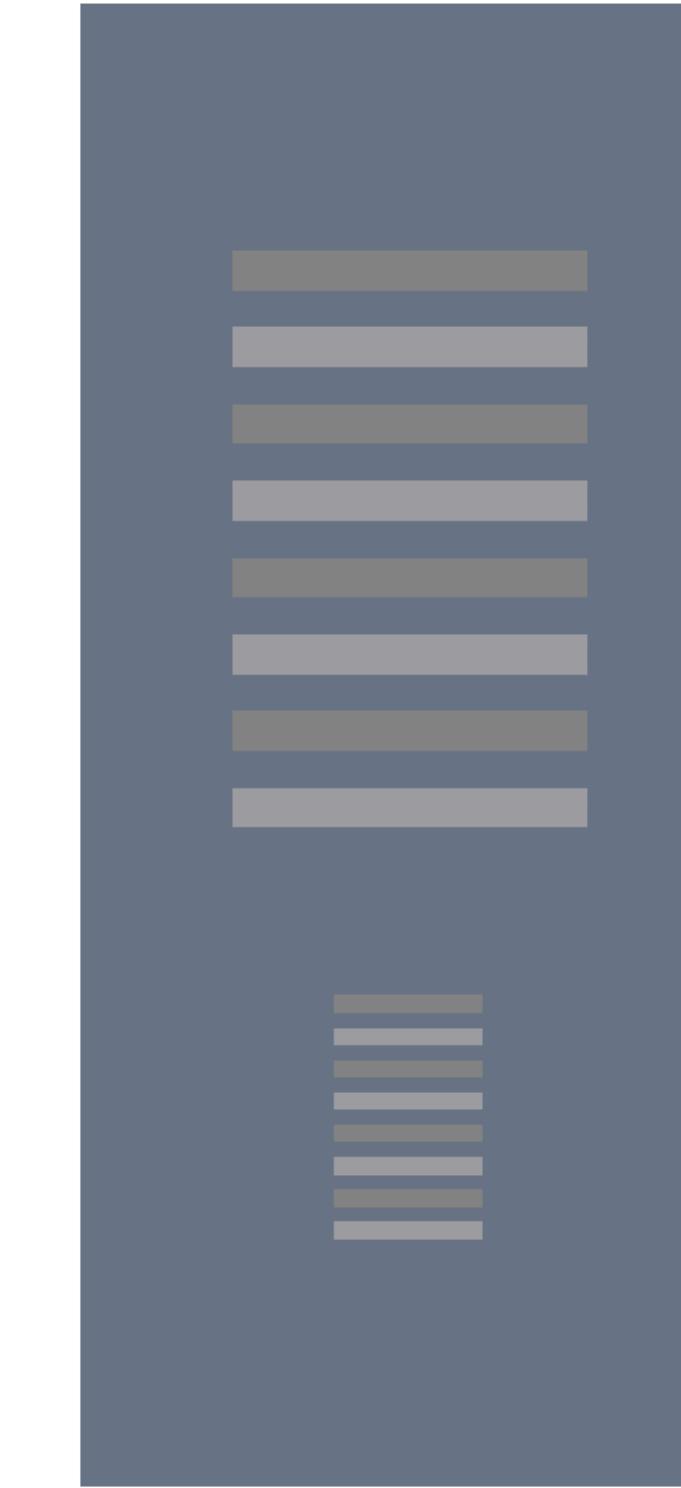
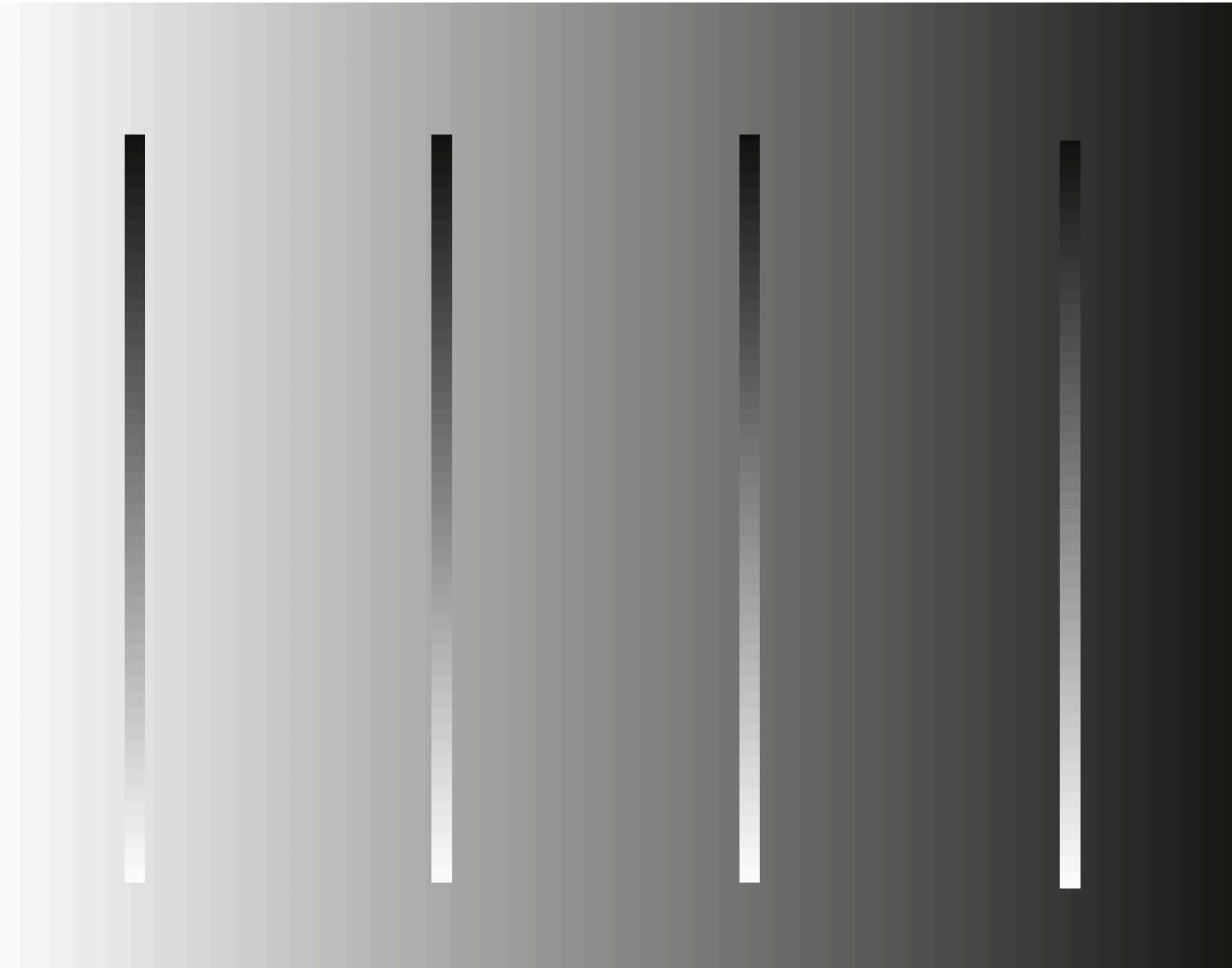
color, background and adjacent luminance can interfere with our perception



**color, background and adjacent luminance can interfere with our perception**

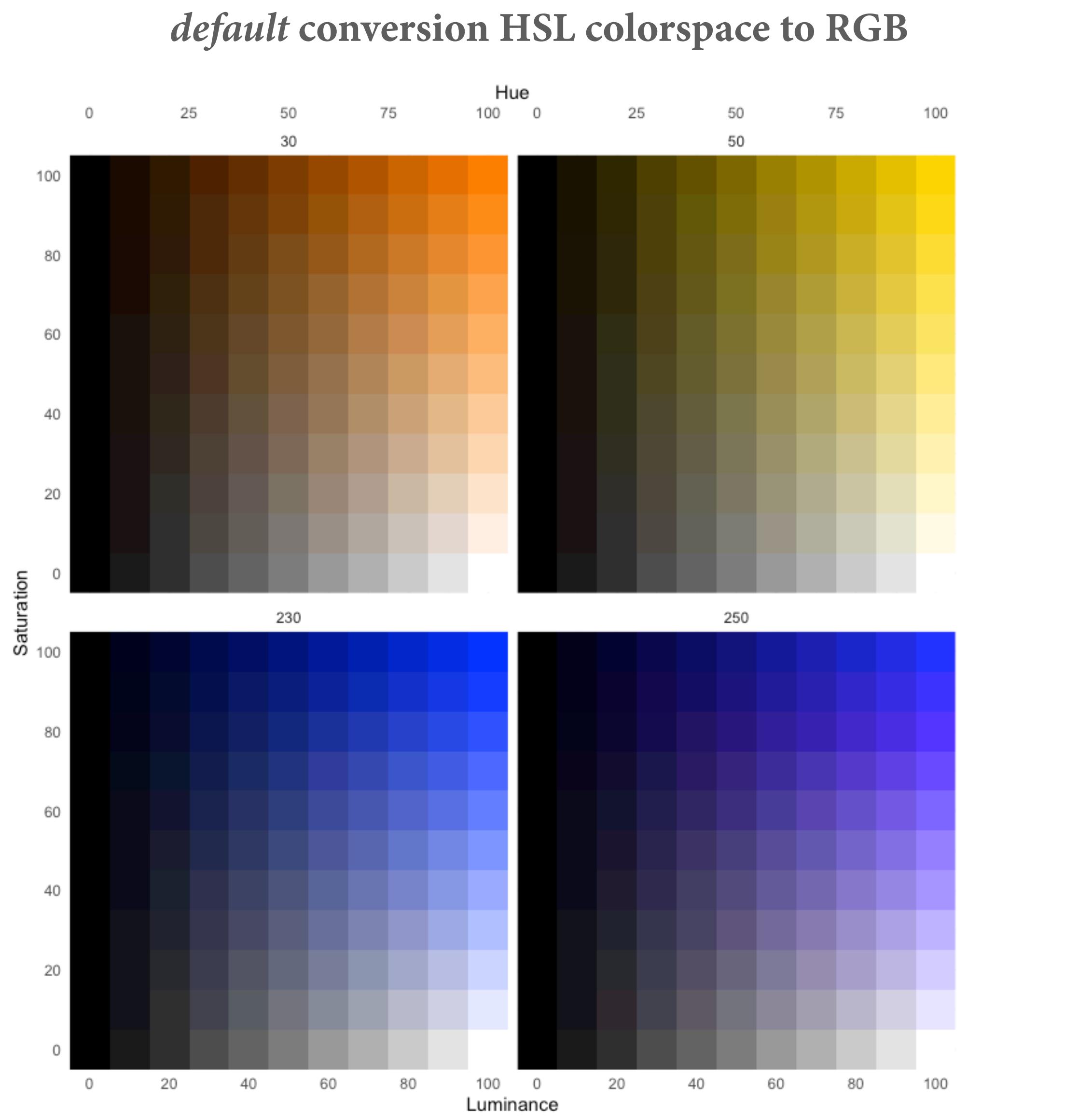


color, high foreground to background luminance contrast enhances shape, lower contrast enhances grayscale



# color, example encoding data into hue, saturation, and luminance

*default conversion HSL colorspace to RGB*

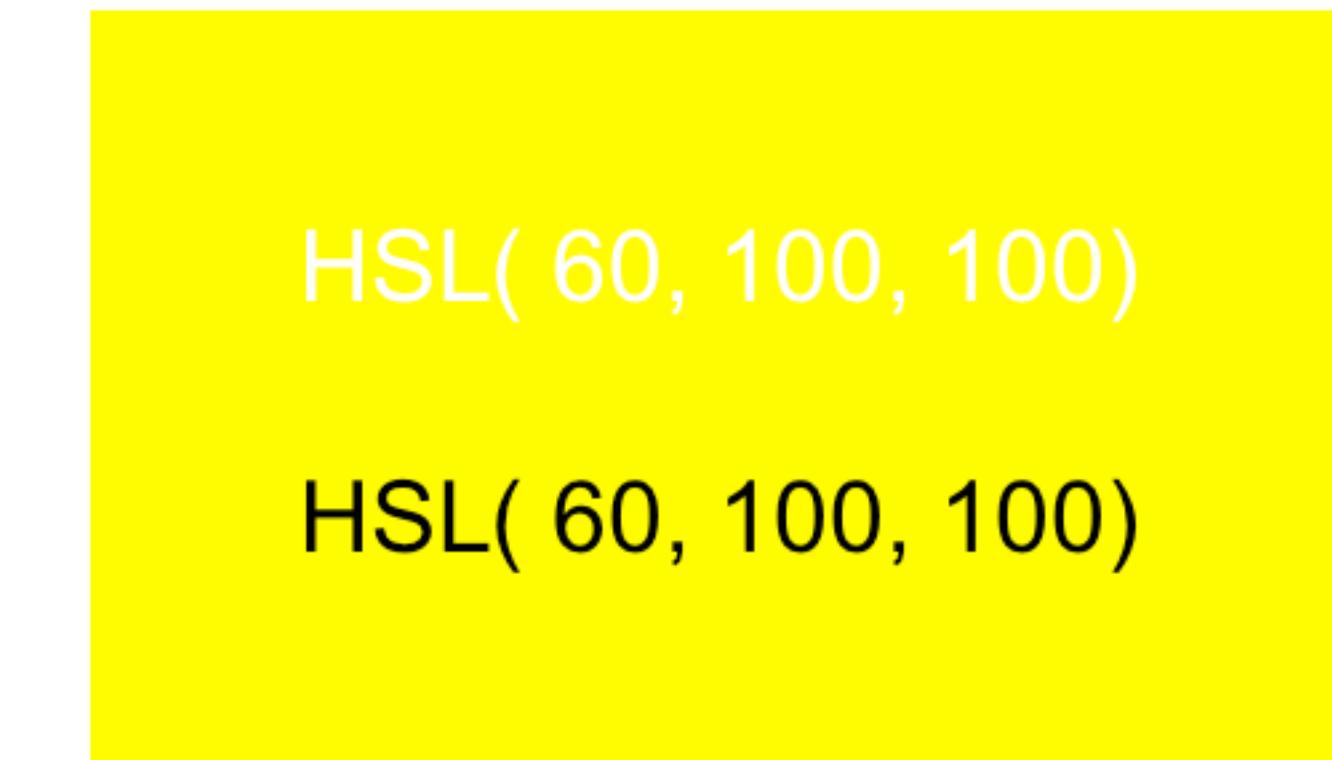
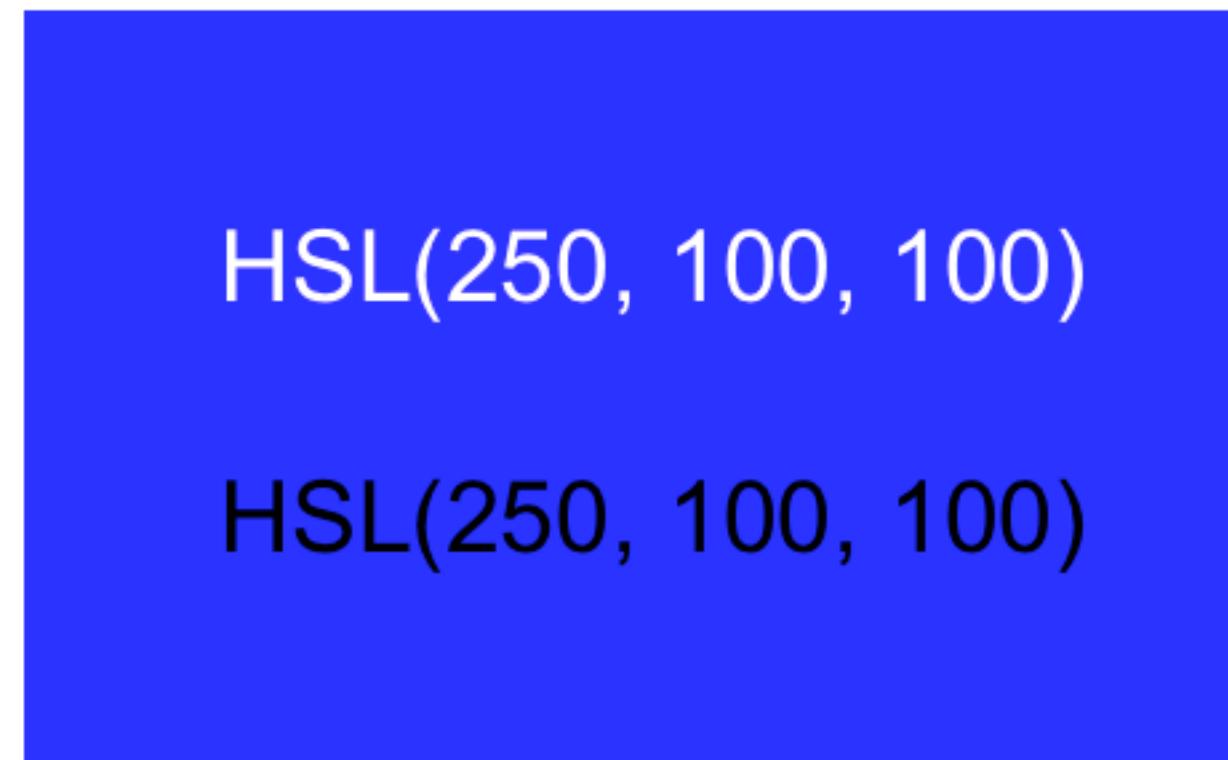


color, as with luminance, *hue* values in the RGB color space fail to uniformly scale across values



