Visualization in R

Intro workshop

Alona O. Dolinsky

COMPTEXT 2024, VU Amsterdam

Introduction

Creating visualizations is one of the most useful and powerful ways to analyse data and tell a convincing story with these data.

With ggplot, R Tidyverse provides a very powerful and well-designed visualization library.

It can be used to make almost all kinds of graphs, from simple line and scatter plots to colorful annotated bubble plots.

Online resources and set up

The R Graph Gallery – lots of examples of pretty plots, including the relevant R code

From data to viz – interactive site showcasing plots based on your own data; includes relevant R code, best practices, common mistakes and alternative suggestions for each plot.

Data Visualization by Kieran Healy – free online book discussing principles of data visualization, best use suggestions and relevant R code

The ggplot cheat sheet – handy overview of ggplot functions and options

Looking at Data

- Think of research questions
- Develop relevant and appropriate hypotheses/tests
- Use corresponding models/methods of analysis
- Communicate the results

Figure 1.1: Plots of Anscombe's Quartet.

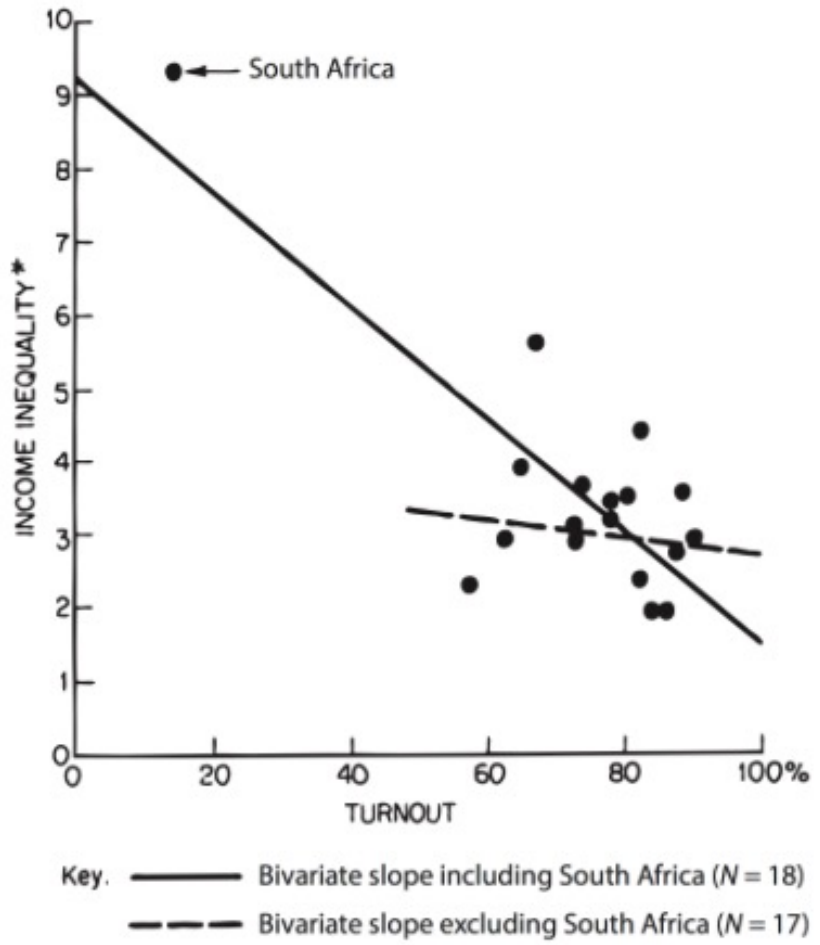


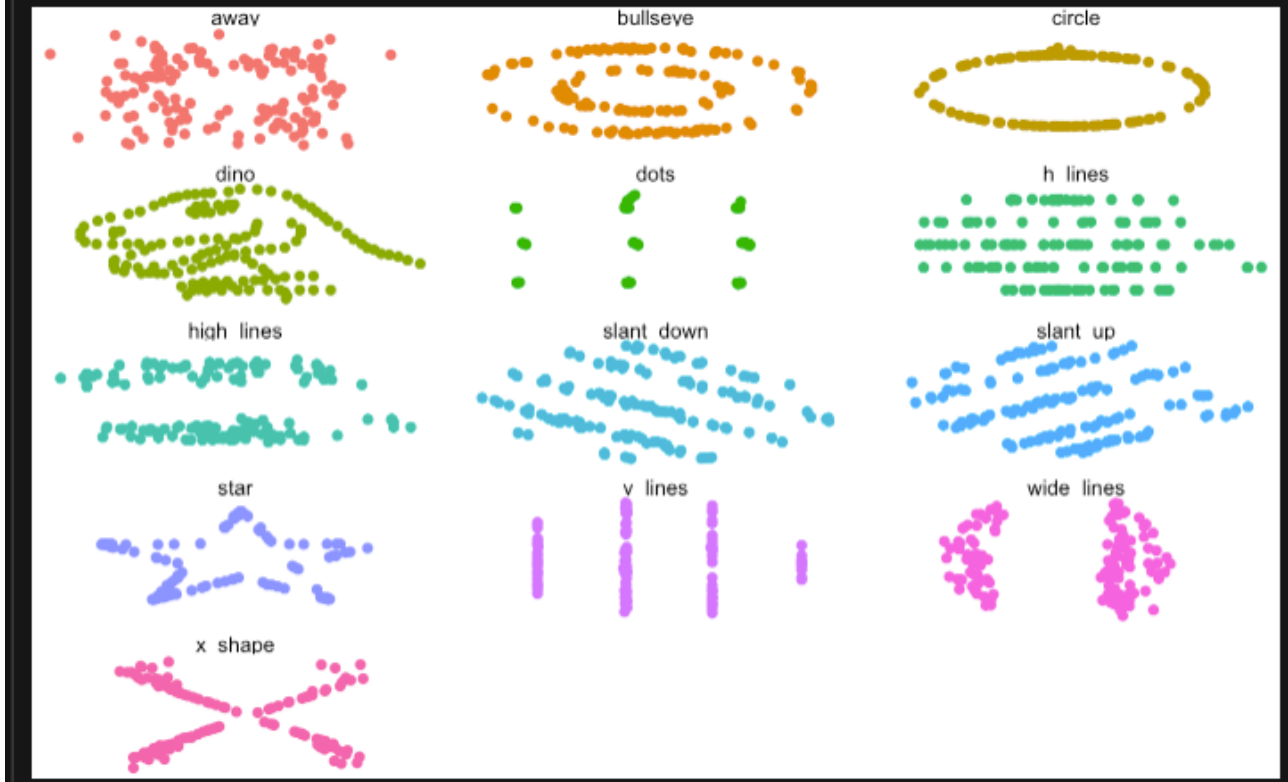
Figure 1.2: Seeing the effect of an outlier on a regression line.

Source: https://socviz.co/lookatdata.html

```r
library(ggplot2)
library(datasauRus)

ggplot(datasaurus_dozen, aes(x = x, y = y, colour = dataset))+
  geom_point() +
  theme_void() +
  theme(legend.position = "none")+
  facet_wrap(~dataset, ncol = 3)

