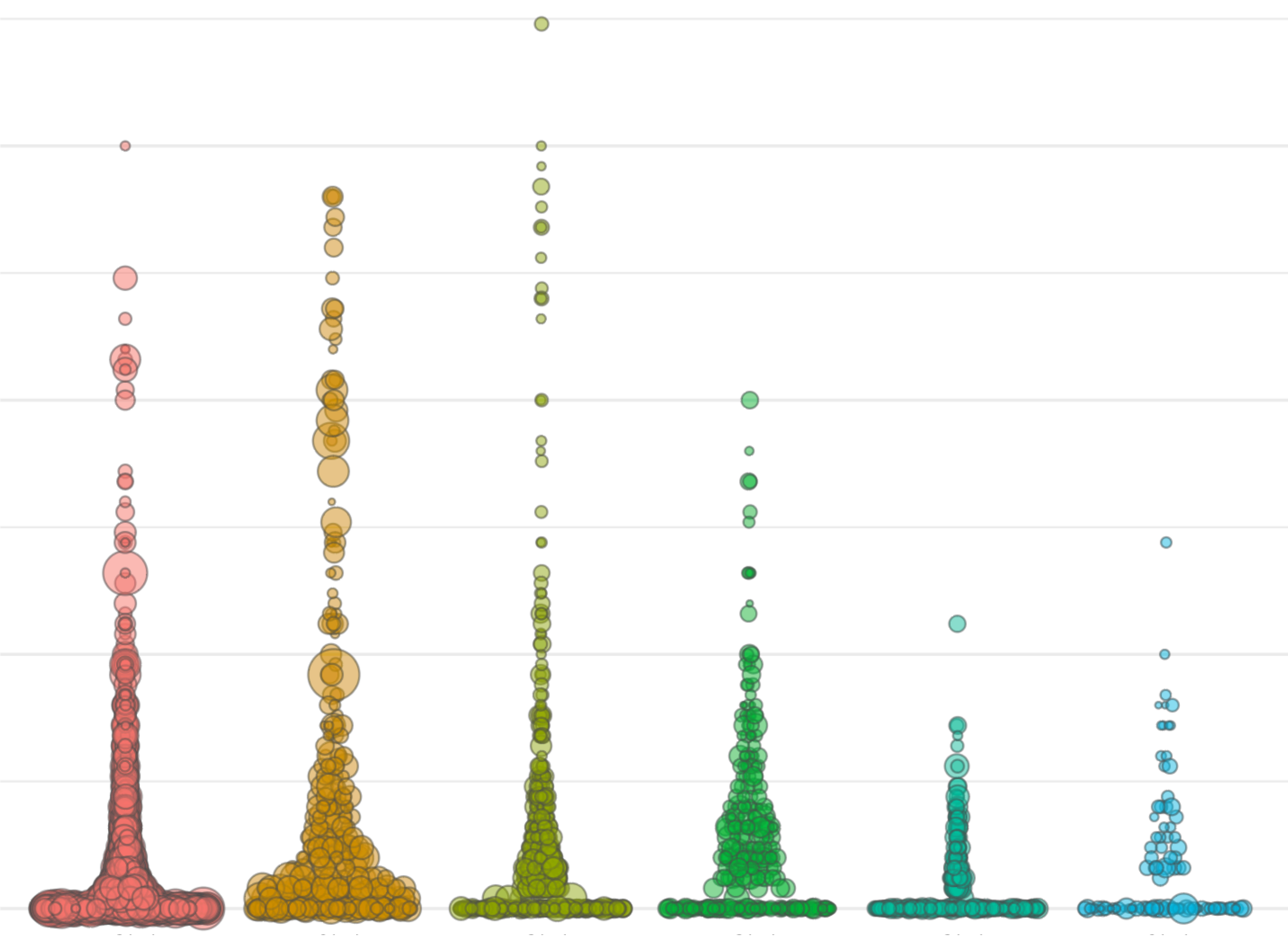


Data Visualization

Spring 2019

Data Management

~ /> _

Spring 2019

http://data880.co

http://socviz880.co

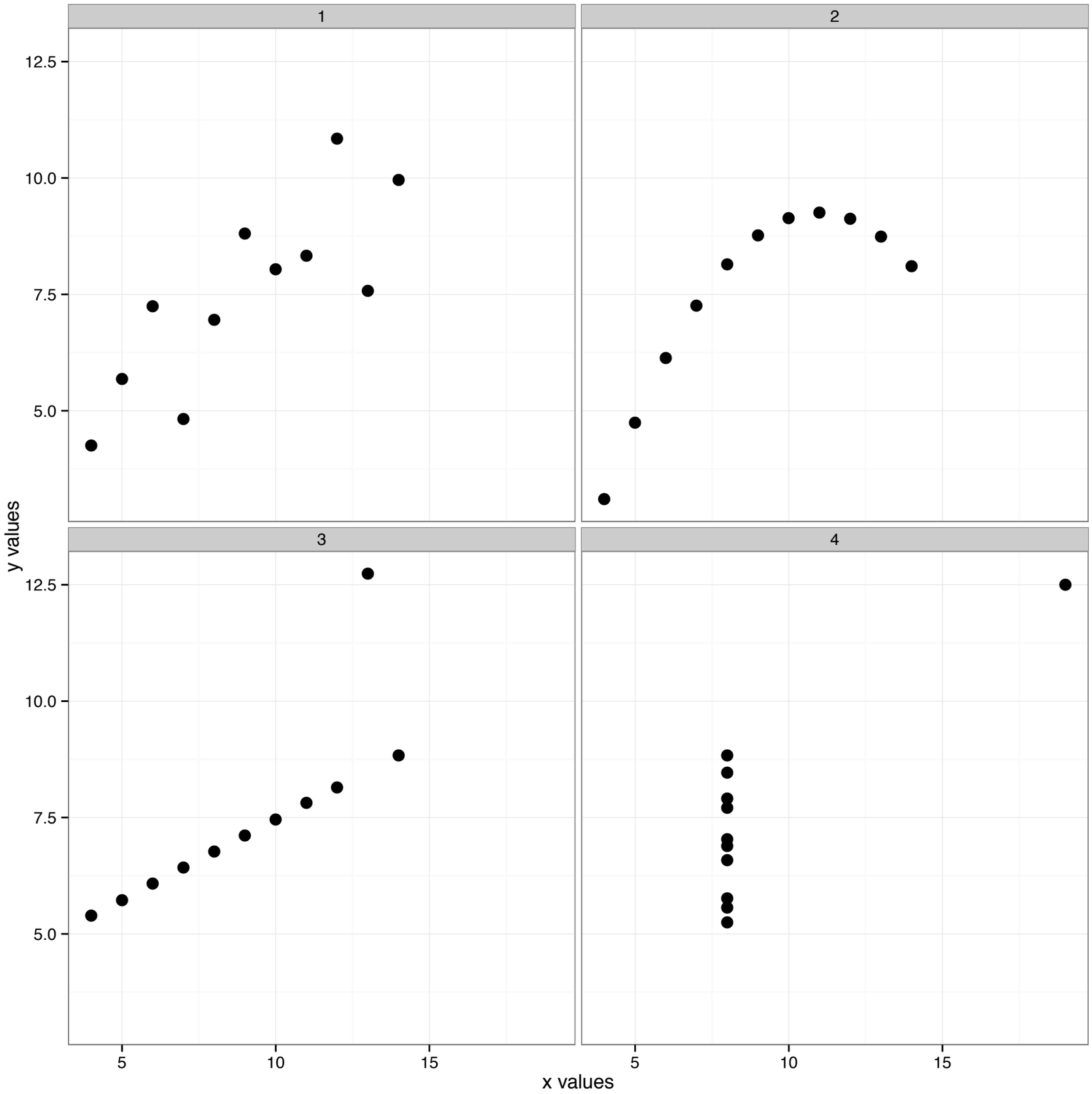
```
my_packages <- c("tidyverse", "fs", "devtools")
install.packages(my_packages)

devtools::install_github("kjhealy/socviz")


library(socviz)
```

```
setup_course_notes()
```

The course packet gets extracted to ~/Desktop/

LOOK AT YOUR DATA

SEEING THINGS



For all panels, N=11; Mean=7.5; Regression: Y=3 + 0.5(X); SE of slope estimate: 0.118, t=4.24; Sum of Squares (X–X): 100; r=0.82.

(01) Normal x, normal residuals

(02) Uniform x, normal residuals

(03) +–skewed x, normal residuals

(04) ––skewed x, normal residuals

(05) Normal x, +–skewed residuals

(06) Normal x, ––skewed residuals

(07) Increasing spread

(08) Decreasing spread

(09) Quadratic trend

(10) Sinusoid relationship

(11) A single positive outlier

(12) A single negative outlier

(13) Bimodal residuals

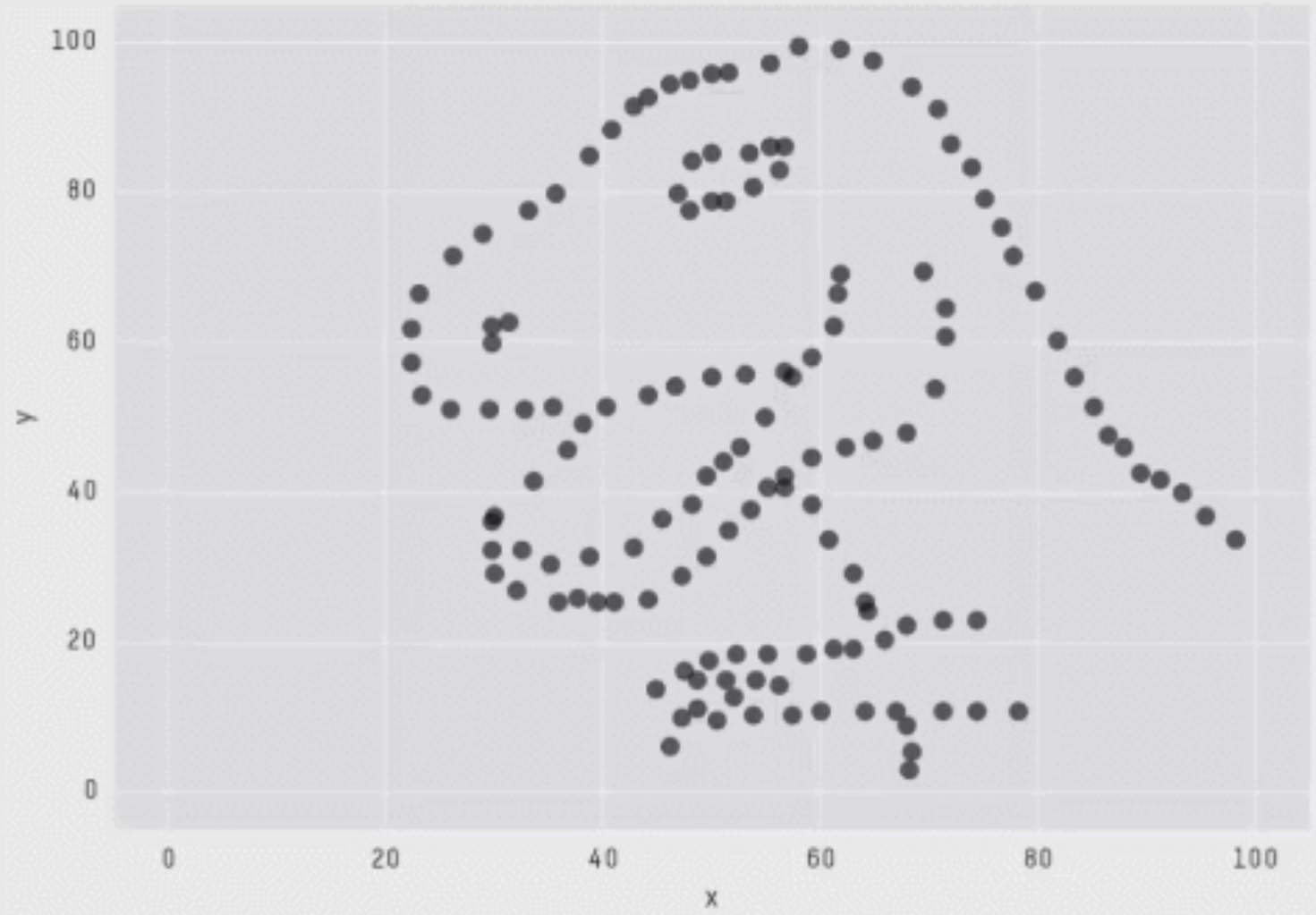
(14) Two groups

(15) Sampling at the extremes

(16) Coarse data

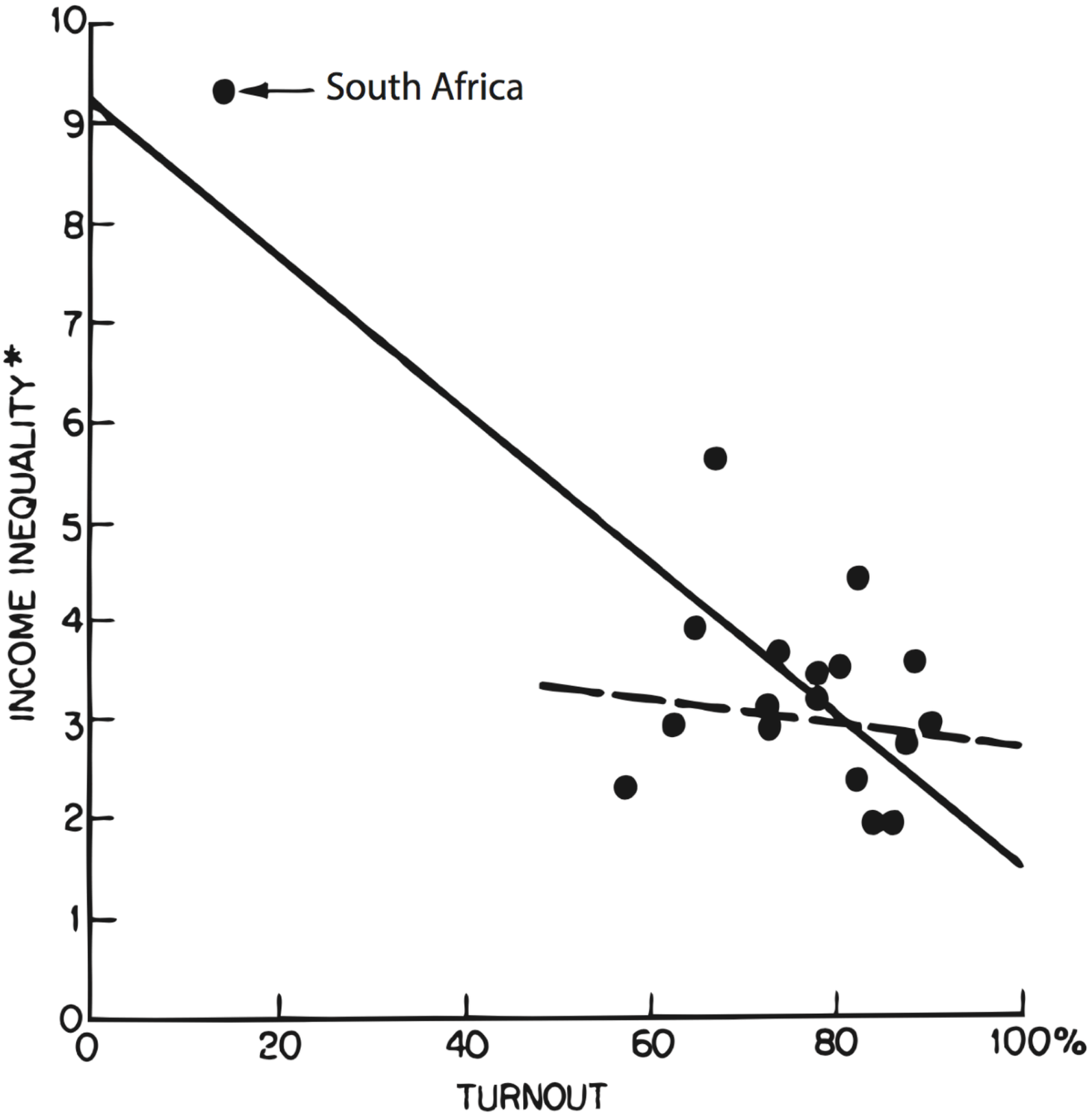
y

x



X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

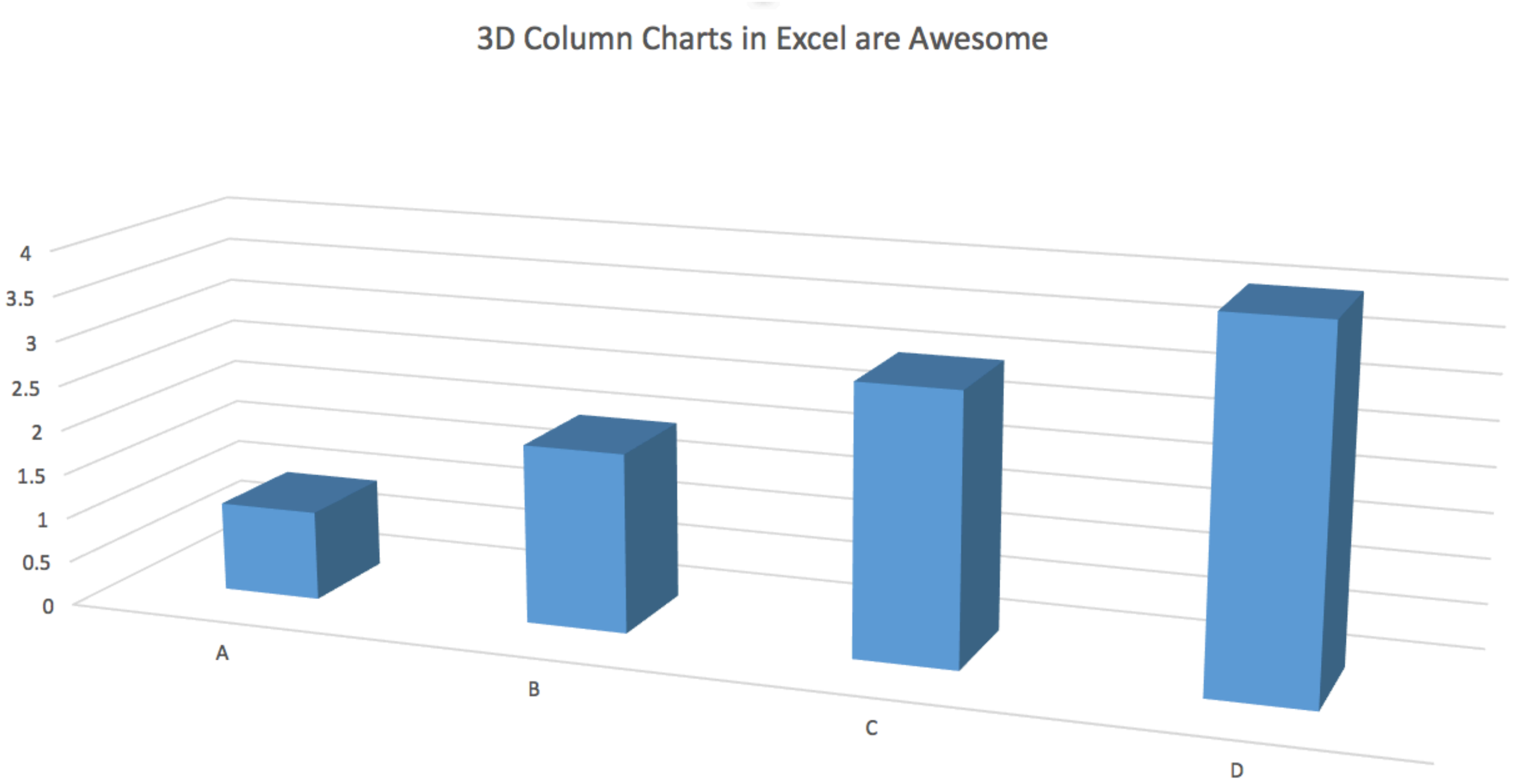
Cairo; Matejka & Fitzmaurice



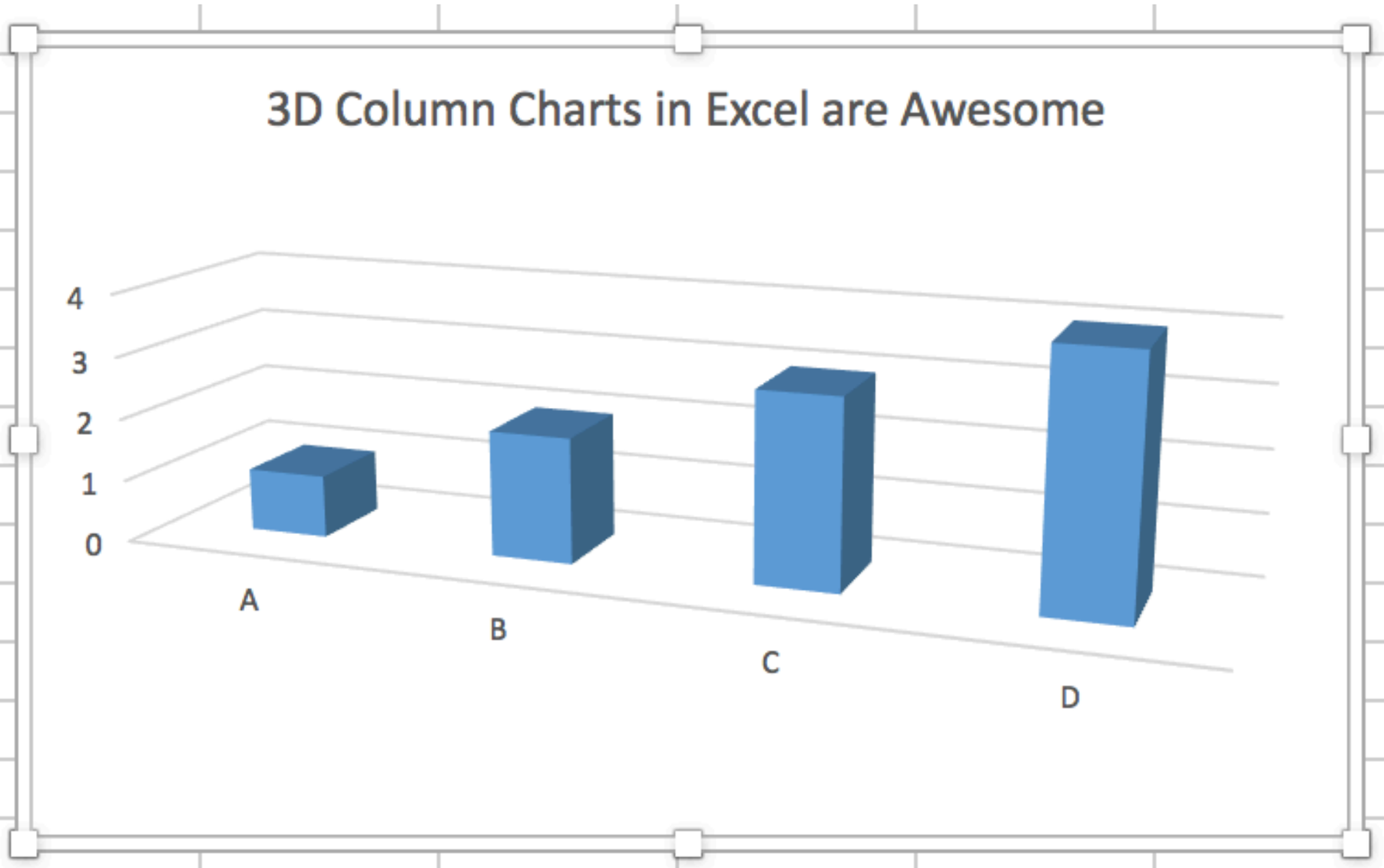
Key. ——— Bivariate slope including South Africa ($N = 18$)

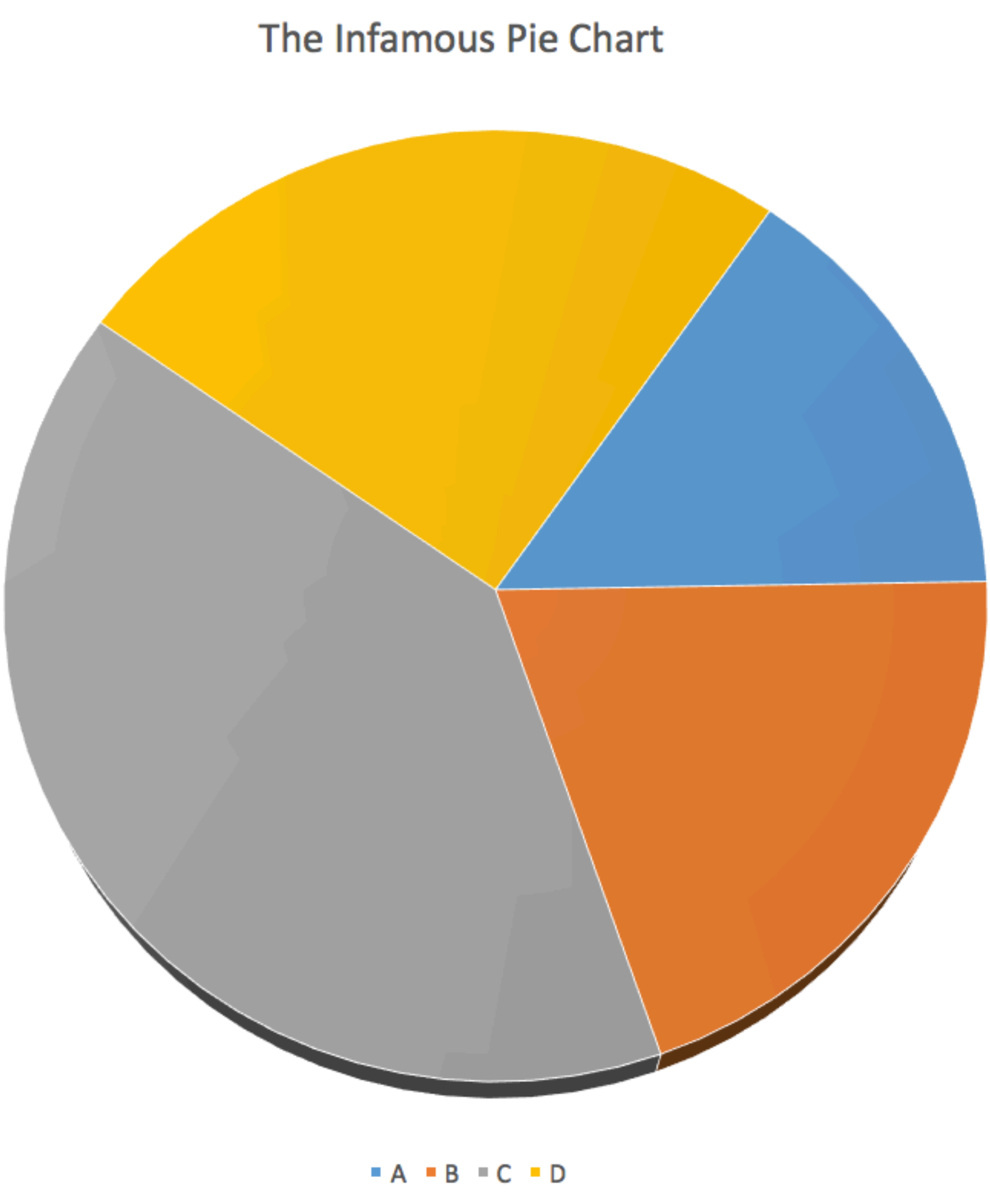
– – – Bivariate slope excluding South Africa ($N = 17$)

NOT SEEING THINGS

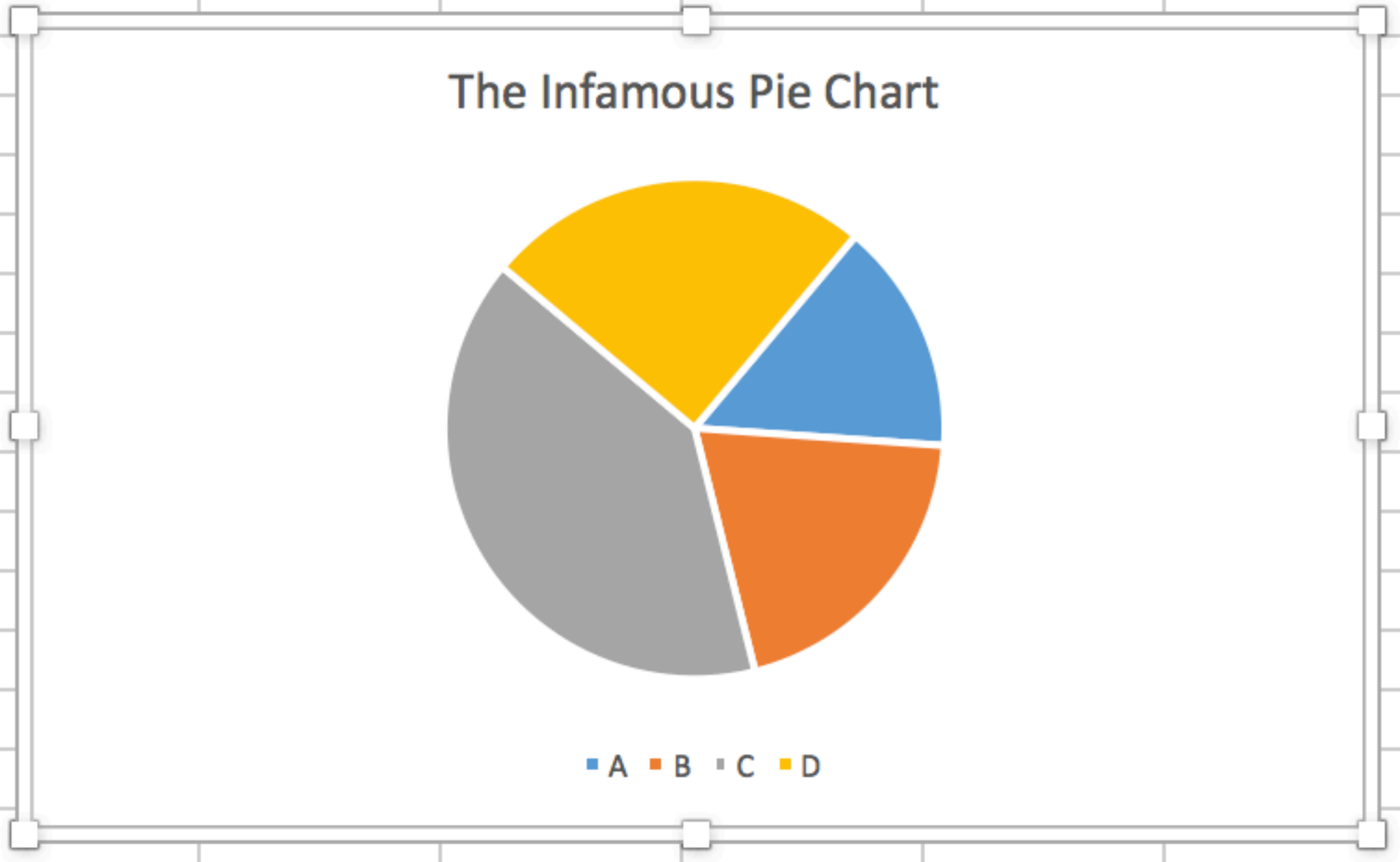

3D Column Charts in Excel are Awesome
4
3.5
3
2.5
2
1.5
1
0.5
0
A
B
C
D

A	1
B	2
C	3
D	4




The Infamous Pie Chart
A
B
C
D

A	15
B	20
C	40
D	25
	100



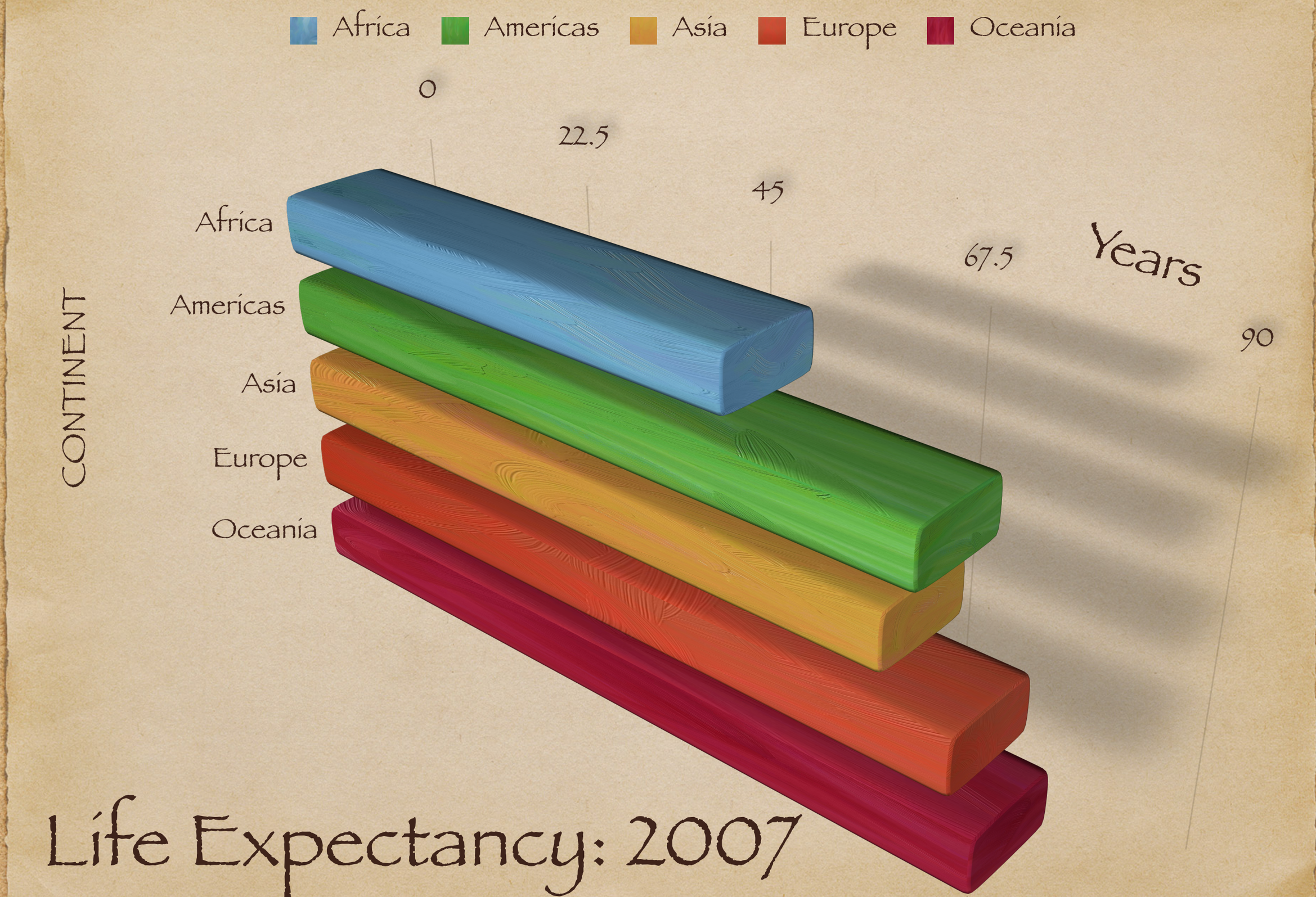
BAD TASTE

BAD DATA

BAD PERCEPTION

TASTE

SIMPLIFY, SIMPLIFY?