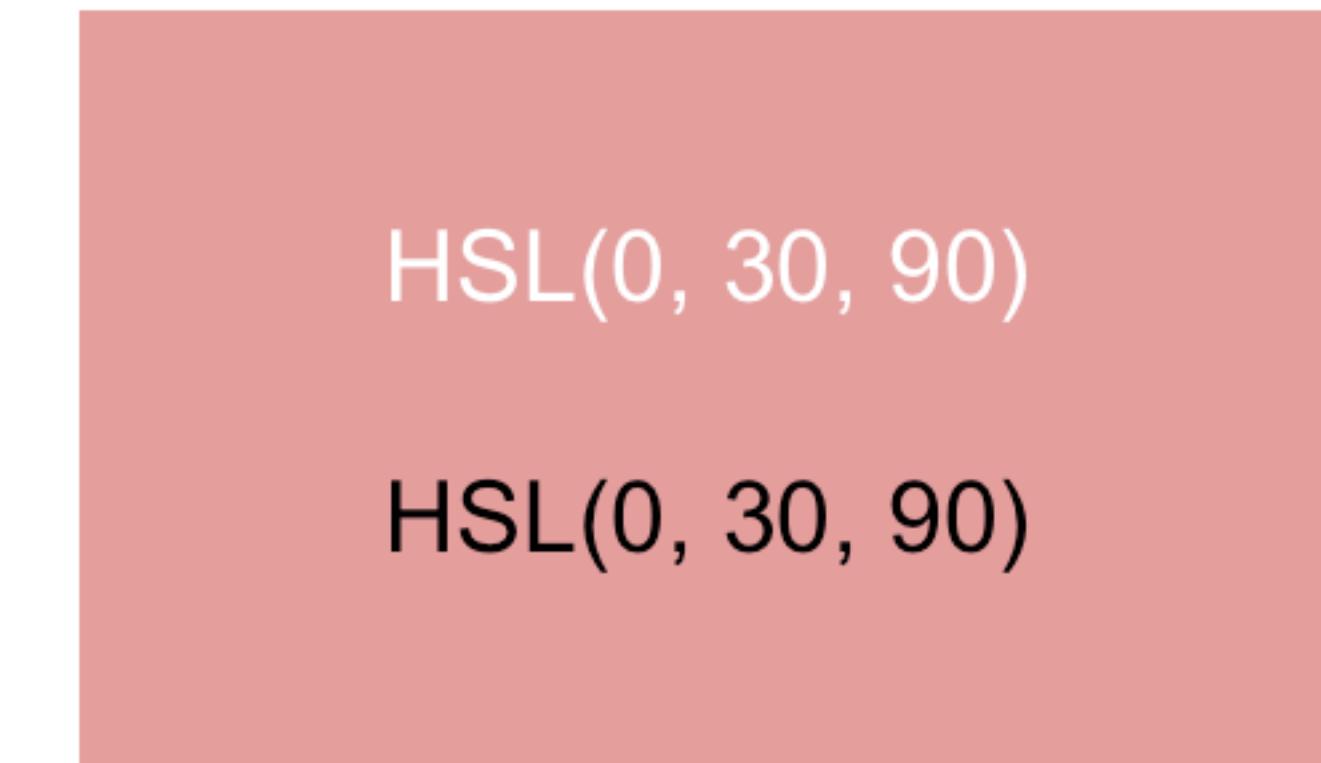
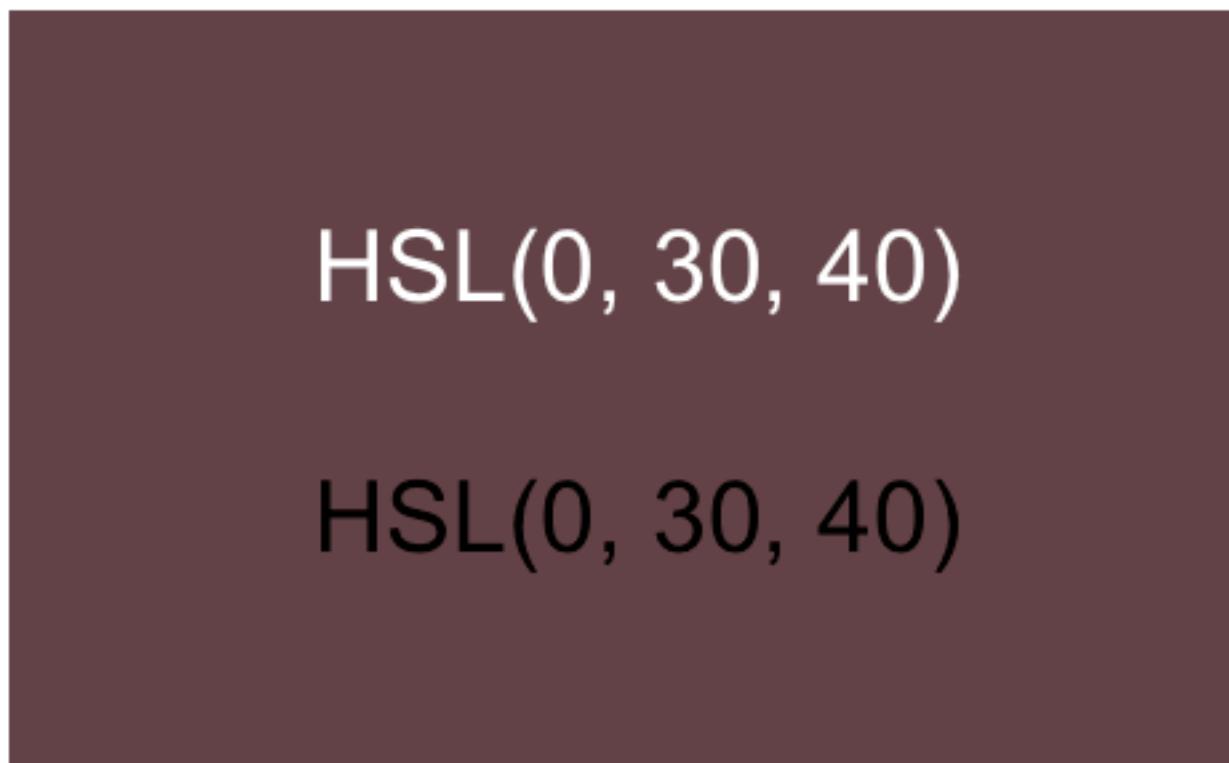
color, HSL colorspace is intuitive, but not perceptually uniform in each attribute

Same luminance or lightness?



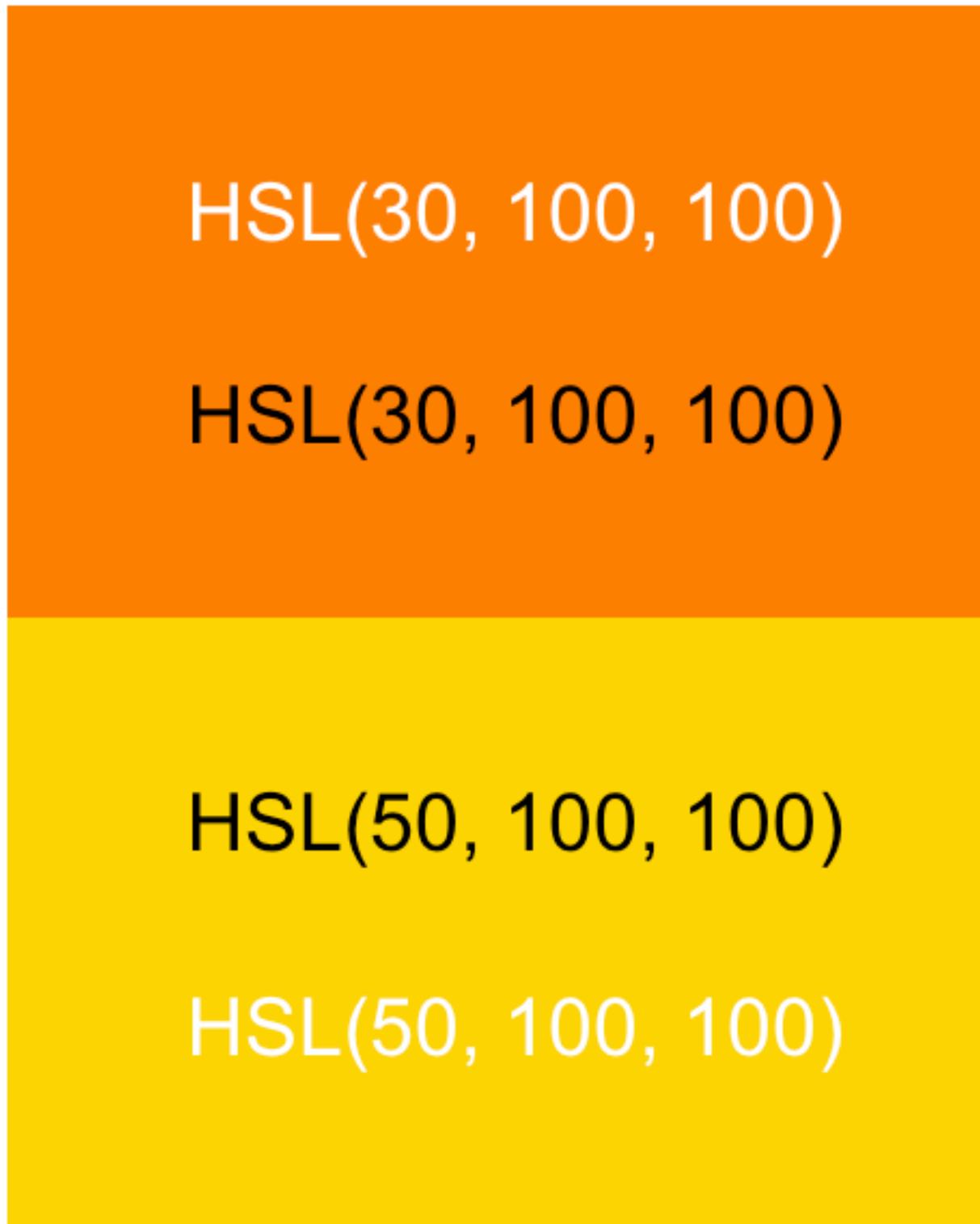
color, HSL colorspace is intuitive, but not perceptually uniform in each attribute

Same saturation?

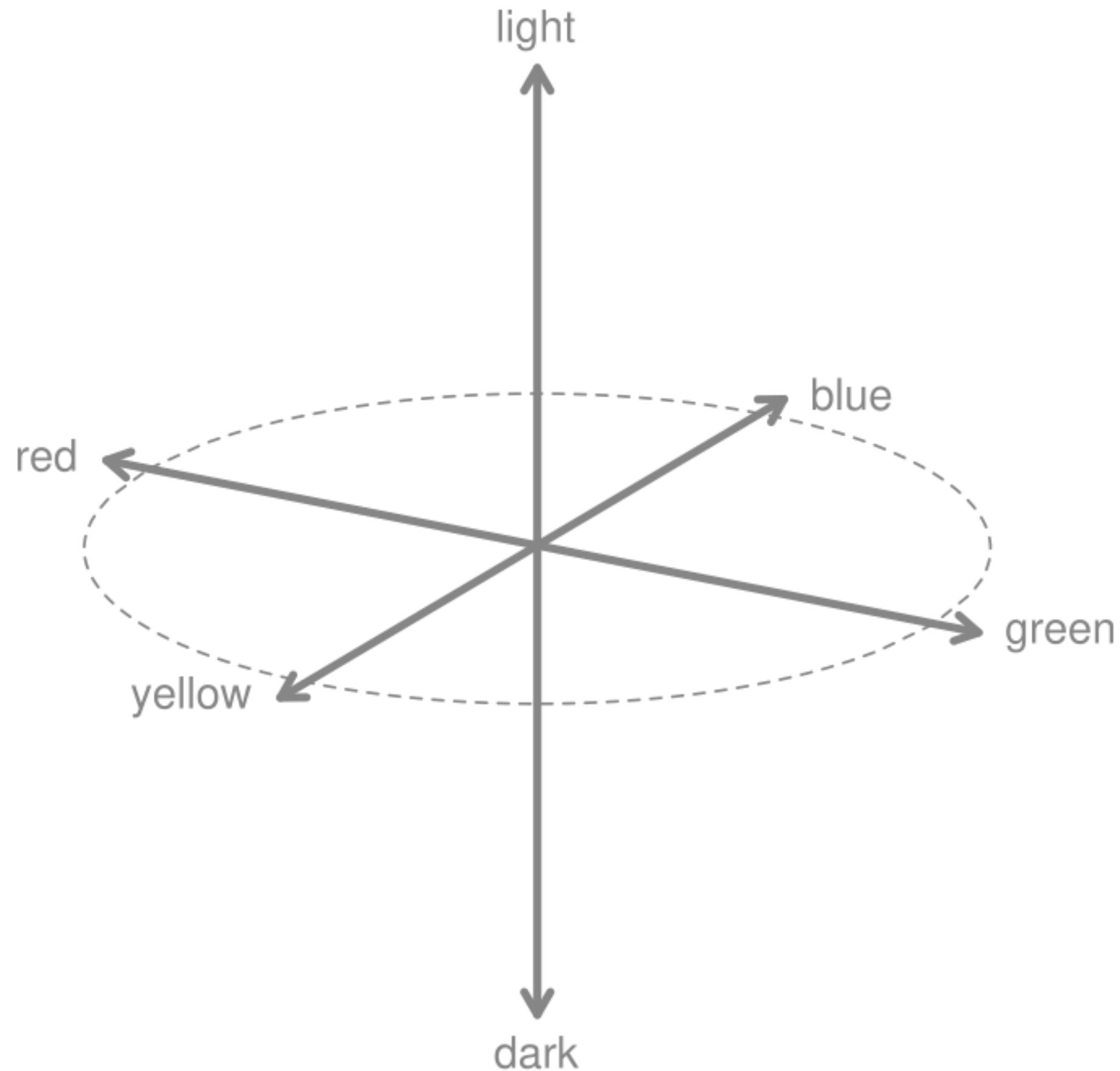


color, HSL colorspace is intuitive, but not perceptually uniform in each attribute

## Equal difference between hues?



# color, HSL colorspace is intuitive, but not perceptually uniform in each attribute



The International Commission on Illumination (CIE) studied human perception and re-mapped color into a space where we perceive color changes uniformly.

Their **CIELuv** color model has two dimensions — u and v — that represent color scales from red to green and yellow to blue.

# color, example encoding data as *perceptually uniform* color attributes: R · ggplot2 · HSLuv

Load functions for mapping data to perceptually-uniform color, from my R package: <https://github.com/ssp3nc3r/hsluv-rcpp>

```
library(HSLuv)
```

Create sample data encoded as hue, saturation, luminance

```
df <- expand.grid(H = c(30, 50, 230, 250),  
                   S = seq(0, 100, by = 10),  
                   L = seq(0, 100, by = 10))
```

Convert HSLuv scaled values to RGB color space as hex code #RRGGBB

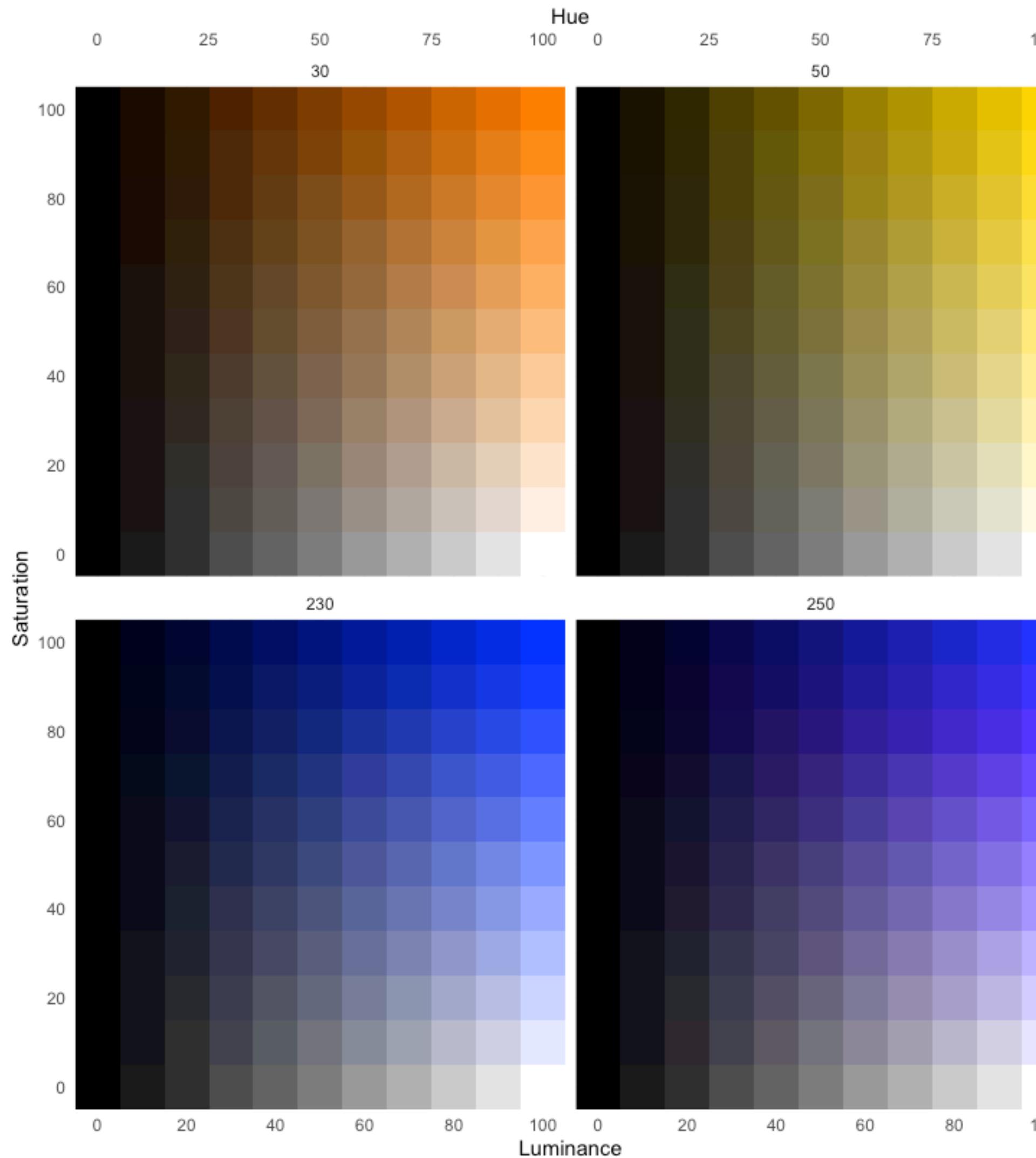
```
df$colors <- with(df, hsluv_hex(H, S, L) )
```

Plot data encoded as colors

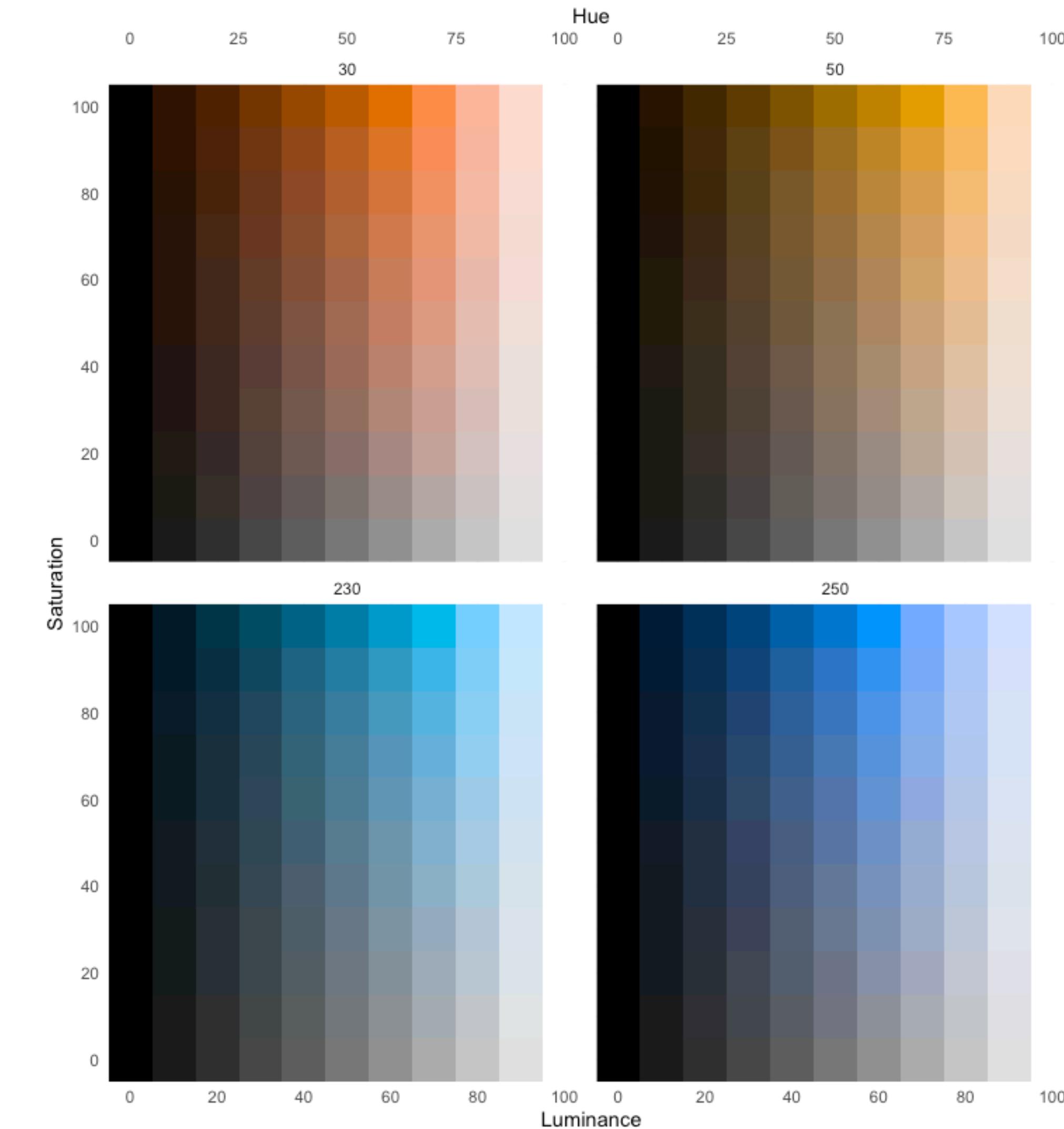
```
ggplot(df) +  
  facet_wrap(~H) +  
  scale_x_continuous(  
    breaks = seq(0, 100, by = 20),  
    expand = c(0,0),  
    sec.axis = sec_axis(~., name = 'Hue')) +  
  scale_y_continuous(  
    breaks = seq(0, 100, by = 20),  
    expand = c(0,0)) +  
  geom_raster(  
    mapping = aes(  
      x = L,  
      y = S),  
    fill = df$colors) +  
  labs(x = 'Luminance',  
       y = 'Saturation')
```