```



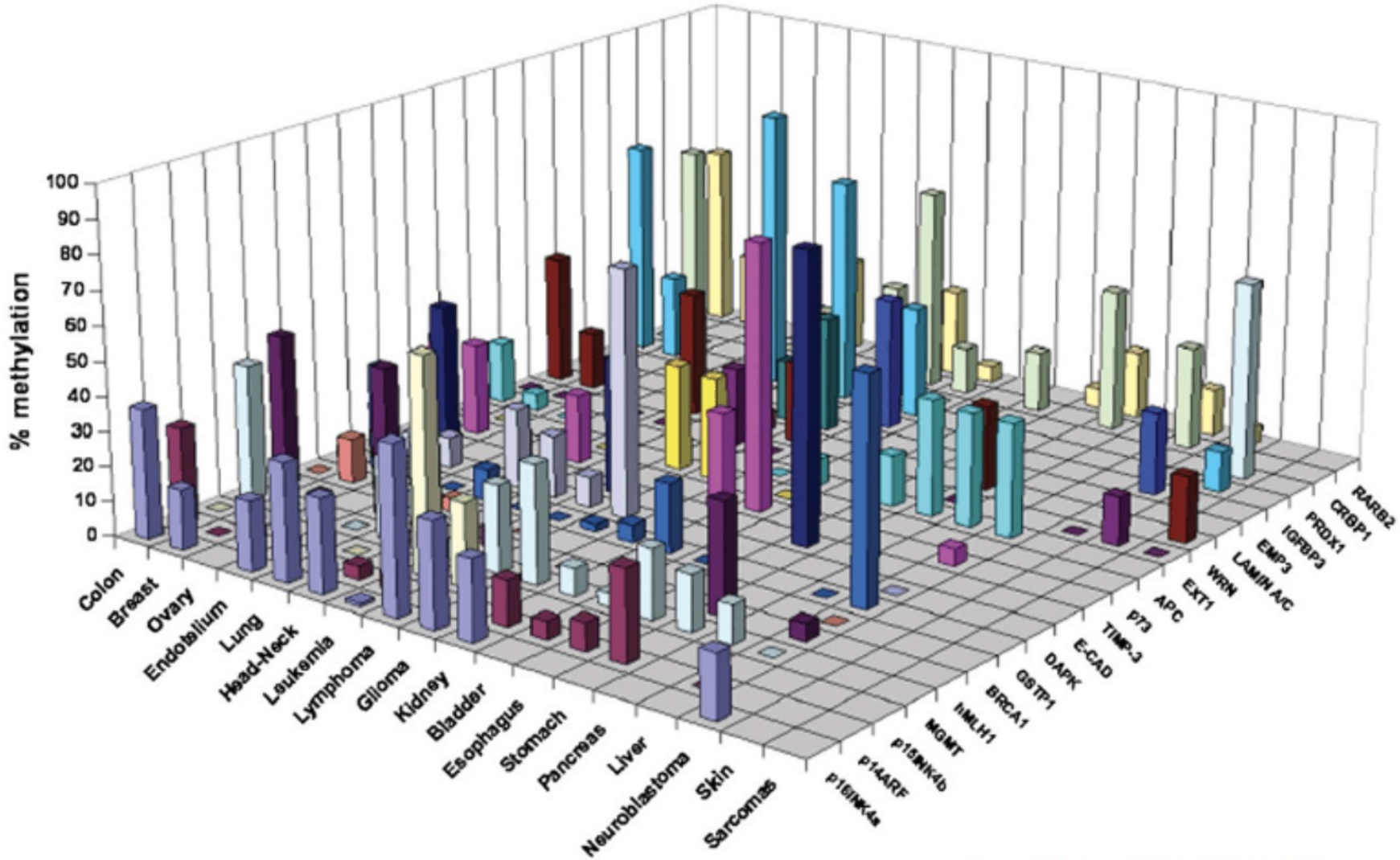
Source: datasauRus

What makes for bad data visualization?

Three varieties of problems:

1. Aesthetics – the figure just doesn't look good or is done in bad taste

A CpG Island Hypermethylation Profile of Human Cancer

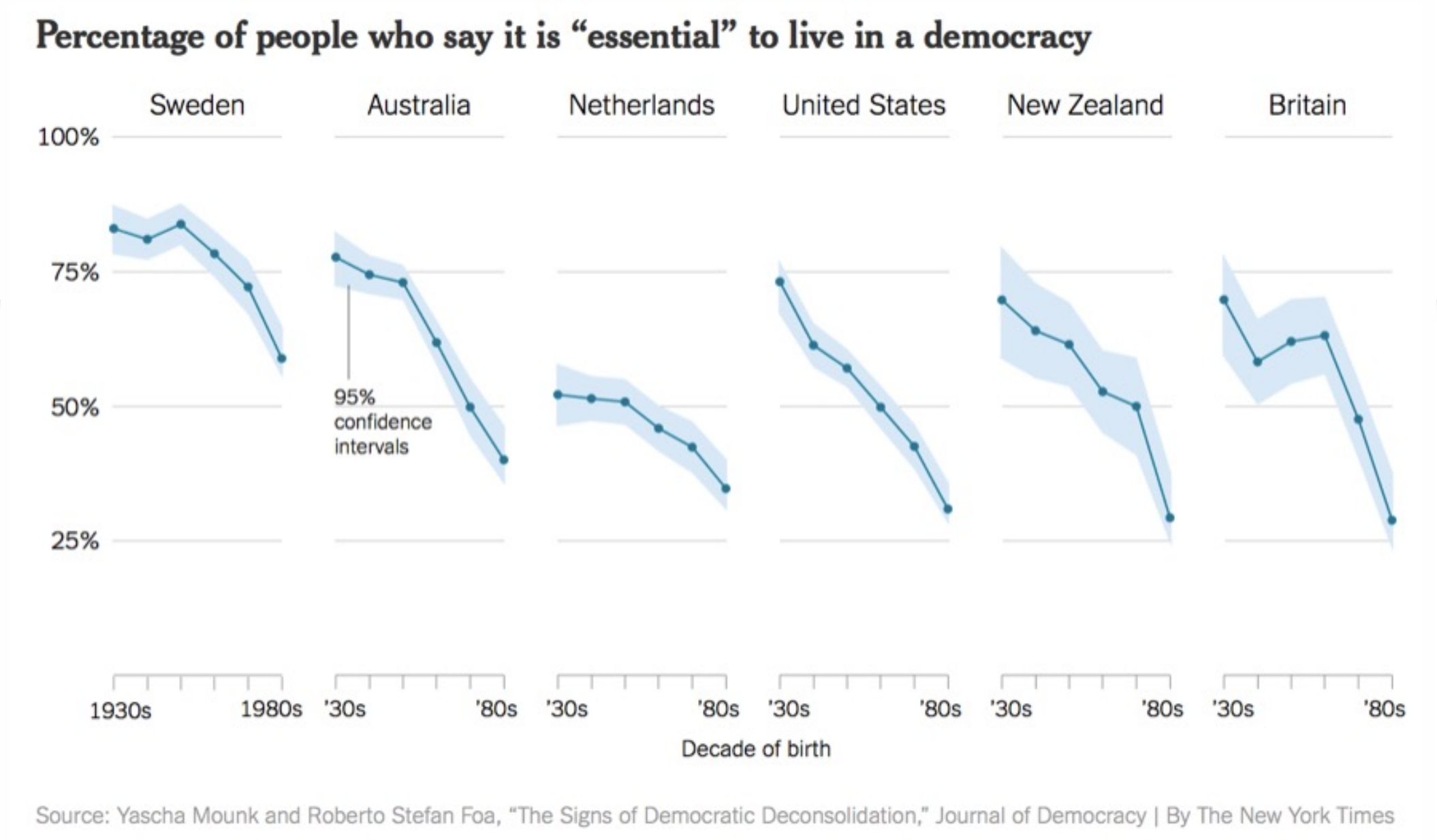


Hum. Mol. Genet. (2007) 16:R50-59

What makes for bad data visualization?