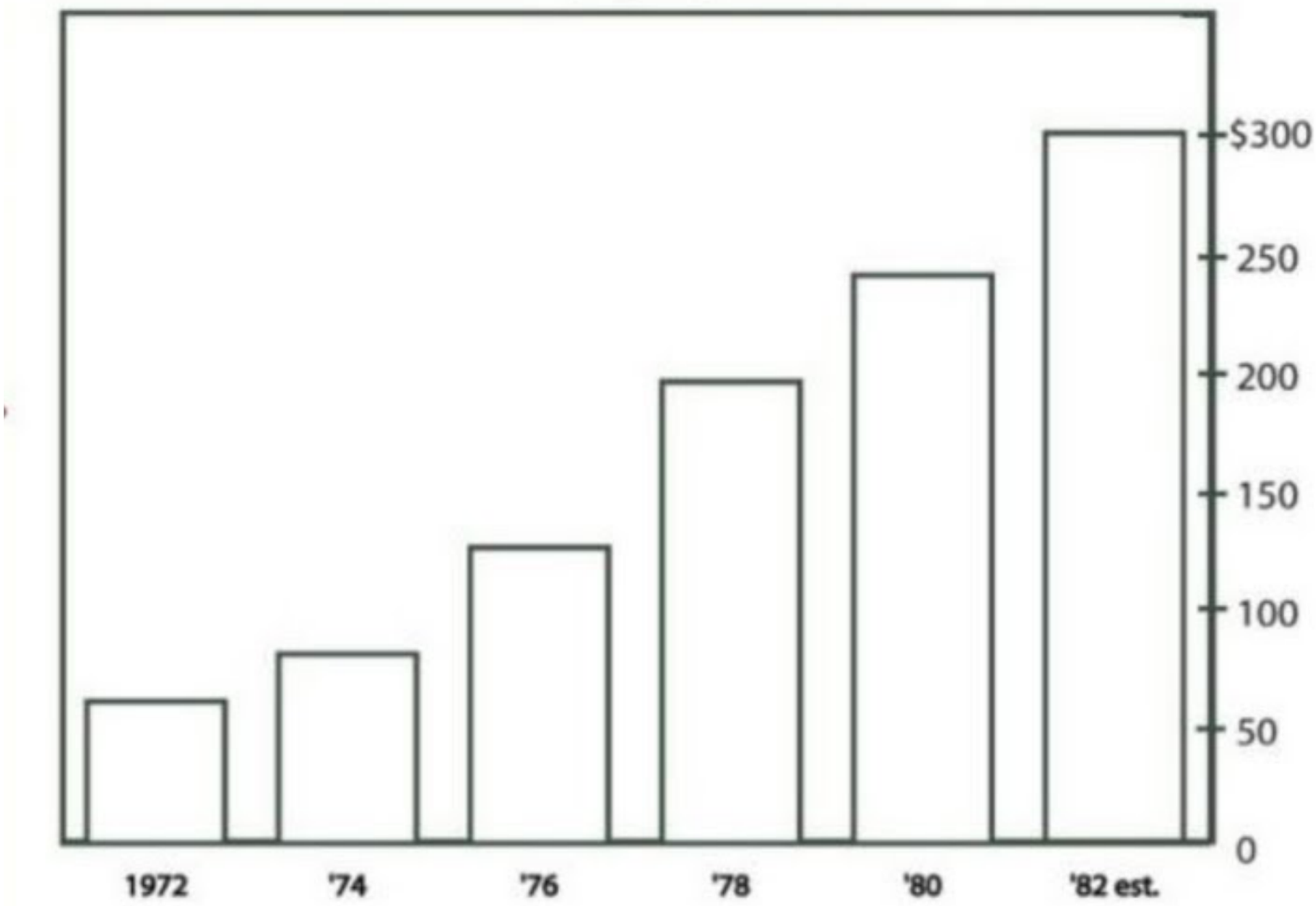
Life Expectancy: 2007

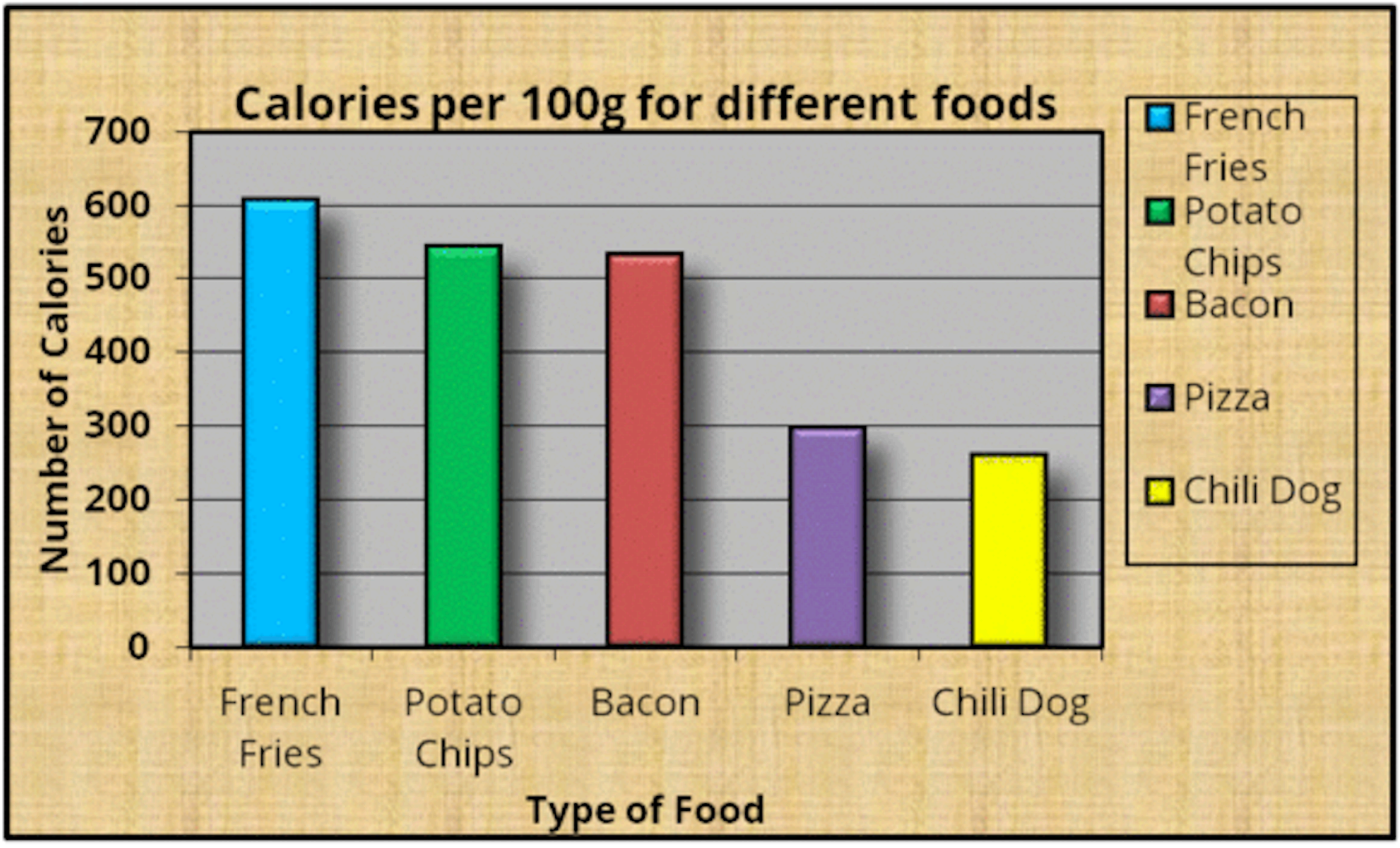
MONSTROUS COSTS

Total House and Senate campaign expenditures, in millions

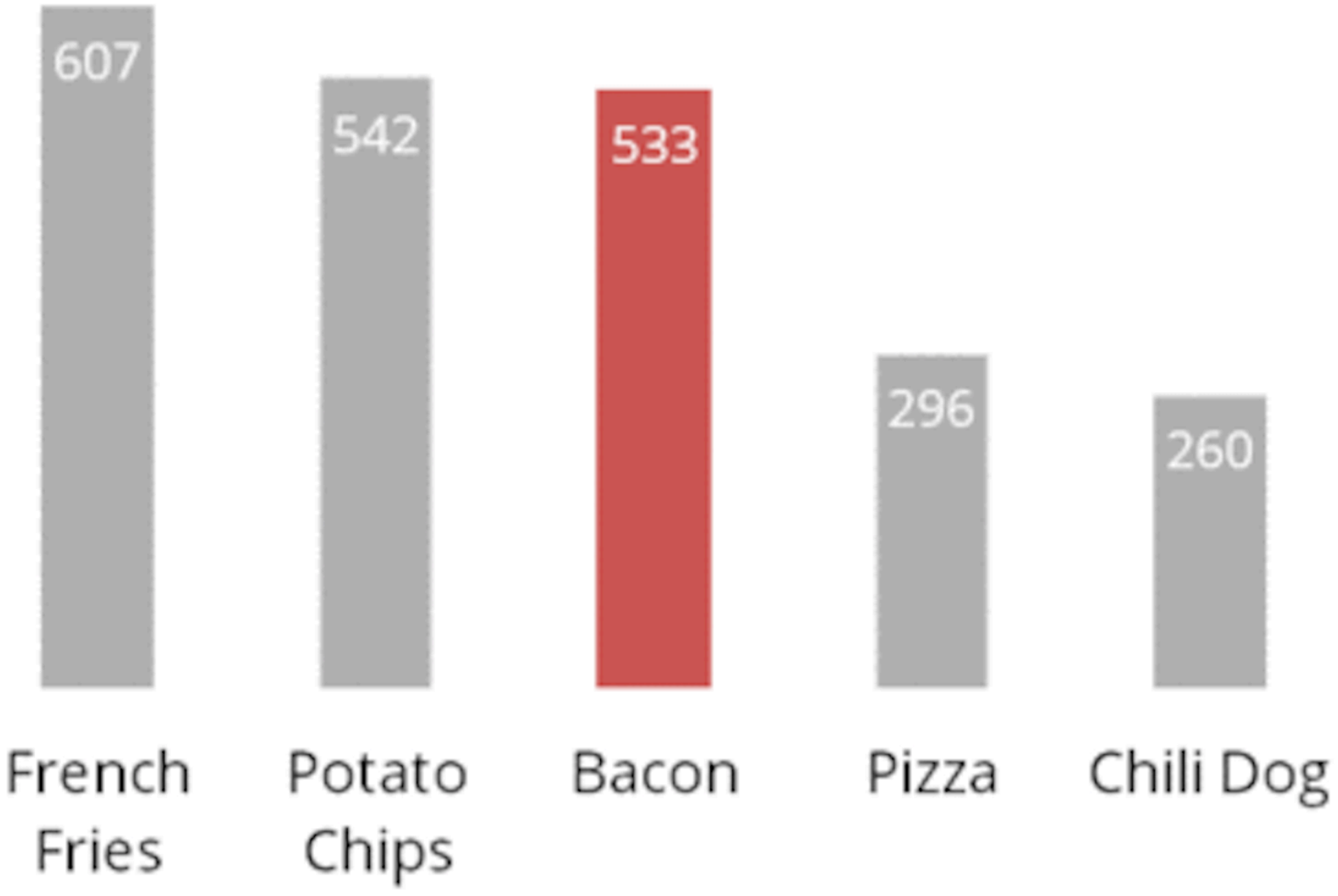
MONSTROUS COSTS

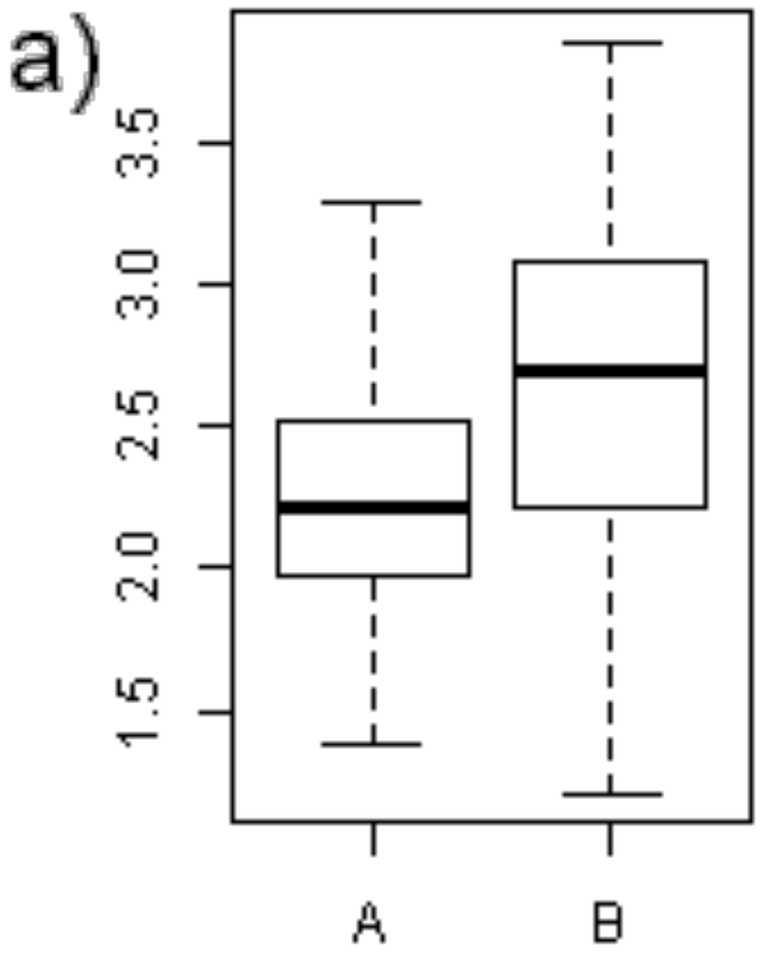
Total House and Senate campaign expenditures, in millions

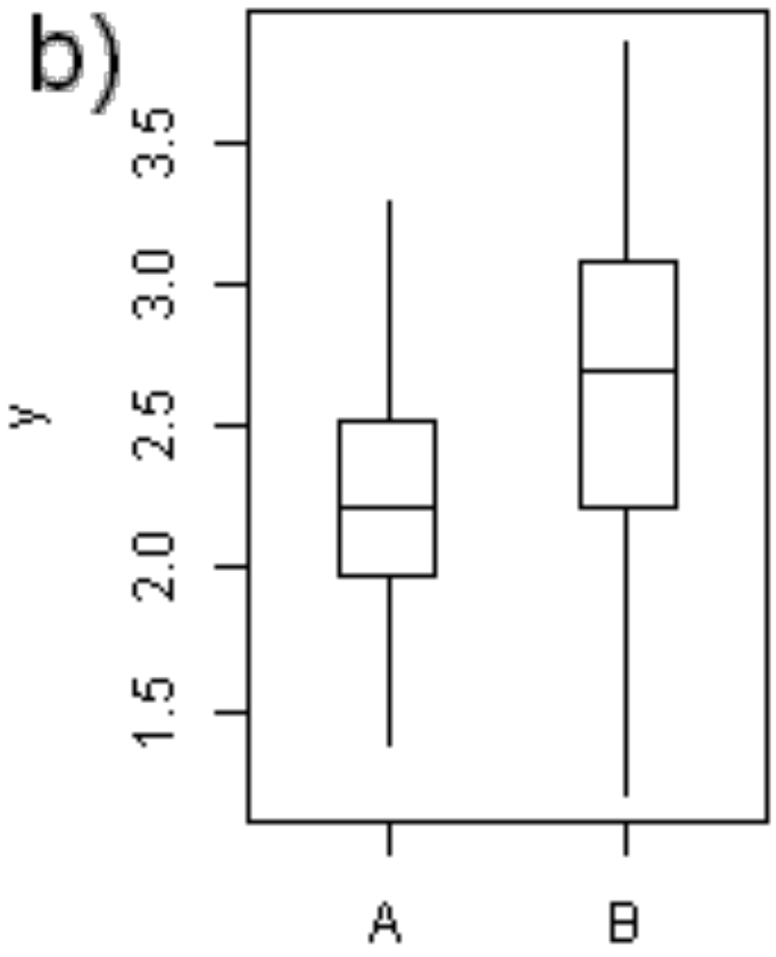


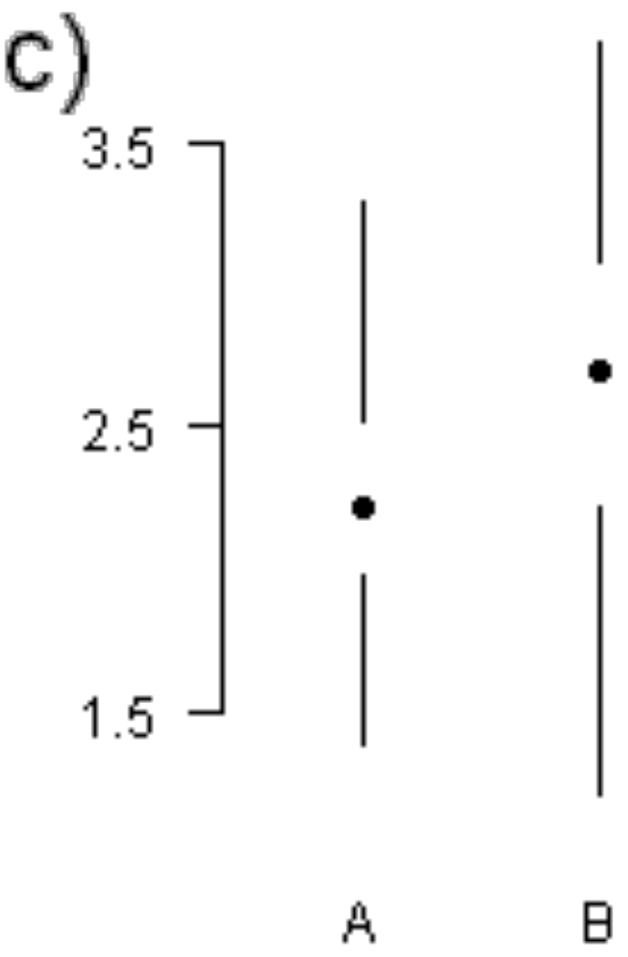


Calories per 100g

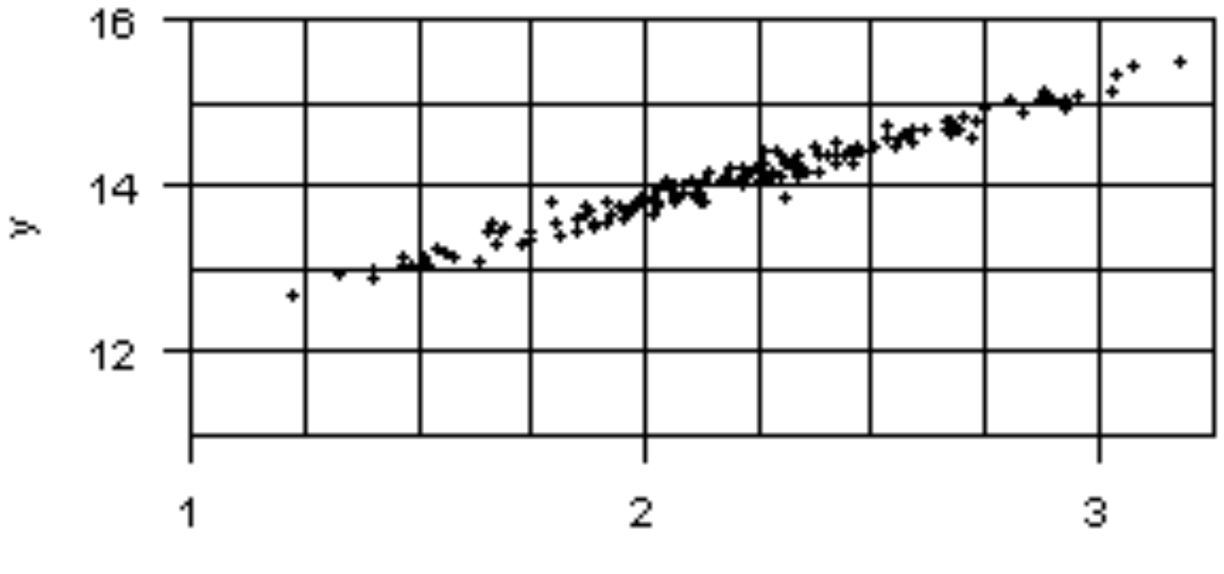



a)
3.5
3.0
2.5
2.0
1.5
A
B

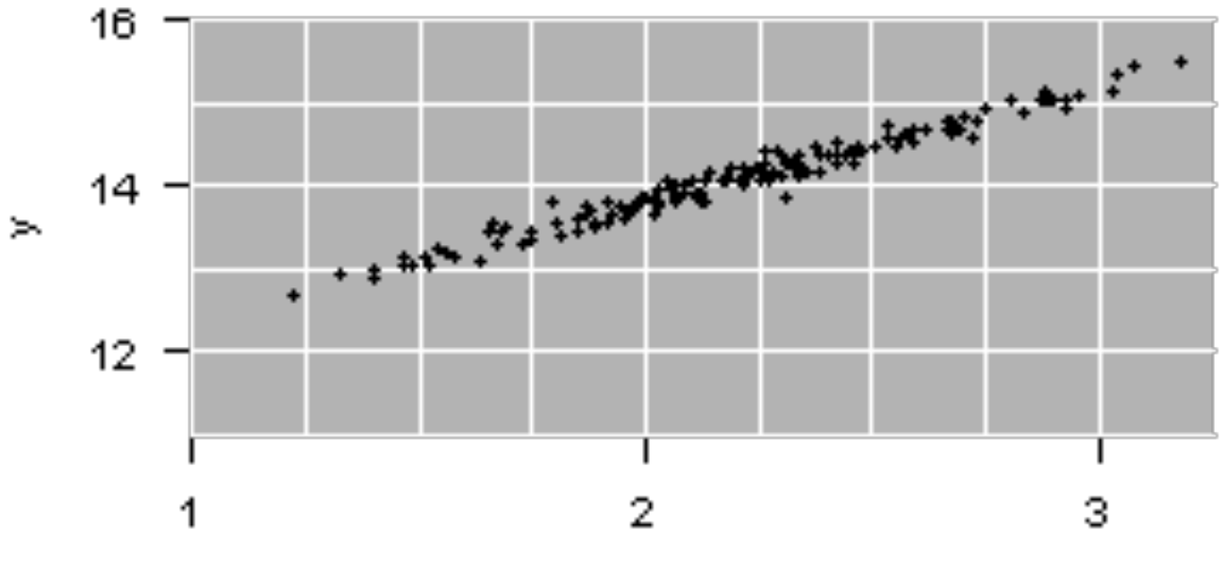

b)
y
3.5
3.0
2.5
2.0
1.5
A
B


c)
3.5
2.5
1.5
A
B

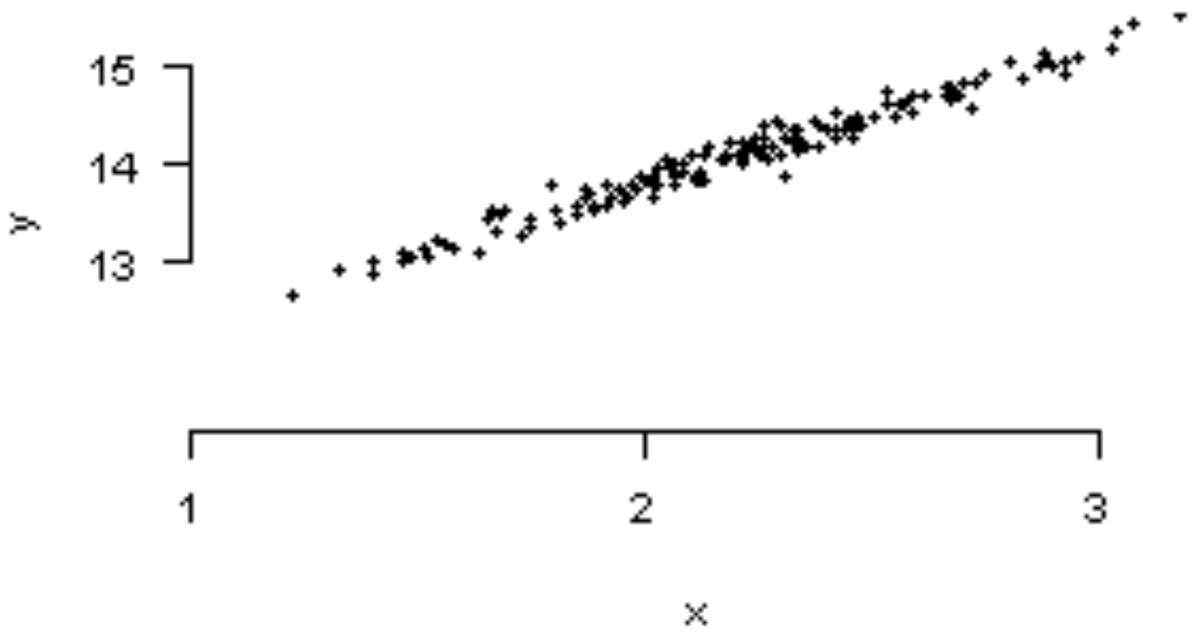
a)


16
14
12
y
1
2
3
x

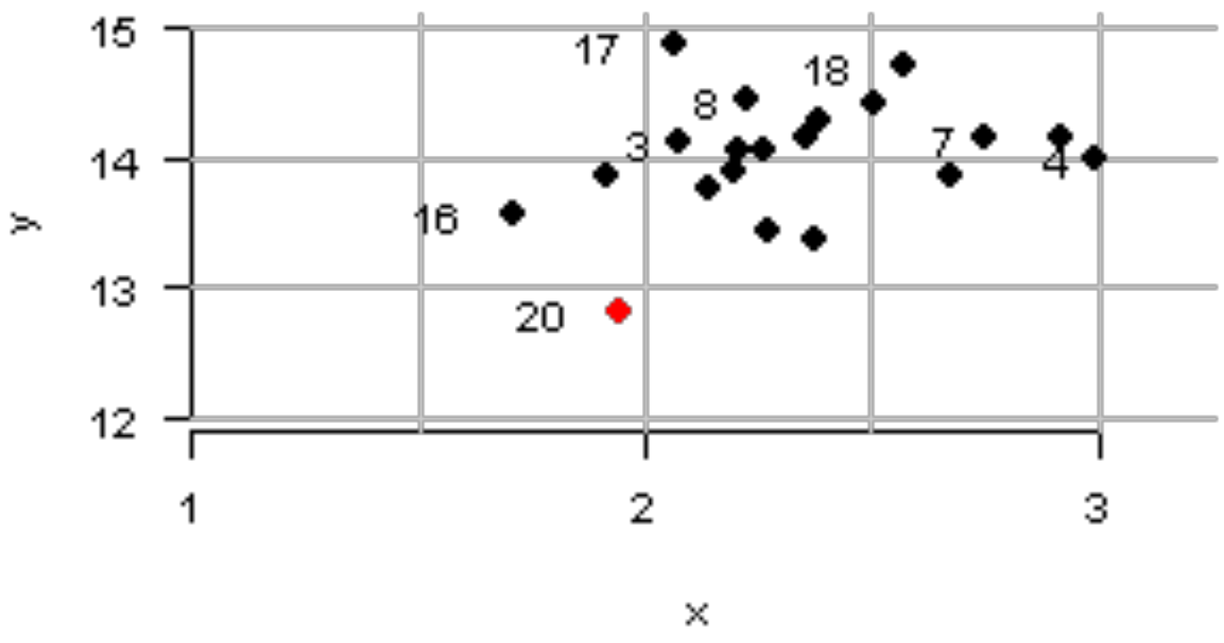
b)


16
14
12
y
1
2
3
x

c)

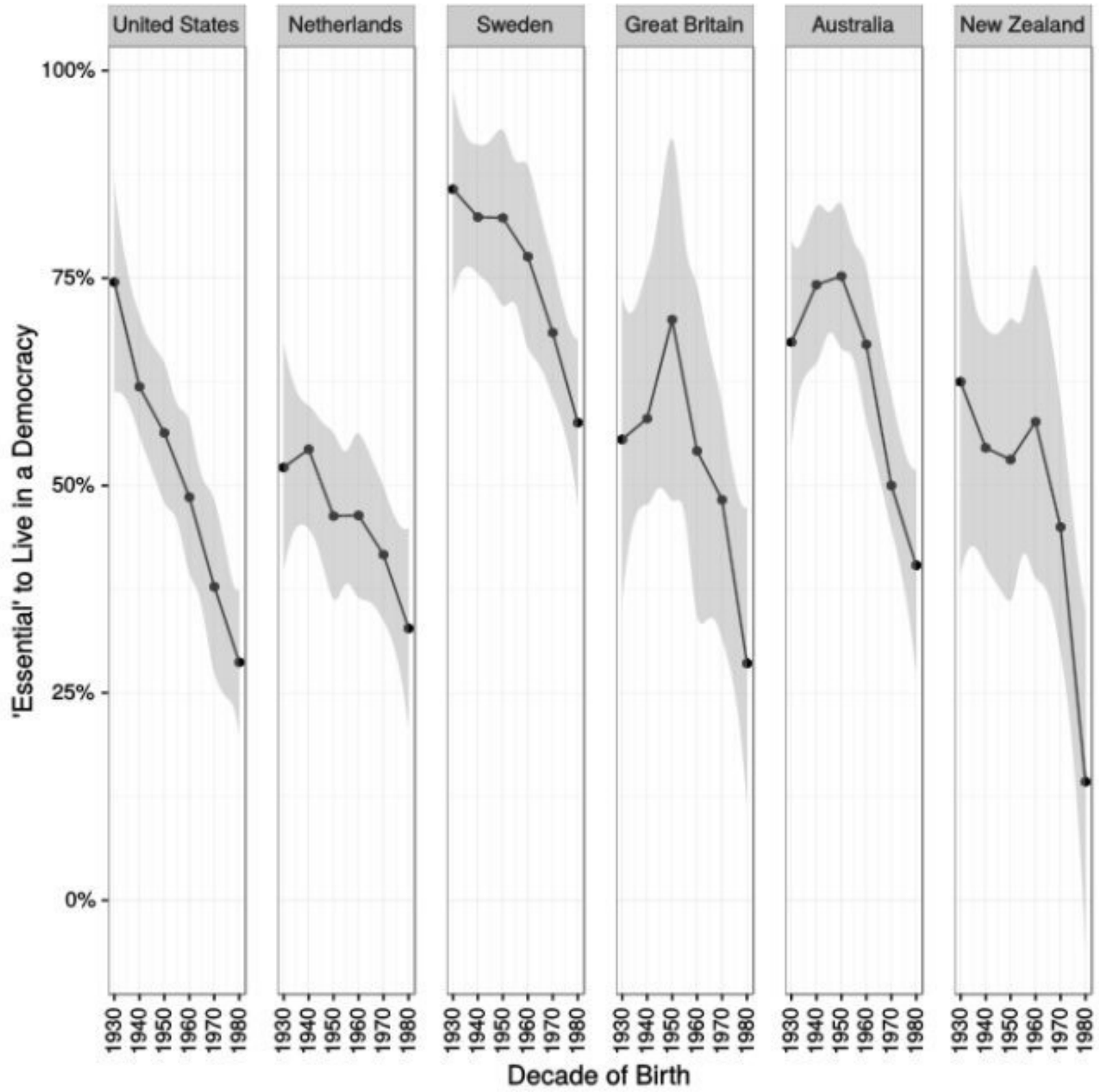

15
14
13
y
1
2
3
x

d)


15
14
13
12
y
17
18
8
3
7
4
16
20
1
2
3
x

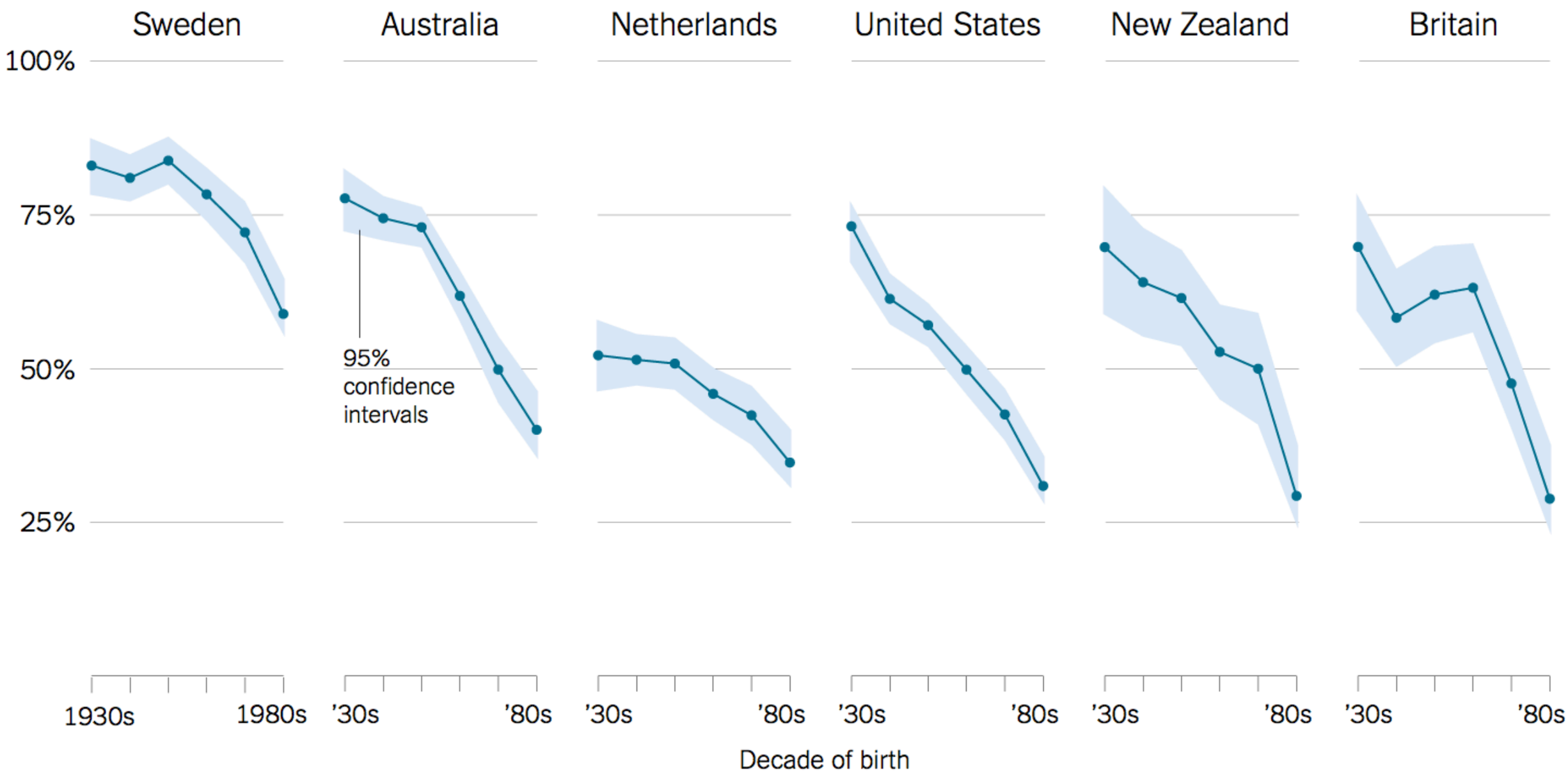
DATA

GOOD DESIGN
BAD DATA



Yascha Mounk and Roberto Stefan Foa, "The Signs of Democratic Deconsolidation," Journal of Democracy

Percentage of people who say it is “essential” to live in a democracy



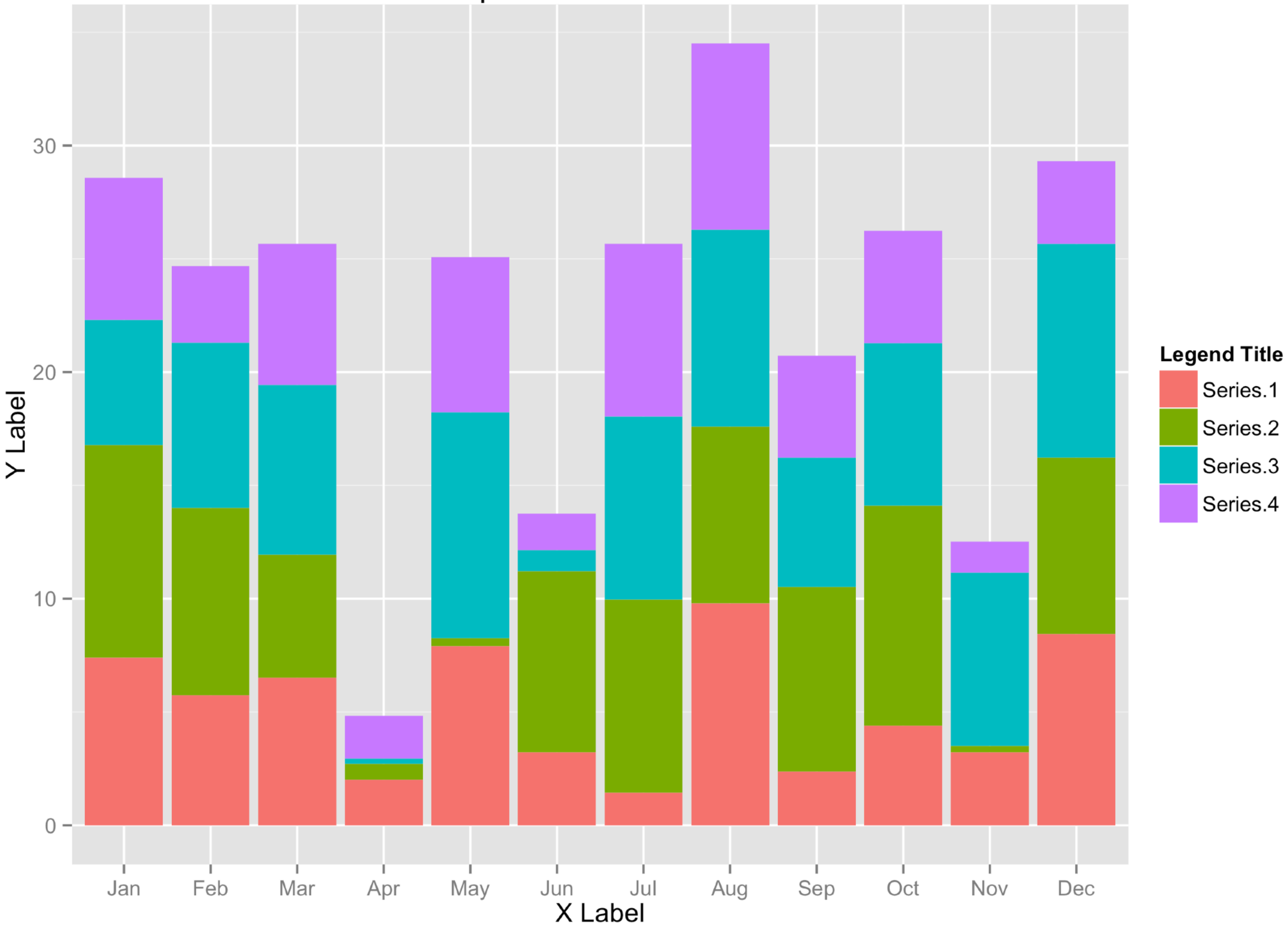
Source: Yascha Mounk and Roberto Stefan Foa, “The Signs of Democratic Deconsolidation,” Journal of Democracy | By The New York Times

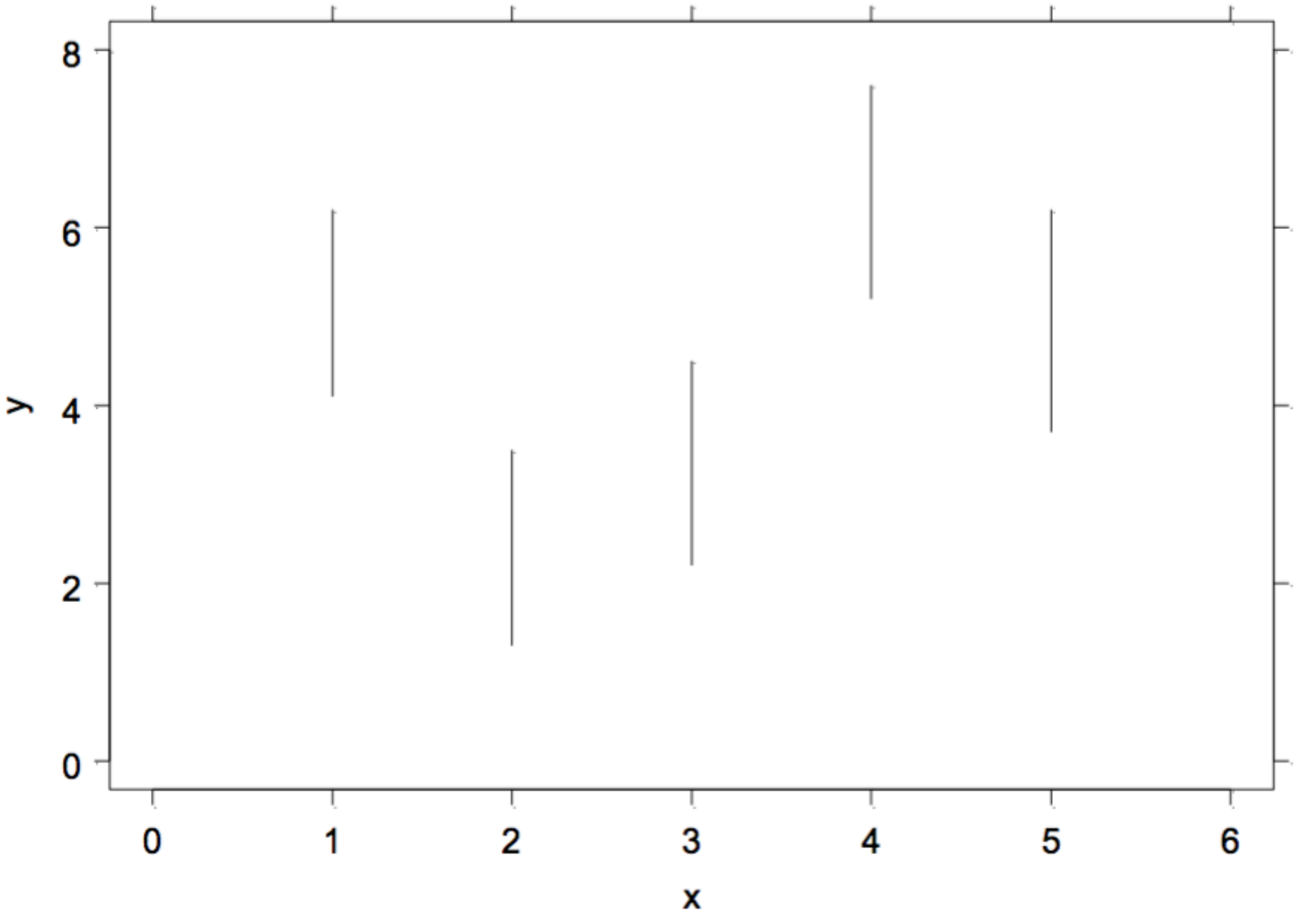
Australia
Netherlands
Sweden
Great Britain
United States
Average importance of democracy
10
9
8
7
6
5
4
3
2
1
30s
50s
70s
90s
Decade of birth
Graph by Erik Voeten, based on WVS 5