# color, example encoding data into hue, saturation, and luminance

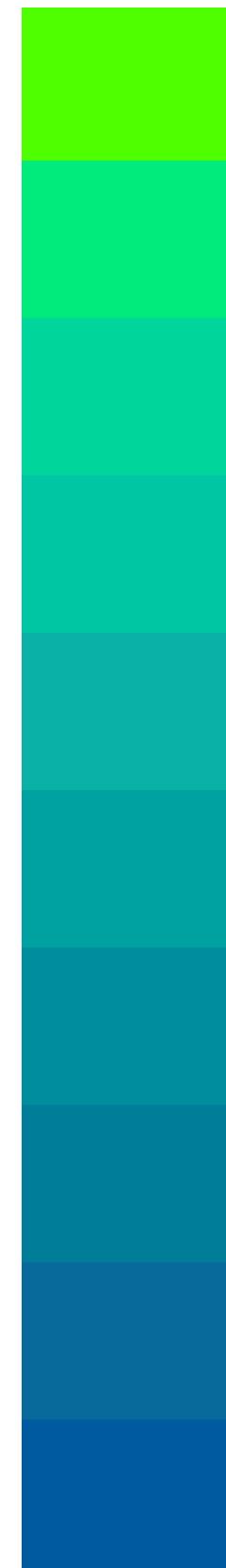
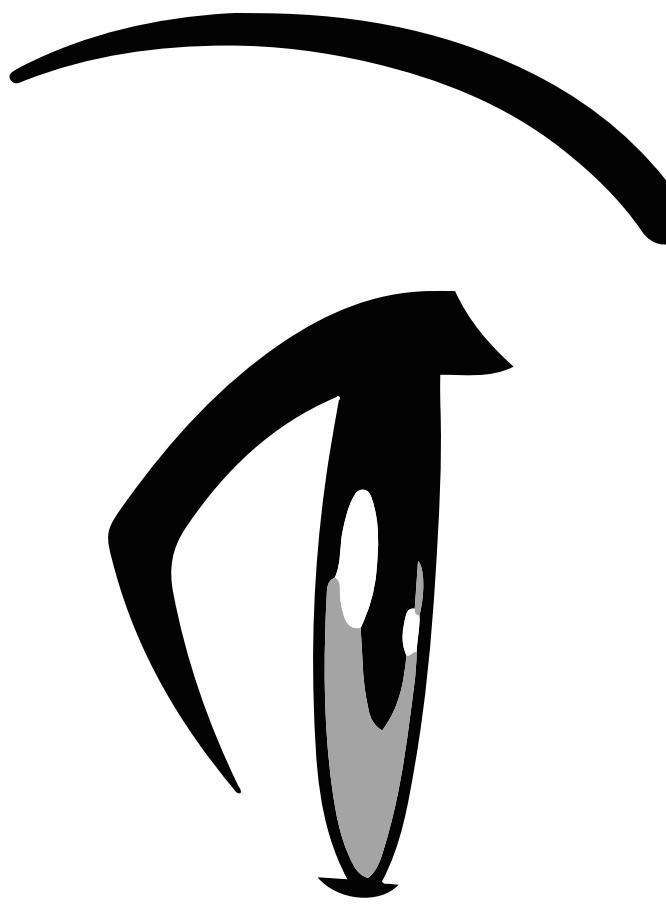
*default conversion HSL colorspace to RGB*



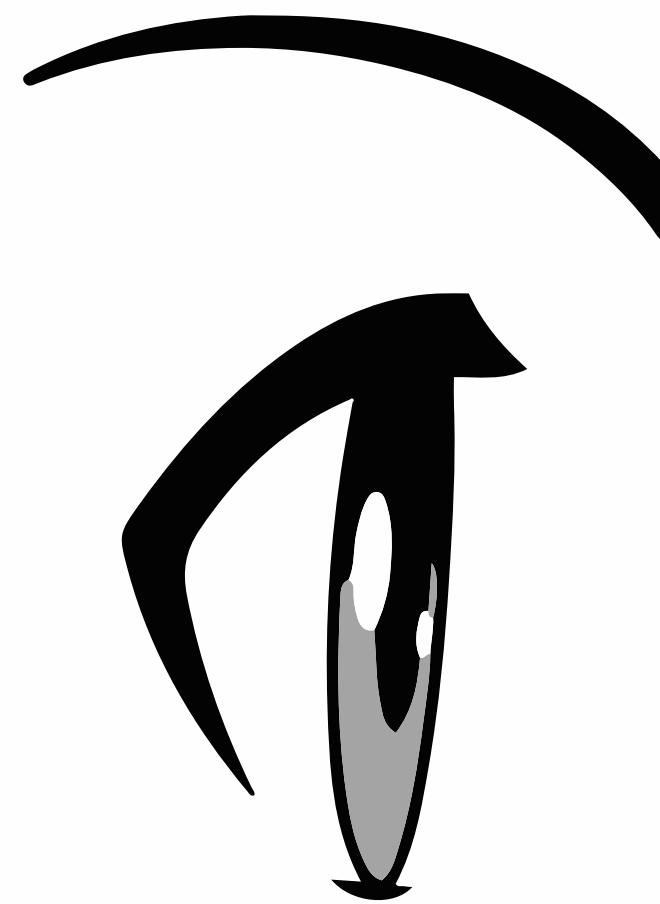
*perceptually uniform conversion HSLuv colorspace to RGB*



color, perceptually uniform color spaces better represent quantity

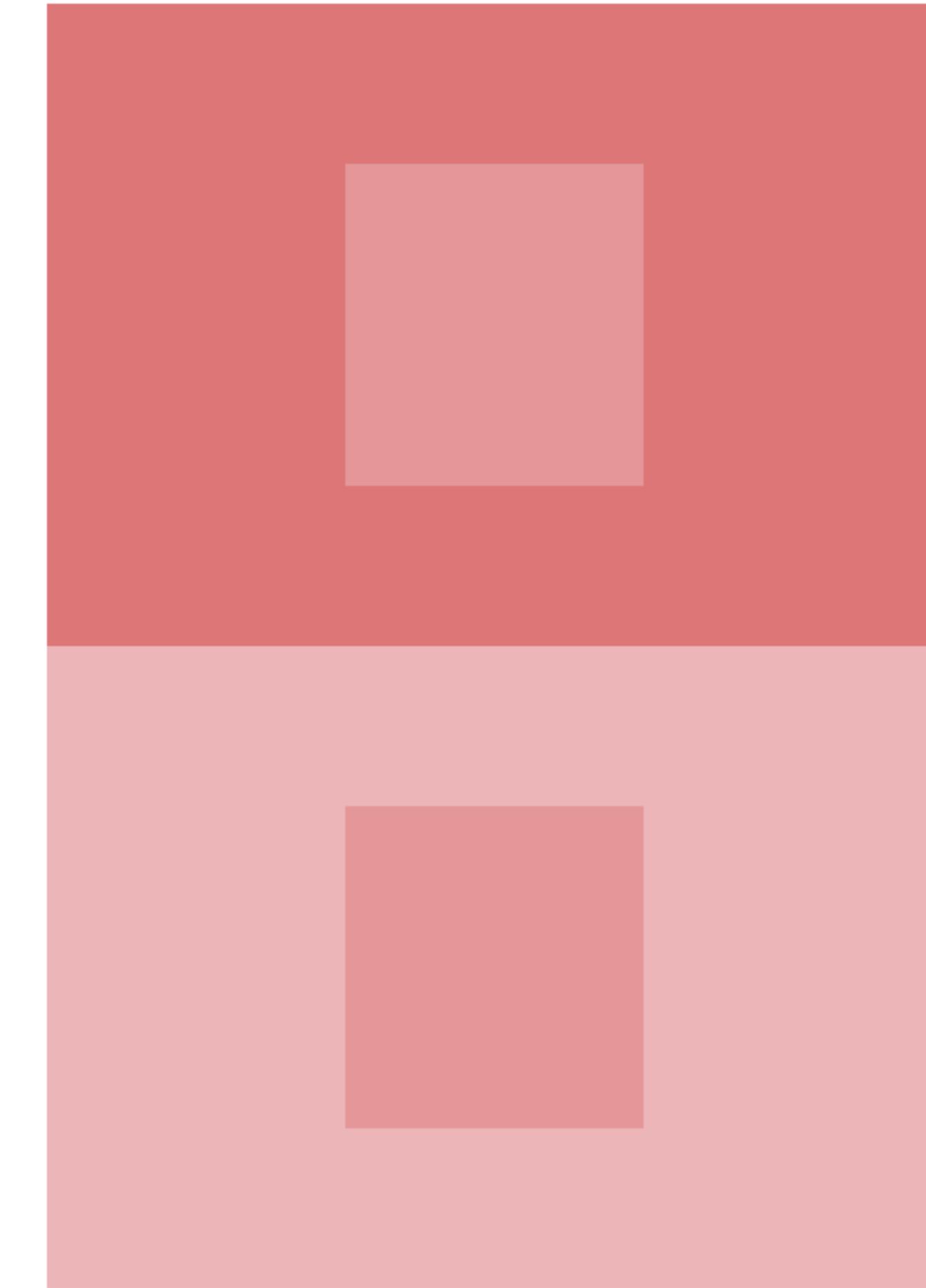
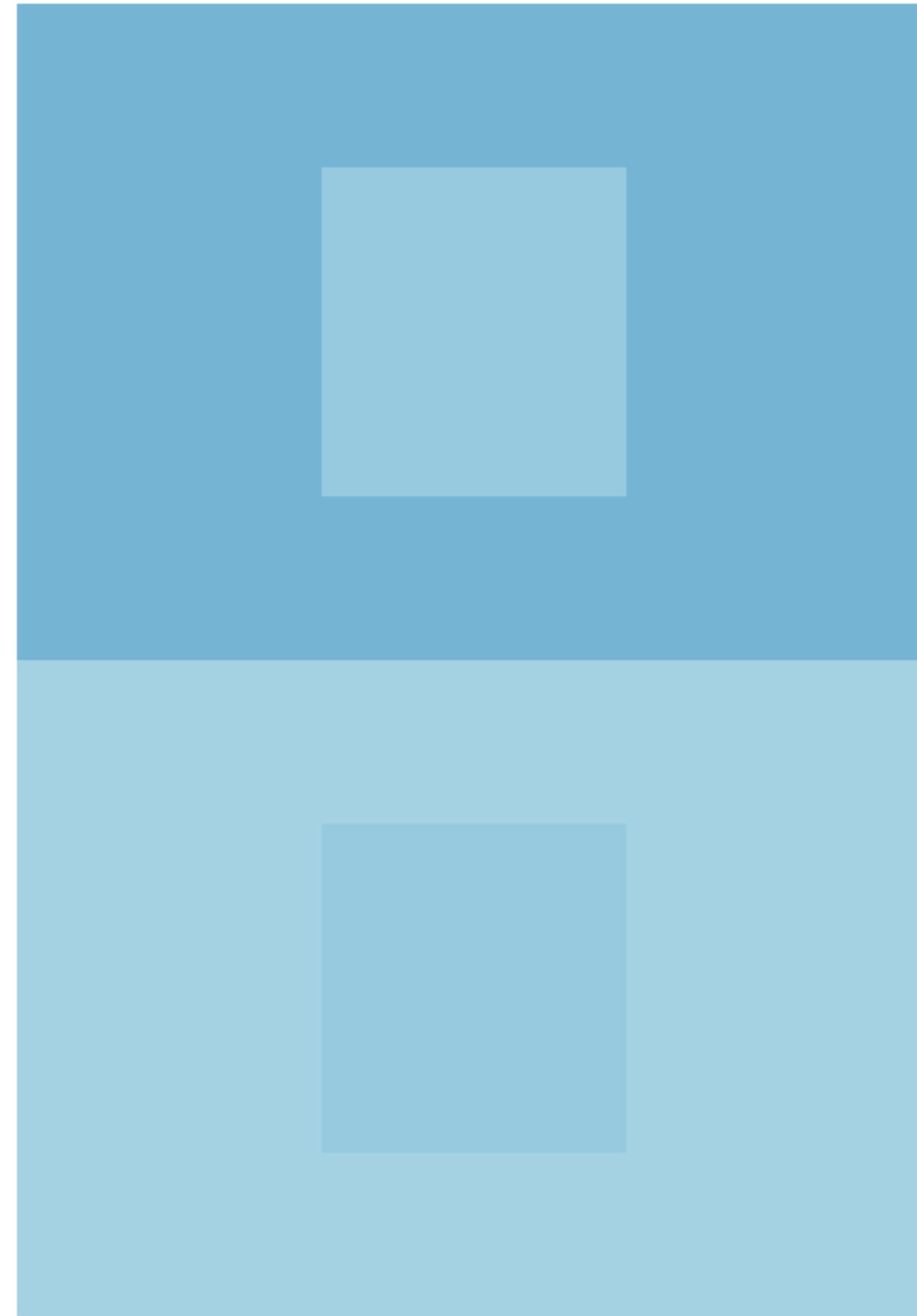


color, perceptually uniform color spaces also help in distinguishing categorical data



**interaction of color**

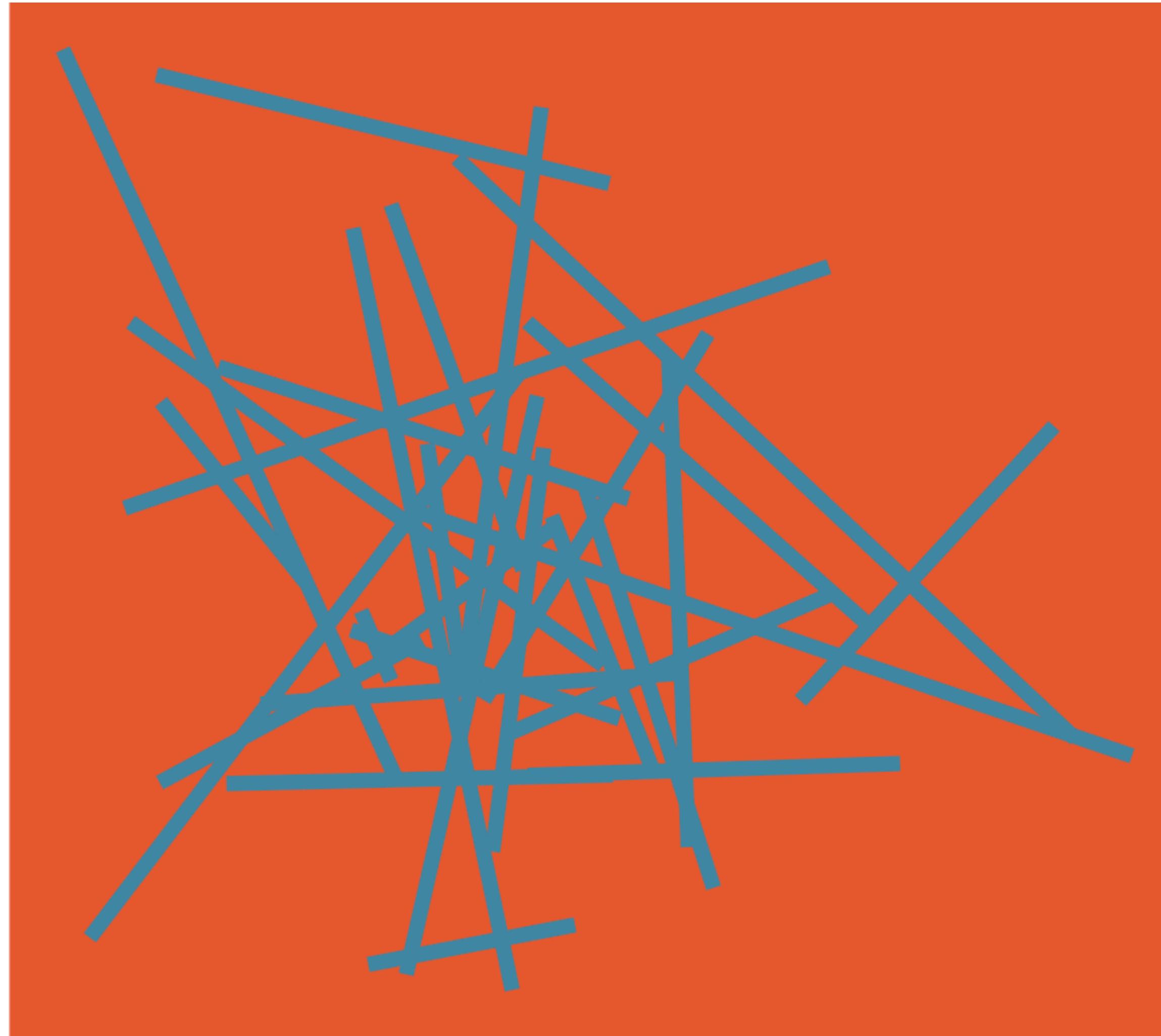
# interaction of color, one color appearing as two



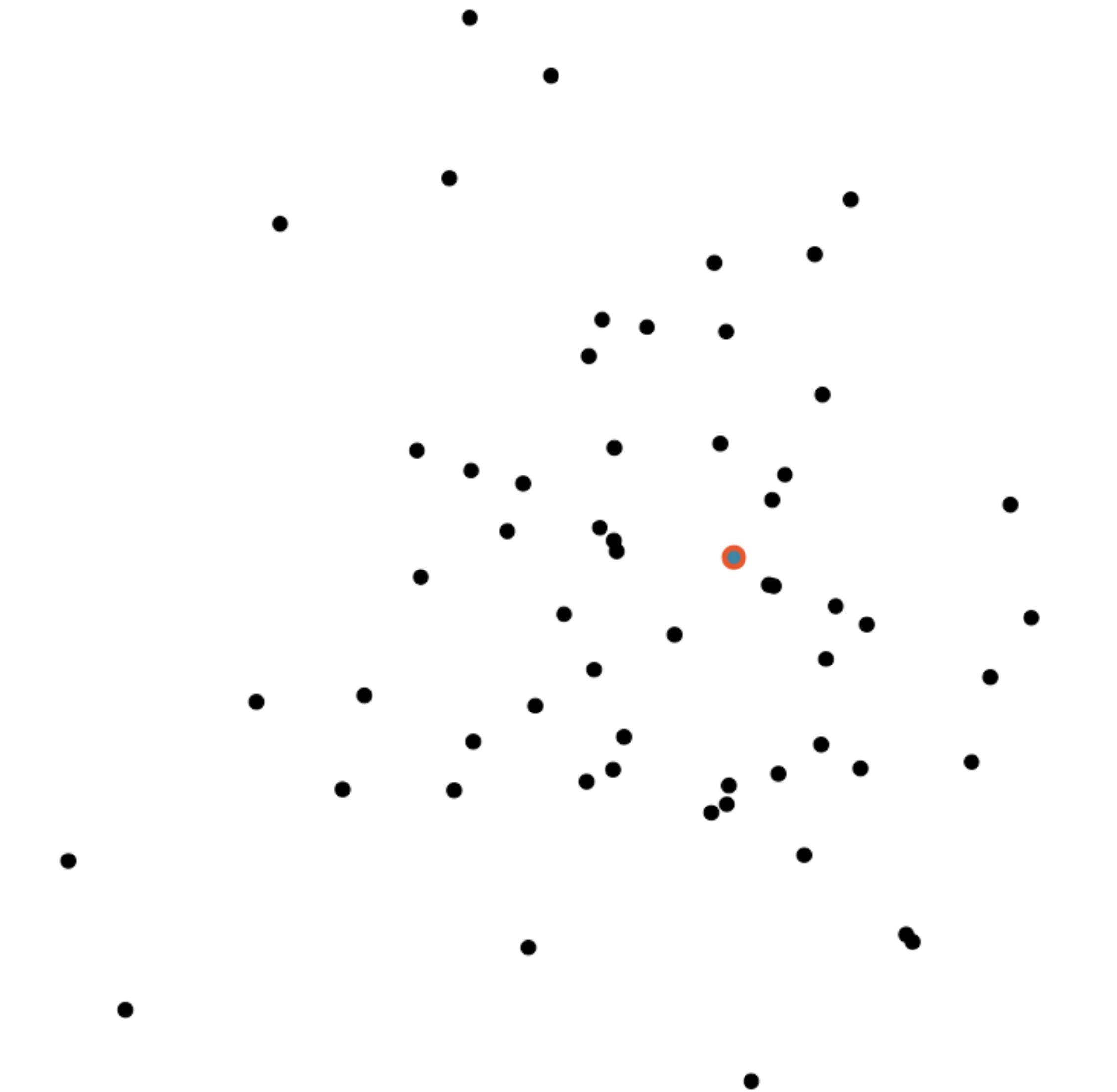
# interaction of color, two different colors look alike



interaction of color, vibrating boundaries, occurs with contrasting hues of similar luminance