Three varieties of problems:

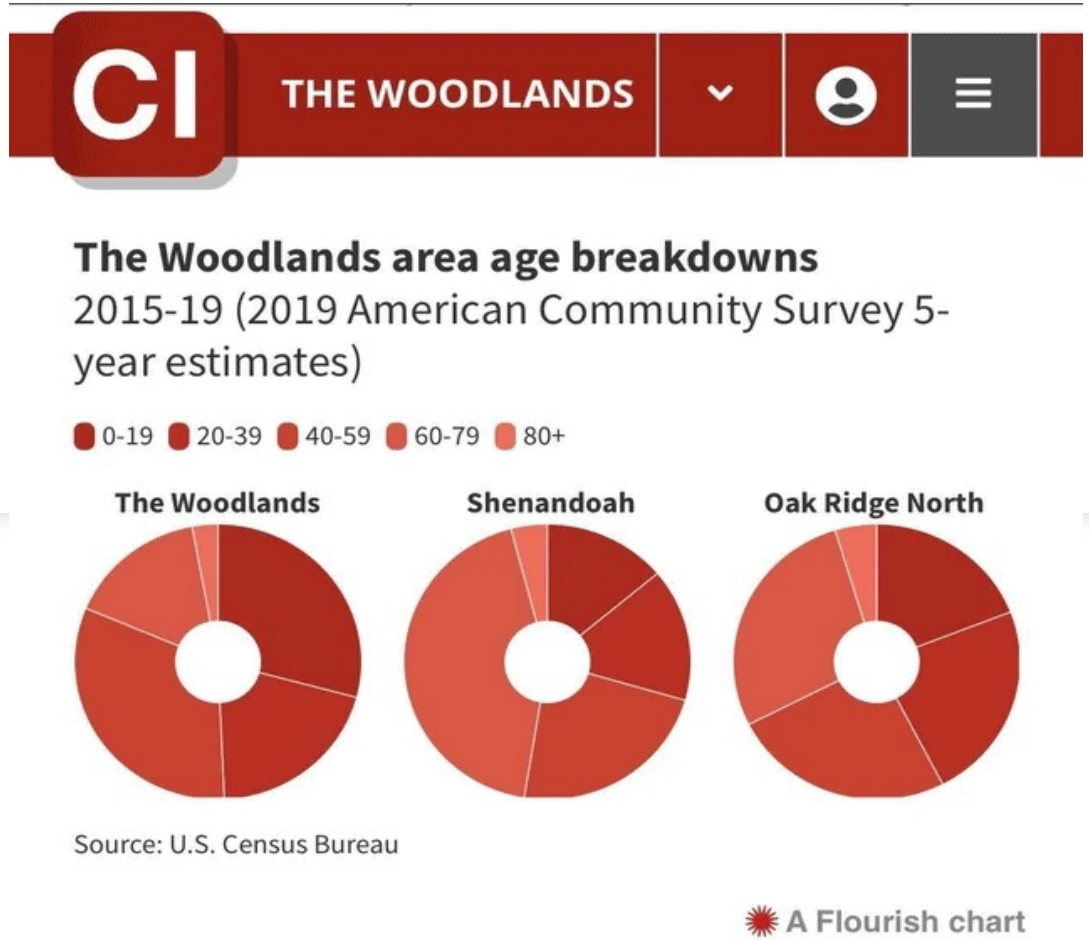
1. Aesthetics – the figure just doesn't look good
2. Substance – the data being presented is the issue


Percentage of people who say it is "essential" to live in a democracy
Sweden
Australia
Netherlands
United States
New Zealand
Britain
100%
75%
50%
25%
95% confidence intervals
1930s
1980s
'30s
'80s
Decade of birth
Source: Yascha Mounk and Roberto Stefan Foa, "The Signs of Democratic Deconsolidation," Journal of Democracy | By The New York Times

What makes for bad data visualization?

Three varieties of problems:

1. Aesthetics – the figure just doesn't look good
2. Substance – the data being presented is the issue
3. Miscomprehension – the figure is confusing or misleading



In Oak Ridge North, median age dropped 0.63% from 47.3 to 47, as the city's population share fell within every age bracket except 20- to 39-year-olds.