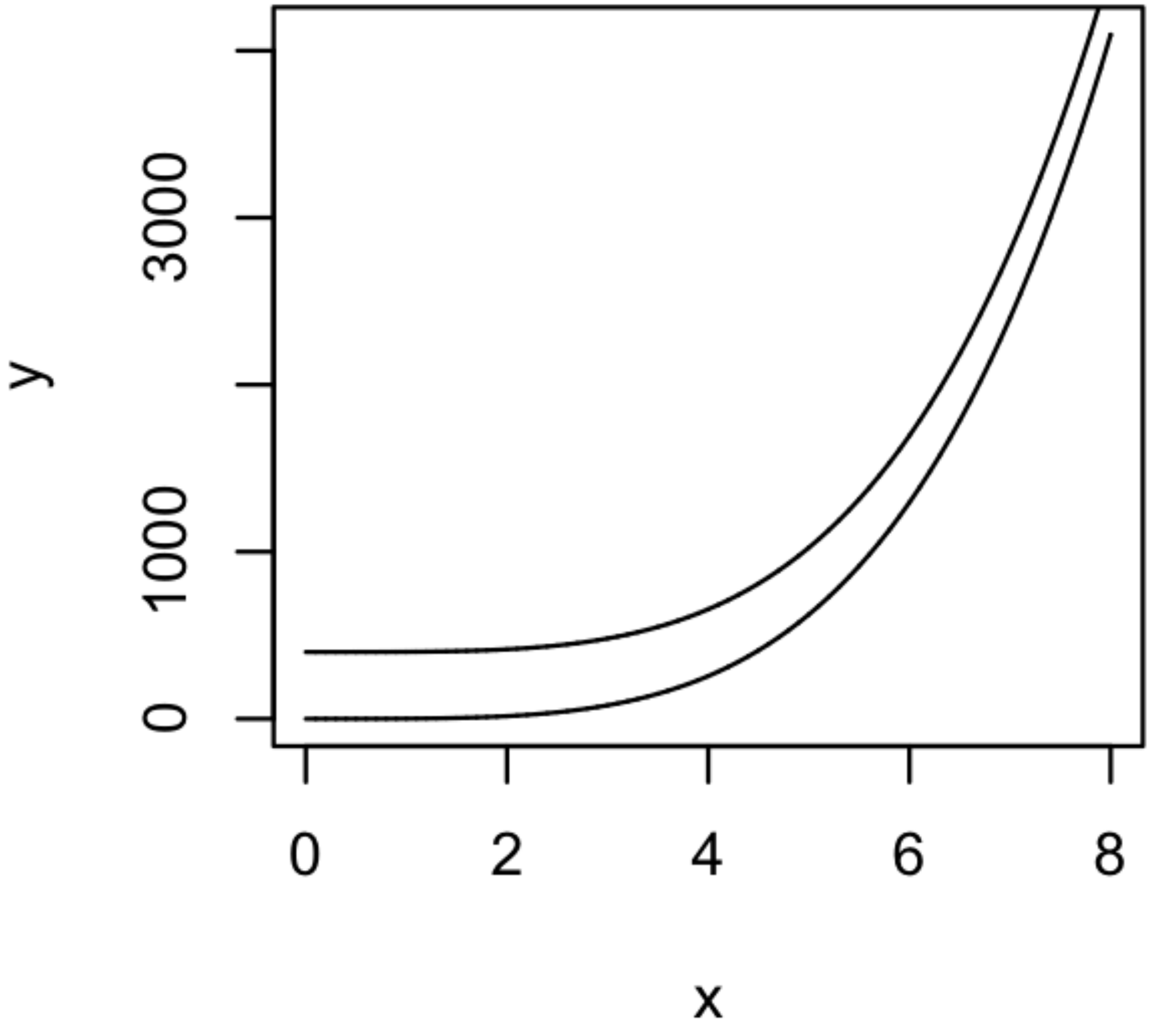
PERCEPTION

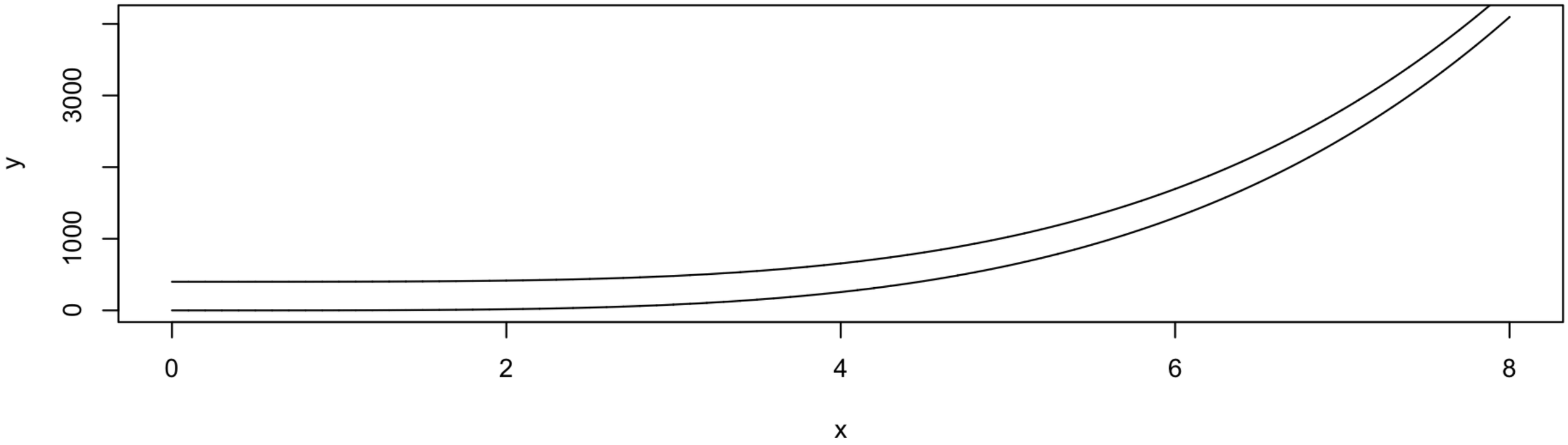
GOOD DESIGN IS MORE THAN GOOD TASTE

An Example Stacked Column Chart




y
8
6
4
2
0
0
1
2
3
4
5
6
x


y
3000
0
1000
0
2
4
6
8
x


y
0
1000
3000
0
2
4
6
8
x

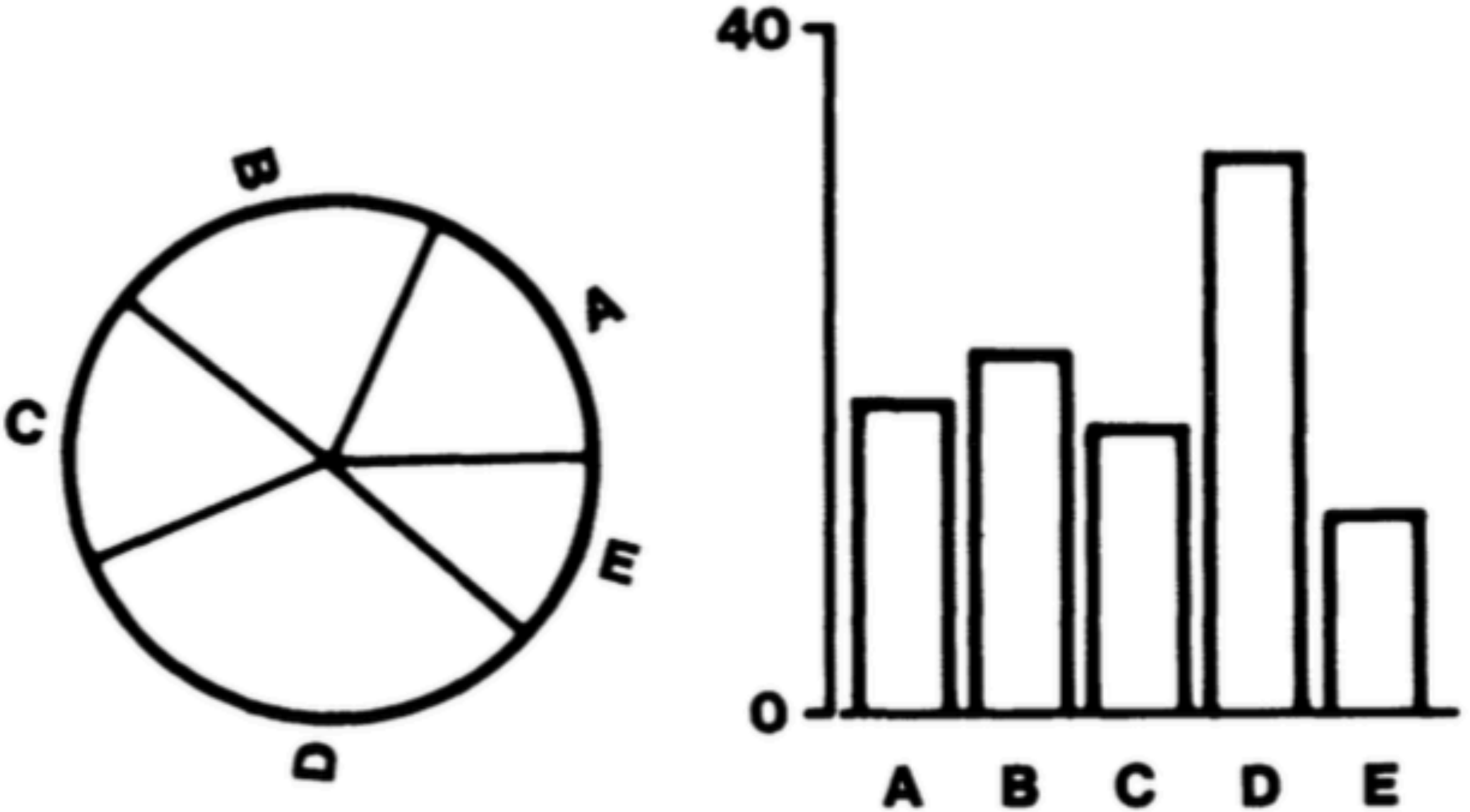
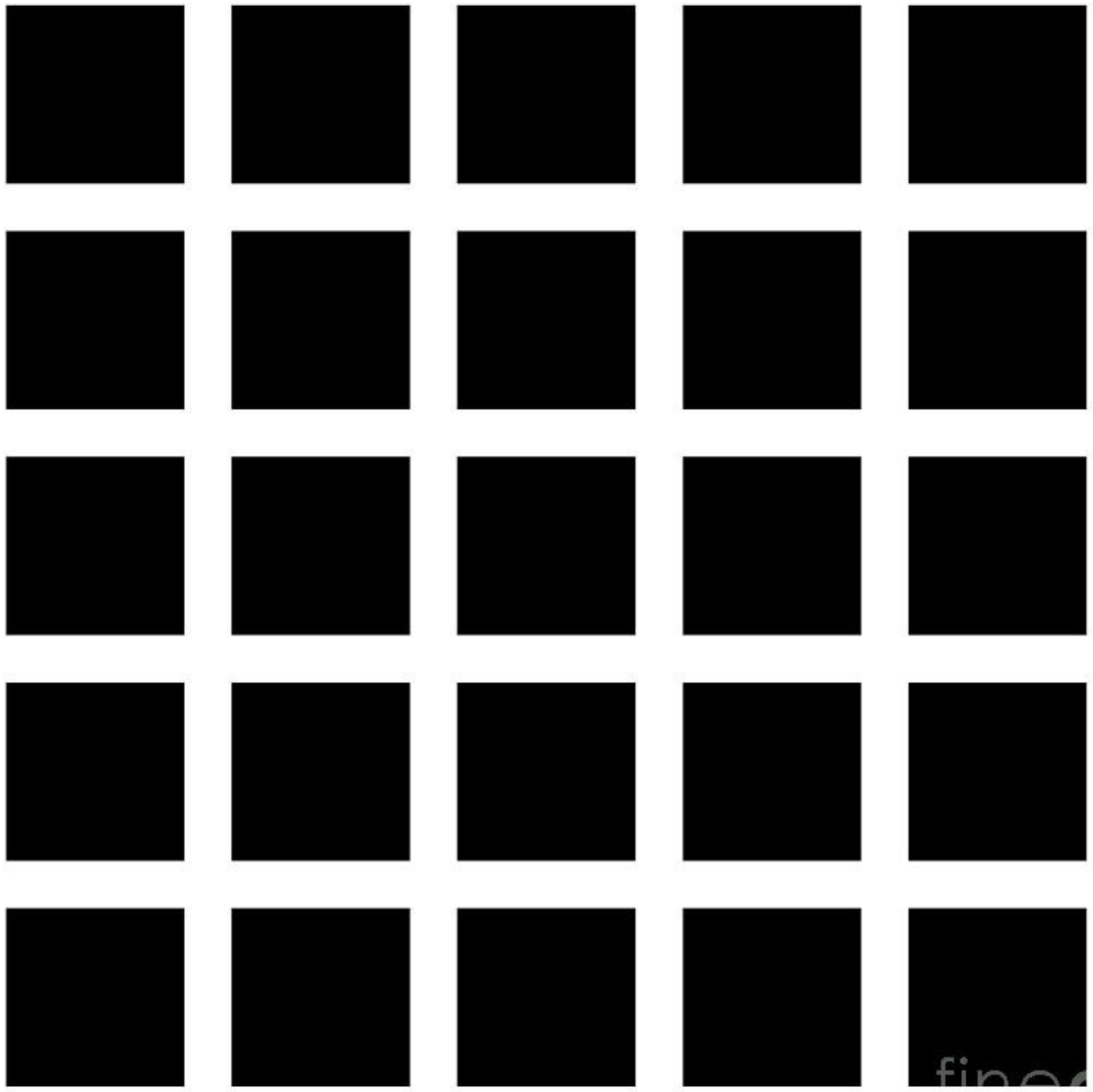
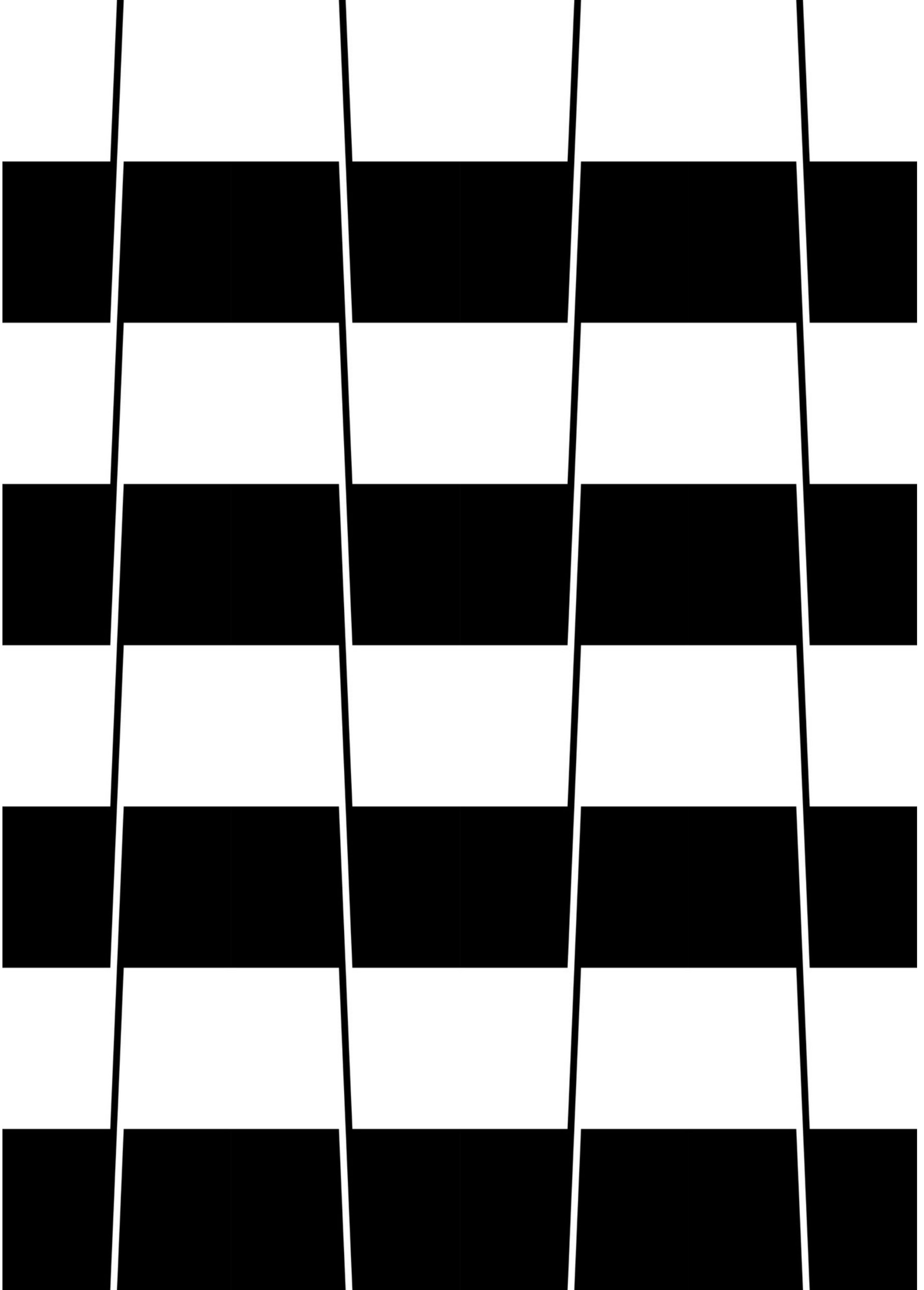
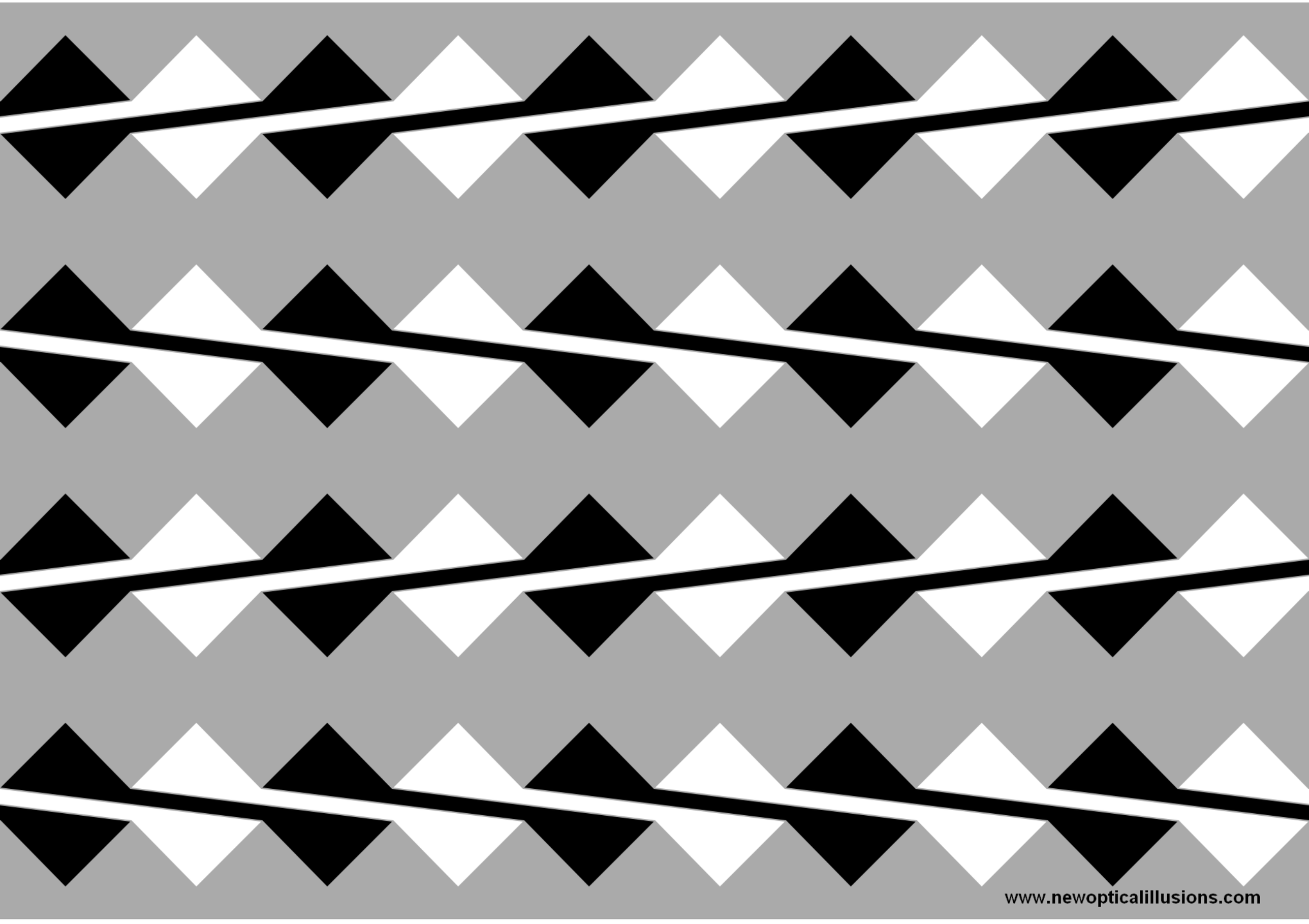
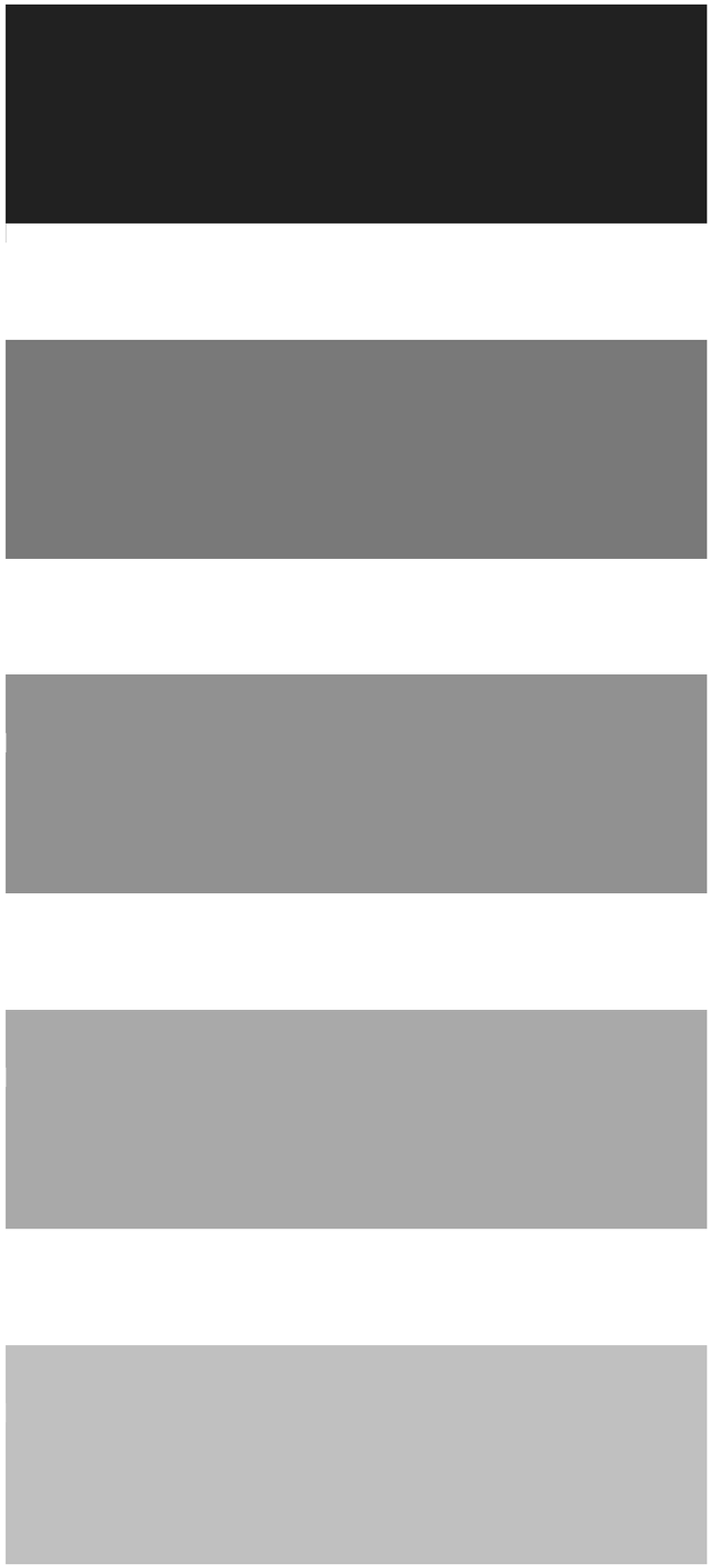


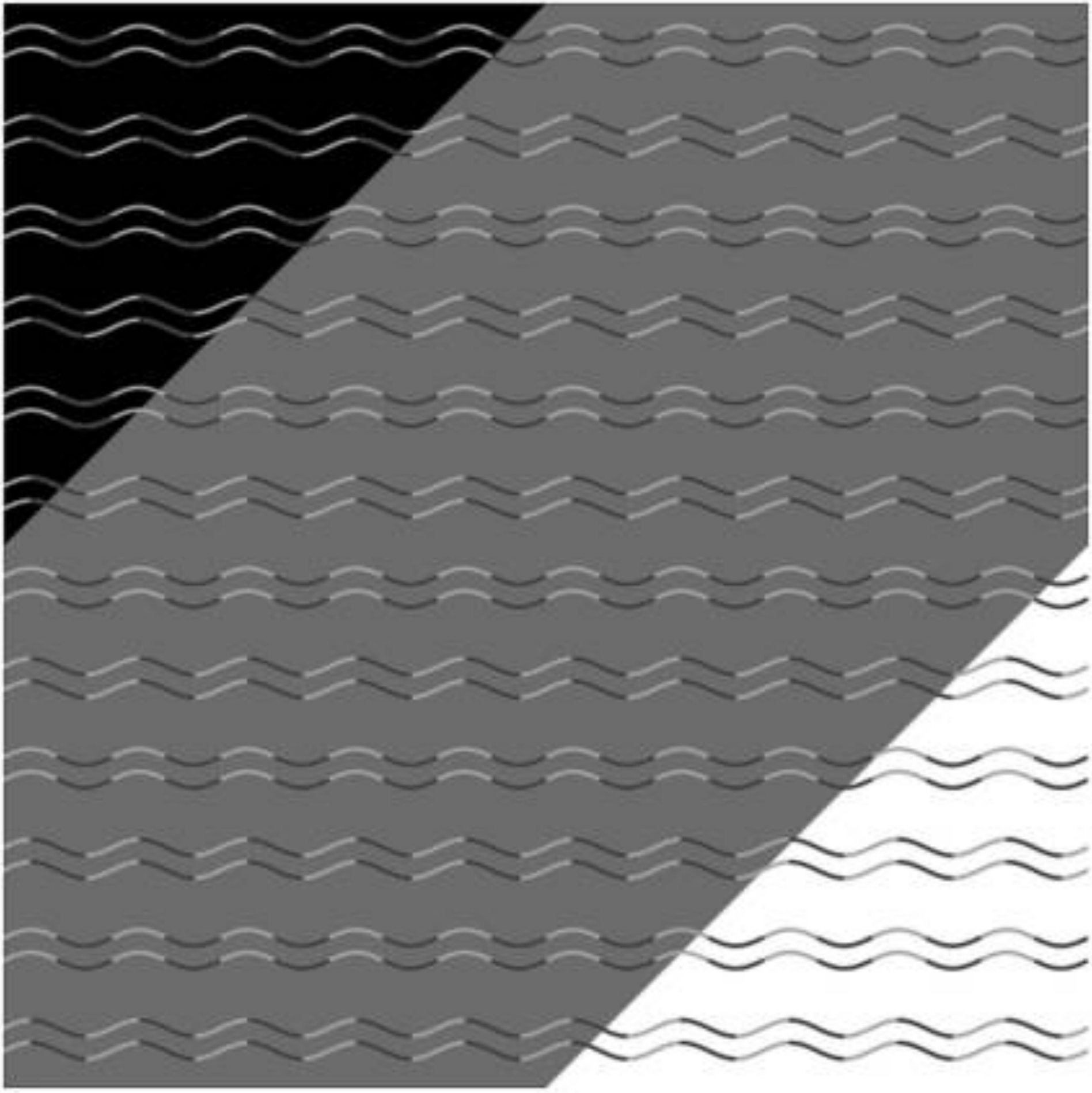
Figure 3. Graphs from position–angle experiment.