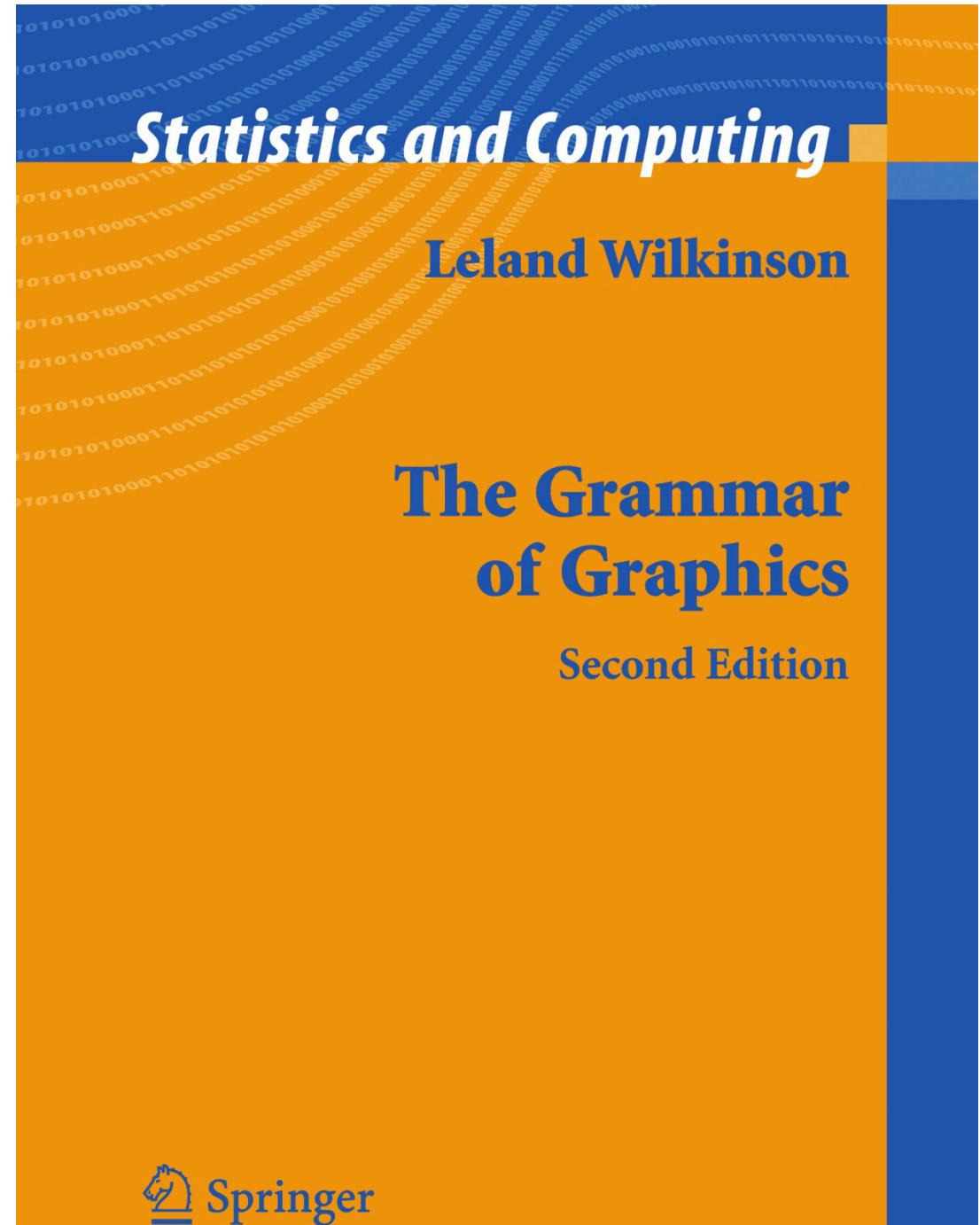
# interaction of color, experimentation with vibrating boundaries to focus attention



**charts are *mere typologies* of graphics — don't limit yourself**

think data encodings, *not* charts ...

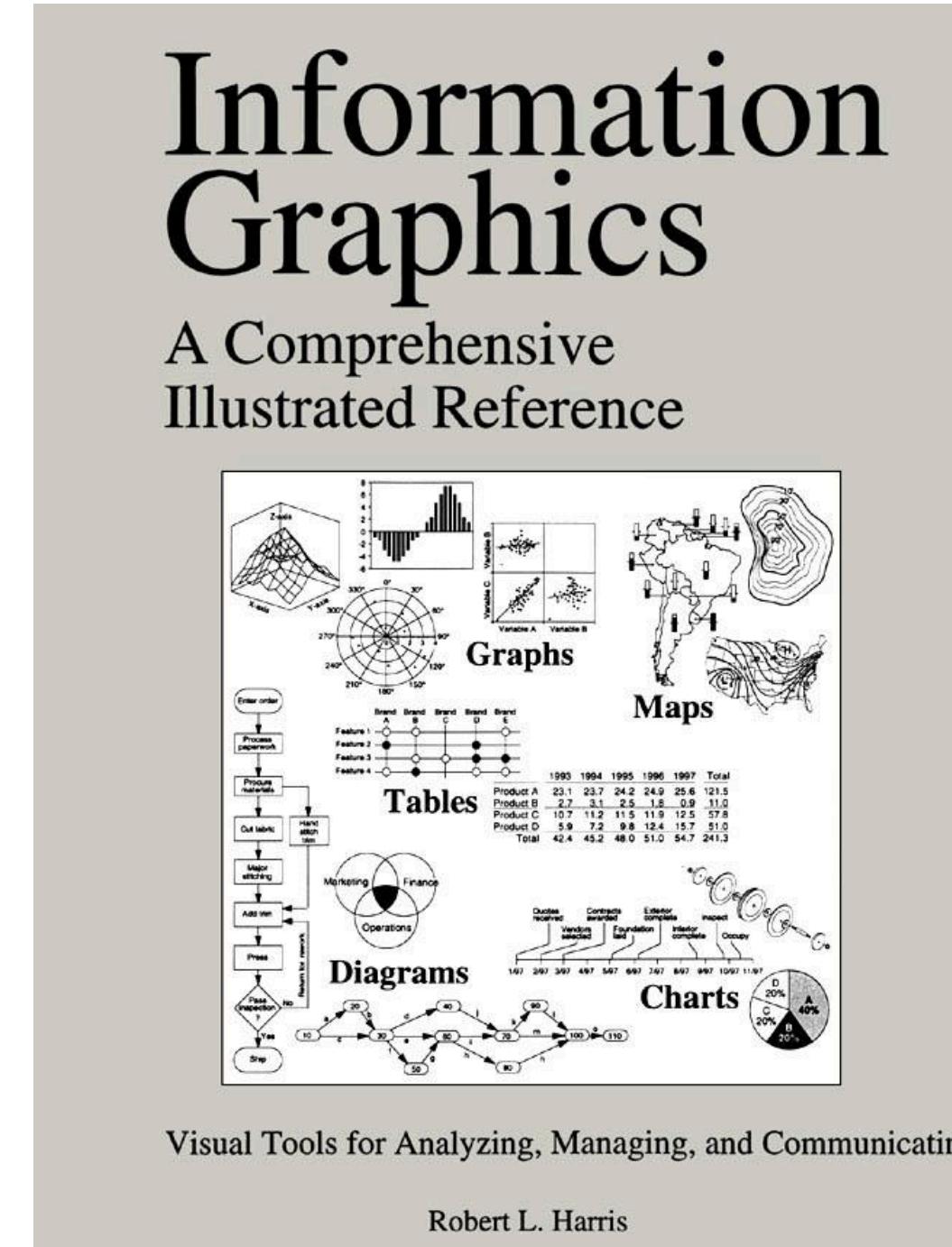
but chart typologies *can* help us learn and discuss encodings



“We often call graphics charts. There are pie charts, bar charts, line charts, and so on. [We should] shun chart typologies. Charts are usually instances of much more general objects.

Once we understand that a pie is a divided bar in polar coordinates, we can construct other polar graphics that are less well known. We will also come to realize why a histogram is not a bar chart and why many other graphics that look similar nevertheless have different grammars.... Elegant design requires us to think about a theory of graphics, not charts.”

— Leland Wilkinson, *The Grammar of Graphics*, Second.



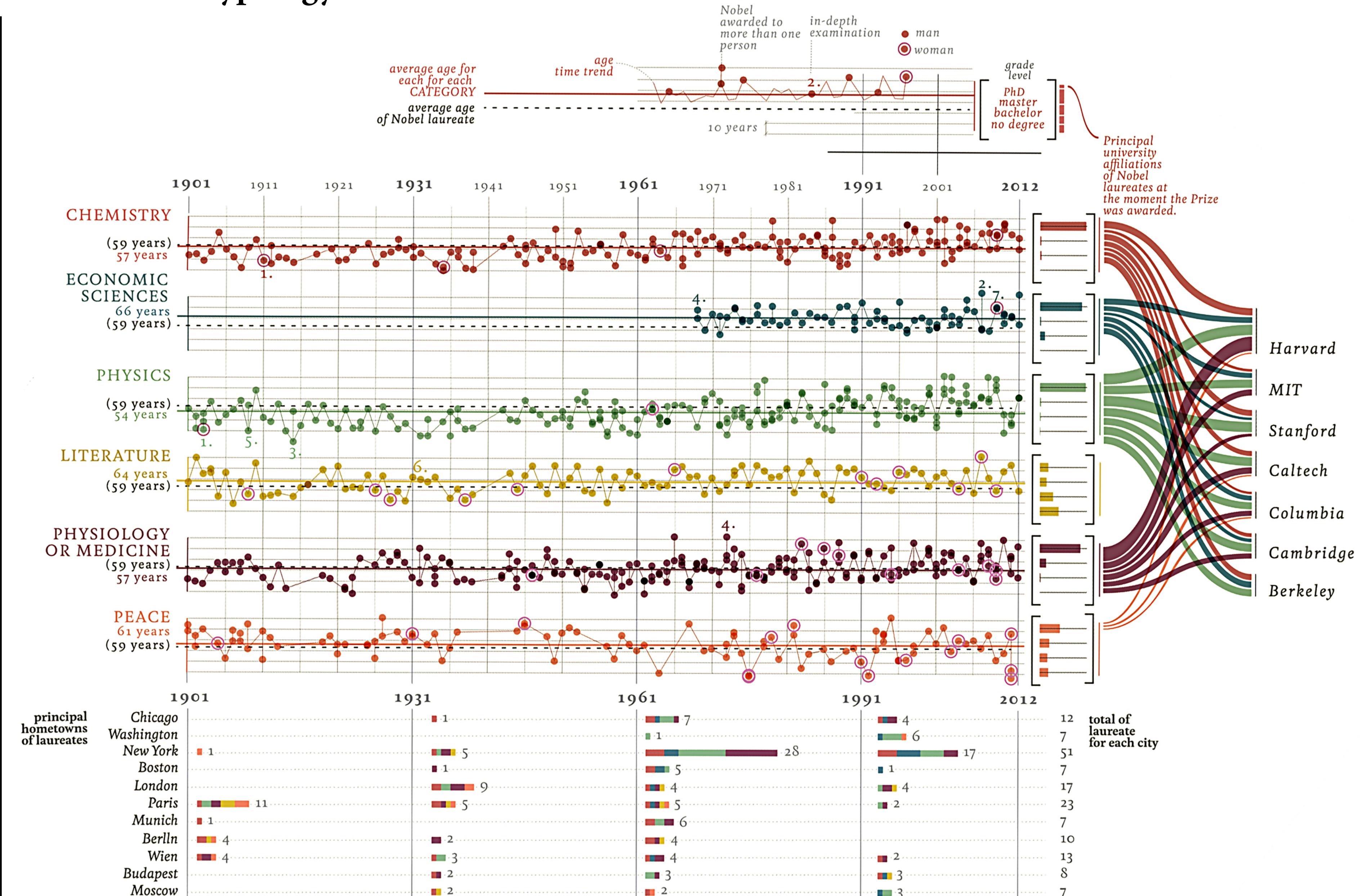
a graphics study — deconstructing Lupi's  
*Nobels, no degrees*, identifying typologies

# a graphics study through deconstruction and typology

If at first, this seems complex, Lupi's graphic is just organized groups of layered data encodings. These even follow typologies commonly used in business communications. We can make something complex like this by creating component parts and carefully arranging them.

Don't be intimidated! — Just methodically experiment with encodings for each data type, then organize them.

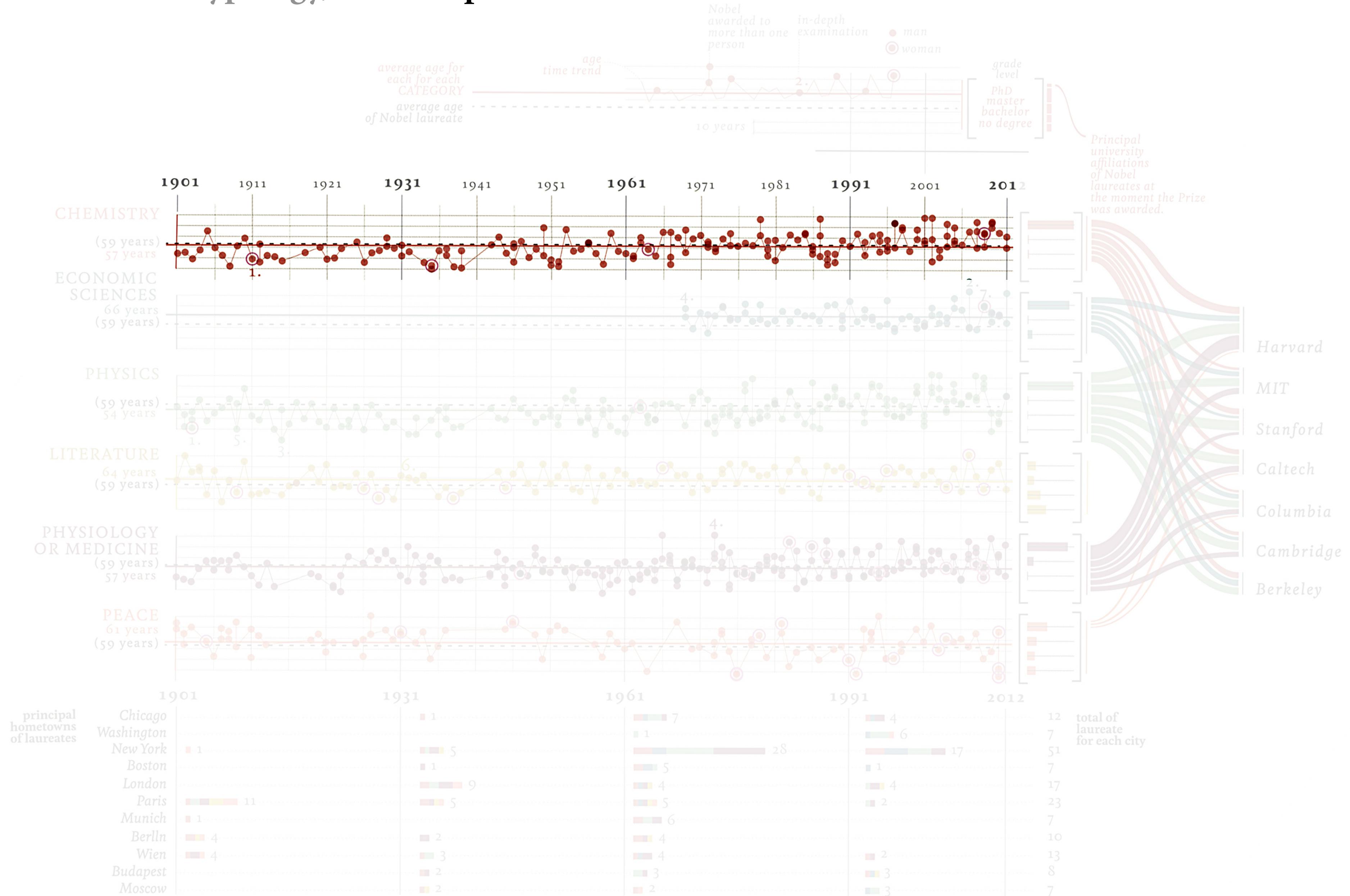
Of note: in Lupi's organization, she aligns graphics by common axis scales. We'll discuss this idea more later.



Lupi, Giorgia. 2016. *Visual Data - La Lettura*. <http://giorgialupi.com/lalettura>

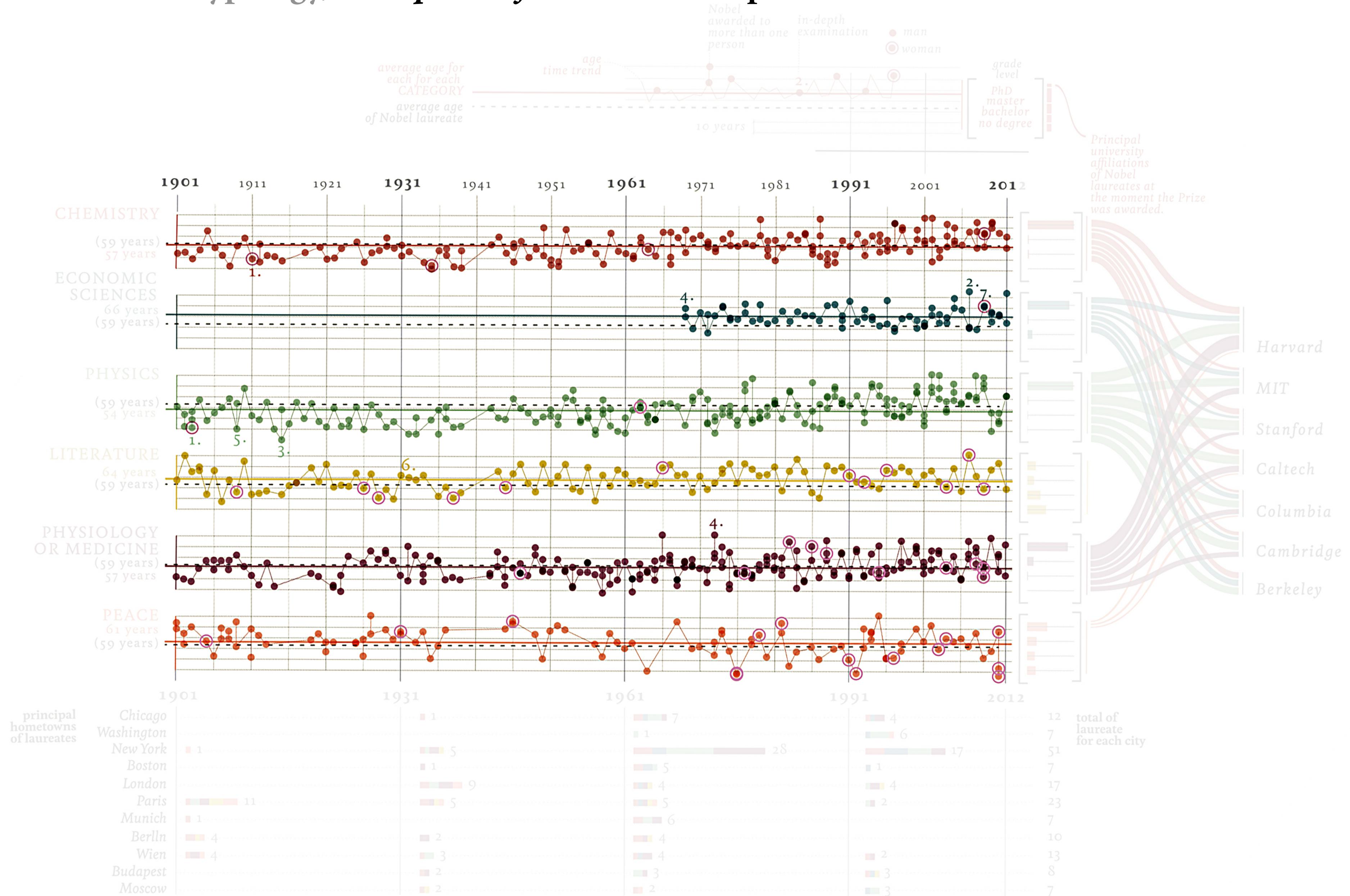
# a graphics study through deconstruction and typology, a scatter plot and line charts

```
ggplot(filter(df, Category == "Chemistry")) +  
  
  scale_color_manual(  
    values = category_colors,  
    breaks = category_names) +  
  
  scale_alpha_manual(  
    values = c(1, 0),  
    breaks = c("Female", "Male")) +  
  
  scale_x_continuous(  
    breaks = seq(1901, 2016, by = 30),  
    minor_breaks = seq(1901, 2016, by = 10),  
    position = "top") +  
  
  geom_hline(  
    mapping = aes(  
      yintercept = mean(Age, na.rm = TRUE)),  
    lwd = 0.2,  
    color = "black",  
    linetype = "dashed") +  
  
  geom_hline(  
    mapping = aes(  
      yintercept = cat_avg_age,  
      color = Category)) +  
  
  geom_line(  
    mapping = aes(  
      x = Year,  
      y = Age,  
      color = Category),  
    lwd = 0.2) +  
  
  geom_point(  
    mapping = aes(  
      x = Year,  
      y = Age,  
      color = Category),  
    size = 1.5,  
    alpha = 0.5) +  
  
  geom_point(  
    mapping = aes(  
      x = Year,  
      y = Age,  
      alpha = Sex),  
    color = "pink",  
    shape = 21,  
    size = 4)
```



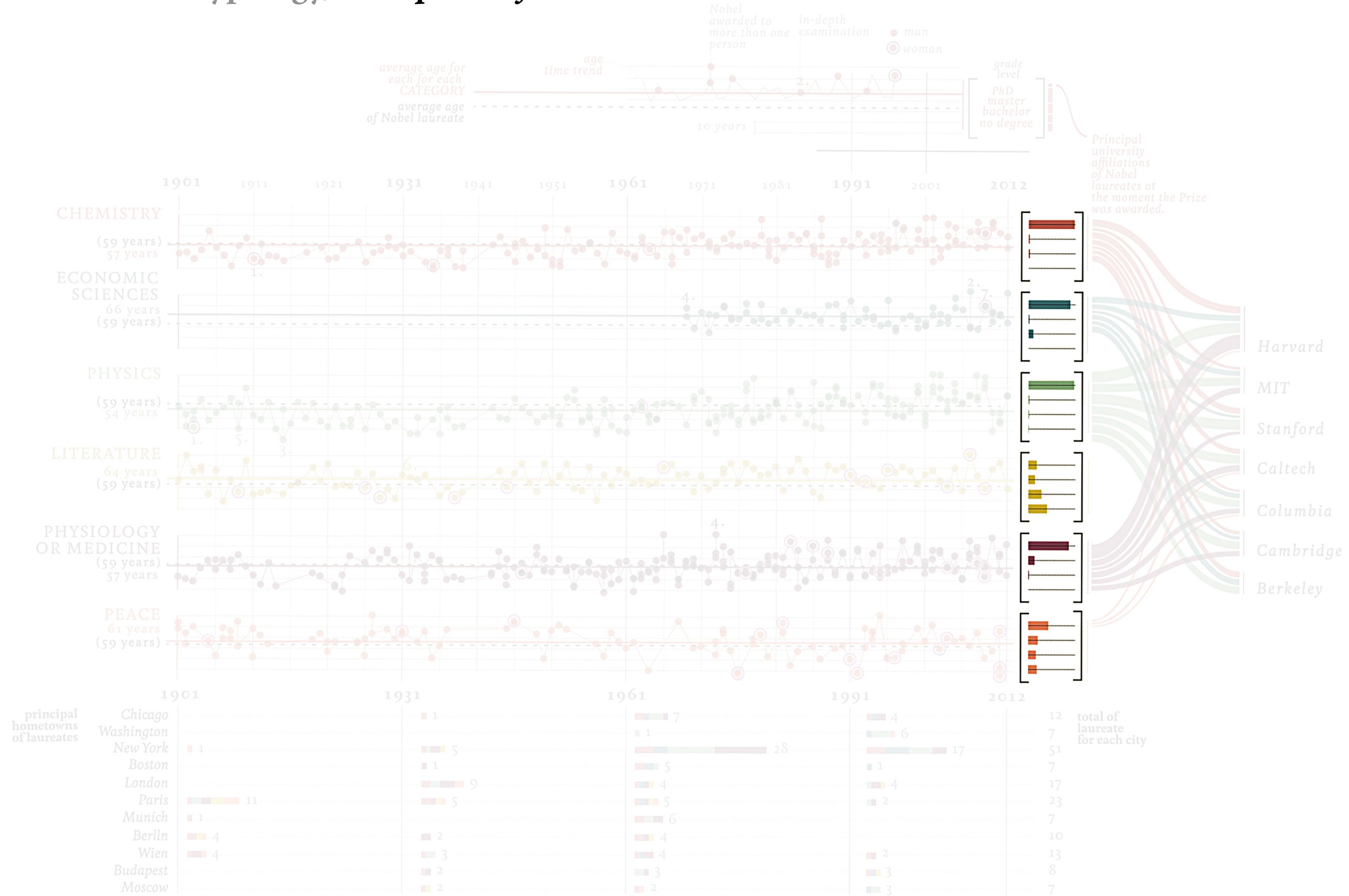
# a graphics study through deconstruction and typology, *multiples or facets* of scatter plots and line charts

```
ggplot(df) +
  facet_wrap(~ Category ~ ., nrow = 6, strip.position = "left") +
  scale_color_manual(values = category_colors, breaks = category_names) +
  scale_alpha_manual(values = c(1, 0), breaks = c("Female", "Male")) +
  scale_x_continuous(breaks = seq(1901, 2016, by = 30),
                     minor_breaks = seq(1901, 2016, by = 10),
                     position = "top") +
  geom_hline(mapping = aes(yintercept = mean(Age, na.rm = TRUE)),
             lwd = 0.2, color = "black", linetype = "dashed") +
  geom_hline(mapping = aes(yintercept = cat_avg_age,
                           color = Category)) +
  geom_line(mapping = aes(x = Year, y = Age, color = Category),
            lwd = 0.2) +
  geom_point(mapping = aes(x = Year, y = Age, color = Category),
             size = 1.5, alpha = 0.5) +
  geom_point(mapping = aes(x = Year, y = Age, alpha = Sex),
             color = "pink", shape = 21, size = 4)
```



# a graphics study through deconstruction and typology, *multiples or facets* of bar charts

```
ggplot(df) +  
  
  facet_wrap(~ Category,  
             ncol = 1) +  
  
  scale_fill_manual(  
    values = category_colors,  
    breaks = category_names) +  
  
  geom_bar(  
    mapping = aes(  
      x = Percent,  
      y = Education,  
      fill = Category),  
    stat = "identity")
```



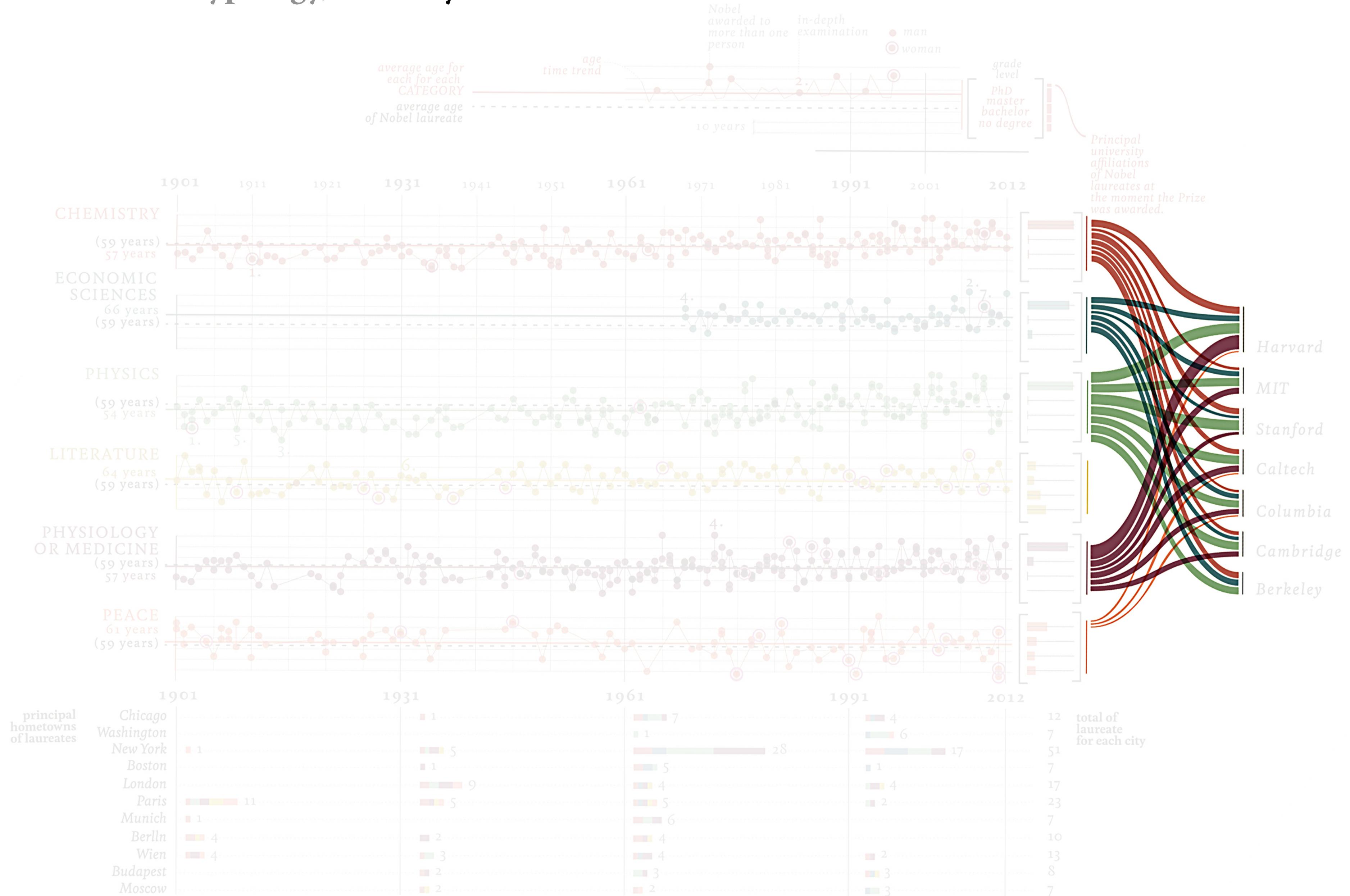
# a graphics study through deconstruction and typology, *multiples or facets* of stacked bar charts

```
ggplot(df) +  
  
  facet_wrap(  
    era~,  
    nrow = 1) +  
  
  scale_fill_manual(  
    values = category_colors,  
    breaks = category_names) +  
  
  geom_bar(  
    mapping = aes(  
      x = n,  
      y = Birth.City,  
      fill = Category),  
    stat = 'identity',  
    width = 0.2) +  
  
  geom_text(  
    mapping = aes(  
      x = n_era_city + 2,  
      y = Birth.City,  
      label = n_era_city),  
    stat = 'identity',  
    width = 0.2)
```



# a graphics study through deconstruction and typology, a sankey chart

```
ggplot(
  data = data,
  mapping = aes(
    x = x,
    id = id,
    split = y,
    value = n)) +
  scale_fill_manual(
    values = category_colors,
    breaks = category_names) +
  geom_parallel_sets(
    mapping = aes(
      fill = Category),
    alpha = 0.6,
    axis.width = 0.05,
    sep = 0.1) +
  geom_parallel_sets_axes(
    axis.width = 0.01,
    fill = "gray80",
    sep = 0.1)
```



**resources**

# References

**Spencer**, Scott. “Color,” “Layers and separation,” “Maximize information in visual displays,” “Layering and opacity,” Sec. 2.1.3.1-2.2.3 In *Data in Wonderland*. 2021. [https://ssp3nc3r.github.io/data\\_in\\_wonderland](https://ssp3nc3r.github.io/data_in_wonderland).

**Albers**, Josef. *Interaction of Color*. Yale University Press, 2006, and interactive app on iPad: <http://interactionofcolor.com>

**Boronine**, Alexei. “Color Spaces for Human Beings.” HSLuv.org (blog), March 26, 2012. <https://www.hsluv.org> and <https://www.boronine.com/2012/03/26/Color-Spaces-for-Human-Beings/>

**Harris**, Robert L. *Information Graphics: A Comprehensive Illustrated Reference*. New York: Oxford University Press, 1999.

**Koponen**, Juuso, and Jonatan Hildén. *Data Visualization Handbook*. First. Finland: Aalto Art Books, 2019.

**Spencer**, Scott. “Approximating the Components of Lupi’s Nobels, No Degrees.” Blog, March 15, 2019. <https://ssp3nc3r.github.io/post/approximating-the-components-of-lupi-s-nobel-no-degrees/>.

———. “Demonstration of Layers in Graphics.” Publications, March 6, 2020. <https://ssp3nc3r.github.io/publications/Spencer-2020-Demonstration-of-layers-in-graphics.pdf>.

**Tufte**, Edward R. “Layers and Separation” and “Color and Information.” In *Envisioning Information*. Graphics Press, 1990.

———. “Data-Ink and Graphical Redesign” and “Data-Ink Maximization and Graphical Design.” In *The Visual Display of Quantitative Information*. Second. Graphics Press, 2001.

**Ware**, Colin. *Information Visualization: Perception for Design*. Fourth. Philadelphia: Elsevier, Inc, 2020.

**Wickham**, Hadley. “A Layered Grammar of Graphics.” *Journal of Computational and Graphical Statistics* 19, no. 1 (January 2010): 3–28.

**Wilke**, C. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. First edition. Sebastopol, CA: O'Reilly Media, 2019.

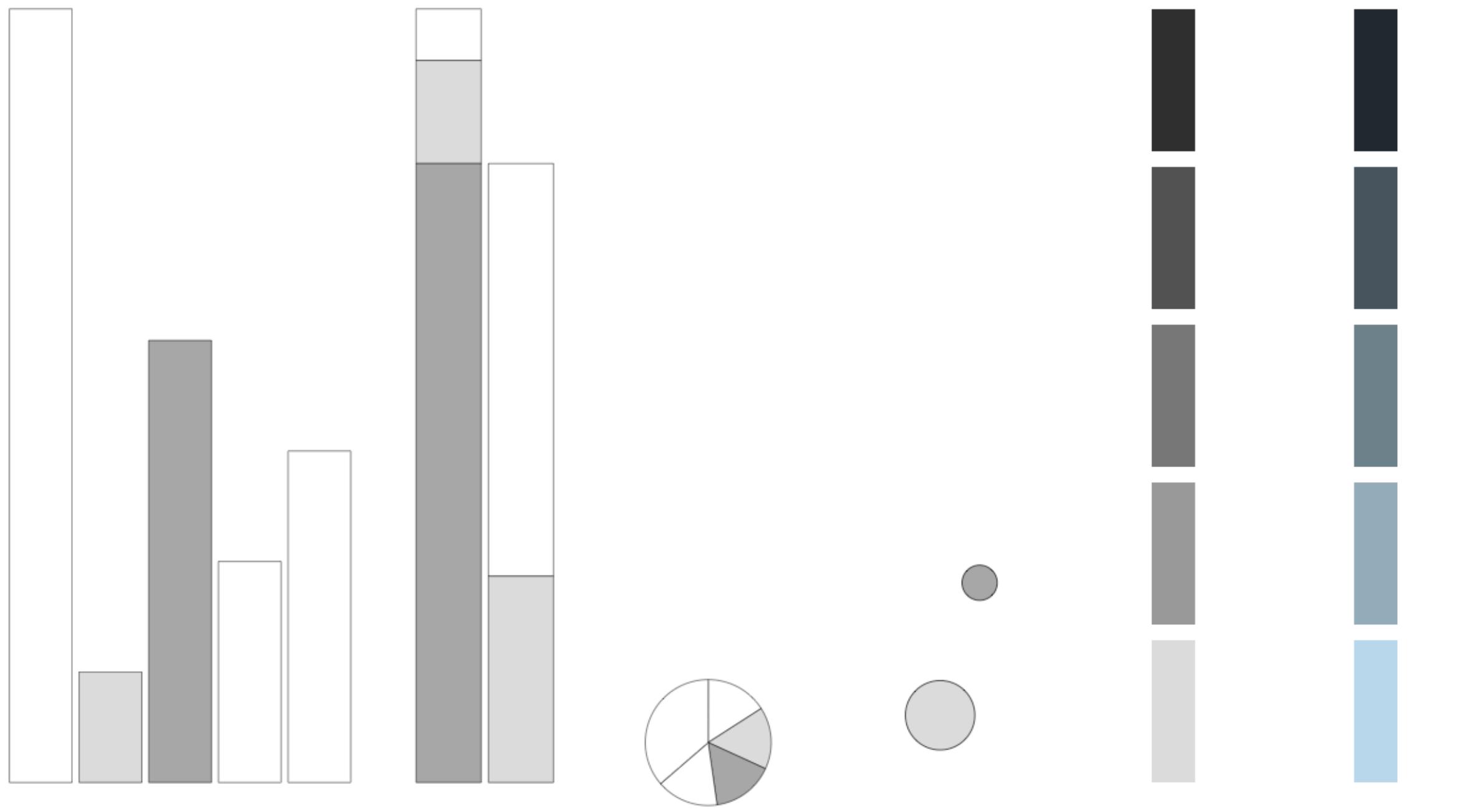
**Wilkinson**, Leland. *The Grammar of Graphics*. Second. Springer, 2005.

## **supplemental visuals**

**channel effectiveness for encoding data**

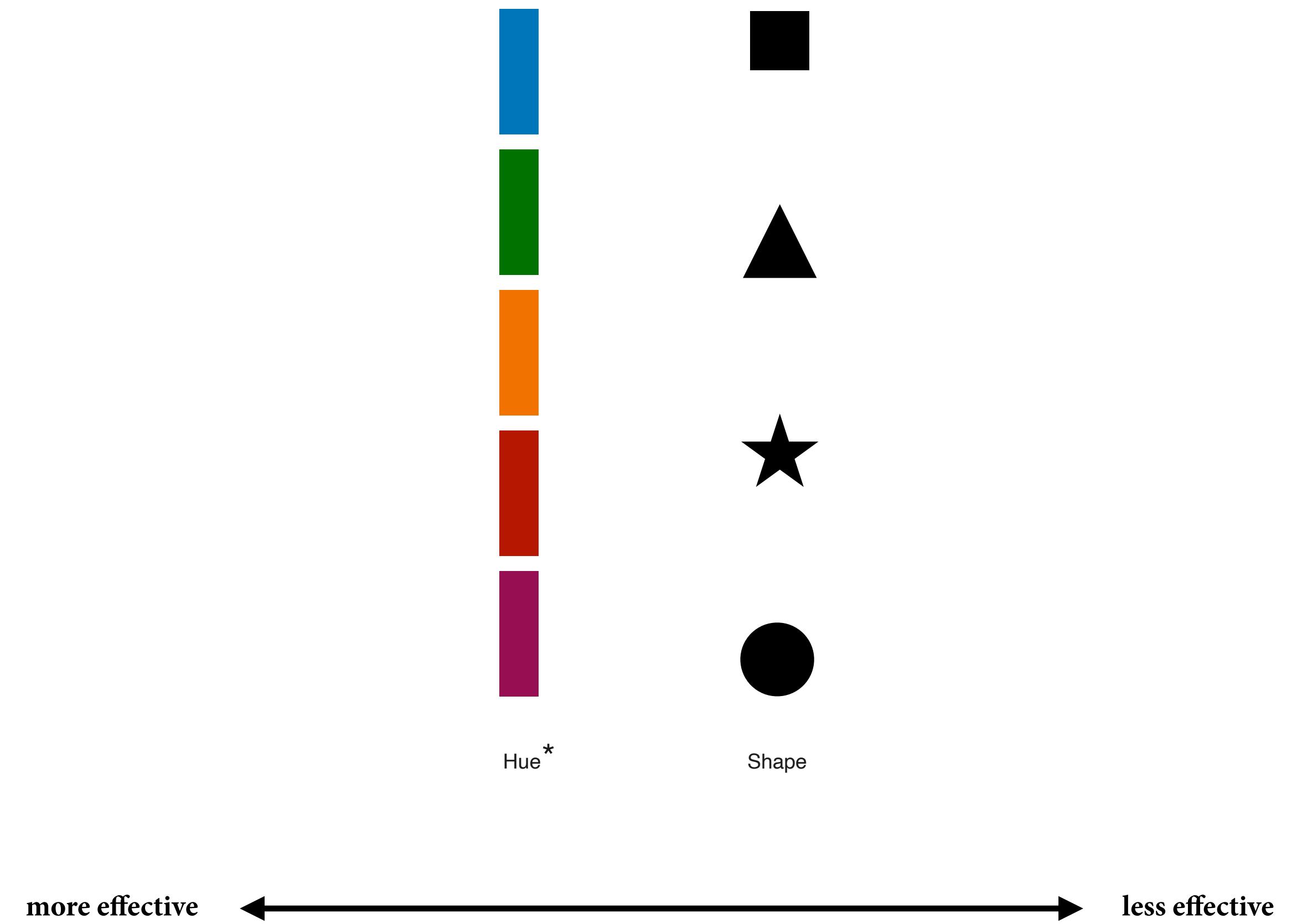
# general channel effectiveness, encoding data