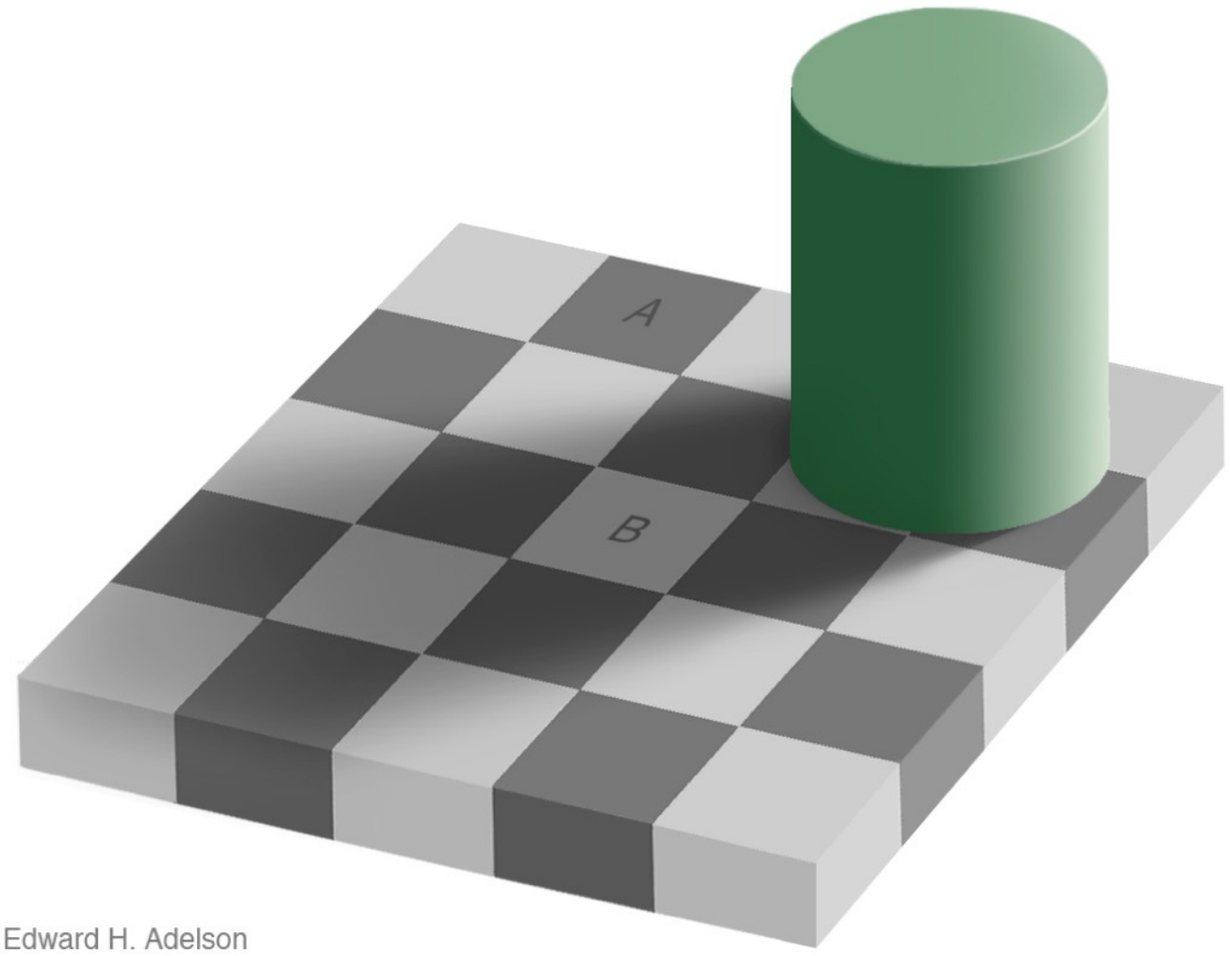


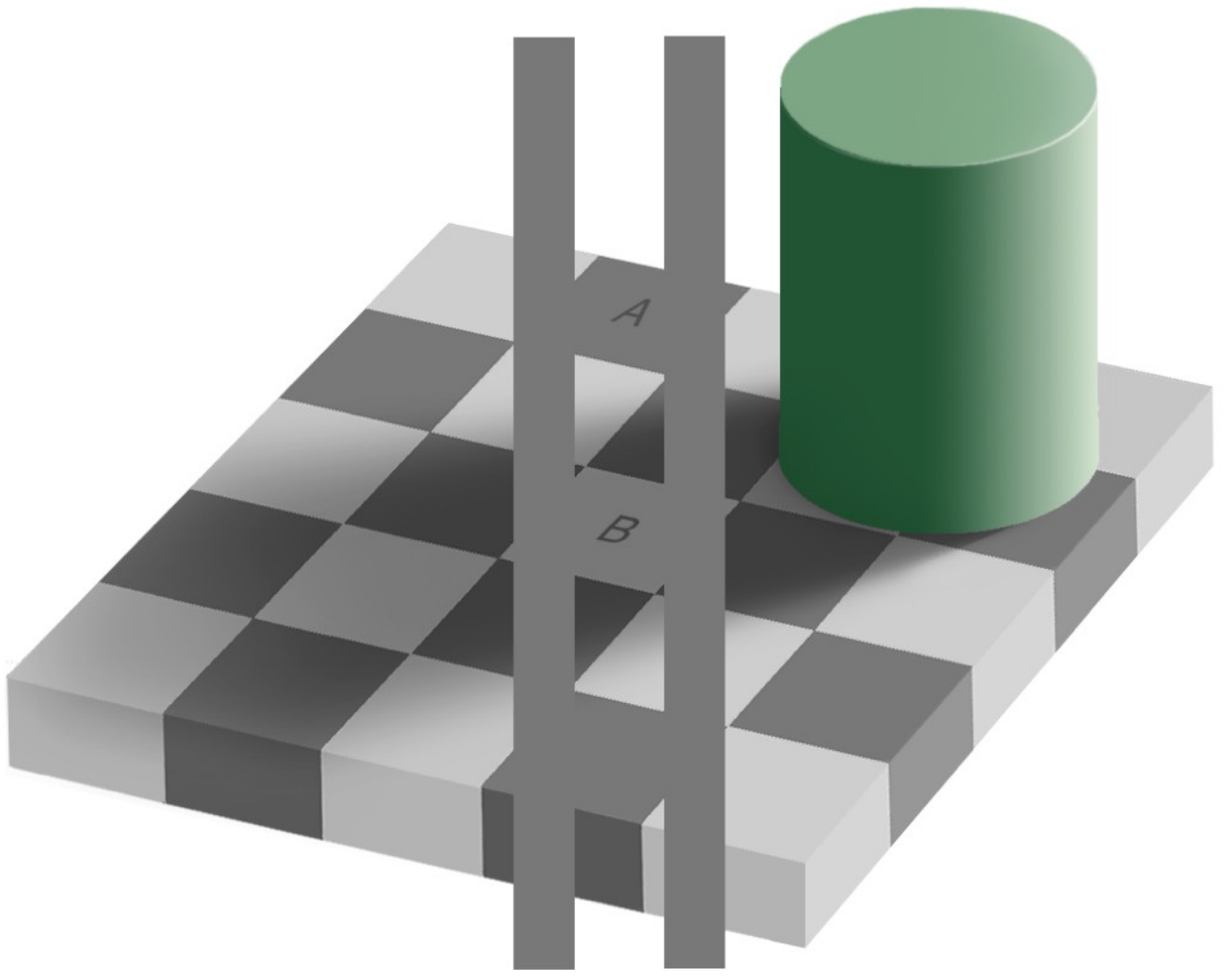



www.newopticalillusions.com

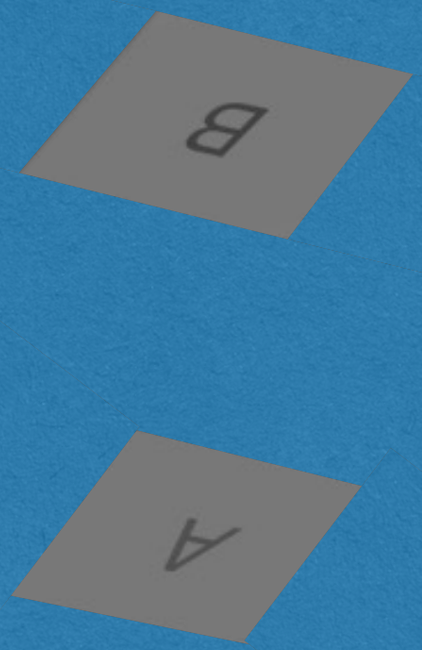


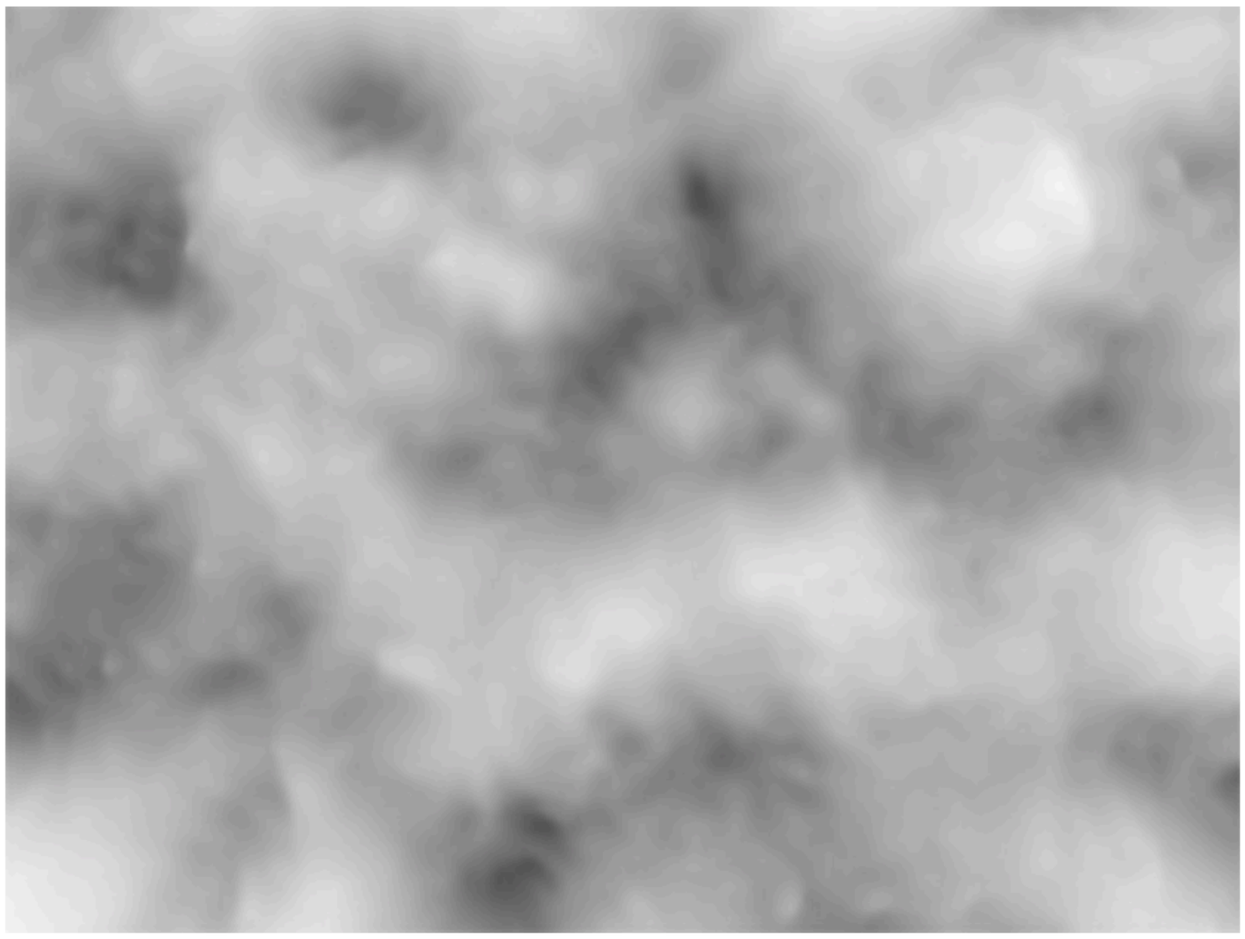


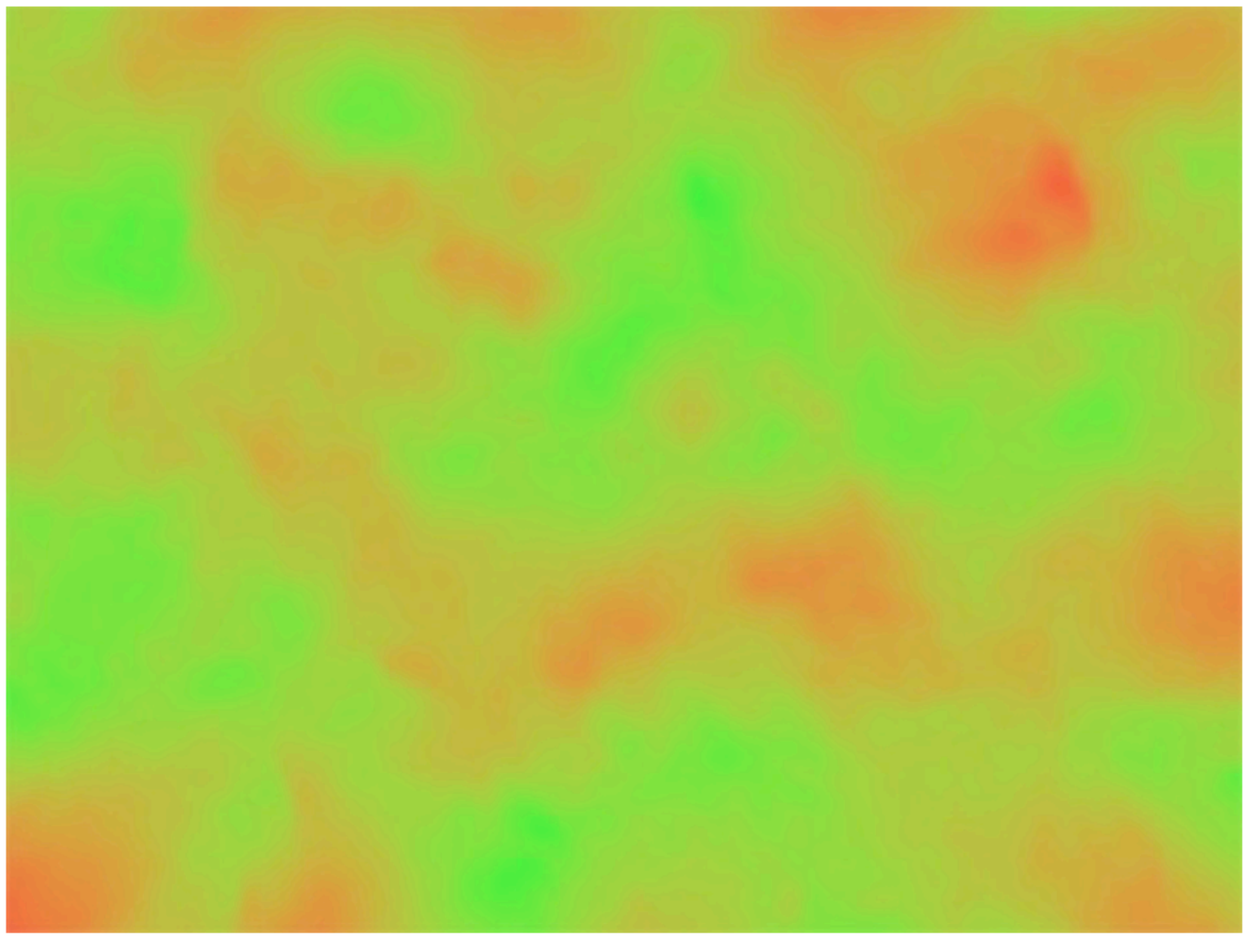




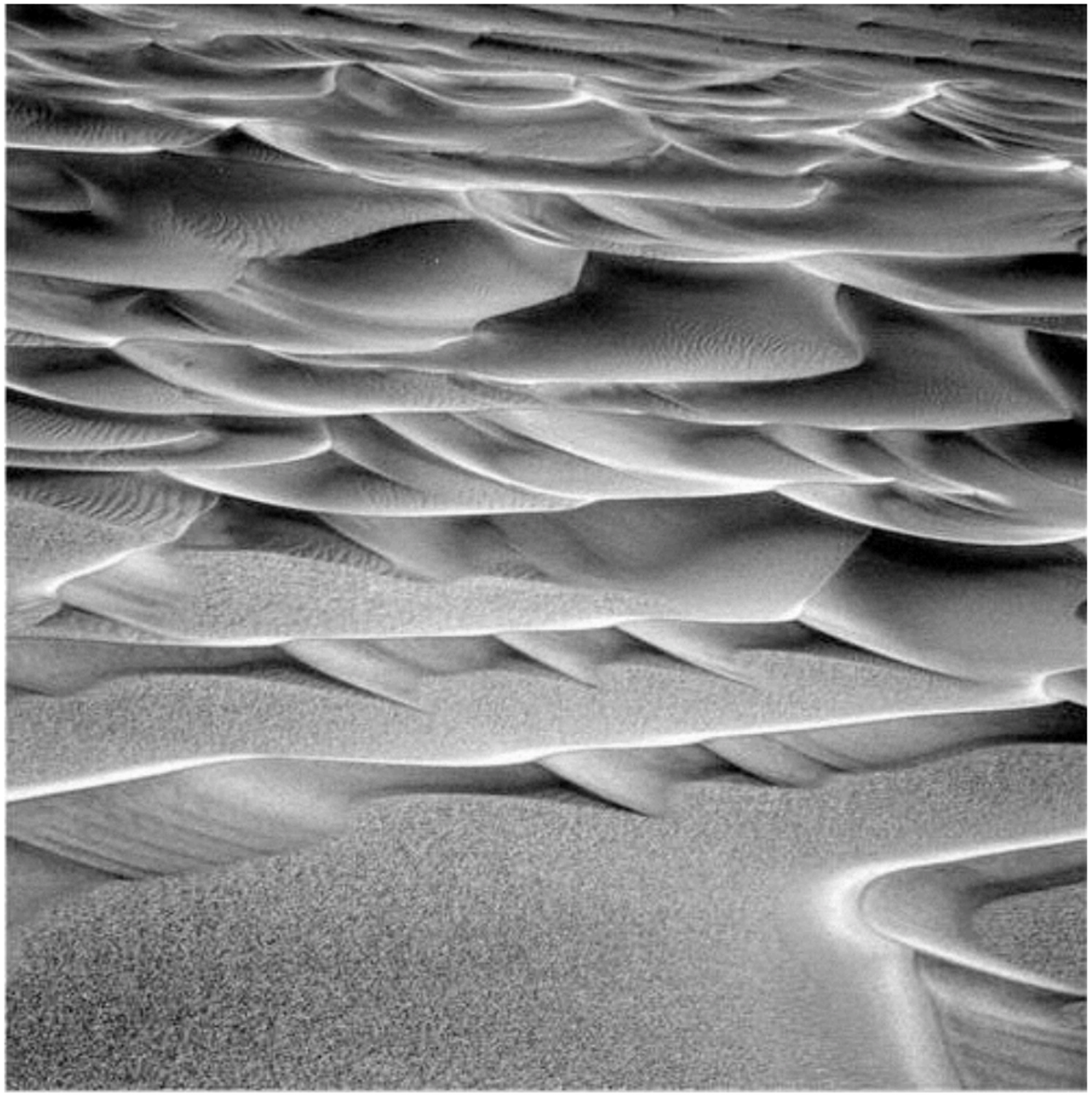
Edward H. Adelson

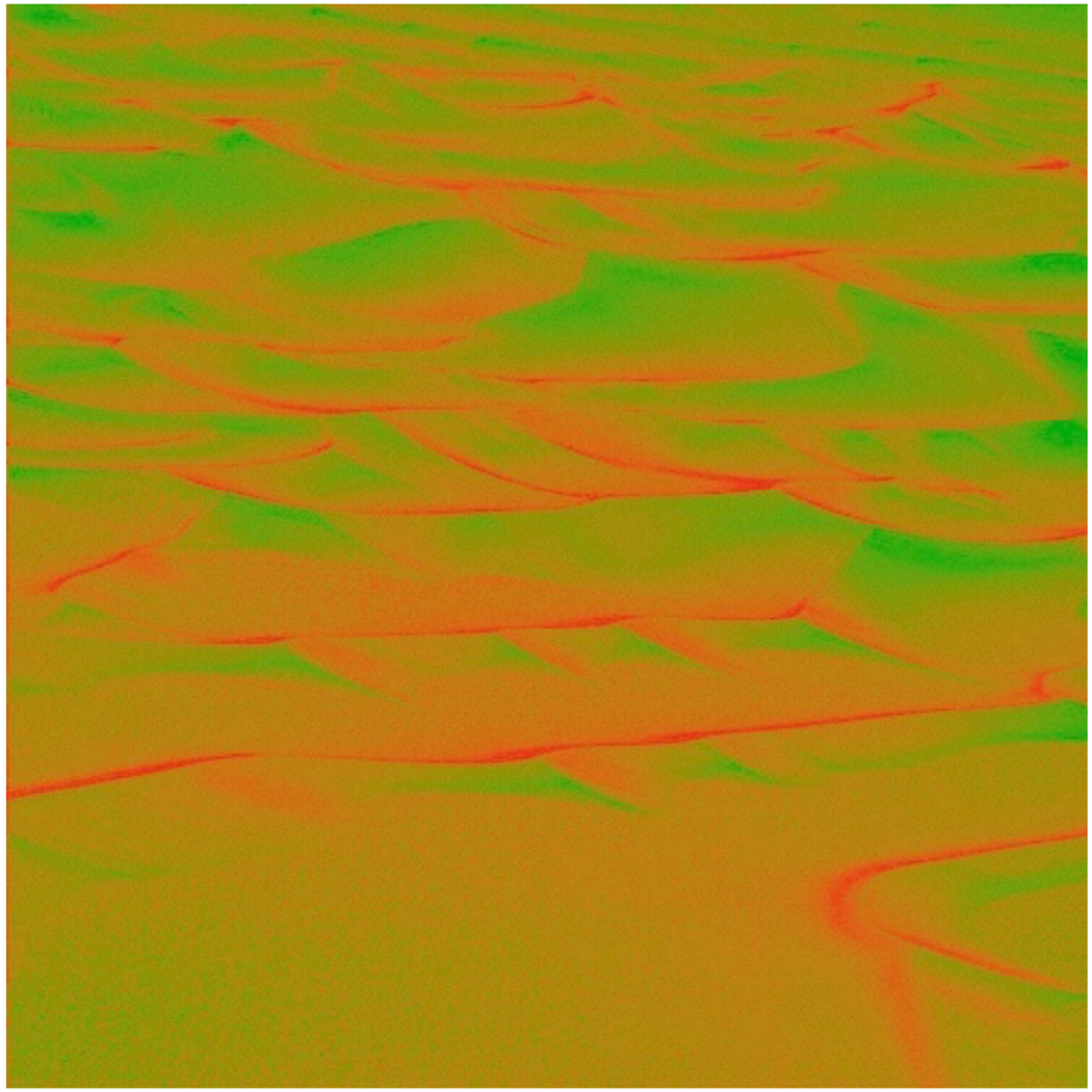

B
A





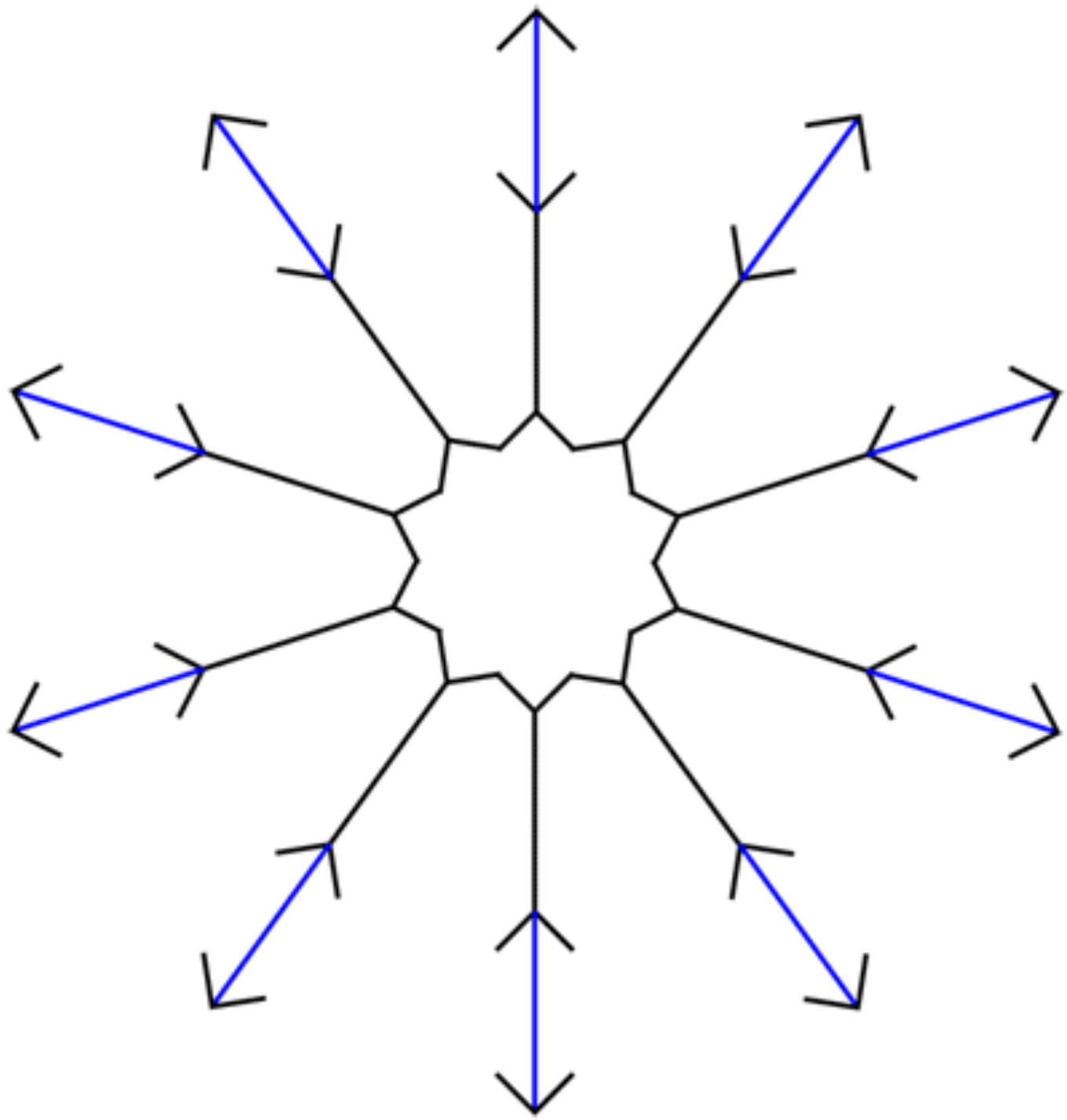
Ware (2008)

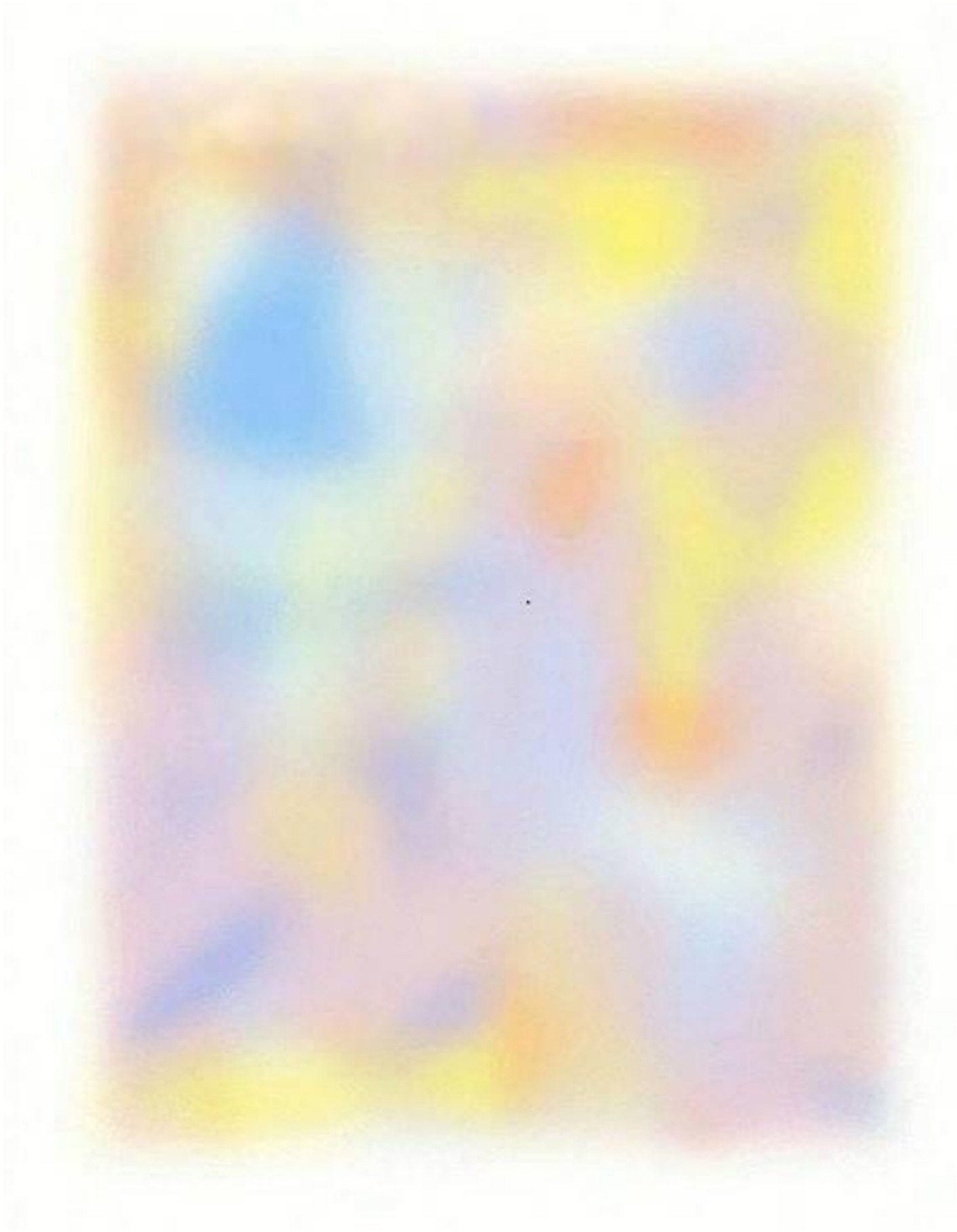


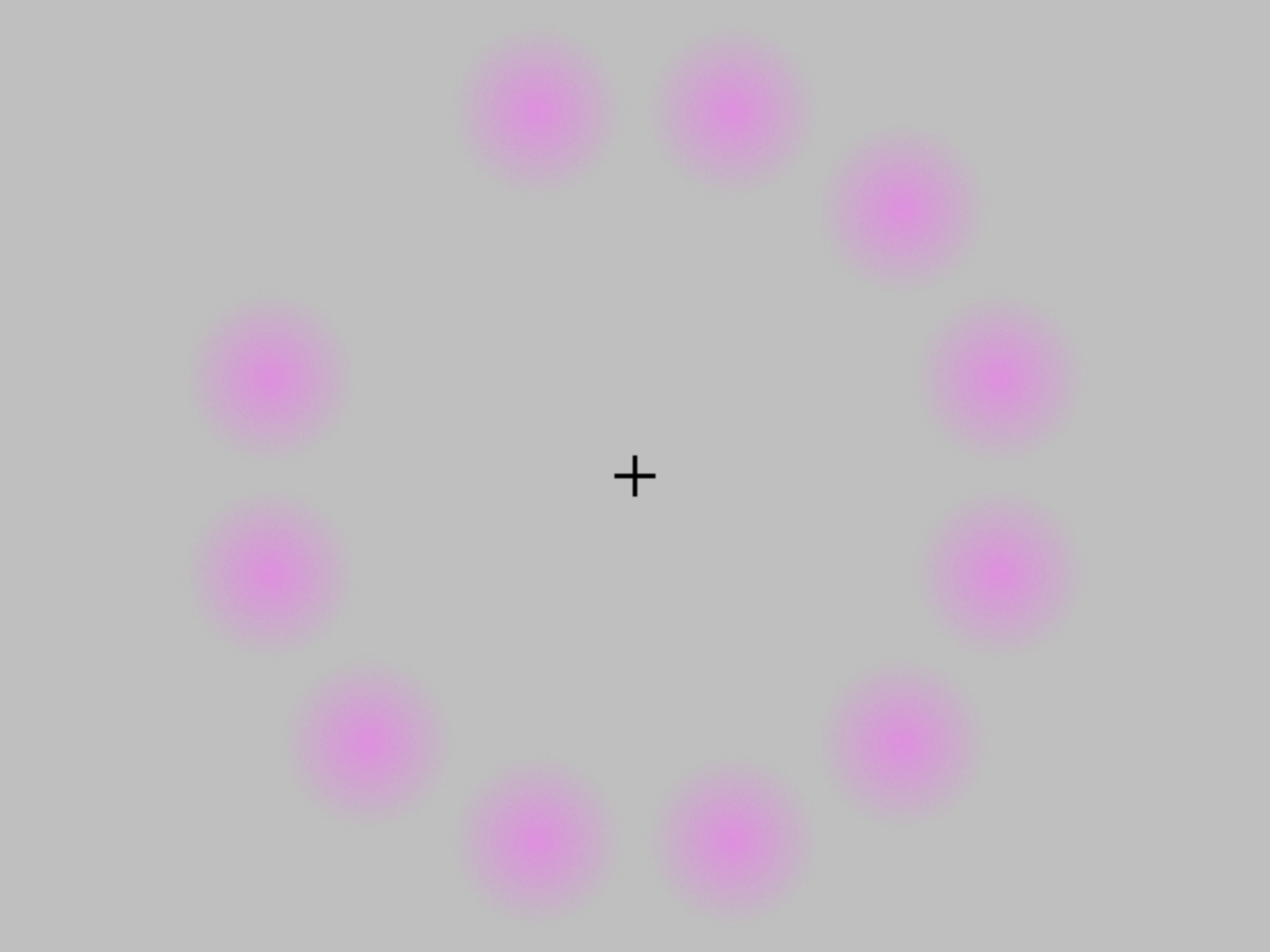


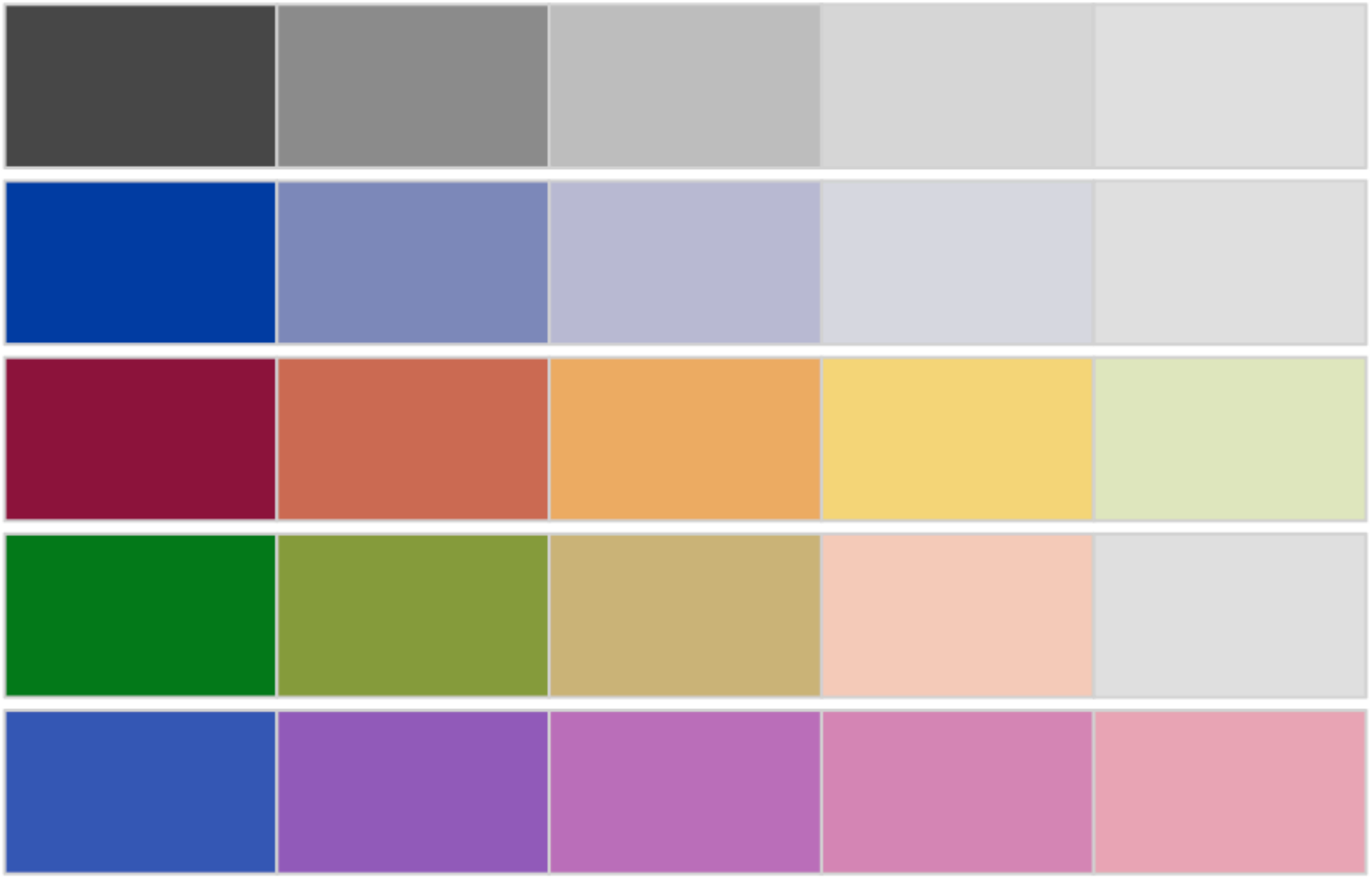
Ware (2008)

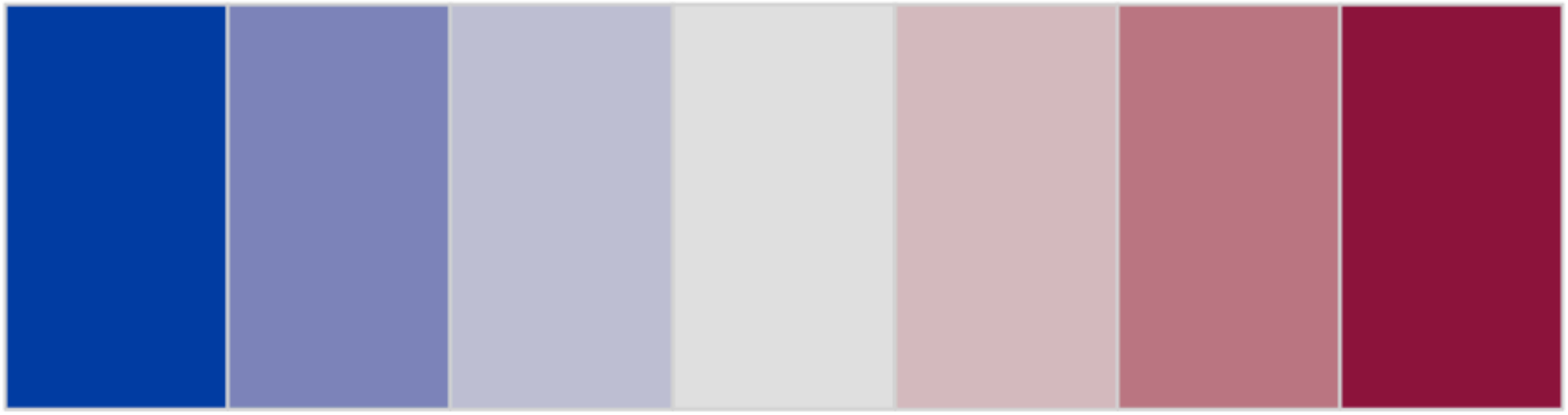
Sarcone's Dynamic Müller-Lyer Illusion



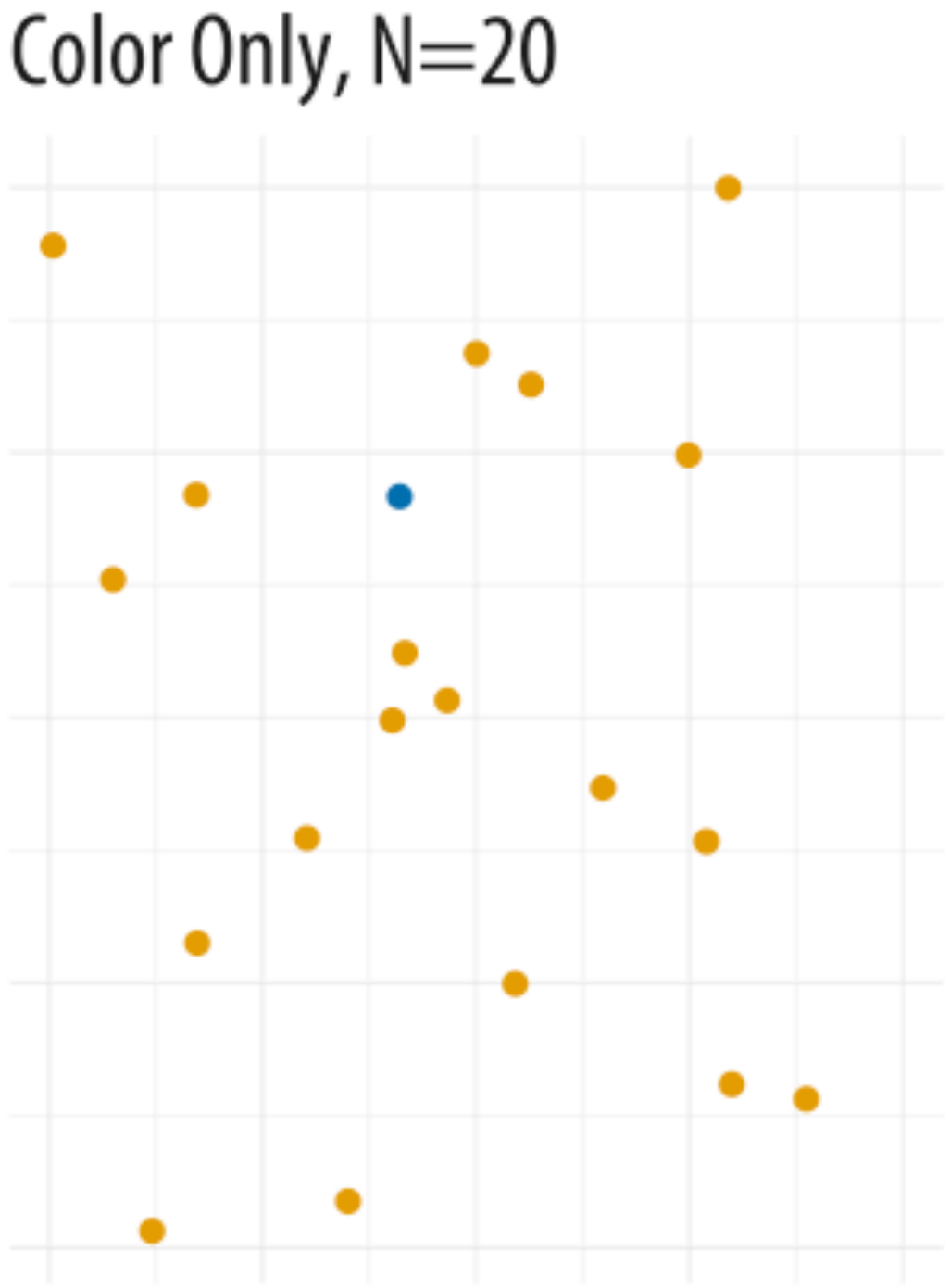


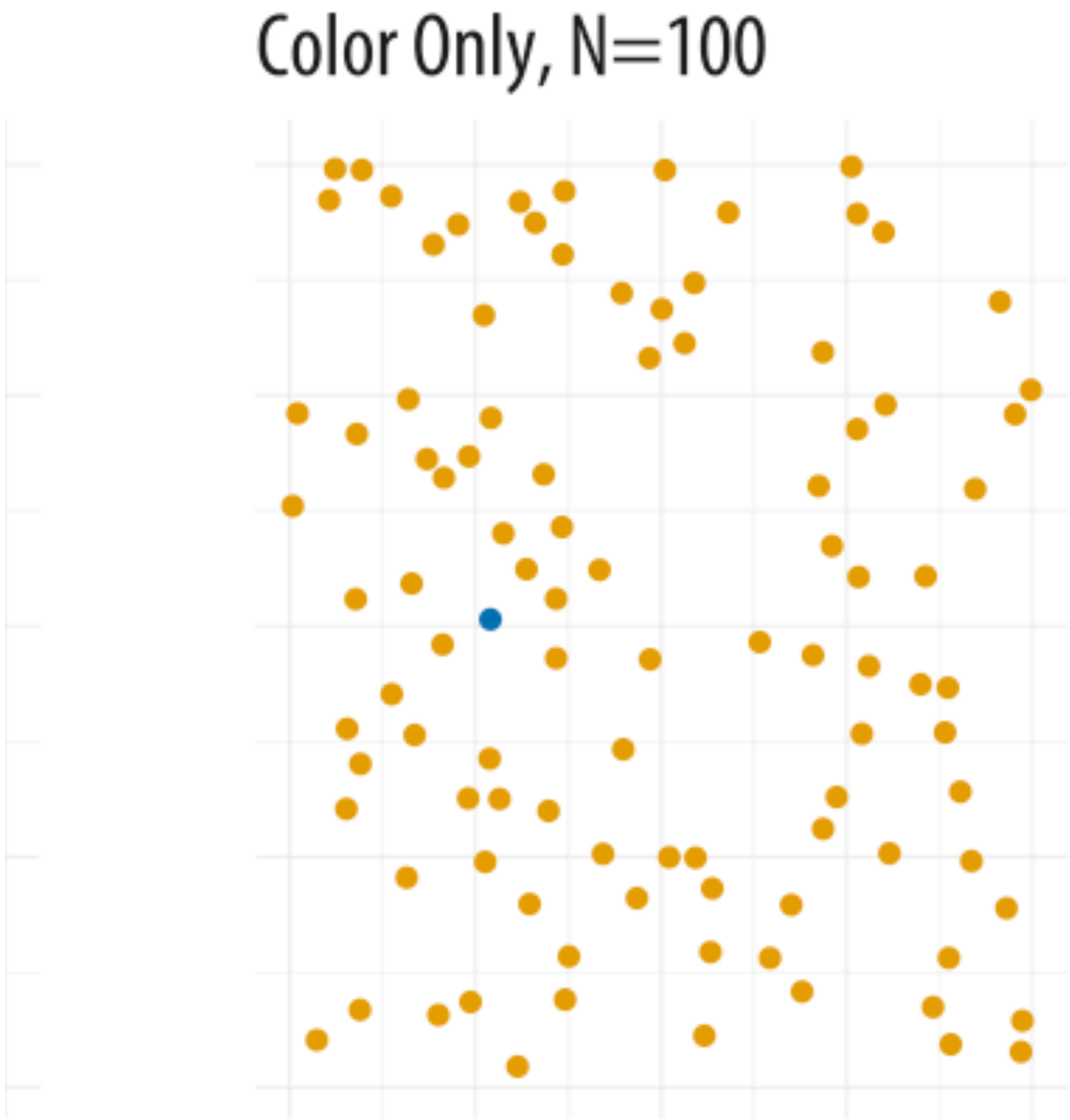


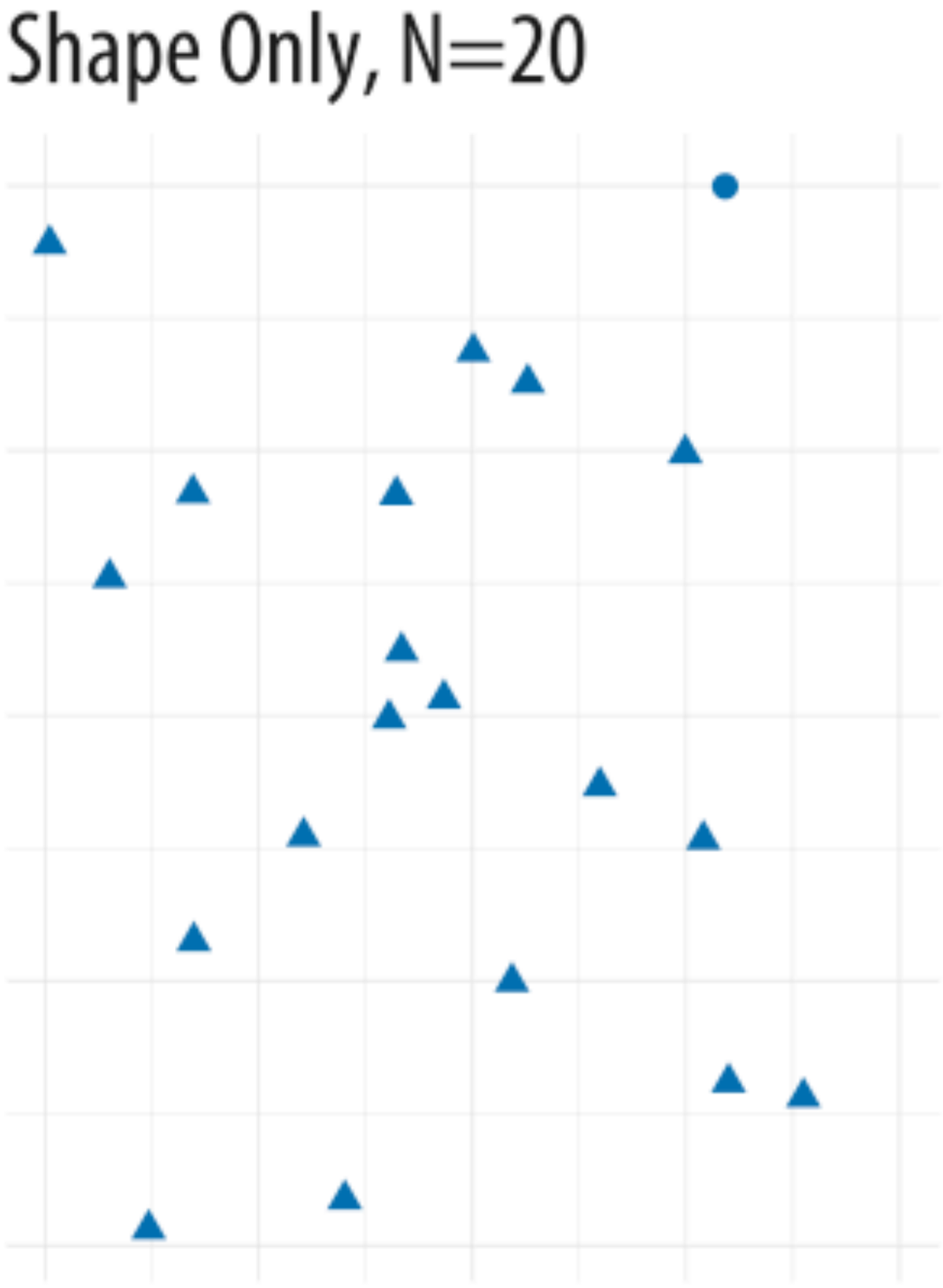


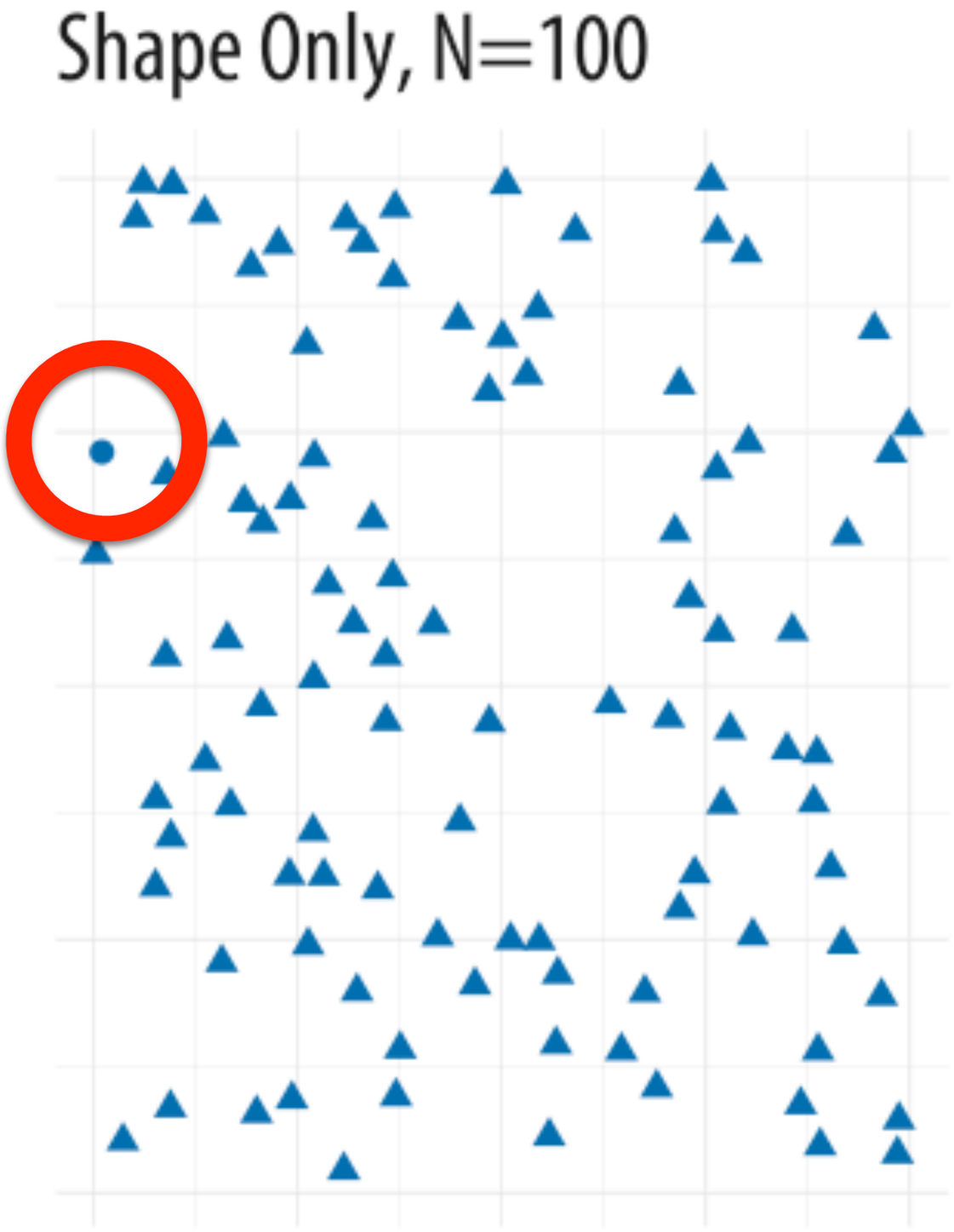


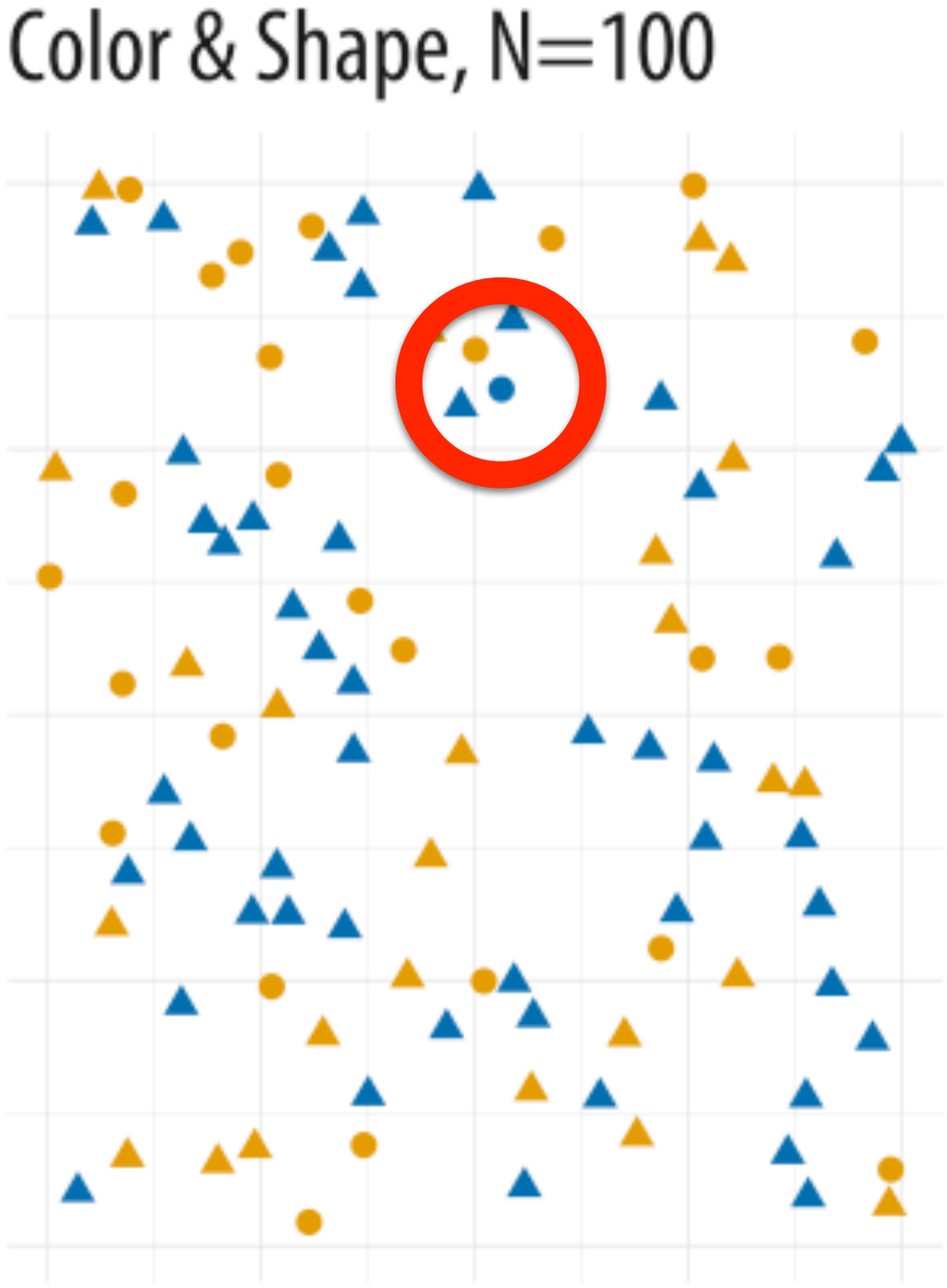
PRE-ATTENTIVE PROCESSING

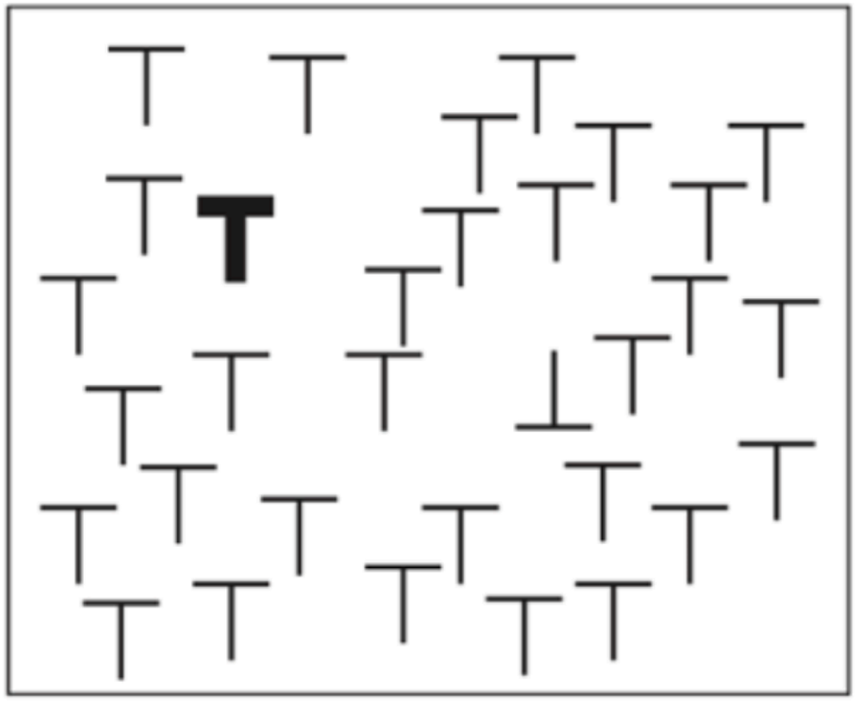

Color Only, N=20

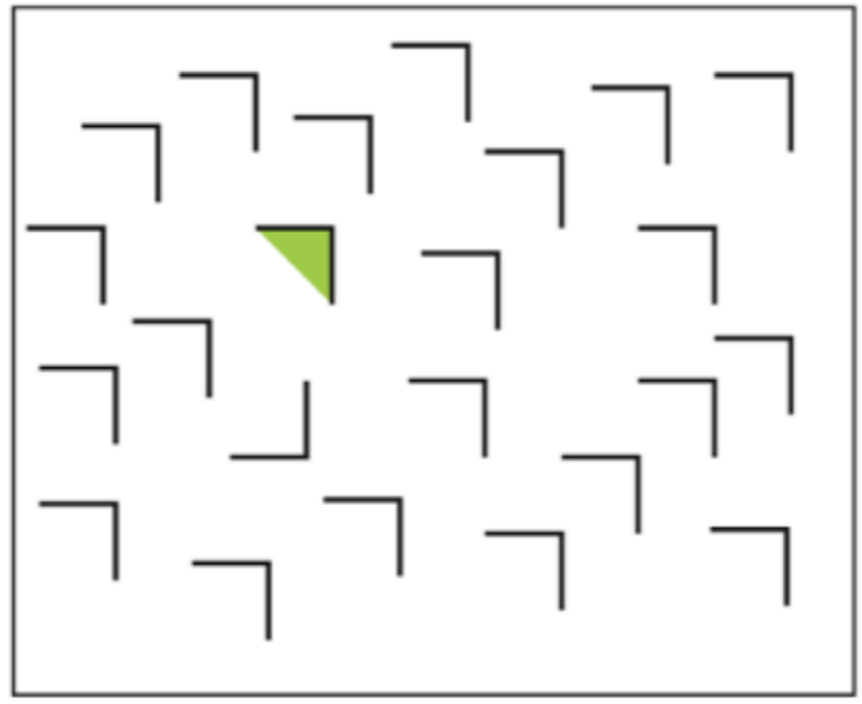

Color Only, N=100

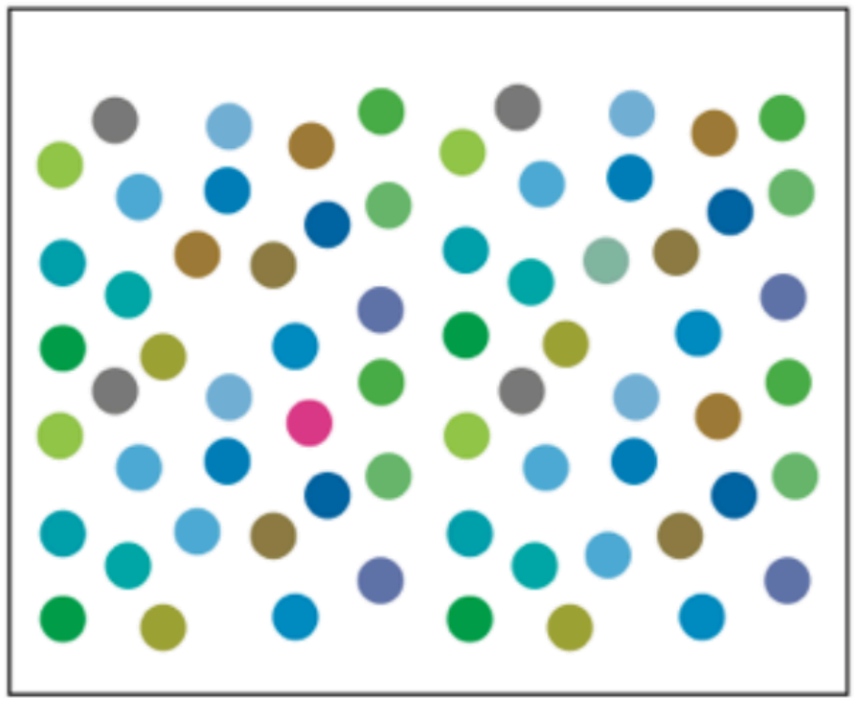

Shape Only, N=20

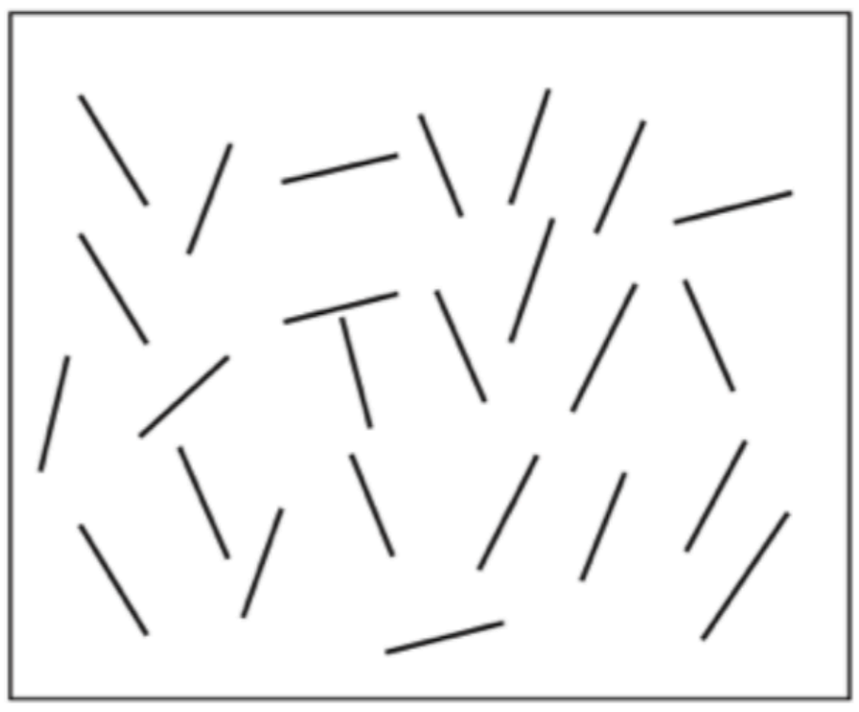

Shape Only, N=100


Color & Shape, N=100







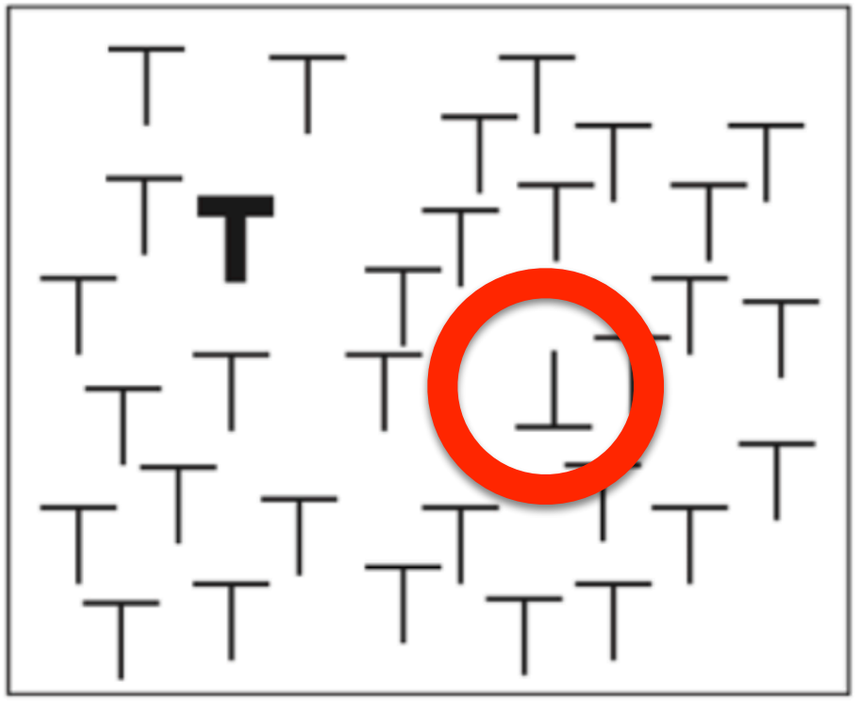


Ware (2008)

difficult

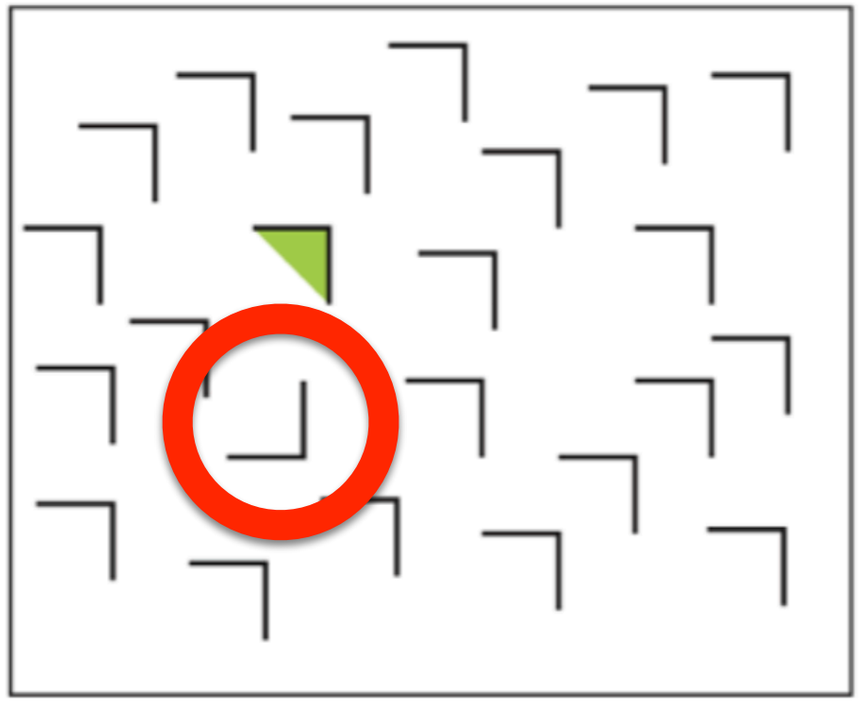
T

easy



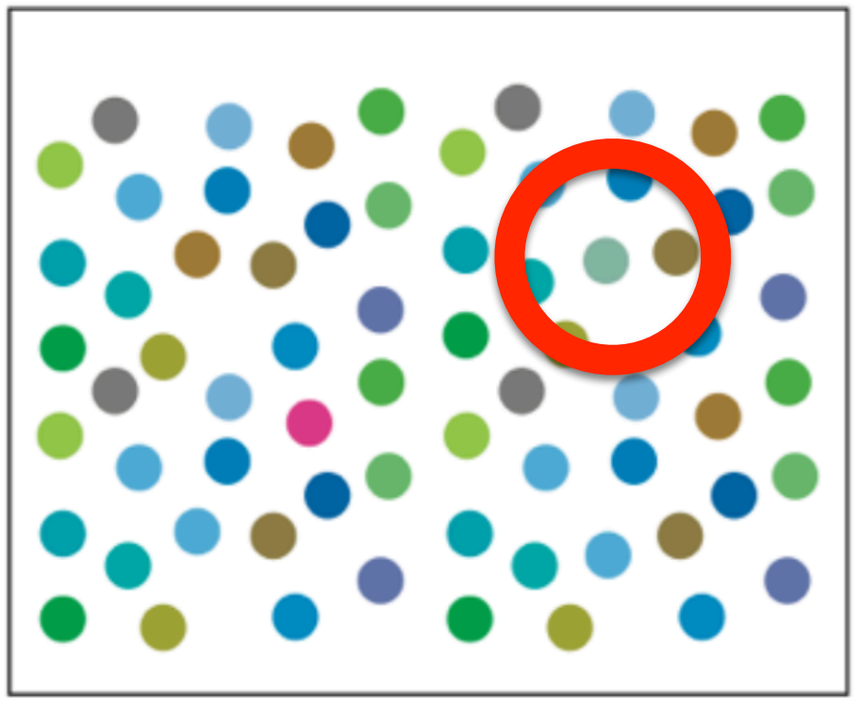
difficult

easy

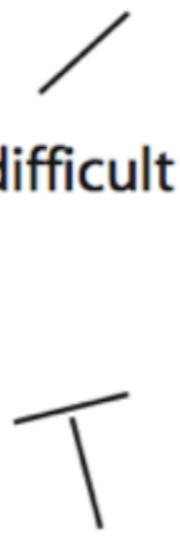


difficult

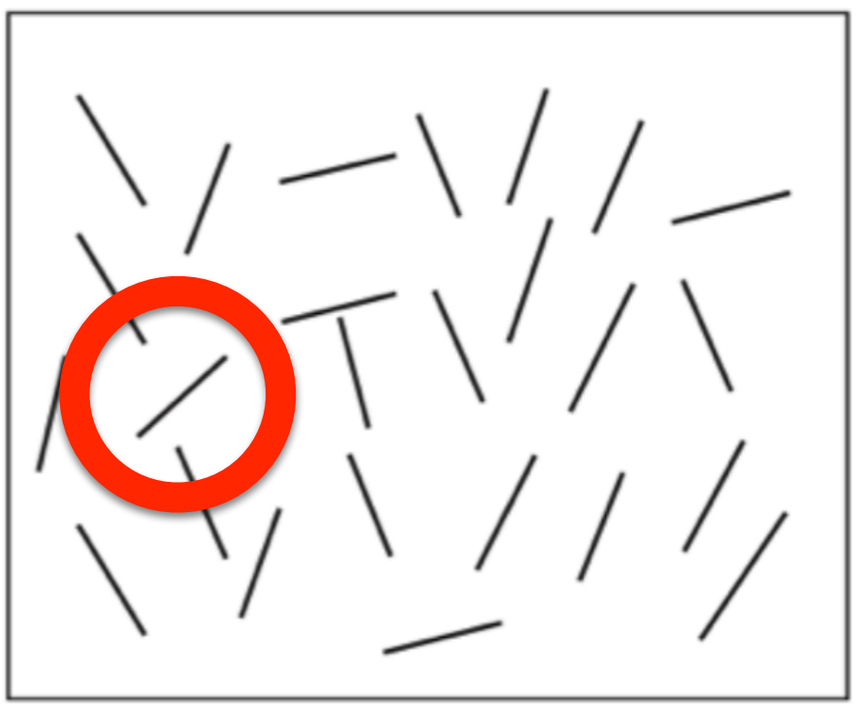
easy



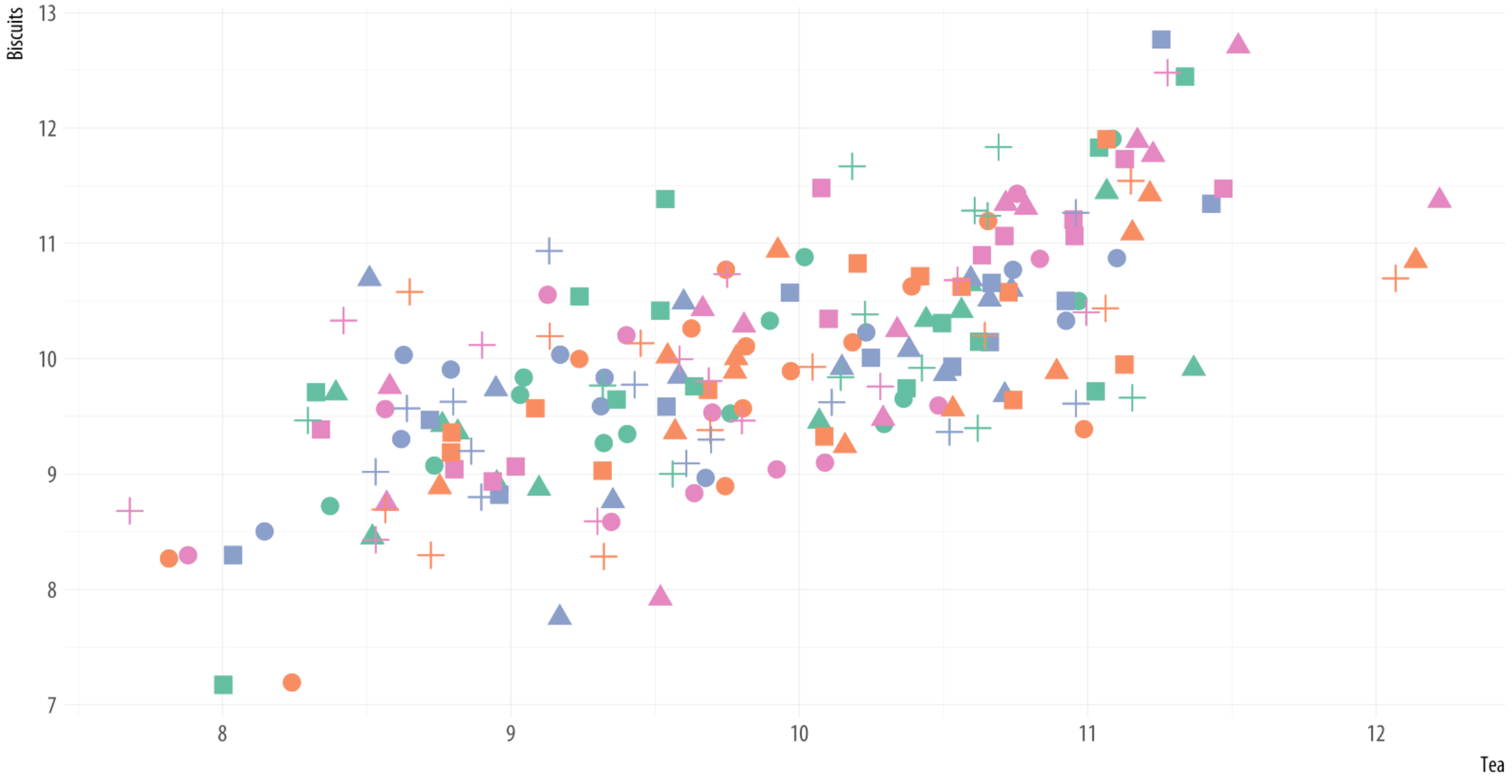
difficult

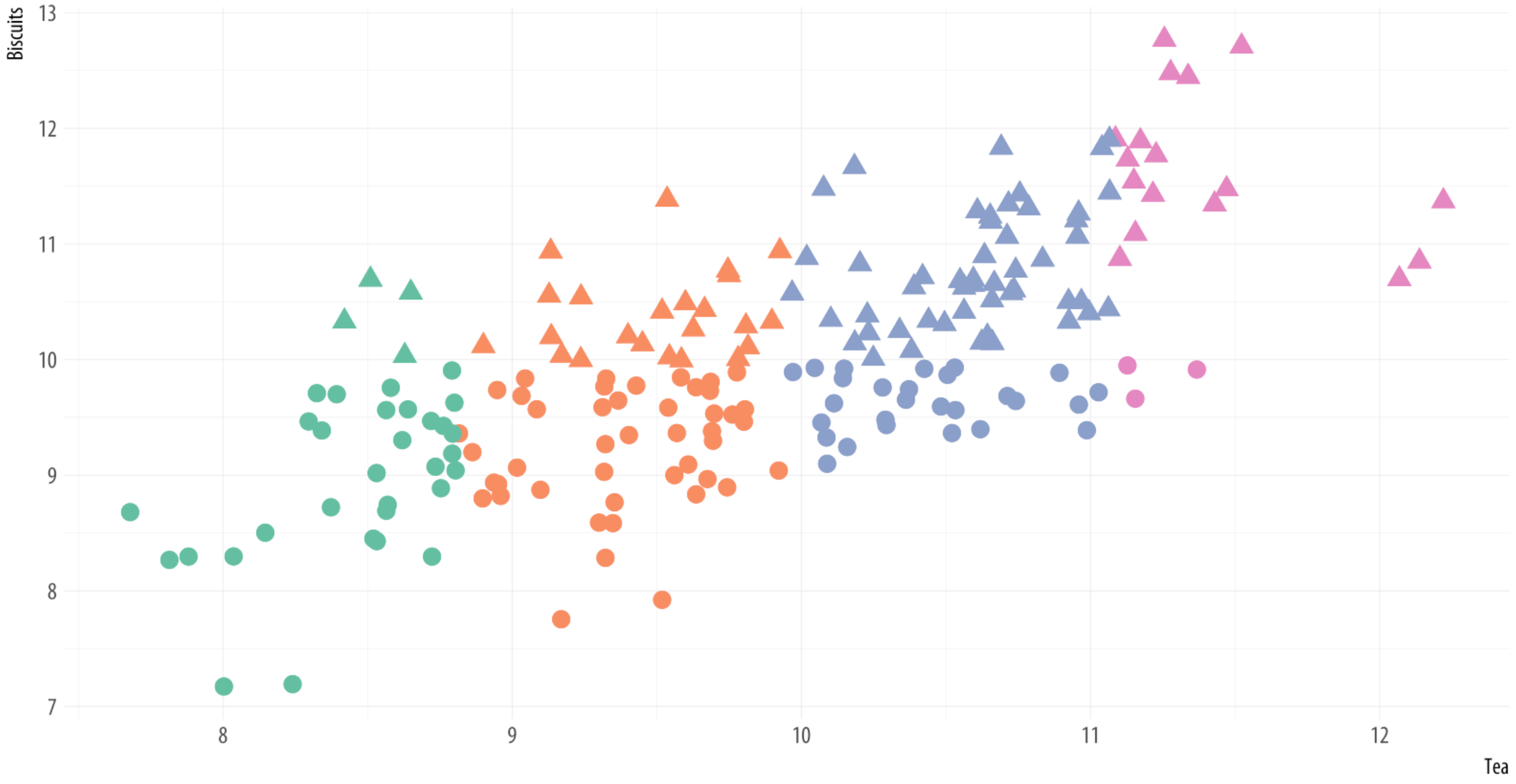


easy

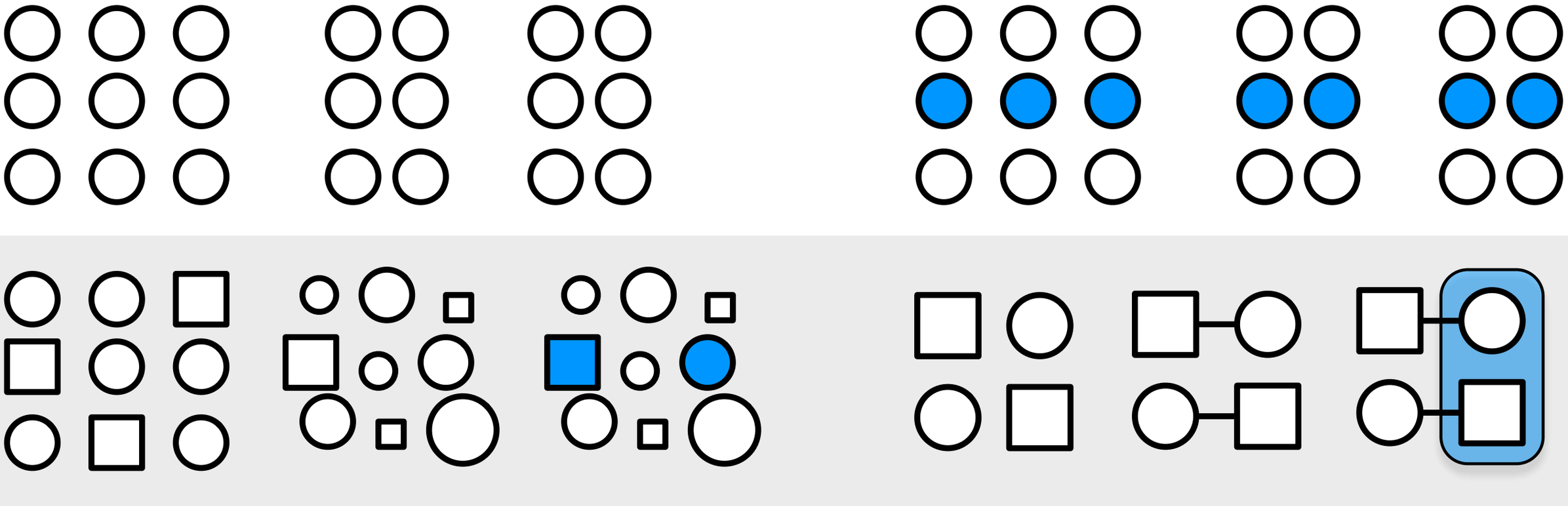


Ware (2008)

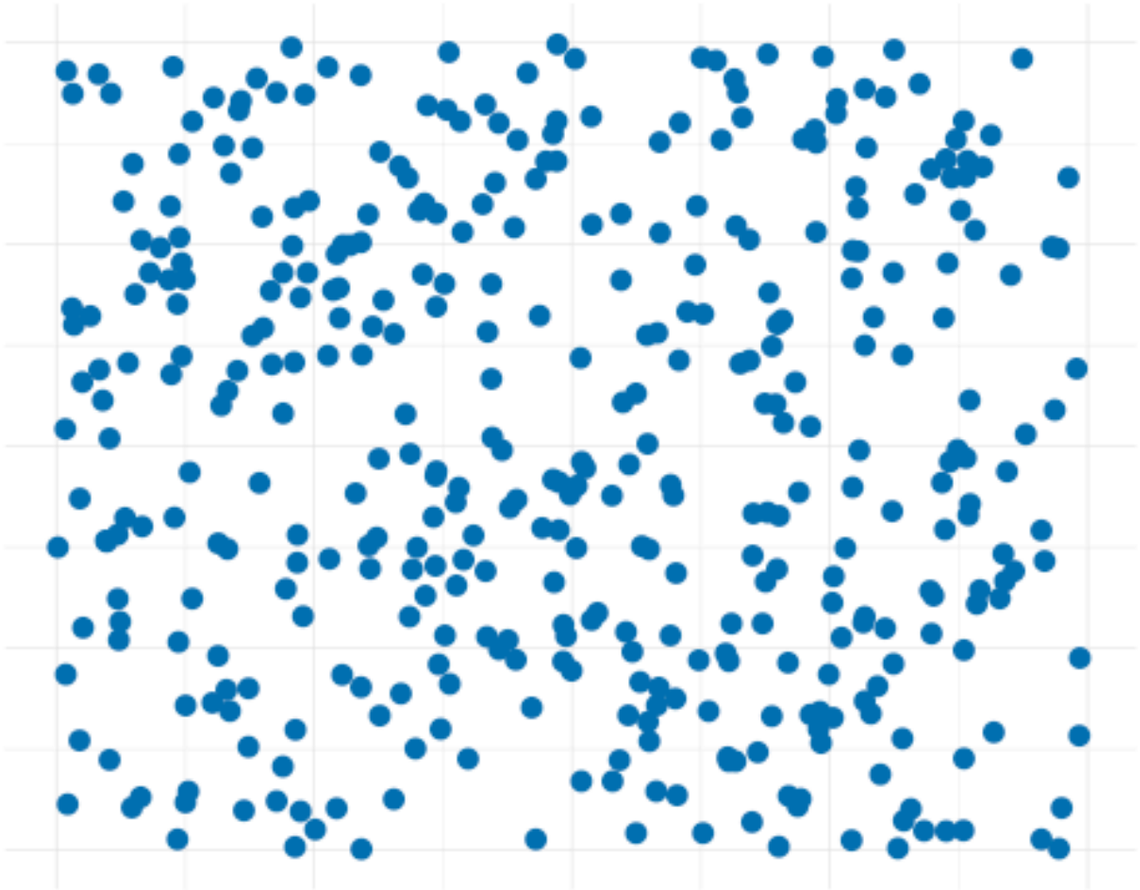

Biscuits
13
12
11
10
9
8
7
8
9
10
11
12
Tea


Biscuits
13
12
11
10
9
8
7
8
9
10
11
12
Tea

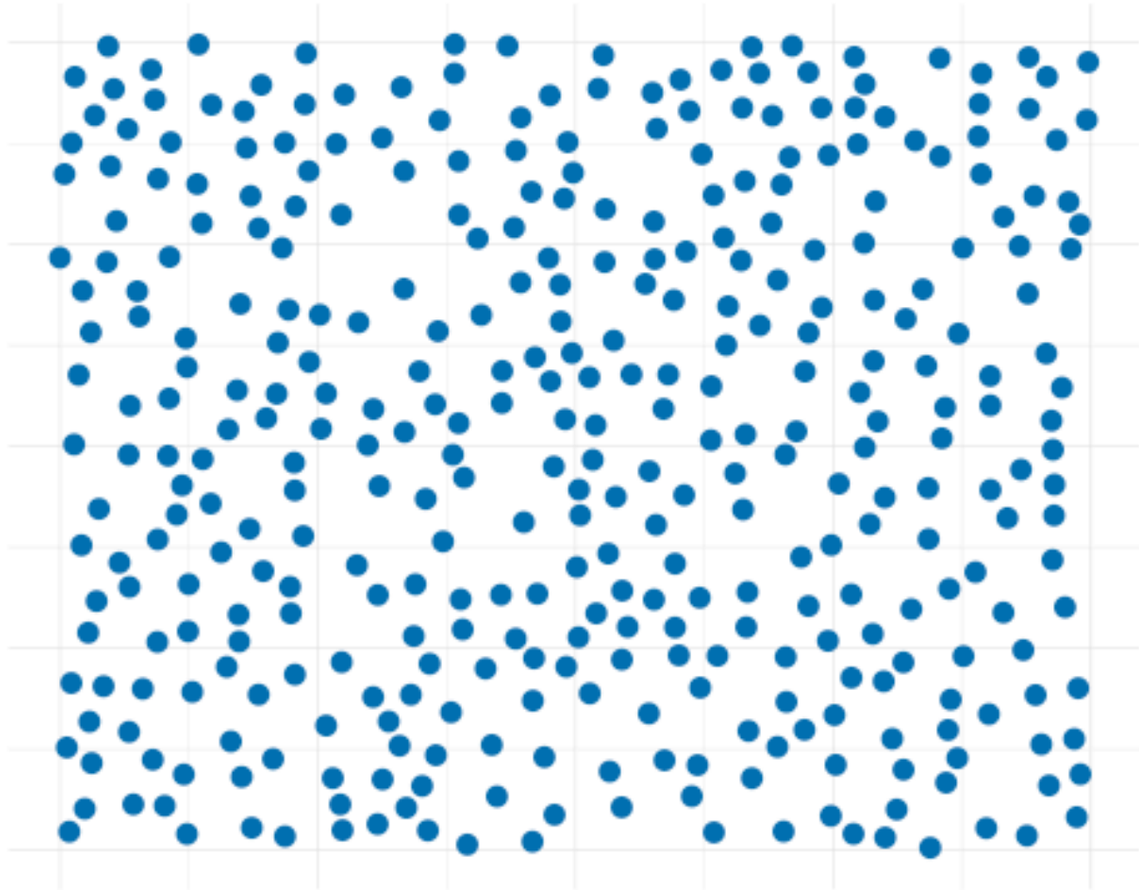
GESTALT INFERENCES



Poisson



Matérn



VISUAL TASKS IN DECODING GRAPHS

GRAPH ELEMENTS ENCODE QUANTITIES

VIEWERS MUST THEN DECODE THEM

QUALITIES & QUANTITIES

nominal
ordinal
interval
ratio

Stevens (1946)

names
grades
ranks
counted fractions
amounts
balances

Mosteller & Tukey (1977)

VISUAL ELEMENTS

position volume color

length shape motion

angle gradient

area pattern

QUALITIES & QUANTITIES

Mapping

Decoding

VISUAL ELEMENTS

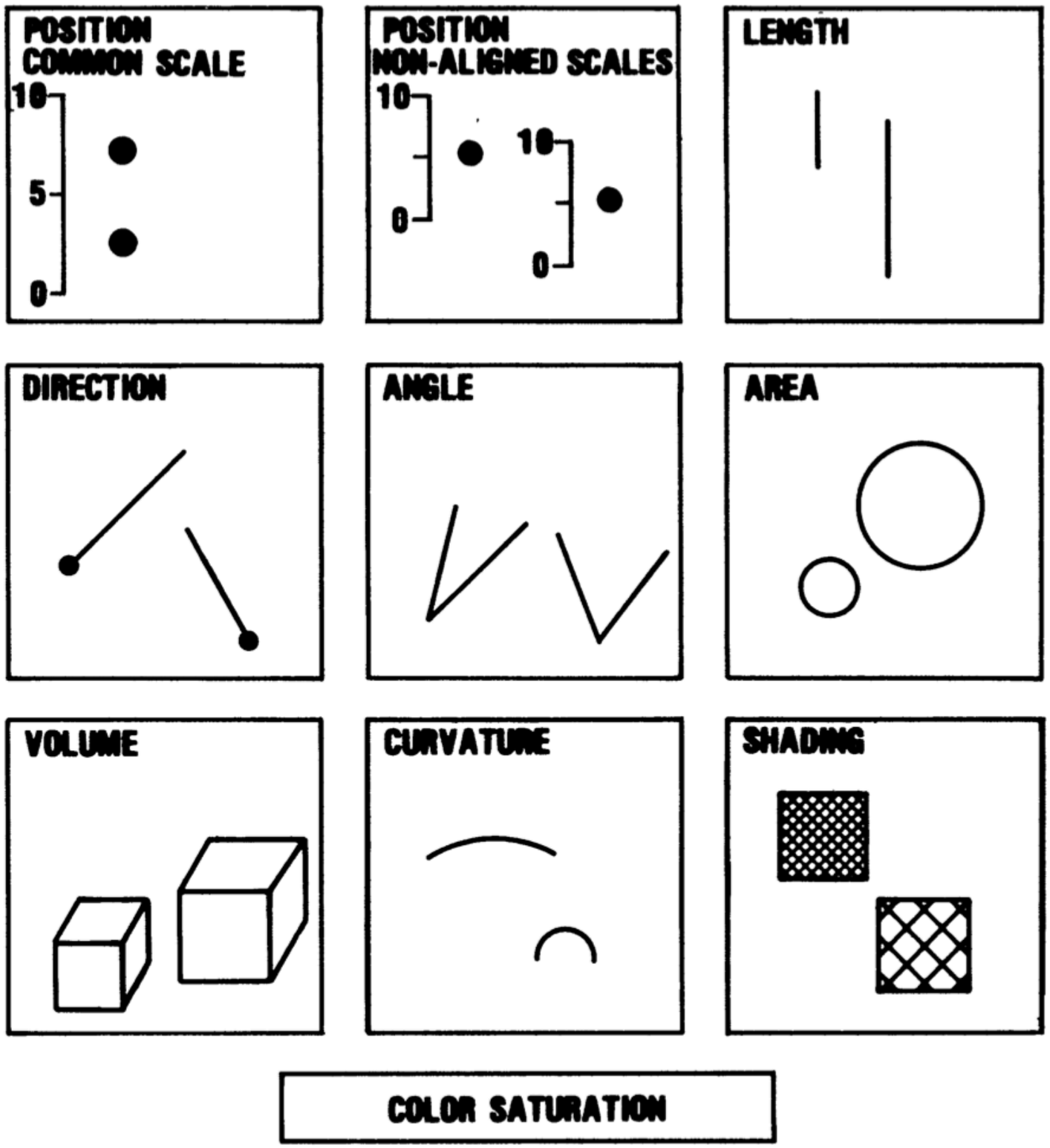


Figure 1. Elementary perceptual tasks.