## ratio, interval, and ordered



more effective ← → less effective

## categorical

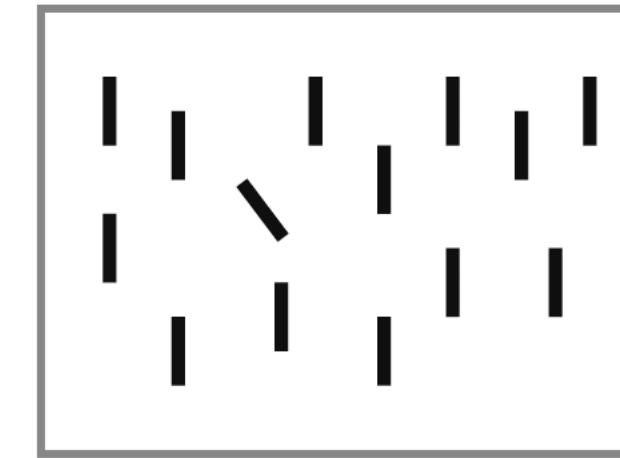


more effective ← → less effective

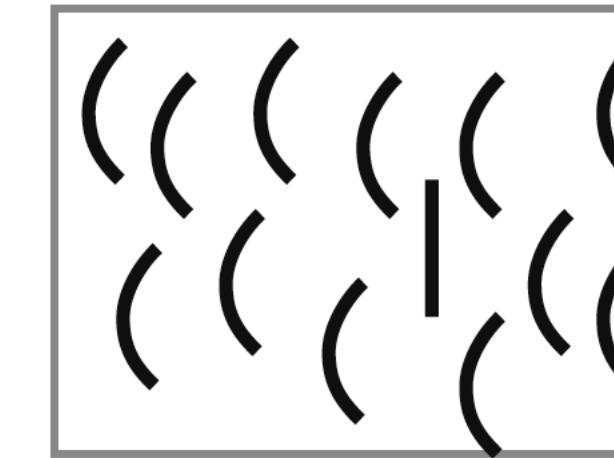
**perceptual psychology**

# perceptual psychology, *pre-attentive attributes*

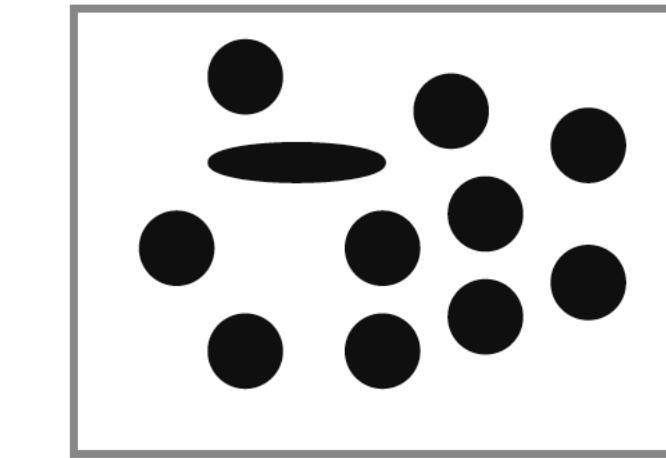
Orientation



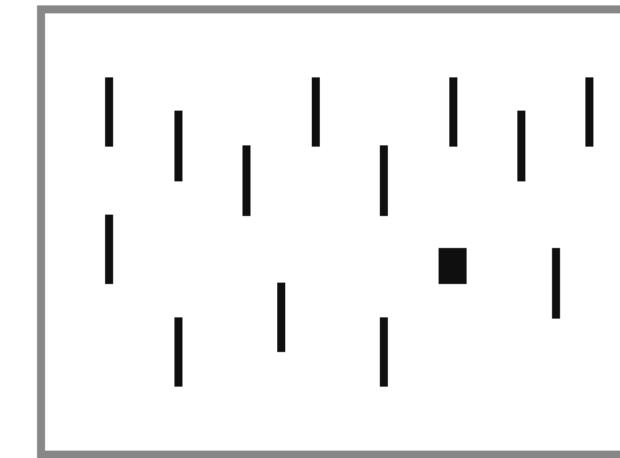
Curved straight



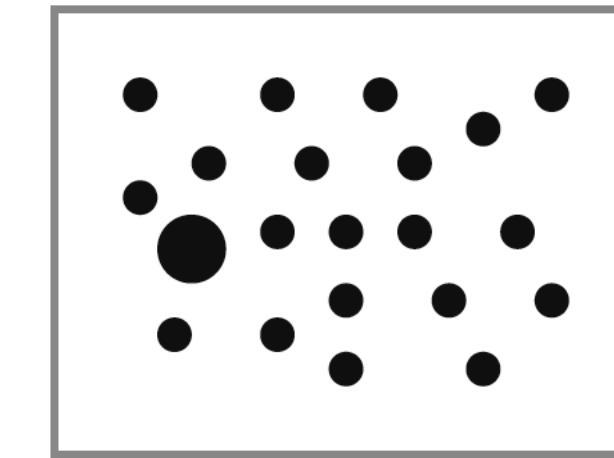
Shape



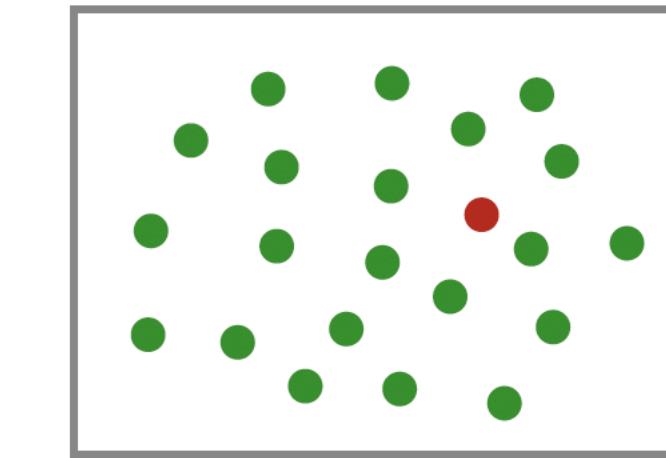
Shape



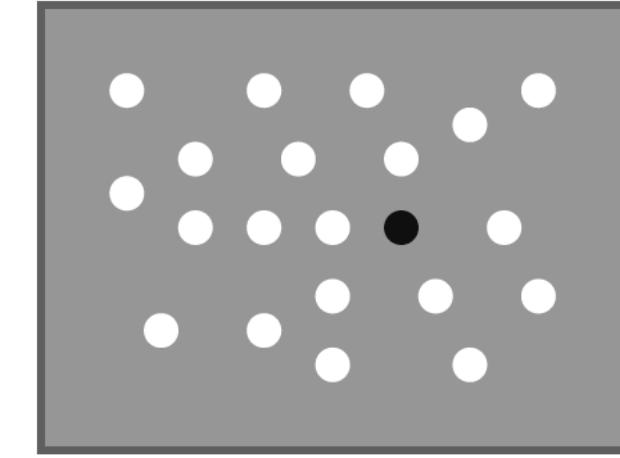
Size



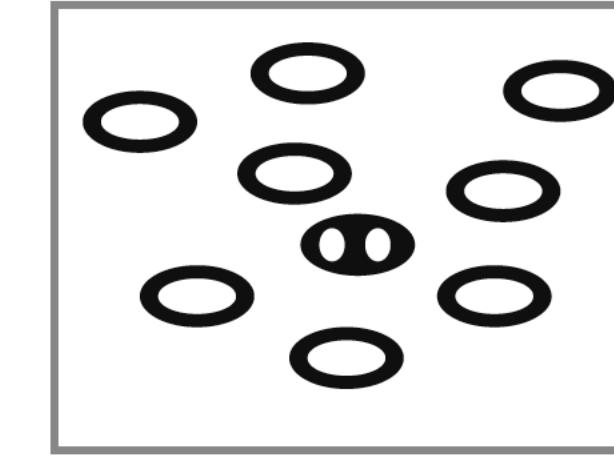
Color



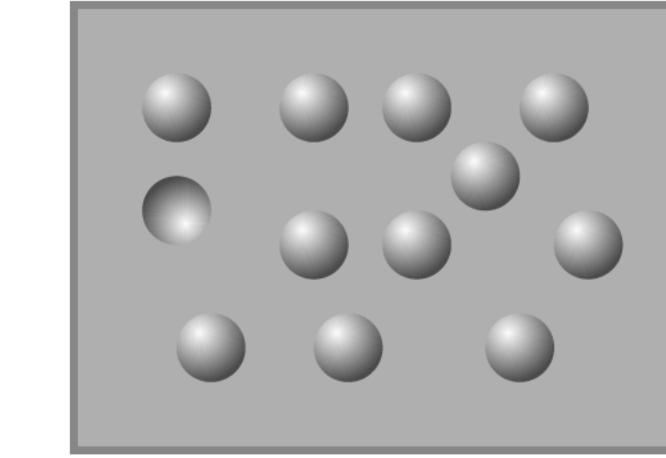
Light/dark



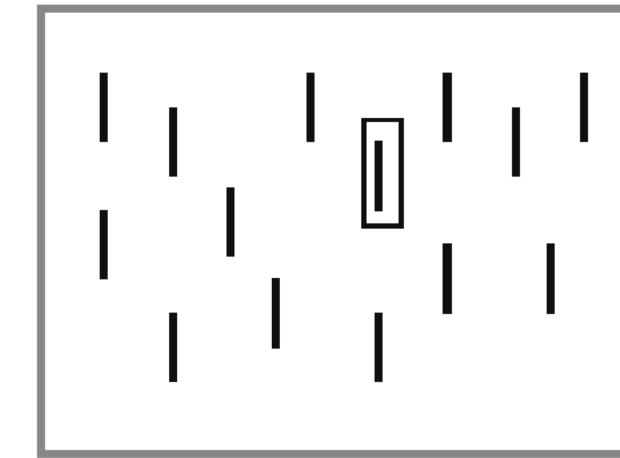
Topology (or count)



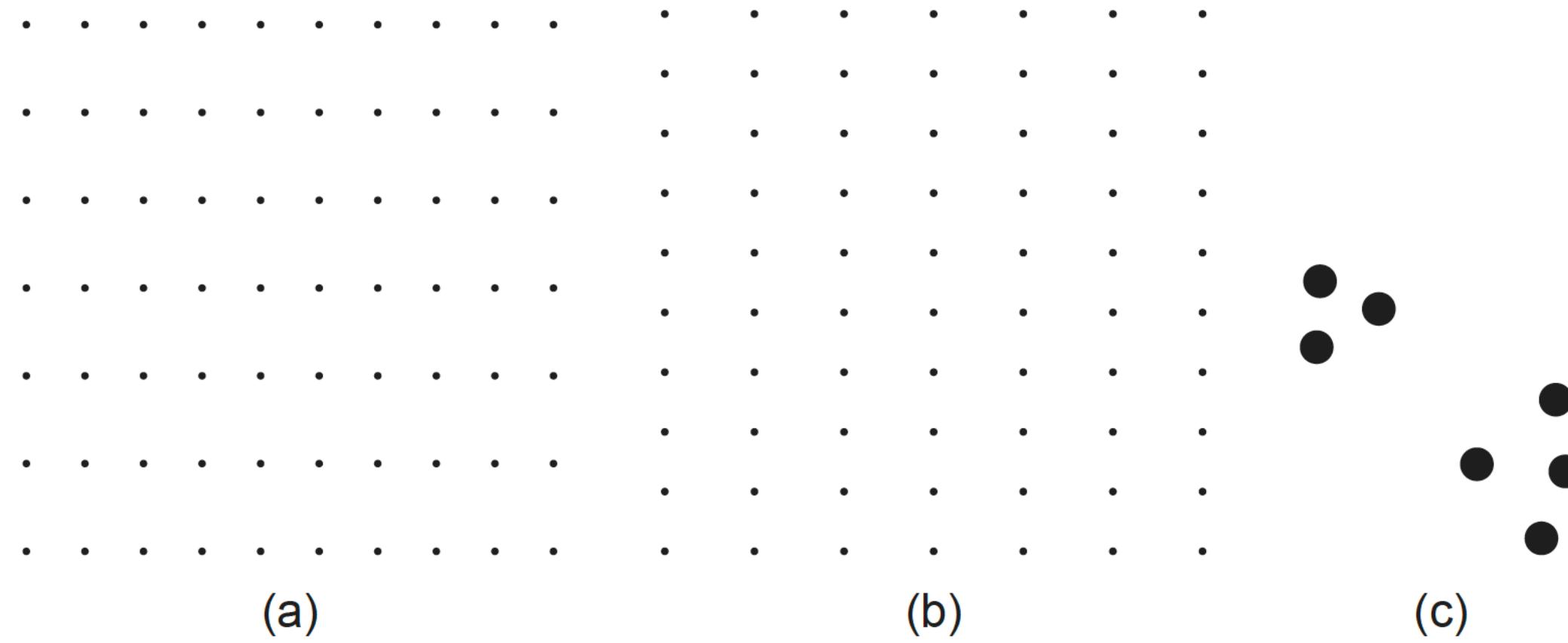
Convex/concave



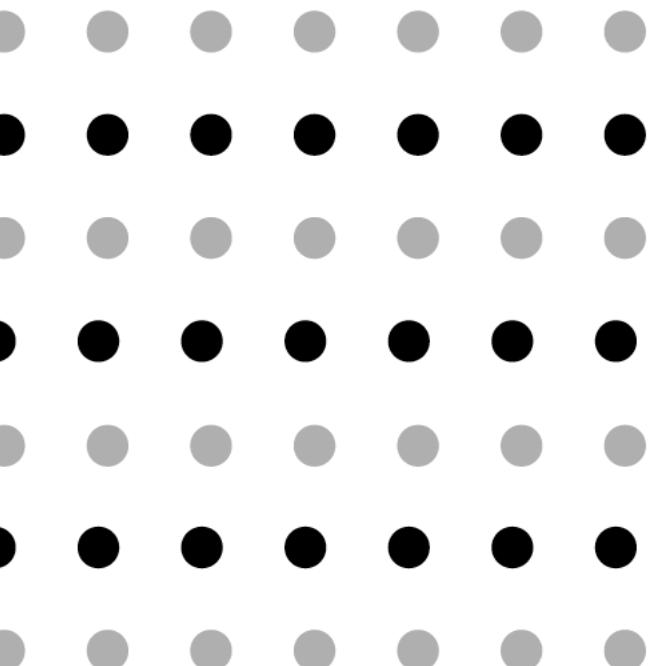
Addition



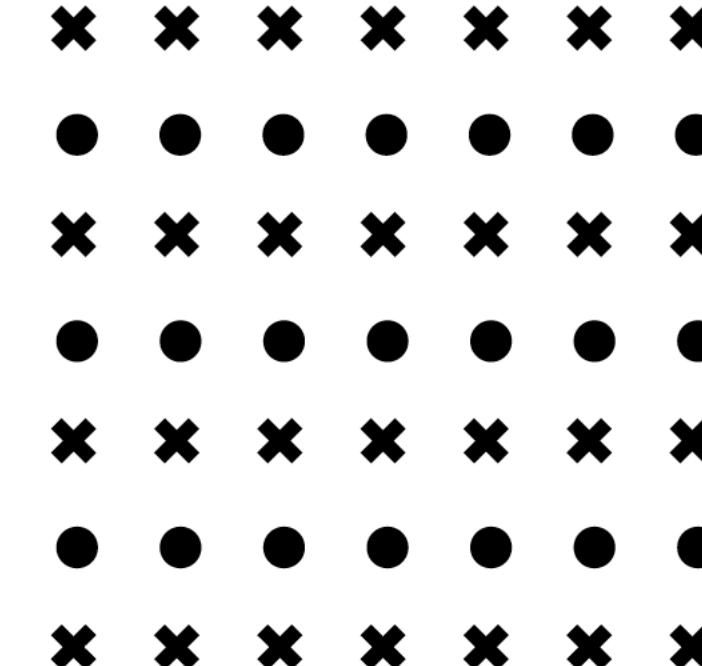
# perceptual psychology, Gestalt principles, *proximity*



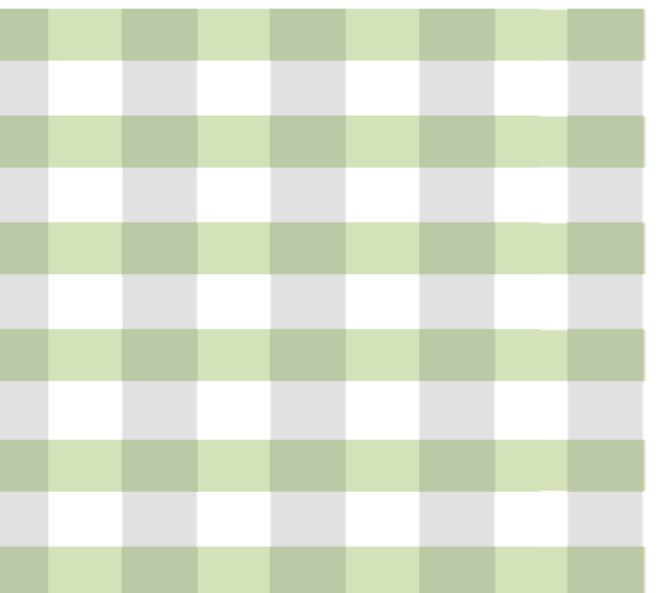
# perceptual psychology, Gestalt principles, *similarity*



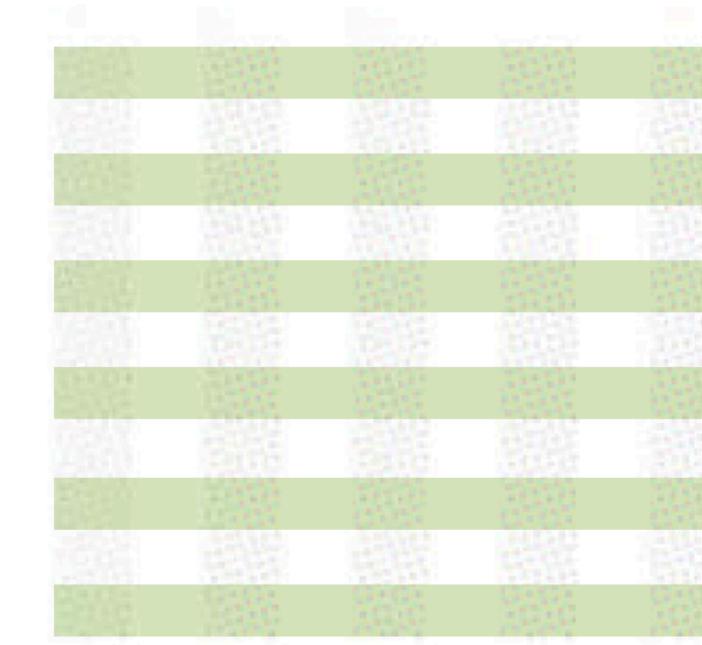
(a)



(b)

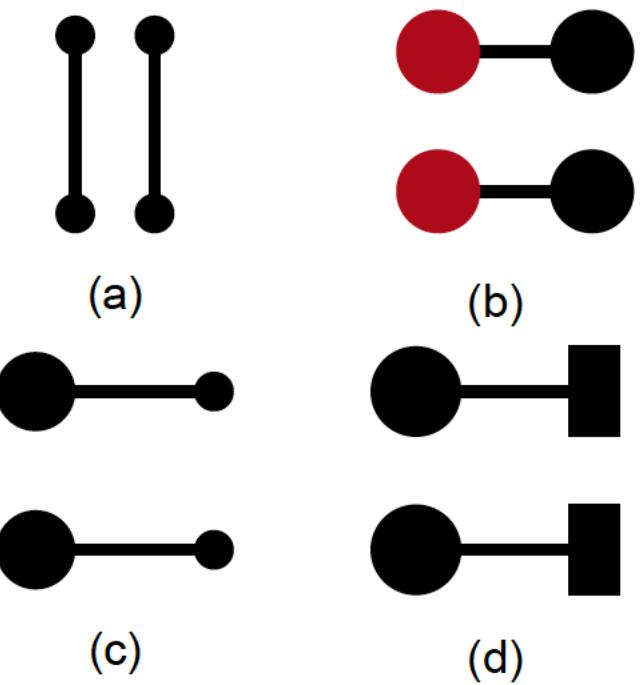


(c)