See: Cleveland & McGill (1984)

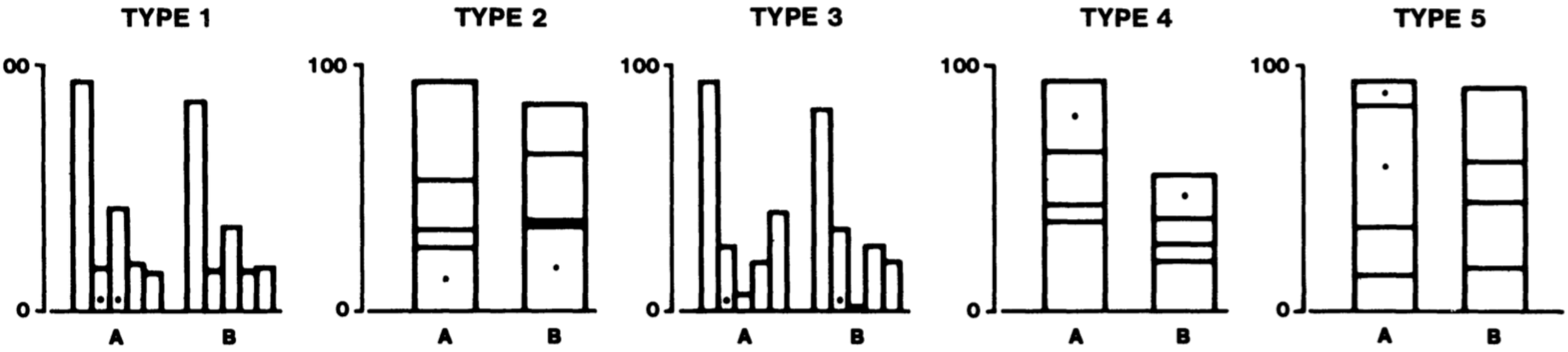
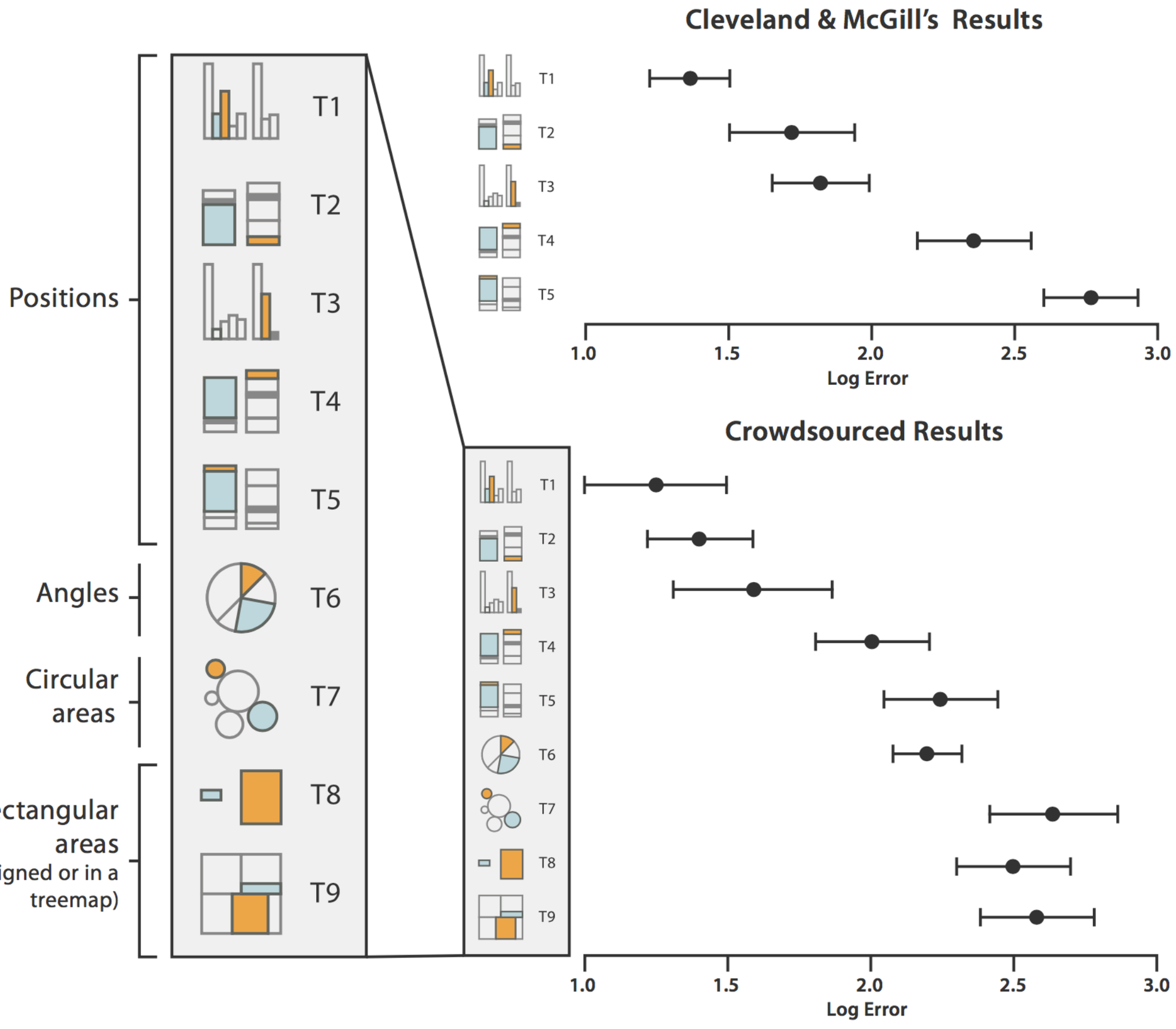


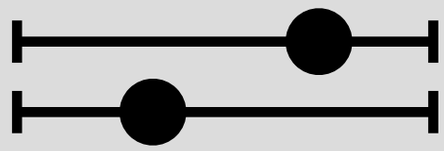
Figure 4. Graphs from position–length experiment.



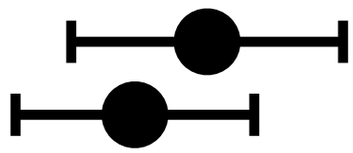
See: Heer & Bostock (2010)

ENCODINGS or MAPPINGS FOR DATA

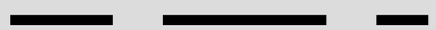
Position on a common scale



Position on unaligned scale

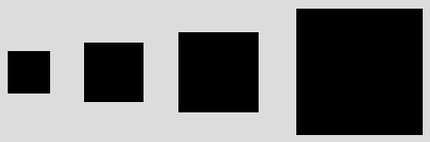


Length (1D as size)



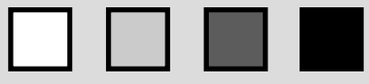
Tilt or Angle

Area (2D as size)

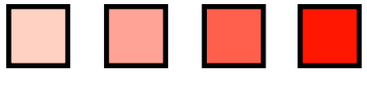


Depth (3D as Position)

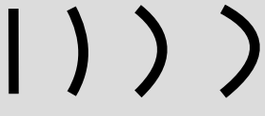
Color luminance [brightness]



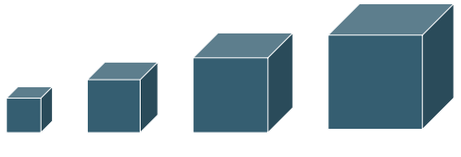
Color saturation [intensity]



Curvature



Volume (3D as size)



Better

Effectiveness

Worse

Spatial Region

Color [hue]

Motion

Shape

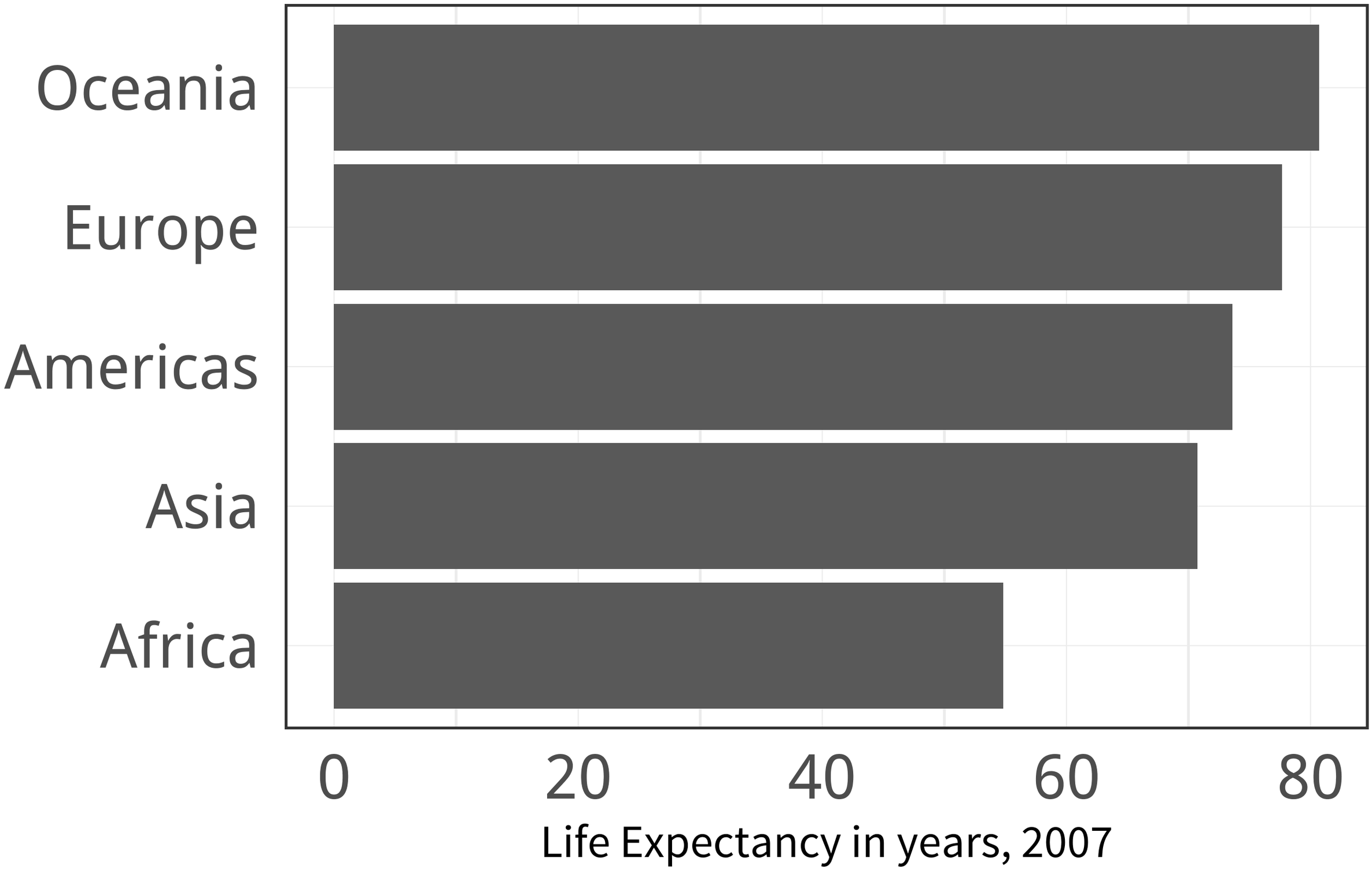
Better

Effectiveness

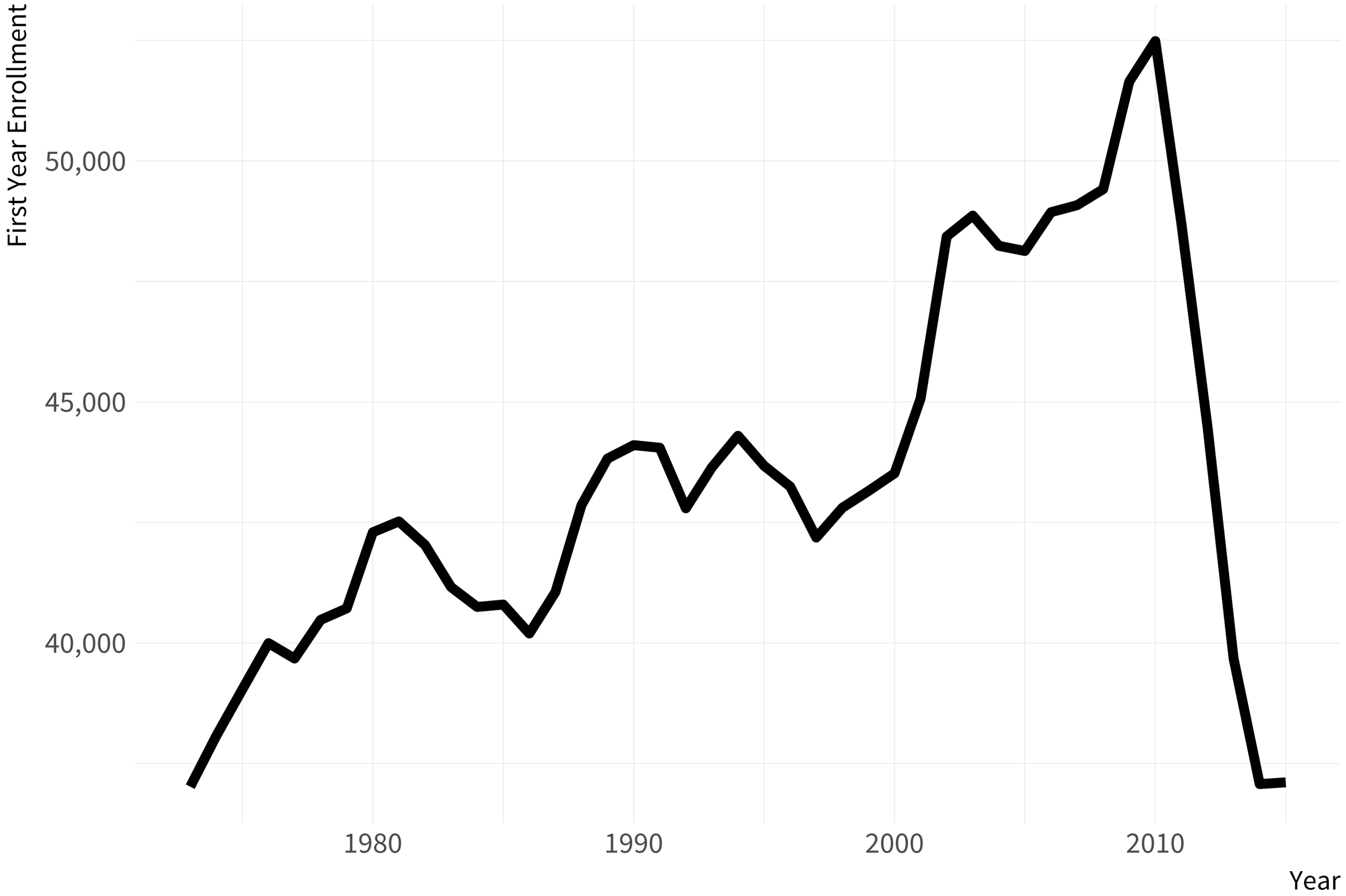
Worse

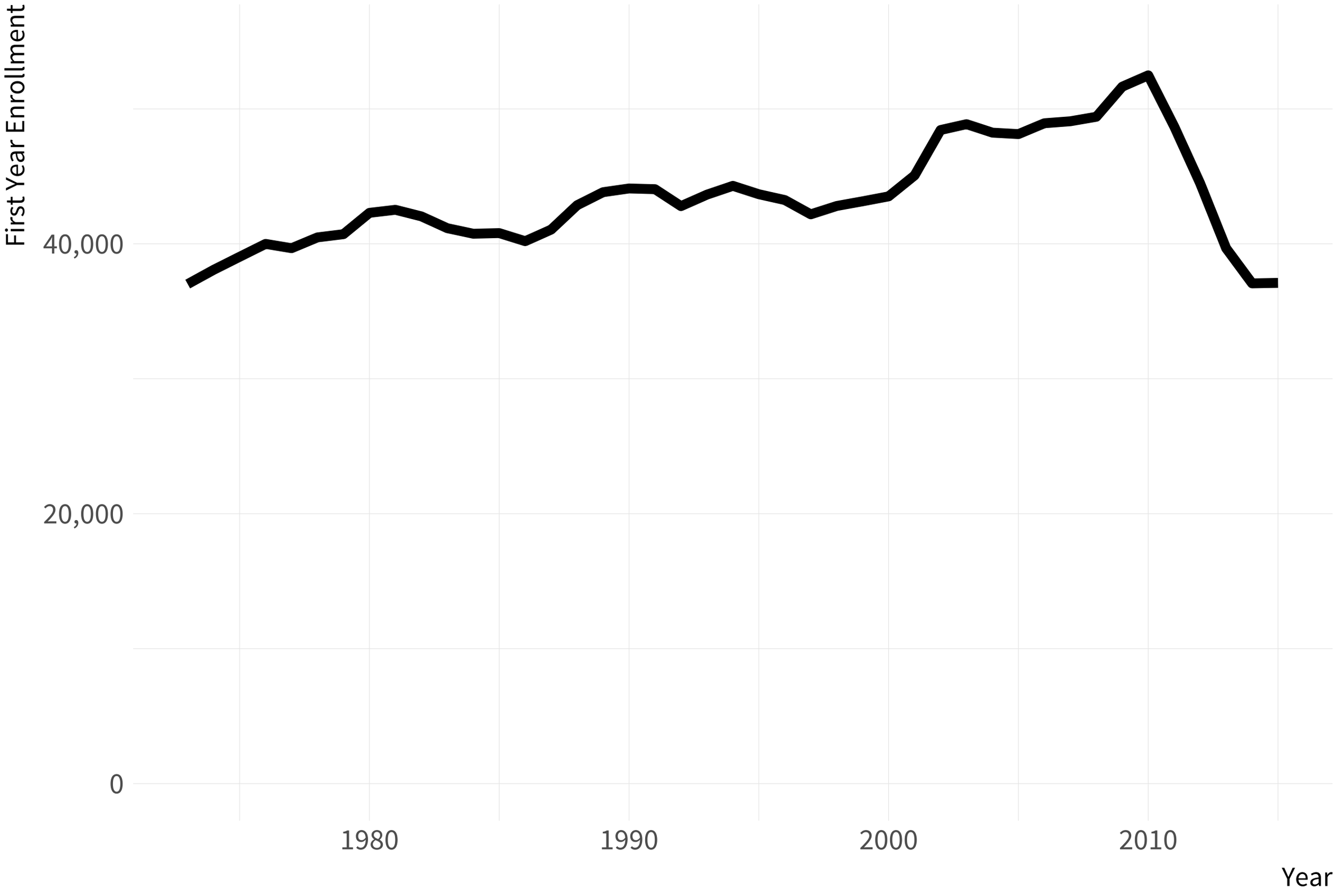
For more see: Munzer (2014)

HONESTY AND GOOD JUDGMENT


Oceania
Europe
Americas
Asia
Africa
0
20
40
60
80
Life Expectancy in years, 2007

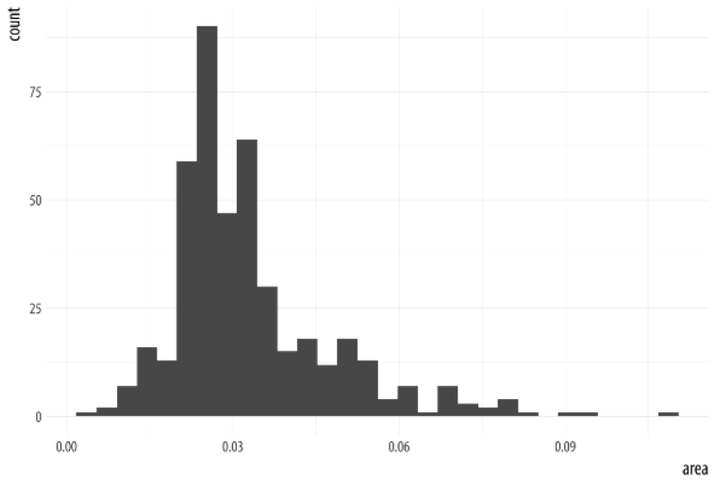

Oceania
Europe
Americas
Asia
Africa
60
70
80
Life Expectancy in years, 2007

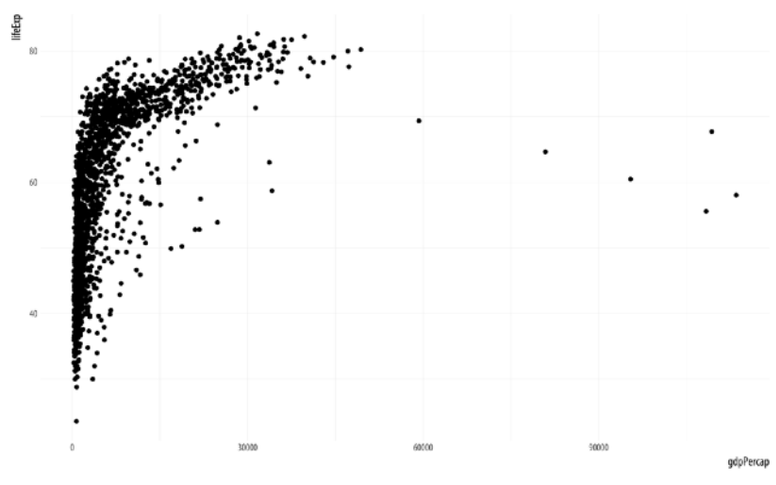

First Year Enrollment
50,000
45,000
40,000
1980
1990
2000
2010
Year

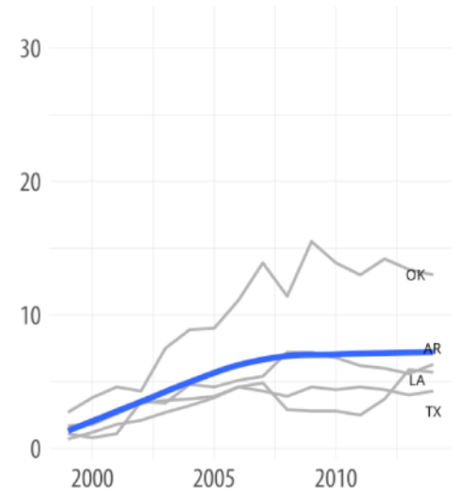

First Year Enrollment
40,000
20,000
0
1980
1990
2000
2010
Year

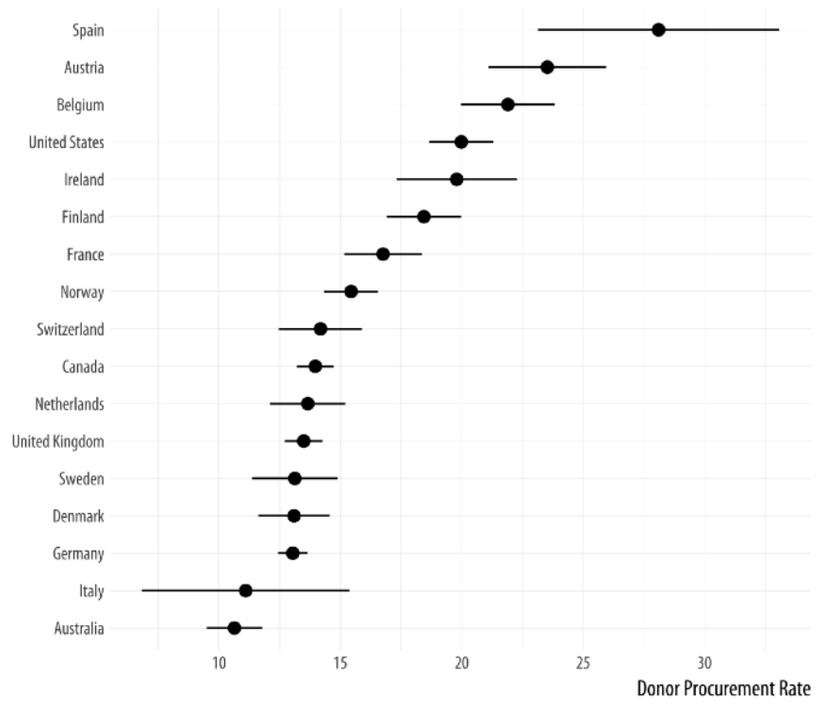
VISUALIZING DATA IN SOCIAL SCIENCE

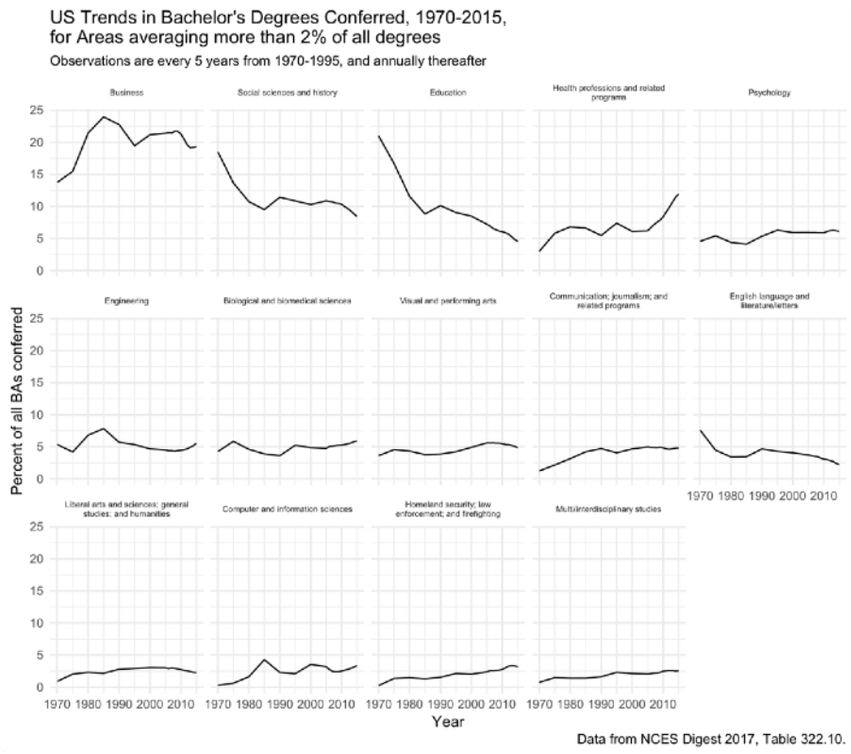
Workhorses


count
75
50
25
0
0.00
0.03
0.06
0.09
area

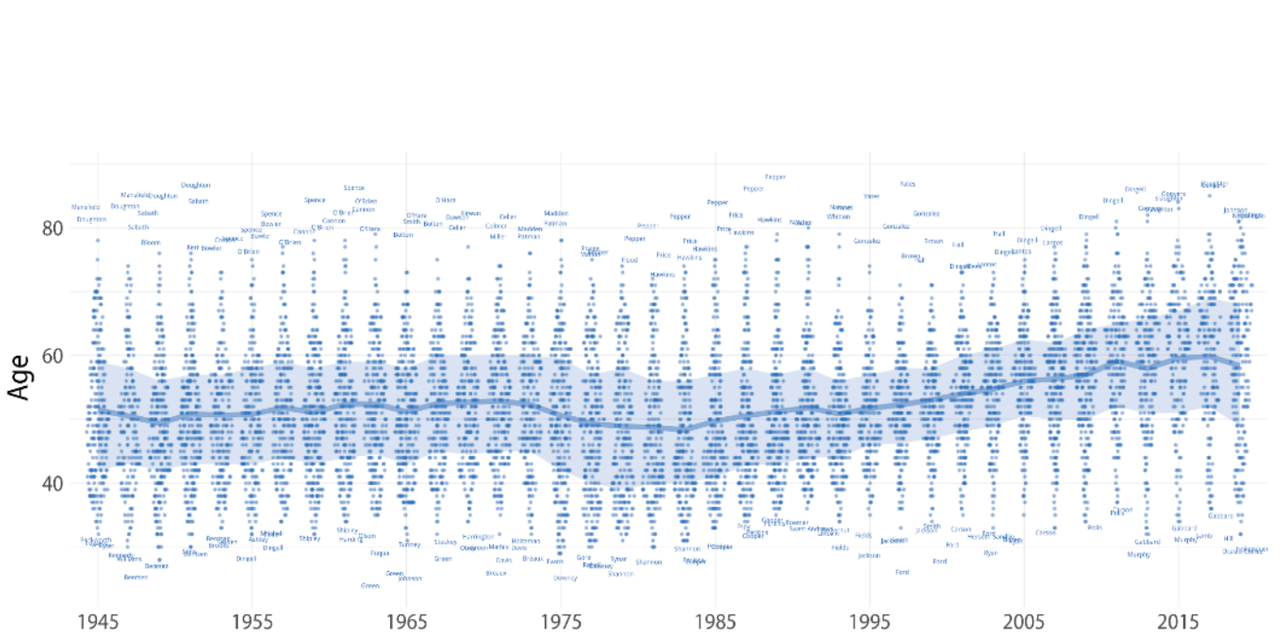

lifeExp
80
60
40
0
30000
60000
90000
gdpPercap

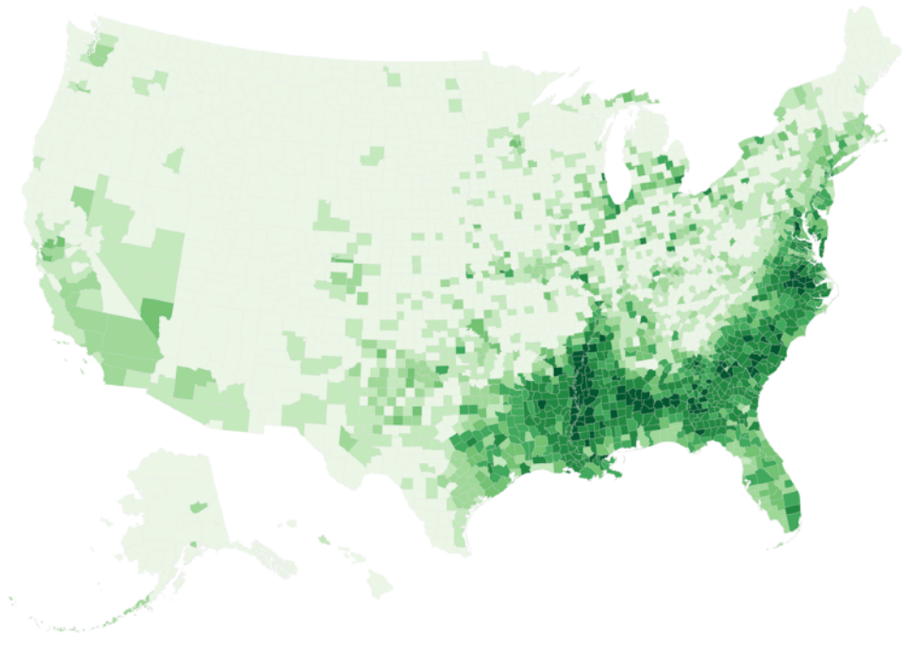

30
20
10
0
OK
AR
LA
TX
2000
2005
2010

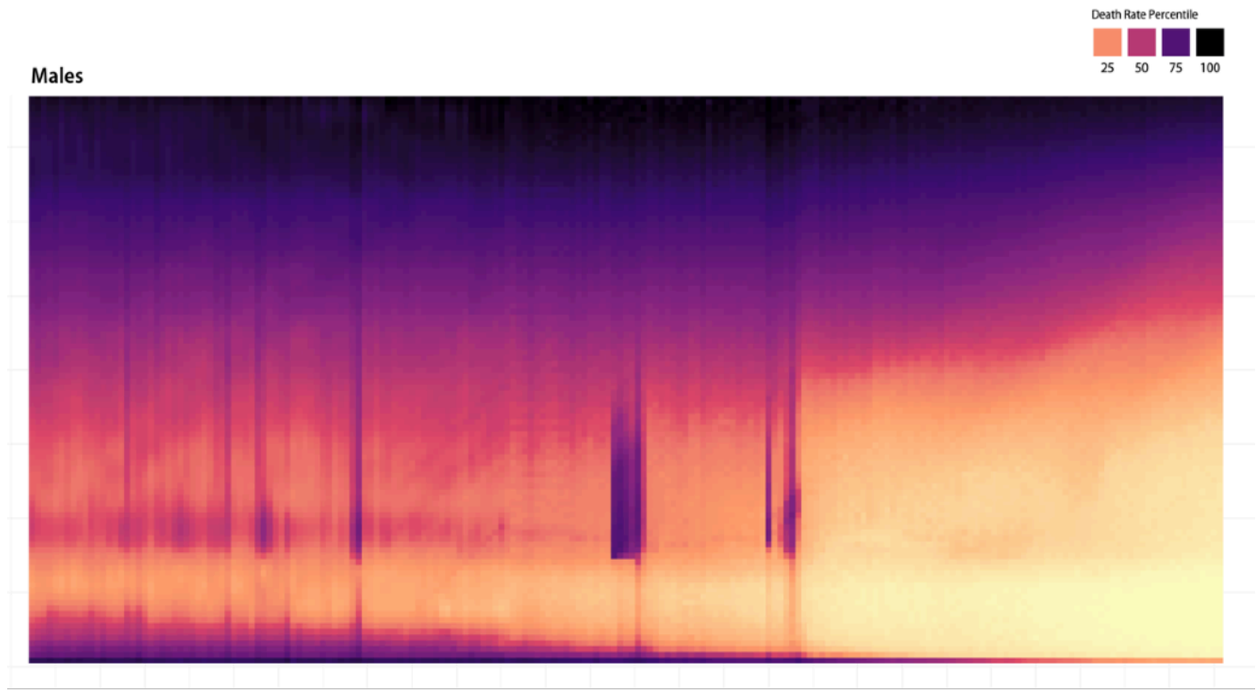

Spain
Austria
Belgium
United States
Ireland
Finland
France
Norway
Switzerland
Canada
Netherlands
United Kingdom
Sweden
Denmark
Germany
Italy
Australia
10
15
20
25
30
Donor Procurement Rate

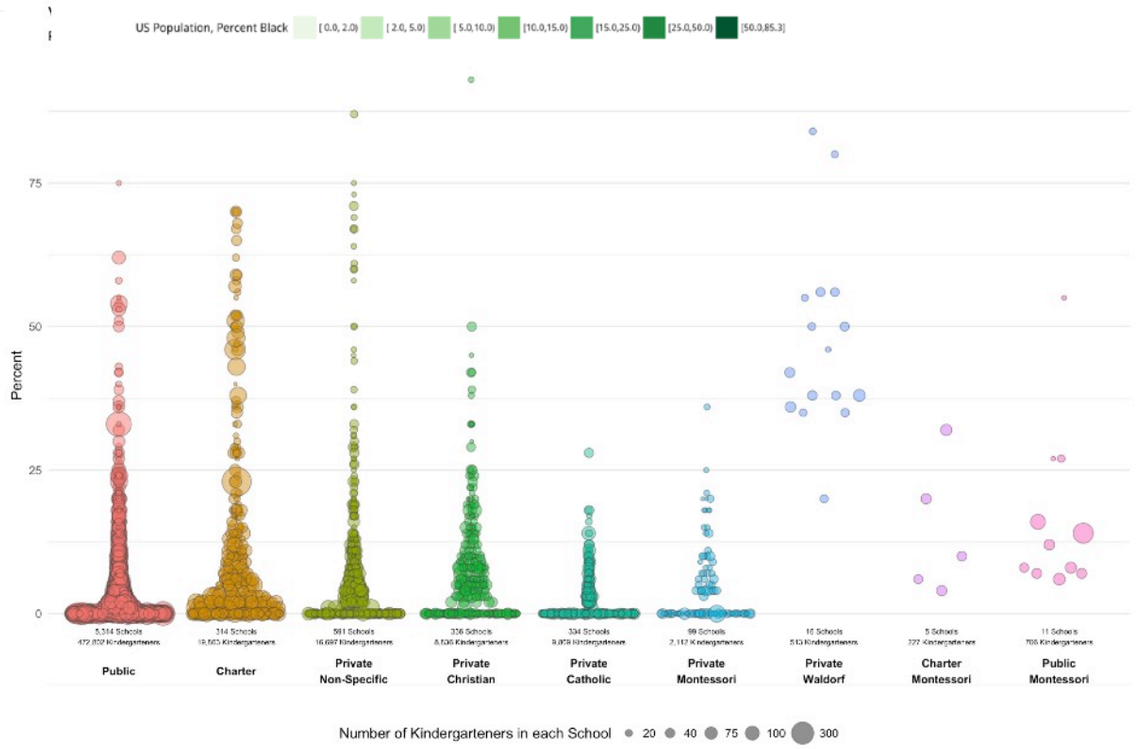

US Trends in Bachelor's Degrees Conferred, 1970-2015,
for Areas averaging more than 2% of all degrees
Observations are every 5 years from 1970-1995, and annually thereafter
Business
Social sciences and history
Education
Health professions and related programs
Psychology
Engineering
Biological and biomedical sciences
Visual and performing arts
Communication, journalism, and related programs
English language and literature/letters
Liberal arts and sciences, general studies, and humanities
Computer and information sciences
Homeland security, law enforcement, and firefighting
Multi/interdisciplinary studies
Percent of all BAs conferred
Year
1970 1980 1990 2000 2010
Data from NCES Digest 2017, Table 322.10.

Show Ponies

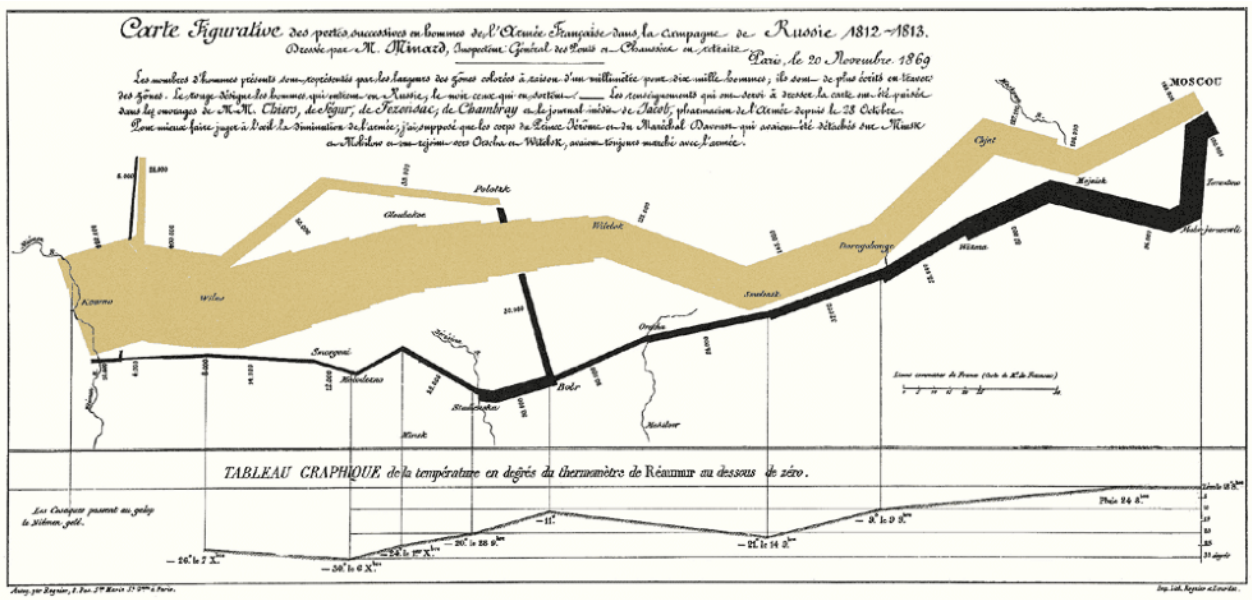

Age
80
60
40
1945
1955
1965
1975
1985
1995
2005
2015

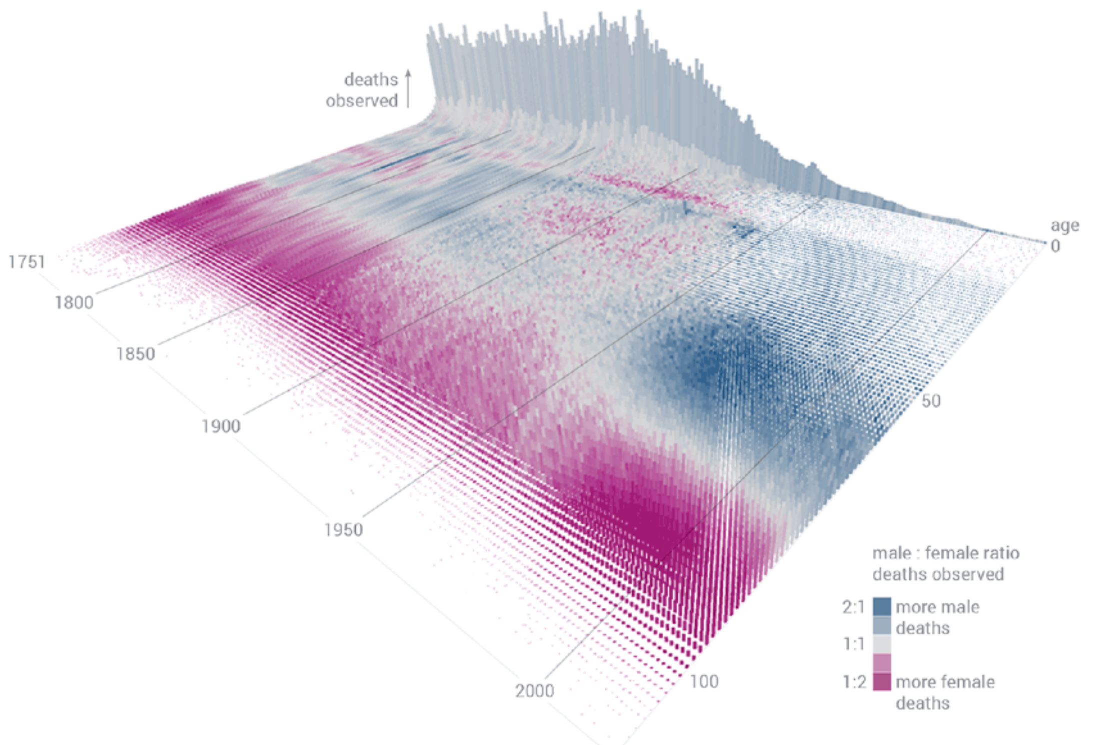


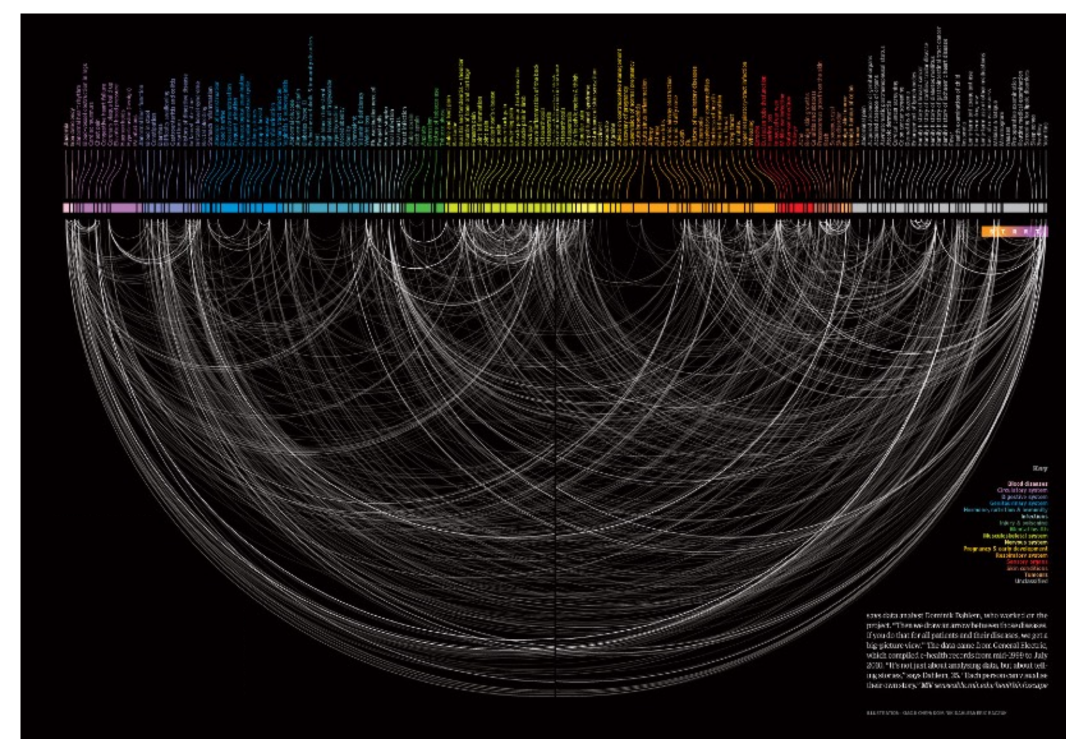

Death Rate Percentile
25
50
75
100
Males


US Population, Percent Black
[0.0, 2.0)
[2.0, 5.0)
[5.0,10.0)
[10.0,15.0)
[15.0,25.0)
[25.0,50.0)
[50.0,85.3]
Percent
75
50
25
0
Public
Charter
Private Non-Specific
Private Christian
Private Catholic
Private Montessori
Private Waldorf
Charter Montessori
Public Montessori
Number of Kindergarteners in each School
20
40
75
100
300

Unicorns


Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
MOSCOU
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.


deaths observed
age
0
50
100
1751
1800
1850
1900
1950
2000
male : female ratio deaths observed
2:1 more male deaths
1:1
1:2 more female deaths



GETTING STARTED

WE WANT TO DRAW GOOD DATA GRAPHICS REPRODUCIBLY

Two ways to use R and ggplot

1. Do Everything in R

Raw Data → Clean & Analyze → Figures

2. Use ggplot only

Tidy CSV of results → Figures

Abstraction in Software

Less → More

Easy things are awkward,

Hard things are doable,

Harder things are difficult

Easy things are trivial,

Hard things are difficult,

Harder things are impossible

D3

Grid

ggplot

Stata

Excel

```
my_packages <- c("tidyverse", "broom", "coefplot", "dotwhisker",
                 "gapminder", "GGally", "ggrepel", "gridExtra",
                 "interplot", "margins", "maps", "mapproj",
                 "mapdata", "MASS", "quantreg", "scales",
                 "survey", "viridis","viridisLite", "devtools")

install.packages(my_packages,
                 repos = "http://cran.rstudio.com")

devtools::install_github("kjhealy/socviz")

devtools::install_github("rstudio/radix")
```

REQUIRED LIBRARIES

LET'S GO

```r
library(gapminder)
```

```r
gapminder
```

```
# A tibble: 1,704 x 6
       country continent  year lifeExp      pop gdpPercap
        <fctr>    <fctr> <int>   <dbl>    <int>     <dbl>
 1 Afghanistan      Asia  1952  28.801  8425333  779.4453
 2 Afghanistan      Asia  1957  30.332  9240934  820.8530
 3 Afghanistan      Asia  1962  31.997 10267083  853.1007
 4 Afghanistan      Asia  1967  34.020 11537966  836.1971
 5 Afghanistan      Asia  1972  36.088 13079460  739.9811
 6 Afghanistan      Asia  1977  38.438 14880372  786.1134
 7 Afghanistan      Asia  1982  39.854 12881816  978.0114
 8 Afghanistan      Asia  1987  40.822 13867957  852.3959
 9 Afghanistan      Asia  1992  41.674 16317921  649.3414
10 Afghanistan      Asia  1997  41.763 22227415  635.3414
# ... with 1,694 more rows
```

named thing gets ...

... with these arguments

```
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp))
```

... the output of this function ...

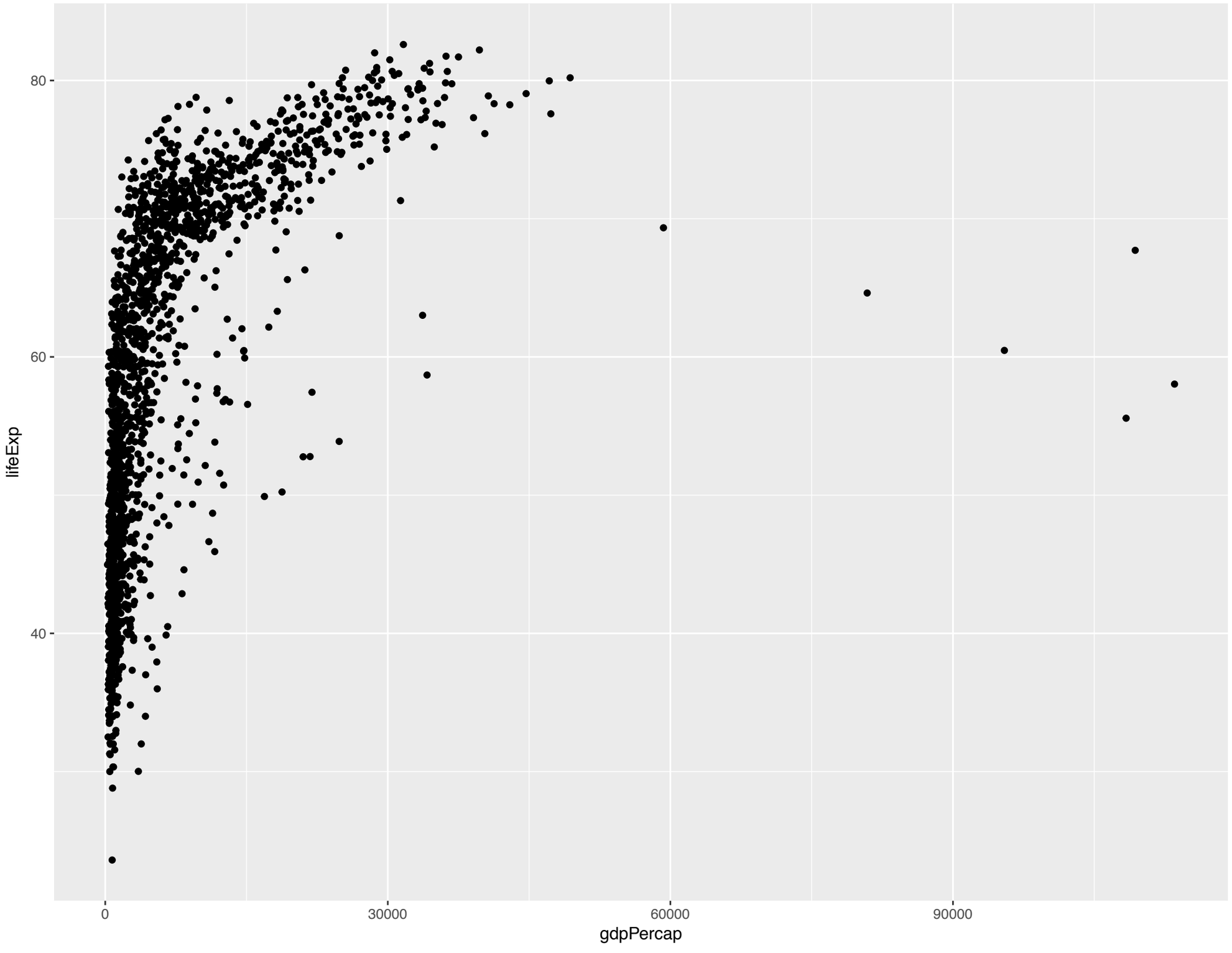
```
p
```

```
p + geom_point()
```

Objects created by ggplot() are unusual in that you can add things to them, and they will work as though you wrote all the code at once.

```
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp))
```

```
p + geom_point()
```


80
60
40
lifeExp
0
30000
60000
90000
gdpPercap

R will be Frustrating

We're going to be adding a lot of objects together.

"+" goes here ◀

```
ggplot(data = mpg,
       mapping = aes(x = displ, y = hwy)) +
    geom_point()
```

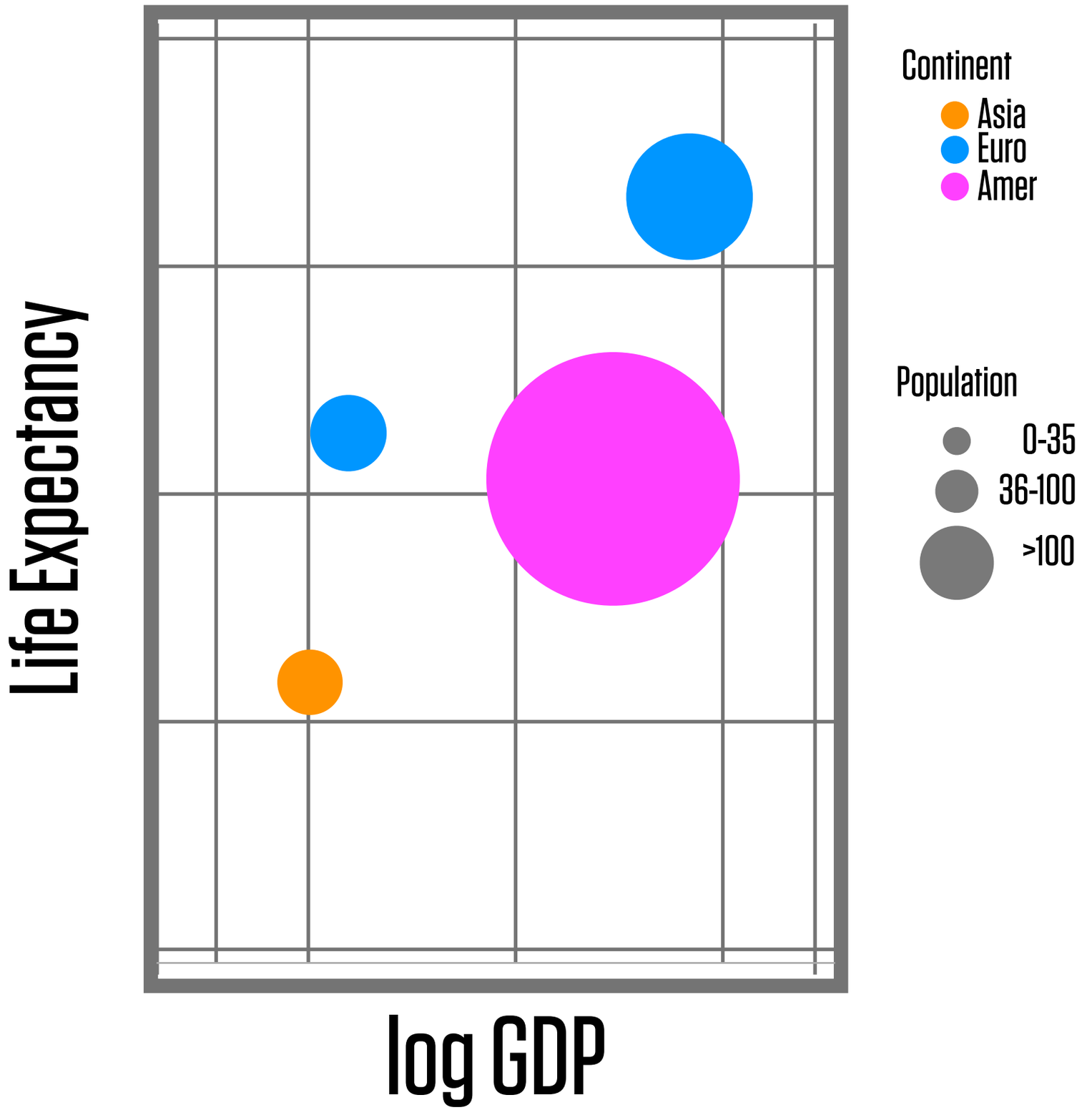
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy))
+ geom_point()
```

▲ not here

HOW ggplot WORKS

gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

A Gapminder Plot



1. Tidy Data

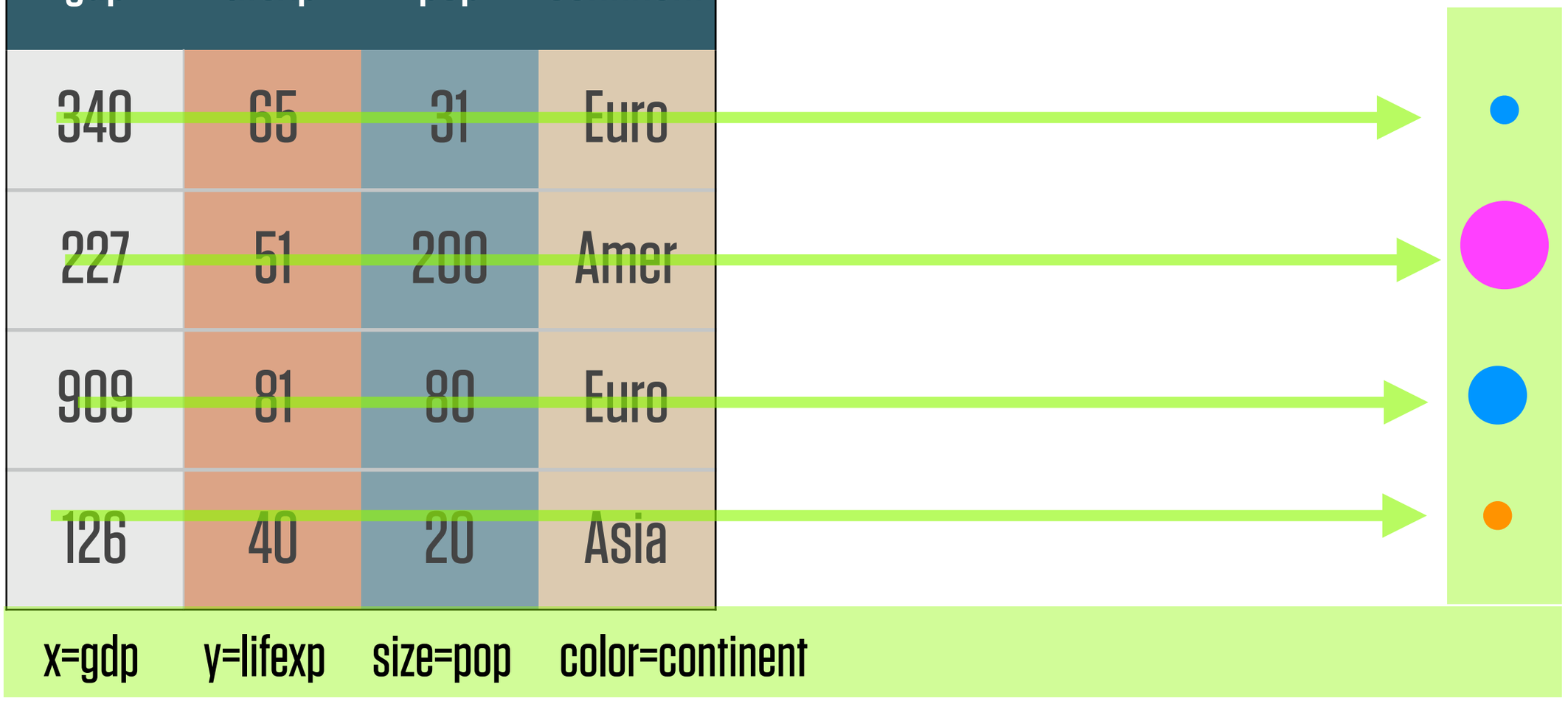
gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

2. Mapping

ggplot(mapping = aes(x = ...))

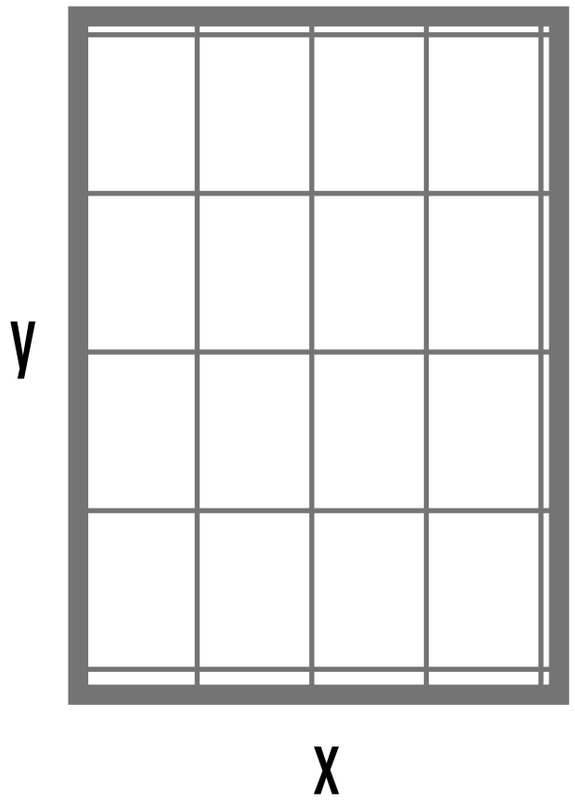
3. Geom

geom_point()

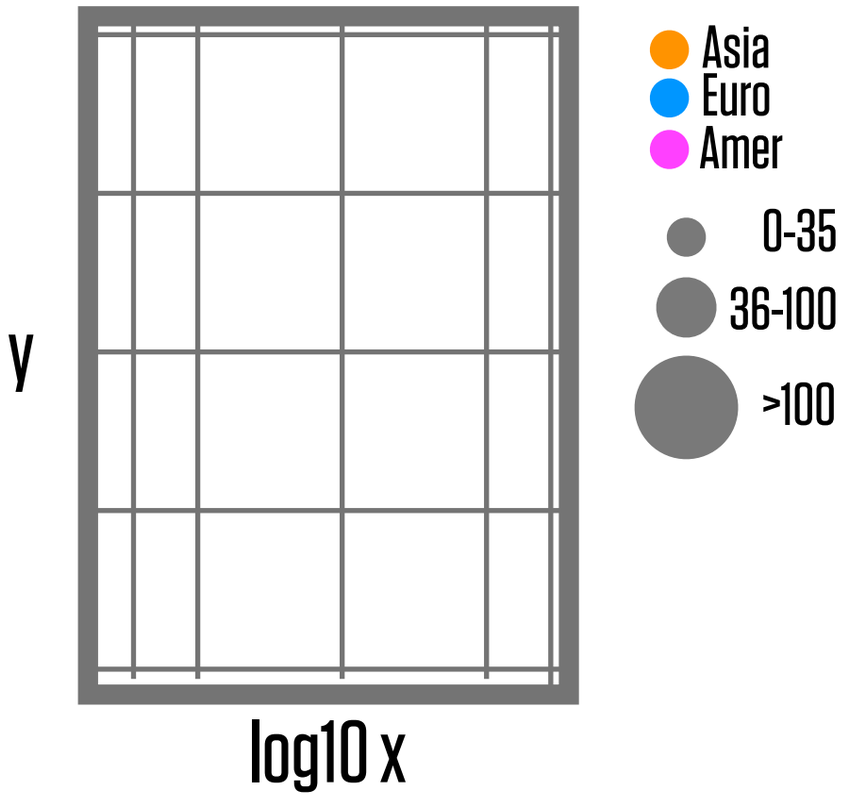


x=gdp y=lifexp size=pop color=continent

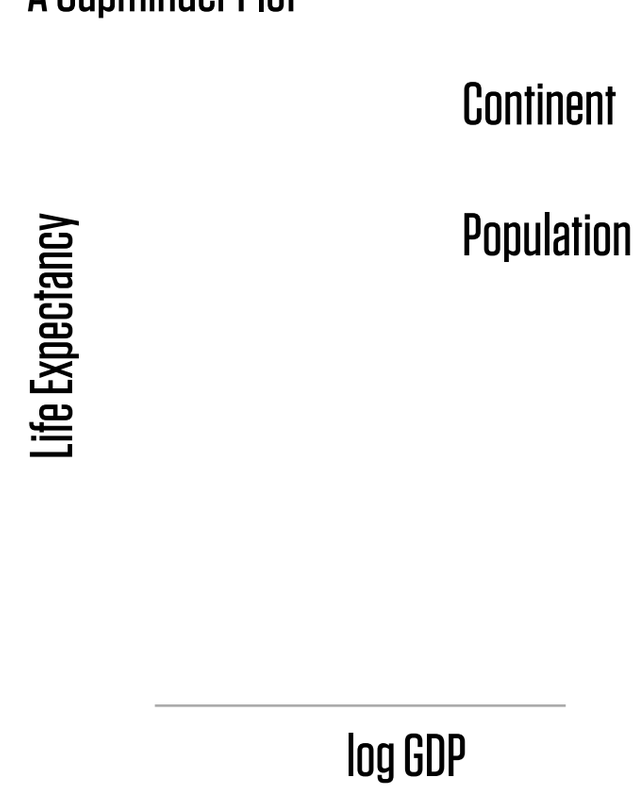
4. Coordinate System



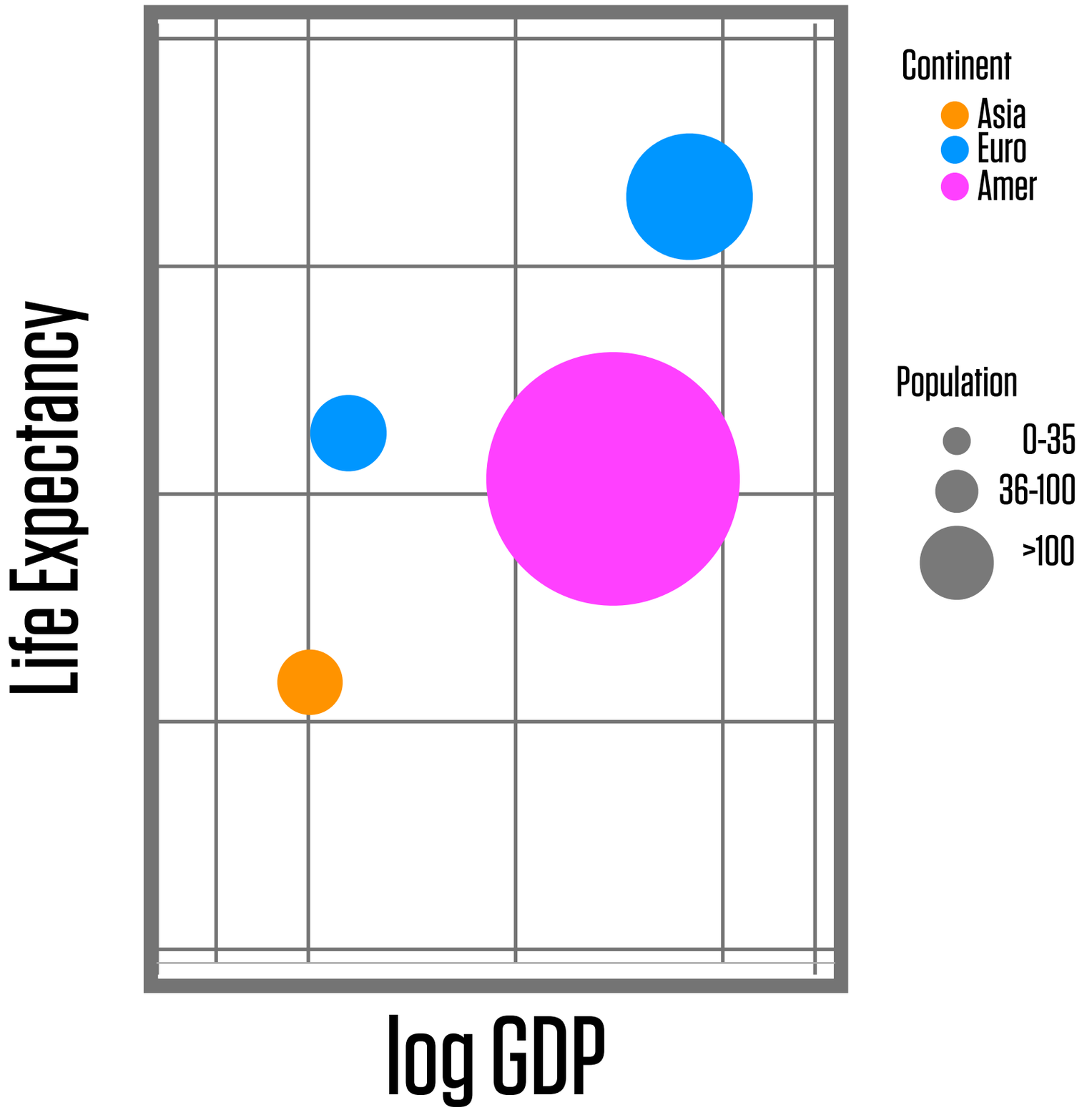
5. Scales



6. Labels & Guides



A Gapminder Plot



LAYER BY LAYER

```r
head(gapminder)
```

```
## # A tibble: 6 × 6
##       country continent  year lifeExp      pop gdpPercap
##        <fctr>    <fctr> <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan      Asia  1952  28.801  8425333  779.4453
## 2 Afghanistan      Asia  1957  30.332  9240934  820.8530
## 3 Afghanistan      Asia  1962  31.997 10267083  853.1007
## 4 Afghanistan      Asia  1967  34.020 11537966  836.1971
## 5 Afghanistan      Asia  1972  36.088 13079460  739.9811
## 6 Afghanistan      Asia  1977  38.438 14880372  786.1134
```

```r
dim(gapminder)
```

```
## [1] 1704    6
```

```
p <- ggplot(data = gapminder)
```

Create a ggplot object
Data is gapminder table

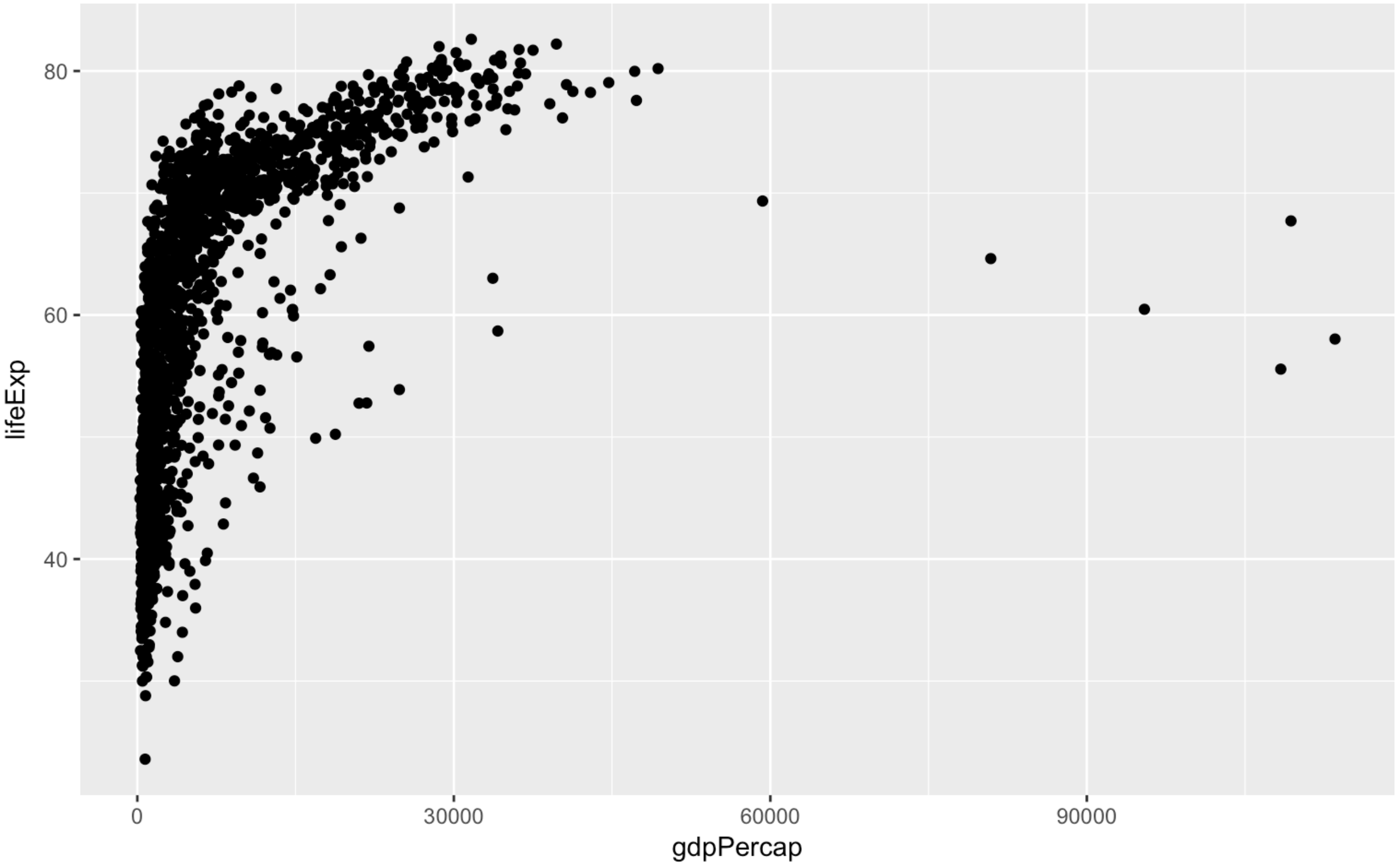
```
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp))
```

Mapping: tell ggplot the relationships you want to see

- The `mapping = aes(...)` instruction *links variables* to *things you will see* on the plot.
- The `x` and `y` values are the most obvious ones.
- Other aesthetic mappings can include, e.g., `color`, `shape`, and `size`.
- These mappings are not *directly* specifying what specific, e.g., colors or shapes will be on the plot. Rather they say which *variables* in the data will be *represented* by, e.g., colors and shapes on the plot.

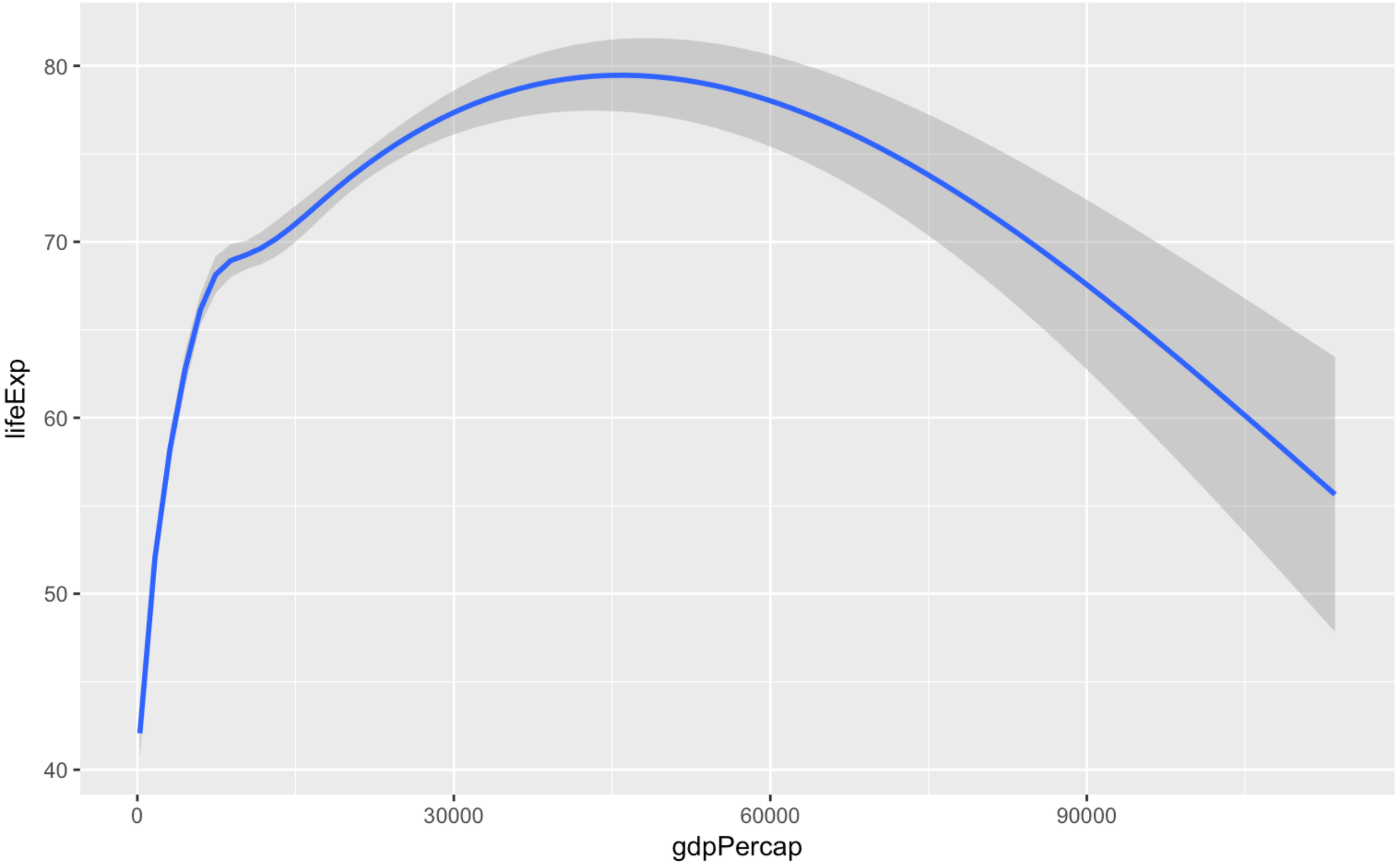
```
p + geom_point()
```

Add a geom layer to the plot


80
60
40
lifeExp
0
30000
60000
90000
gdpPercap

```
p + geom_smooth()
```

Try a different geom

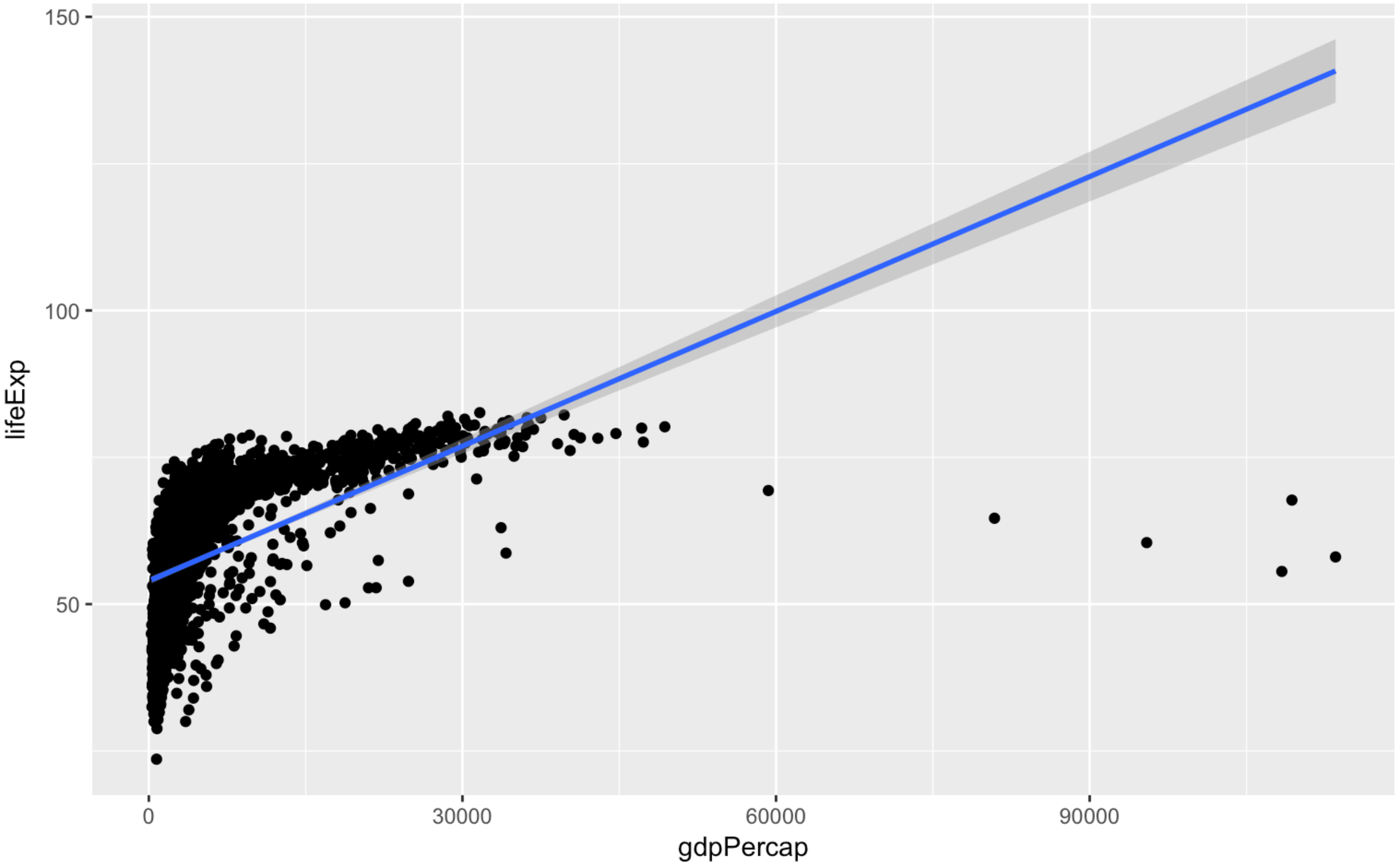

lifeExp
80
70
60
50
40
0
30000
60000
90000
gdpPercap