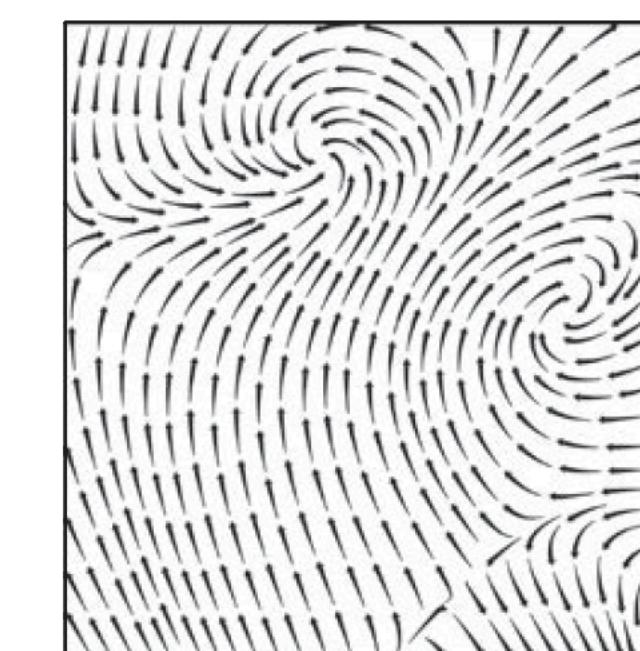
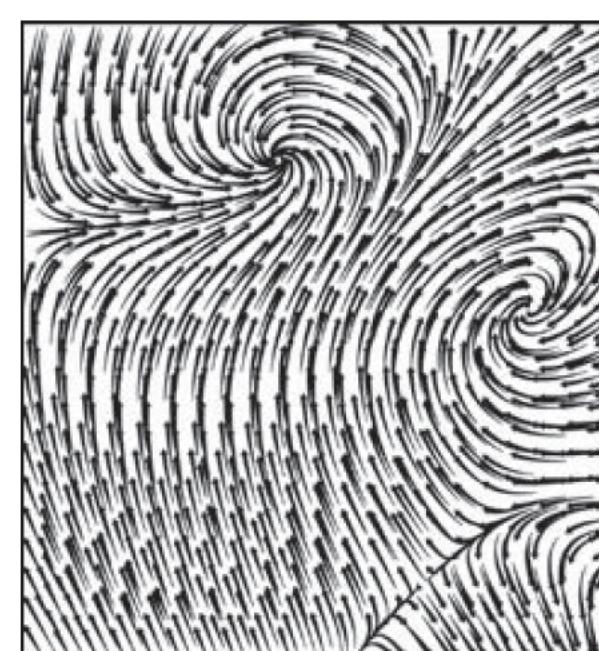
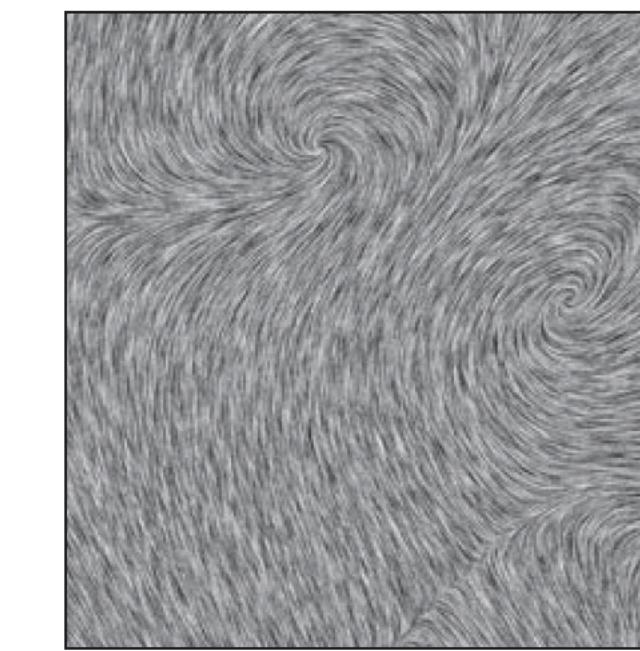
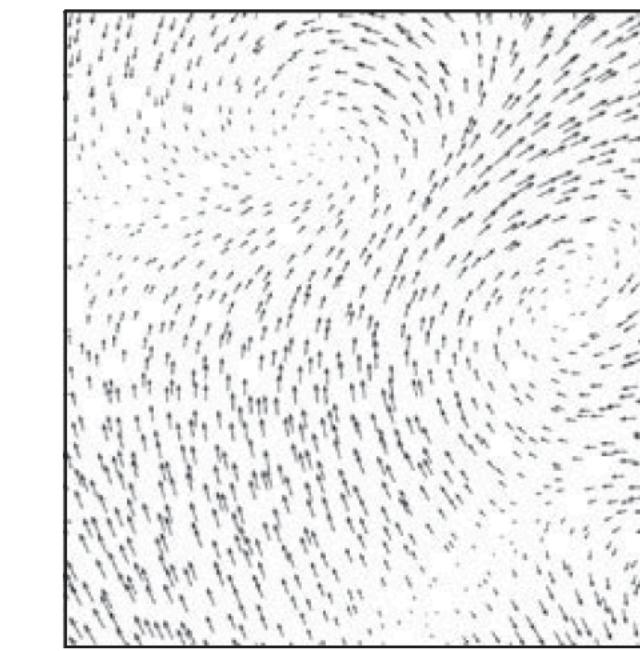
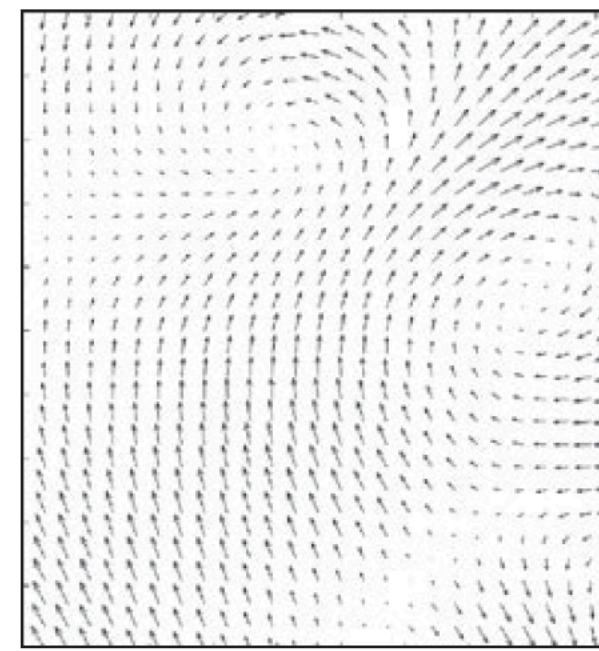


(d)