```
p + geom_point() +
    geom_smooth(method = "gam") +
    scale_x_log10(labels = scales::dollar)
```

This process is literally additive

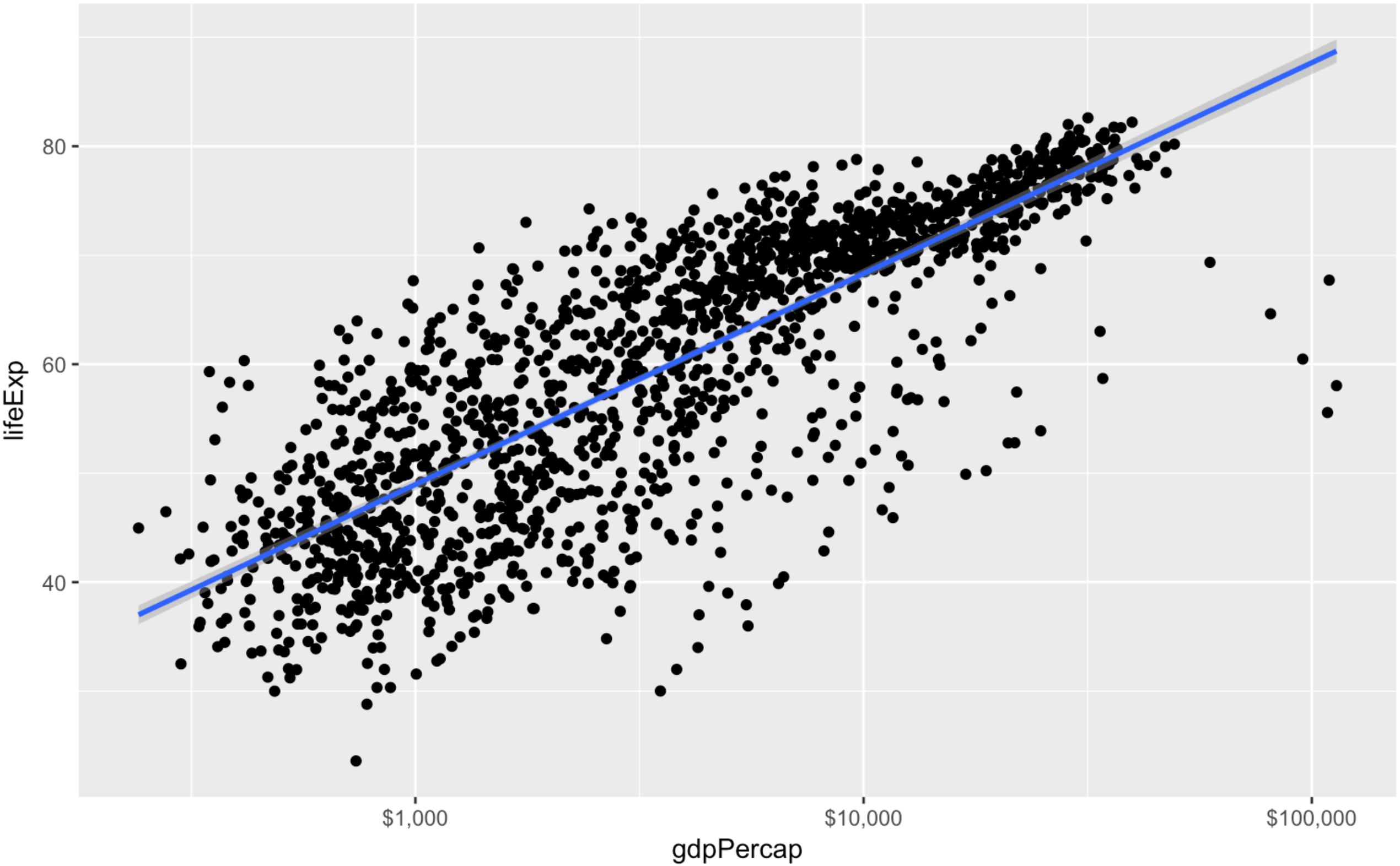
```
p + geom_point() +
geom_smooth(method = "lm")
```

Every geom is a function. It can take arguments.


150
100
50
lifeExp
0
30000
60000
90000
gdpPercap

```
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp))
p + geom_point() +
    geom_smooth(method = "lm") +
    scale_x_log10(label = scales::dollar)
```

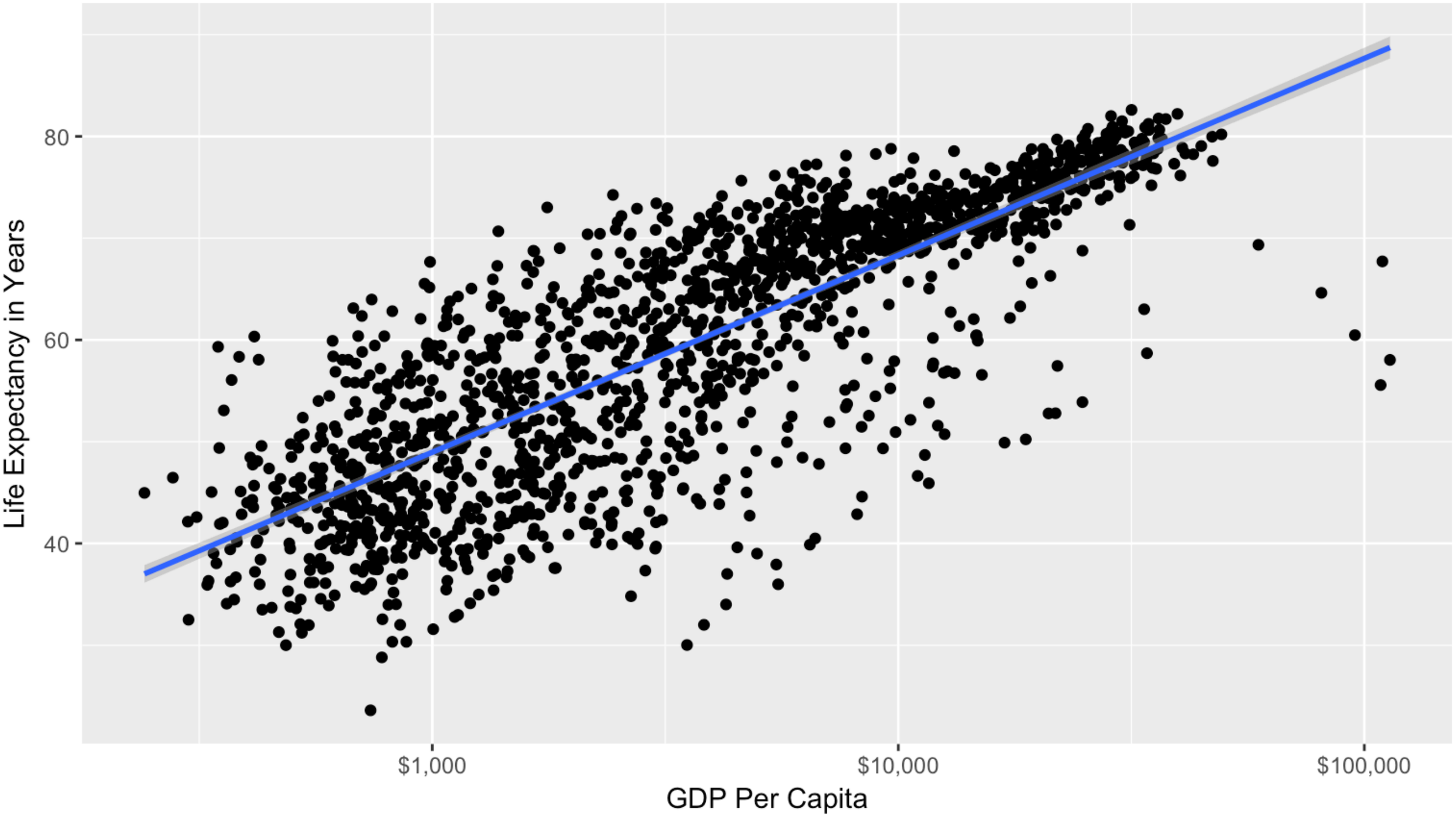
Keep Layering


80
60
40
lifeExp
$1,000
$10,000
$100,000
gdpPercap

```
p + geom_point() +
    geom_smooth(method = "gam") +
    scale_x_log10(labels = scales::dollar) +
    labs(x = "GDP Per Capita",
         y = "Life Expectancy in Years",
         title = "Economic Growth and Life Expectancy",
         subtitle = "Data points are country-years",
         caption = "Data source: Gapminder")
```

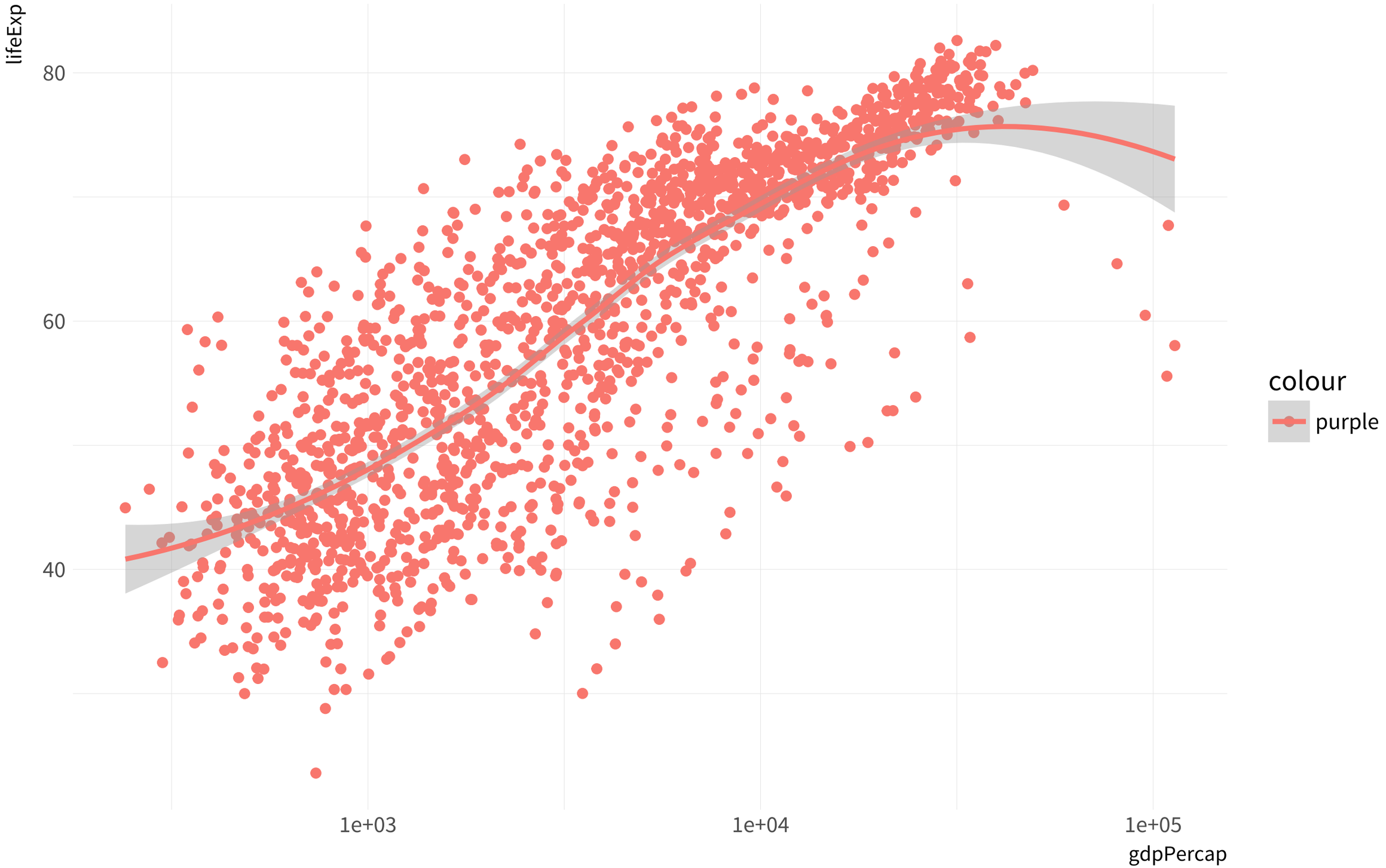
Economic Growth and Life Expectancy

Data points are country-years



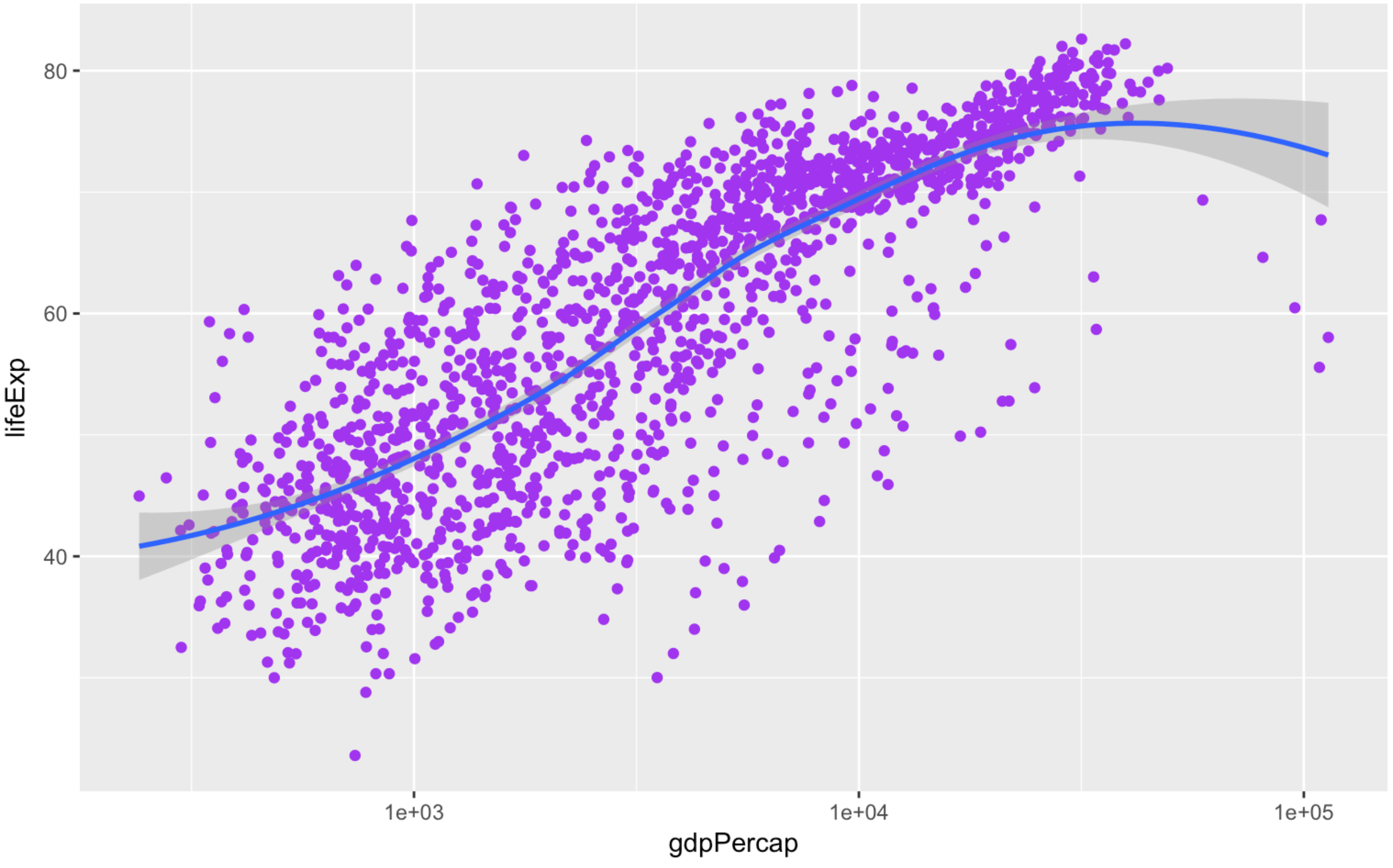
MAPPING VS SETTING AESTHETICS

```r
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp,
                          color = "purple"))
p + geom_point() +
    geom_smooth(method = "loess") +
    scale_x_log10()
```



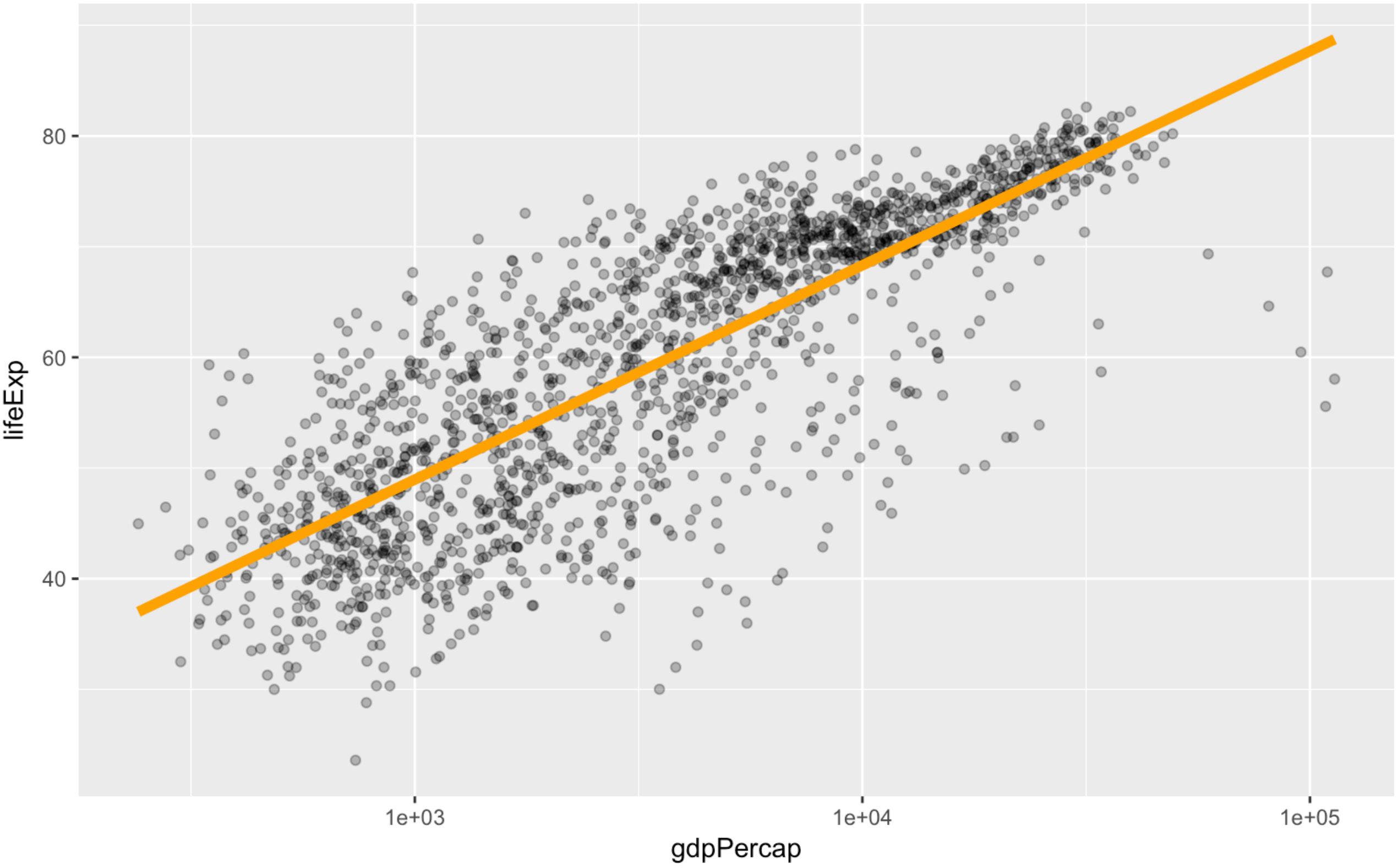
What has gone wrong here?

```
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp)
p + geom_point(color = "purple") +
    geom_smooth(method = "loess")) +
    scale_x_log10()
```

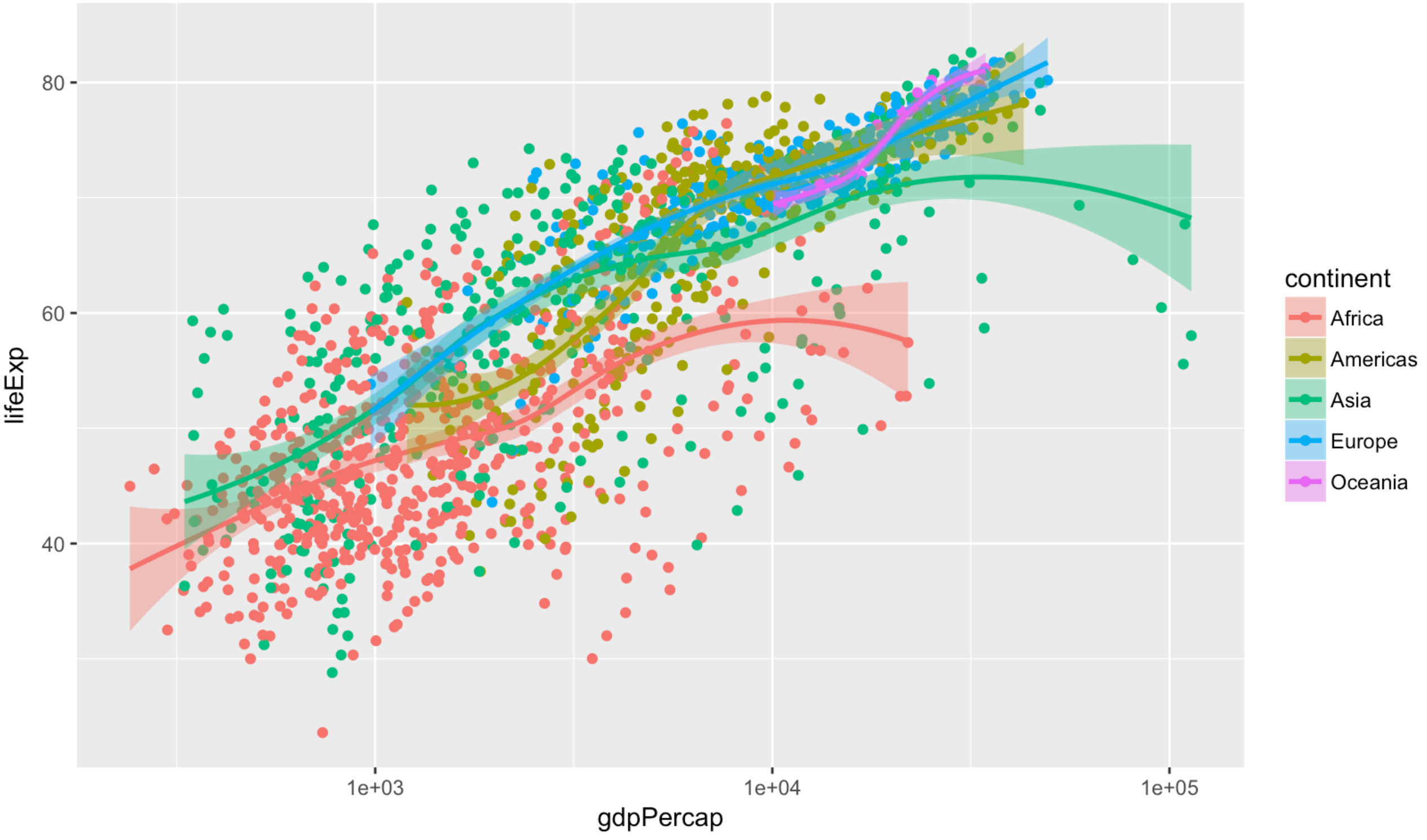

80
60
40
lifeExp
1e+03
1e+04
1e+05
gdpPercap

```r
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp))
p + geom_point(alpha = 0.3) +
    geom_smooth(color = "orange",
      se=FALSE, size=2, method = "lm") +
    scale_x_log10()
```

Here, some aesthetics are mapped, and some are set

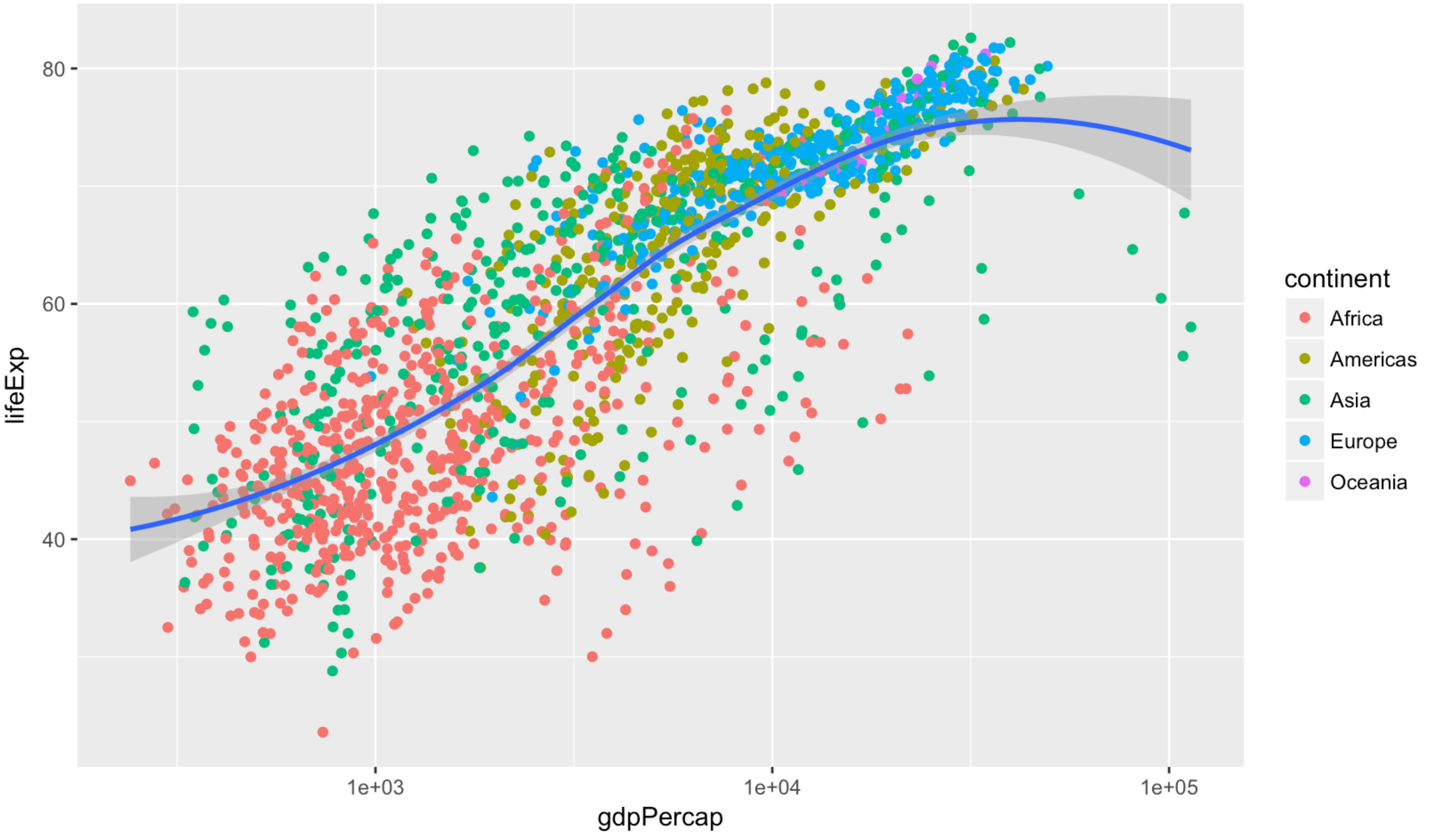

80
60
40
lifeExp
1e+03
1e+04
1e+05
gdpPercap

```
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp,
                          color = continent,
                          fill = continent))
p + geom_point() +
    geom_smooth(method = "loess") +
    scale_x_log10()
```

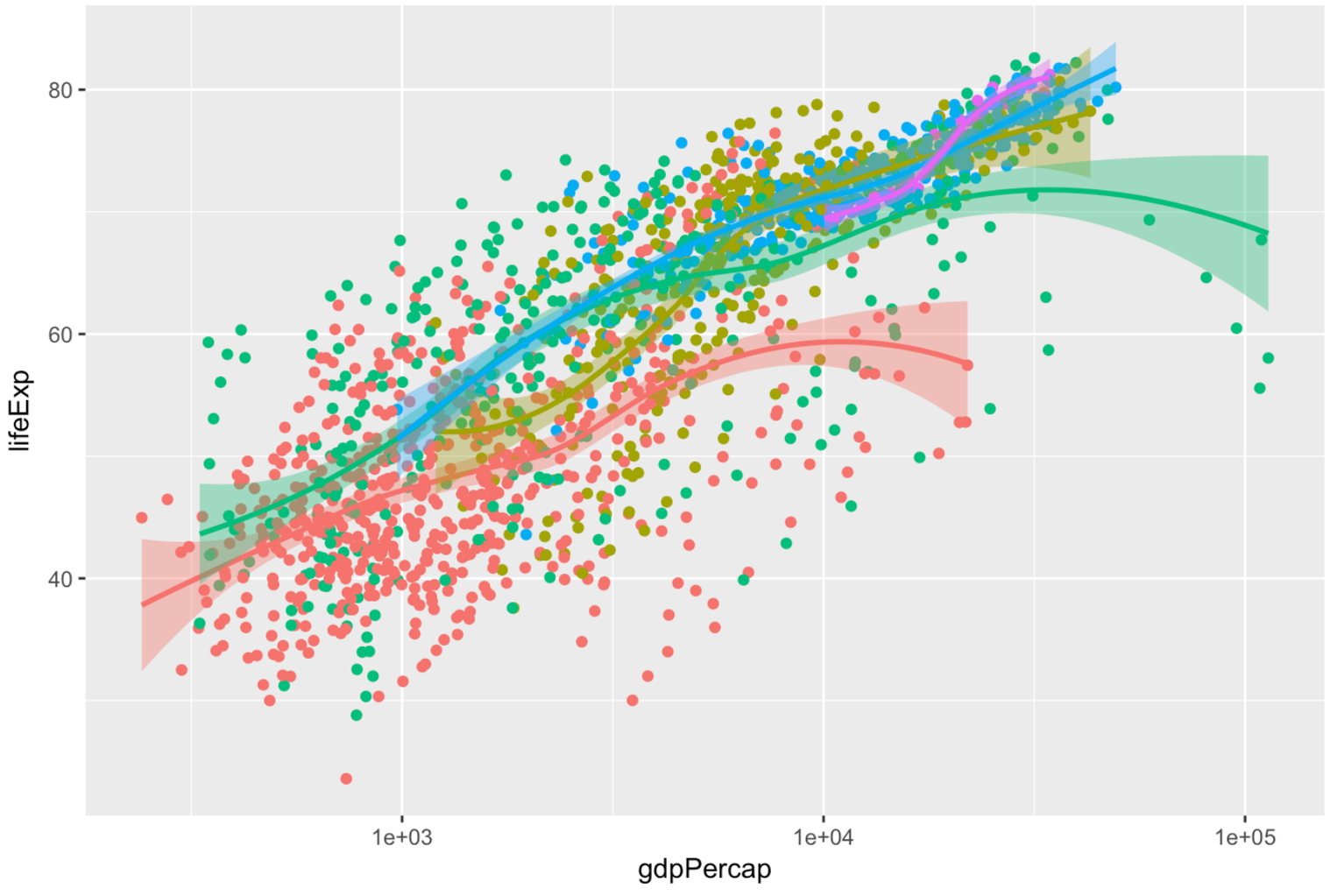

lifeExp
80
60
40
1e+03
1e+04
1e+05
gdpPercap
continent
Africa
Americas
Asia
Europe
Oceania

MAP or SET AESTHETICS per geom

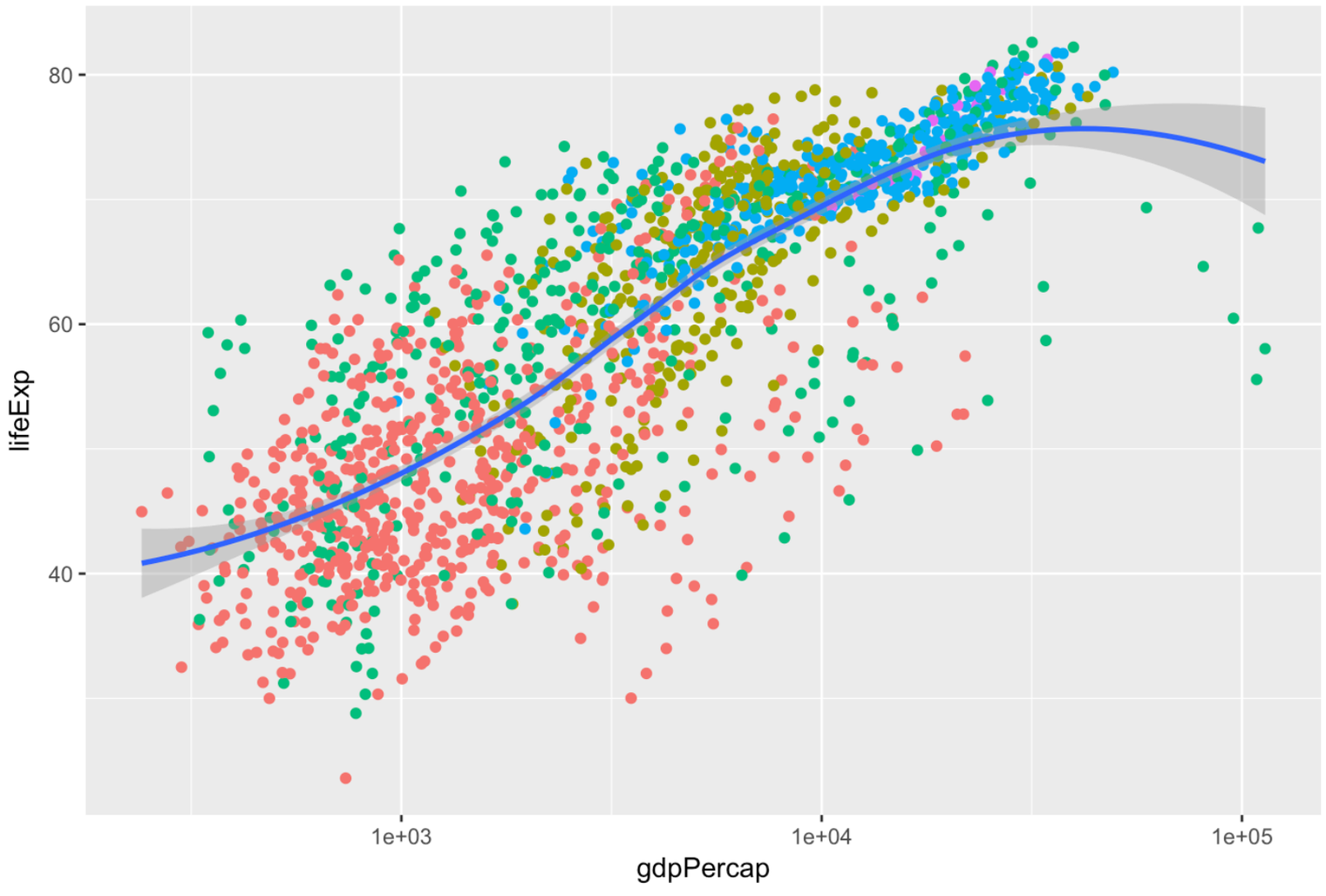
```
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp))
p + geom_point(mapping =
                 aes(color = continent)) +
    geom_smooth(method = "loess") +
    scale_x_log10()
```

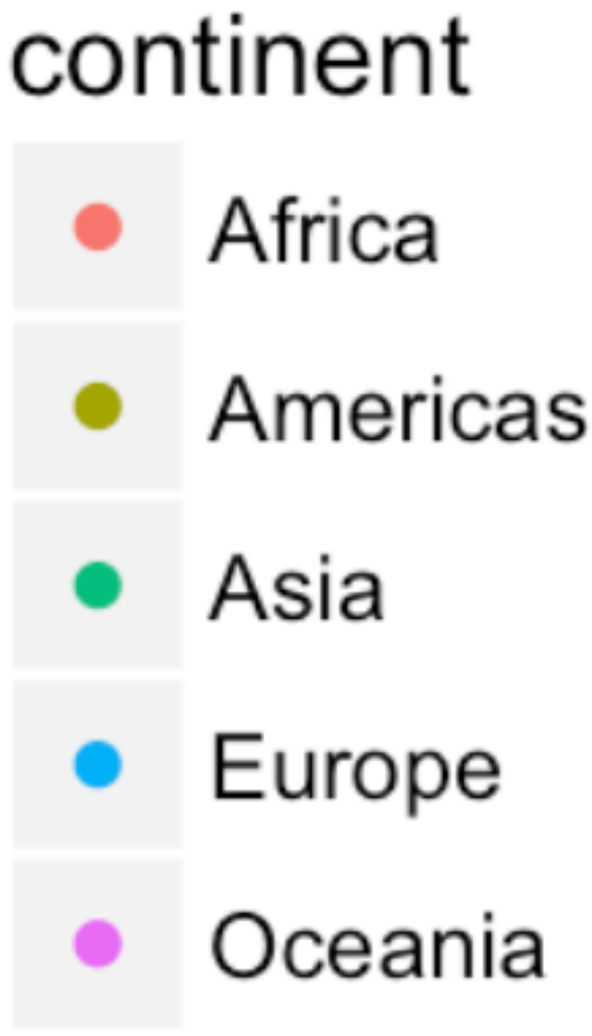

80
60
40
lifeExp
1e+03
1e+04
1e+05
gdpPercap
continent
Africa
Americas
Asia
Europe
Oceania

```r
p <- ggplot(data = gapminder,
            mapping =
              aes(x = gdpPercap,
                  y = lifeExp,
                  color = continent,
                  fill = continent))
p + geom_point() +
    geom_smooth(method = "loess") +
    scale_x_log10()
```



```
p <- ggplot(data = gapminder,
            mapping =
              aes(x = gdpPercap,
                  y = lifeExp))
p + geom_point(mapping =
           aes(color = continent)) +
    geom_smooth(method = "loess") +
    scale_x_log10()
```





Saving Your Work

With ggsave

```
ggsave()

ggsave("figures/my_figure.png")

ggsave("my_figure.pdf")

ggsave("my_figure.pdf",
       plot = p5,
       scale = 1.2)

ggsave("figures/my-figure.pdf",
       plot = p5,
       width = 8,
       height = 5)
```

With pdf() or other graphics devices

```
pdf(file = "plot.pdf", height = 5in,
    width = 5in)
```

Open device ...

```
print(p5)

dev.off()
```

... and close when done

Within an Rmd chunk

````
```{r my_plot, fig.cap="My Plot", fig.width=9, fig.height=8}
p + geom_point(mapping =
          aes(color = continent)) +
    geom_smooth(method = "loess") +
    scale_x_log10()
```
````

Set defaults in your first code chunk

```
knitr::opts_chunk$set(fig.width=8, fig.height=5)
```