# perceptual psychology, Gestalt principles, *connectedness*

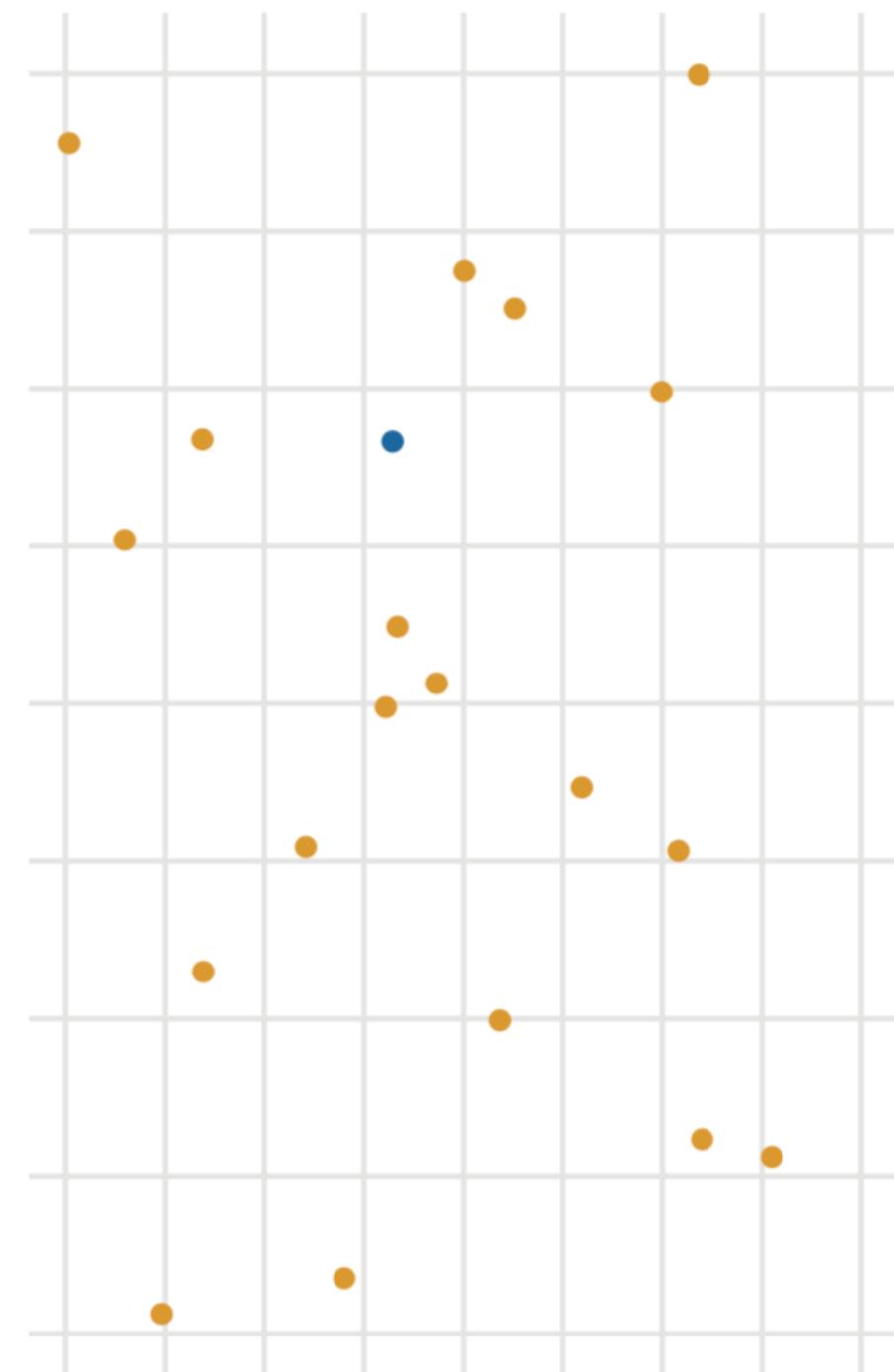


# perceptual psychology, Gestalt principles, *orientation*, *magnitude*, *direction*

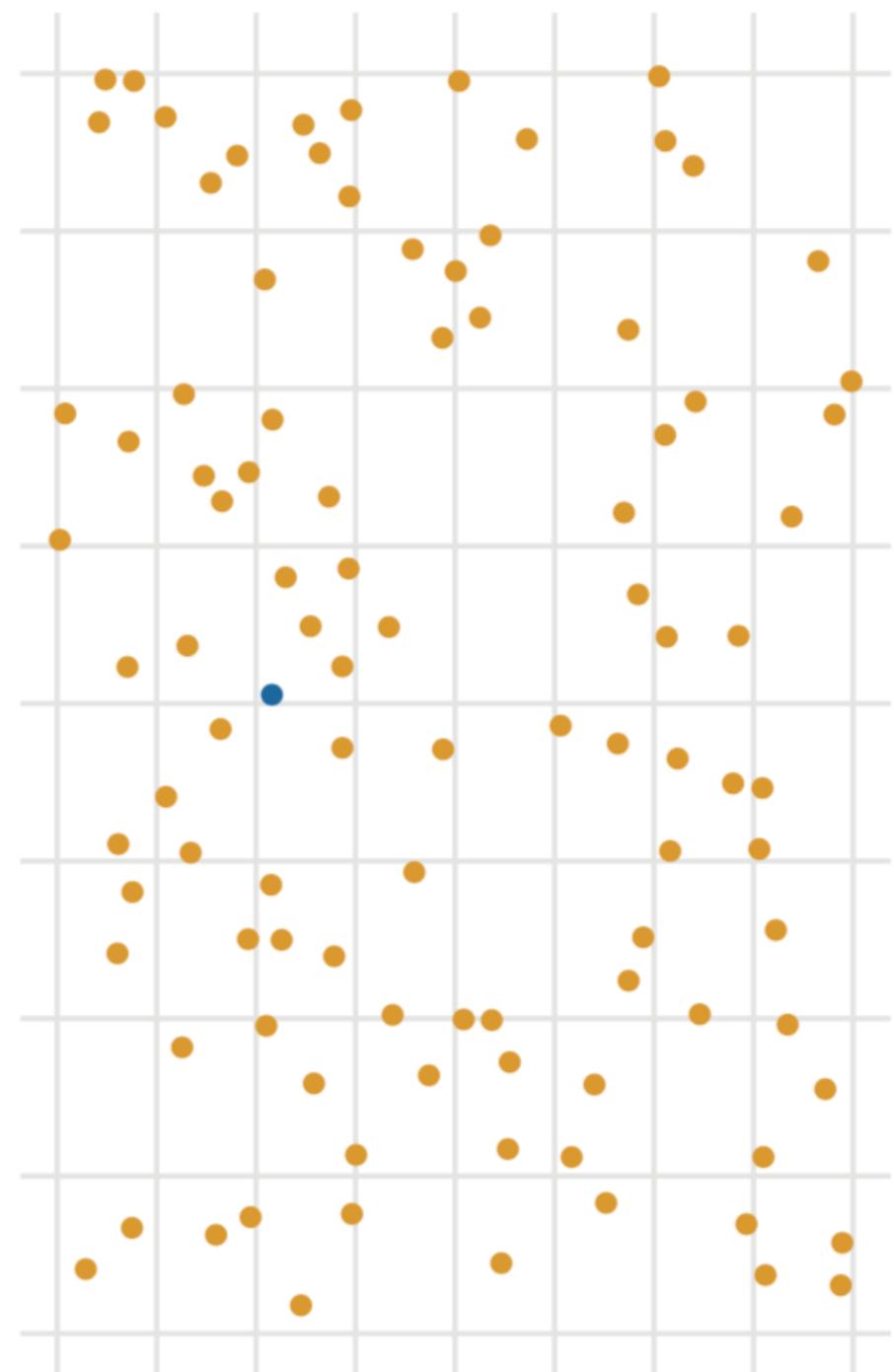


# perceptual psychology, example — *focusing visual attention*

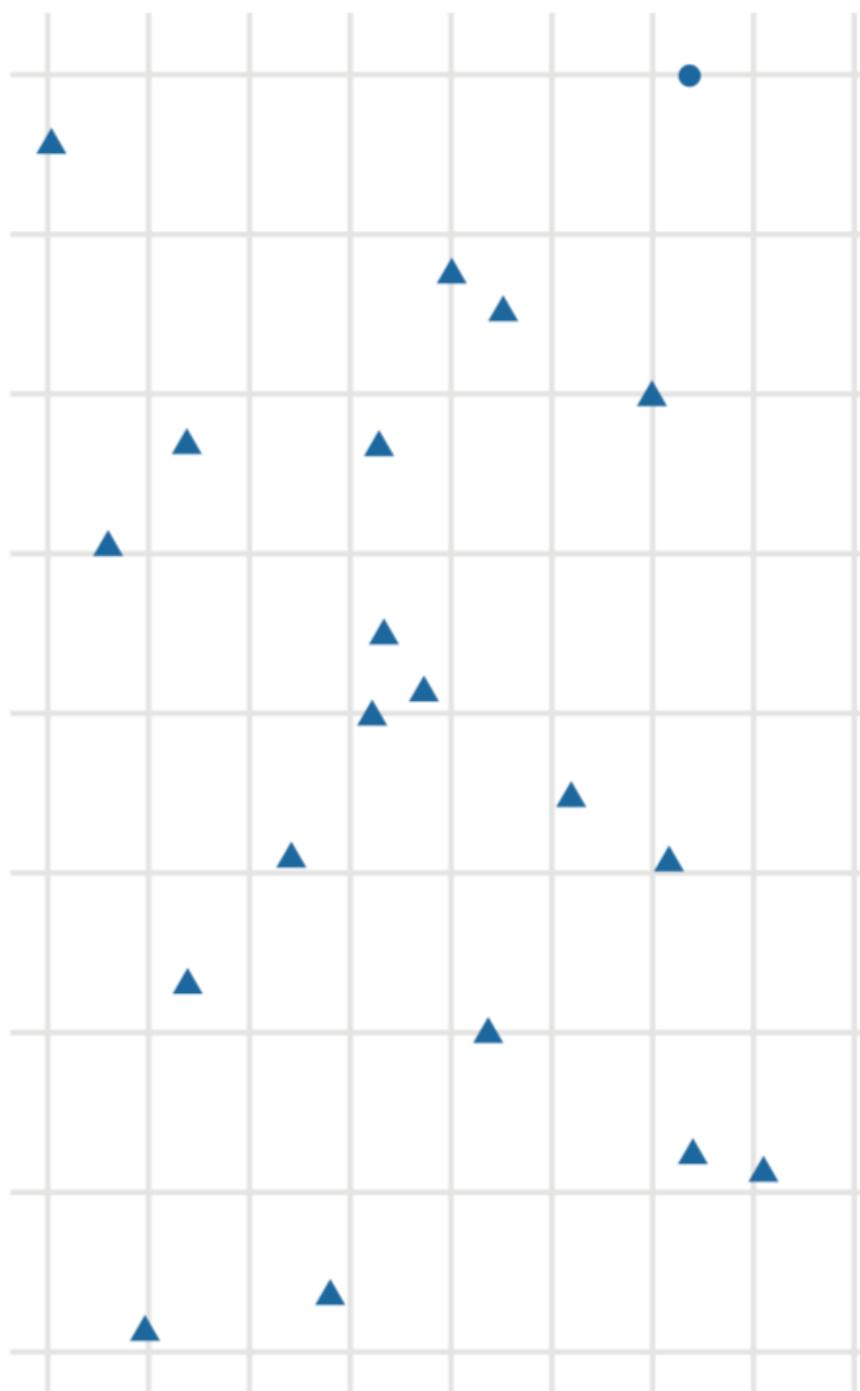
Color only,  $N = 20$



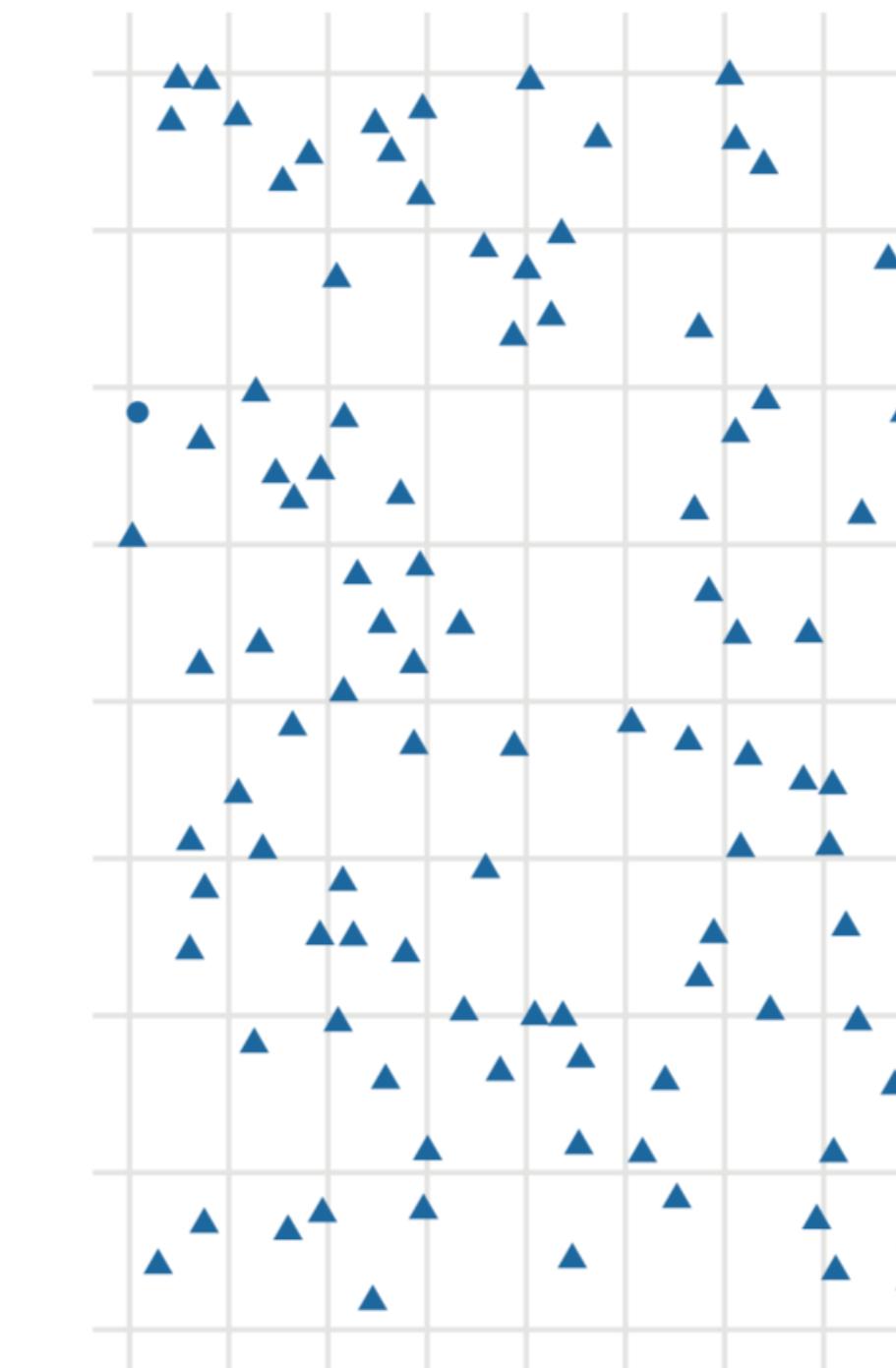
Color only,  $N = 100$



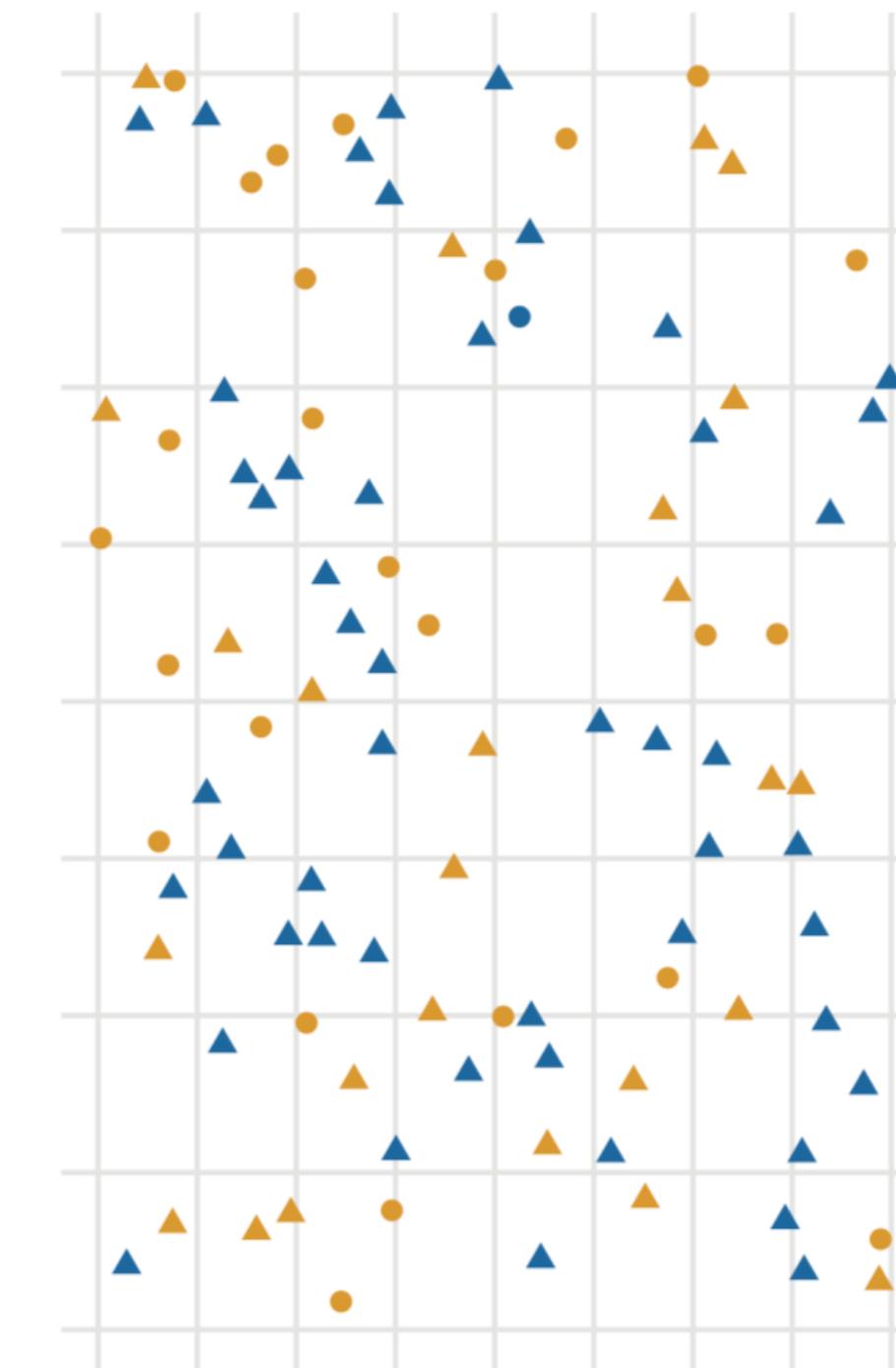
Shape only,  $N = 20$



Shape only,  $N = 100$

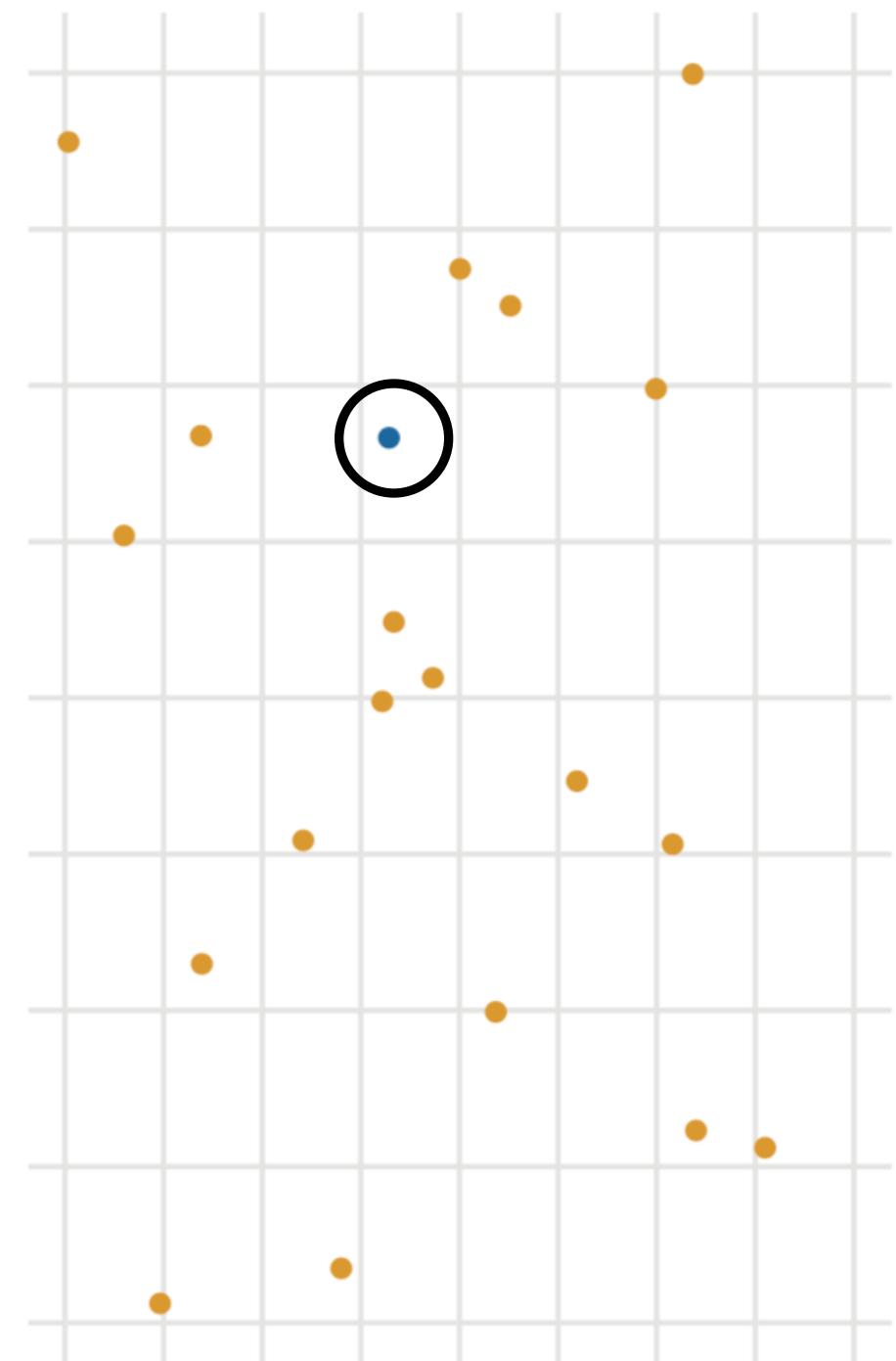


Color & shape,  $N = 100$

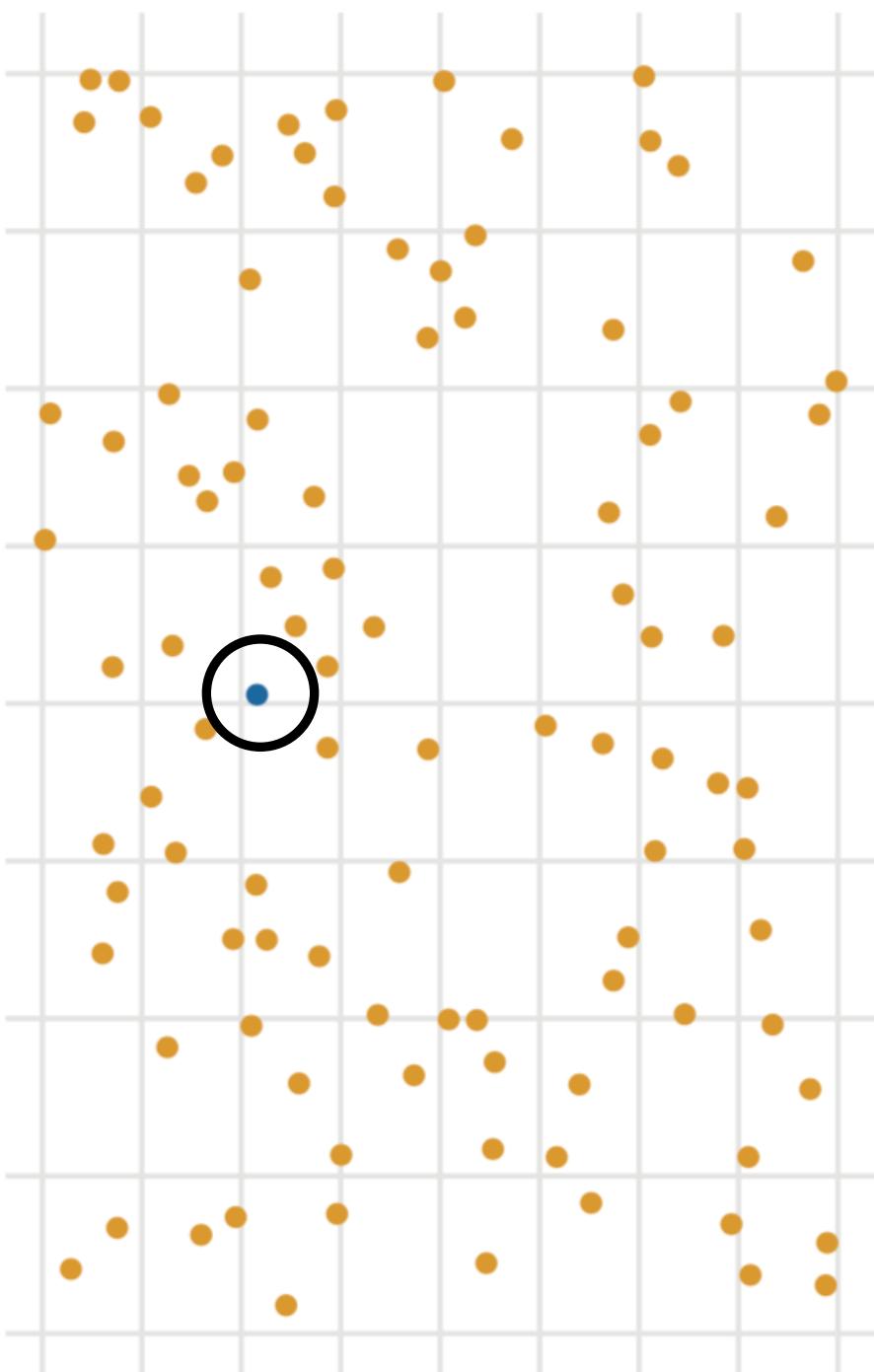


# perceptual psychology, example — *focusing visual attention*

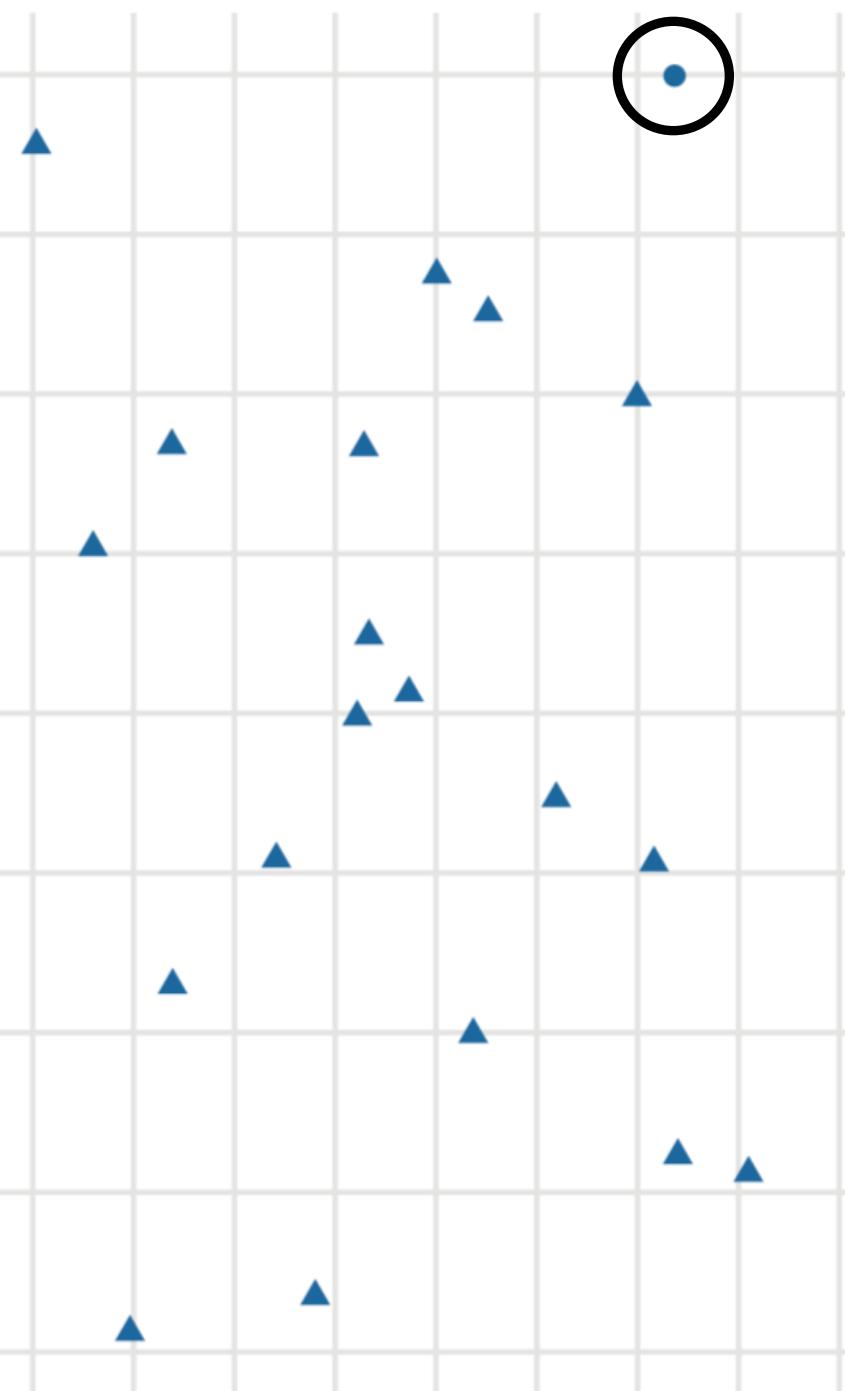
Color only,  $N = 20$



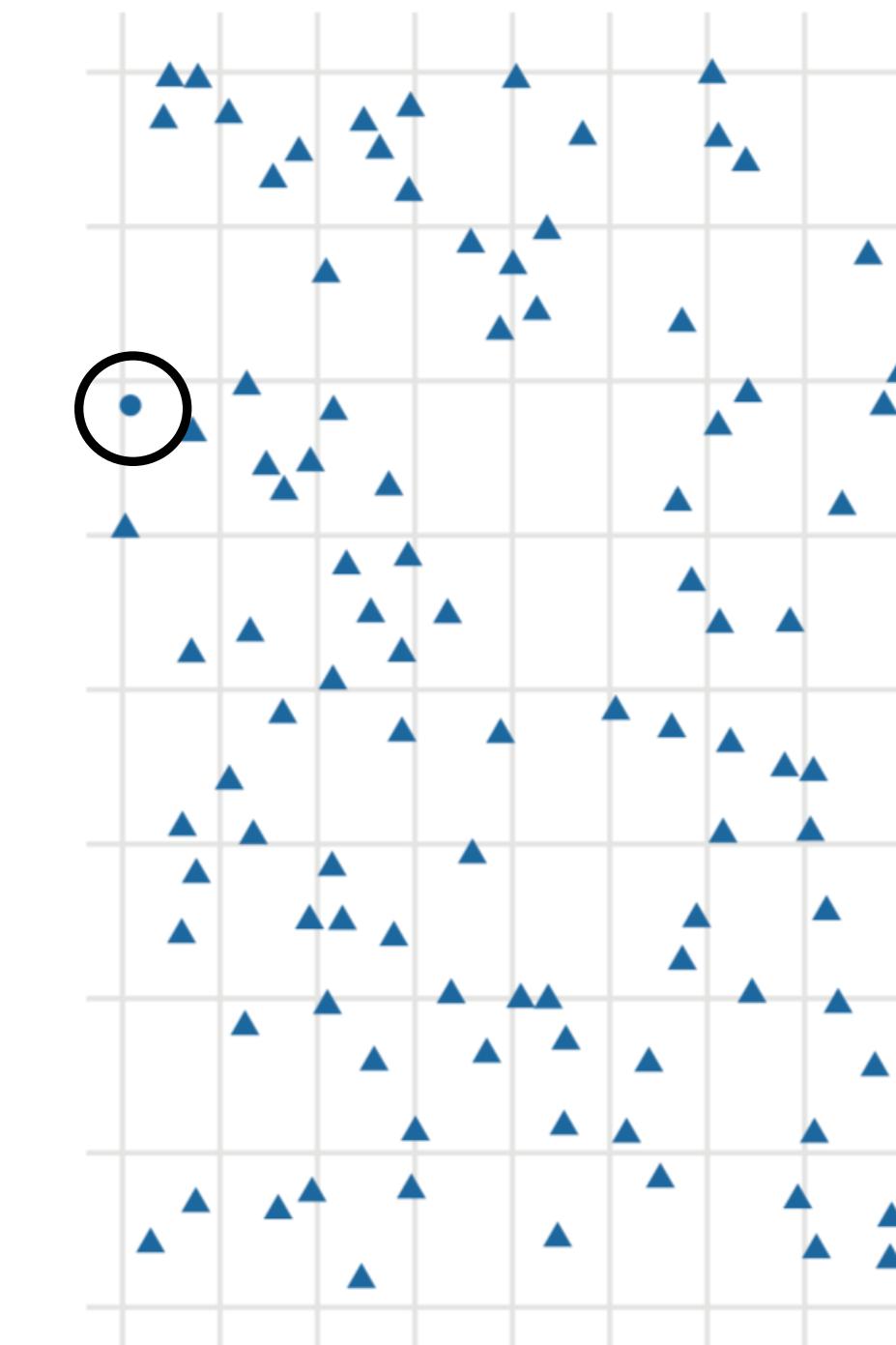
Color only,  $N = 100$



Shape only,  $N = 20$



Shape only,  $N = 100$



Color & shape,  $N = 100$