Source: When Data Visualization Really Is not Useful (and When It Is)

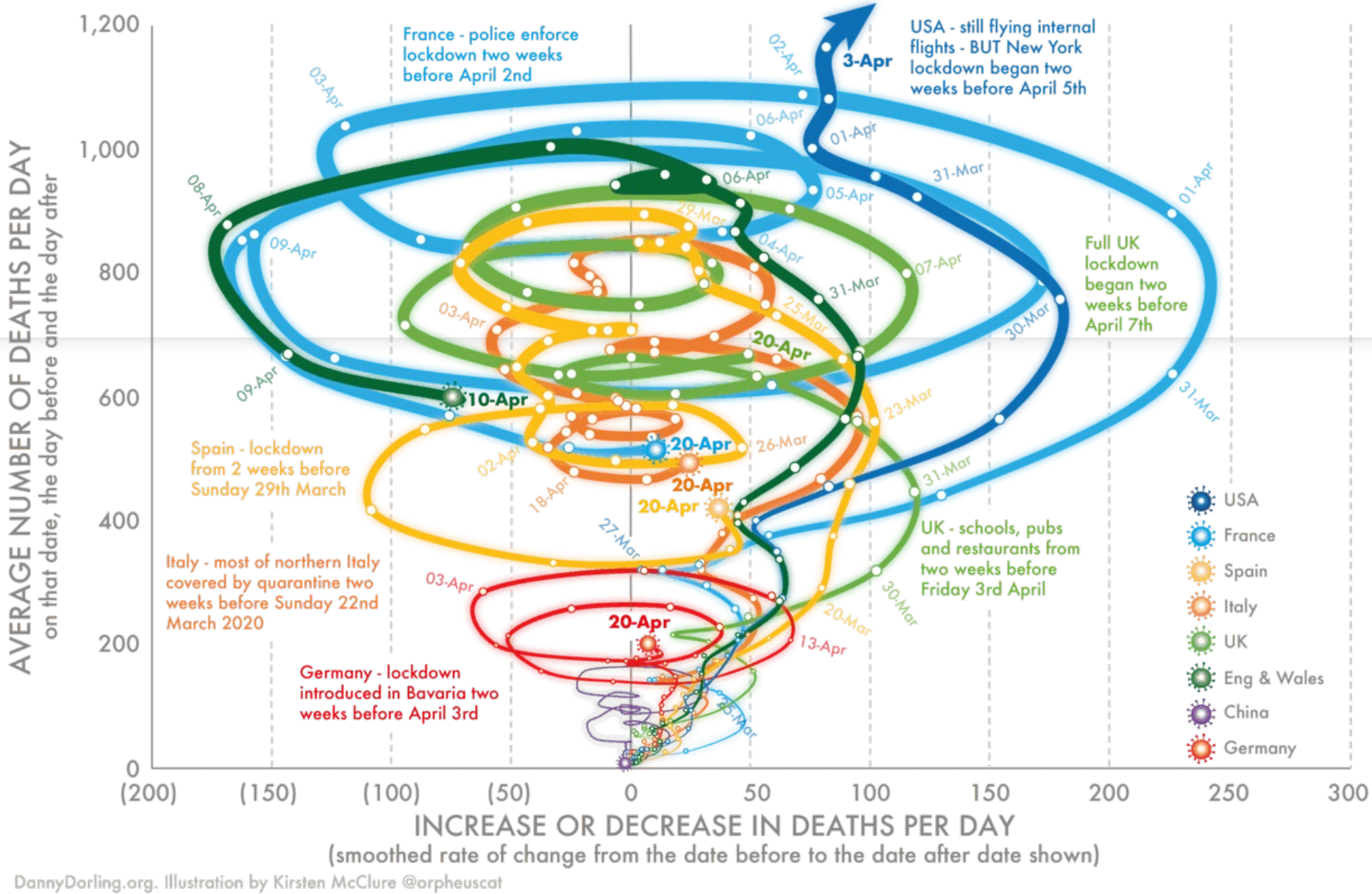
Principles of Data Visualization

1. Show as much data as possible
2. Tell the Truth

Source: https://medium.com/fourier-media/covid-19-chart-crimes-b9765194ef6c

Principles of Data Visualization

1. Tell the Truth
2. Show as much data as possible
3. Keep it simple

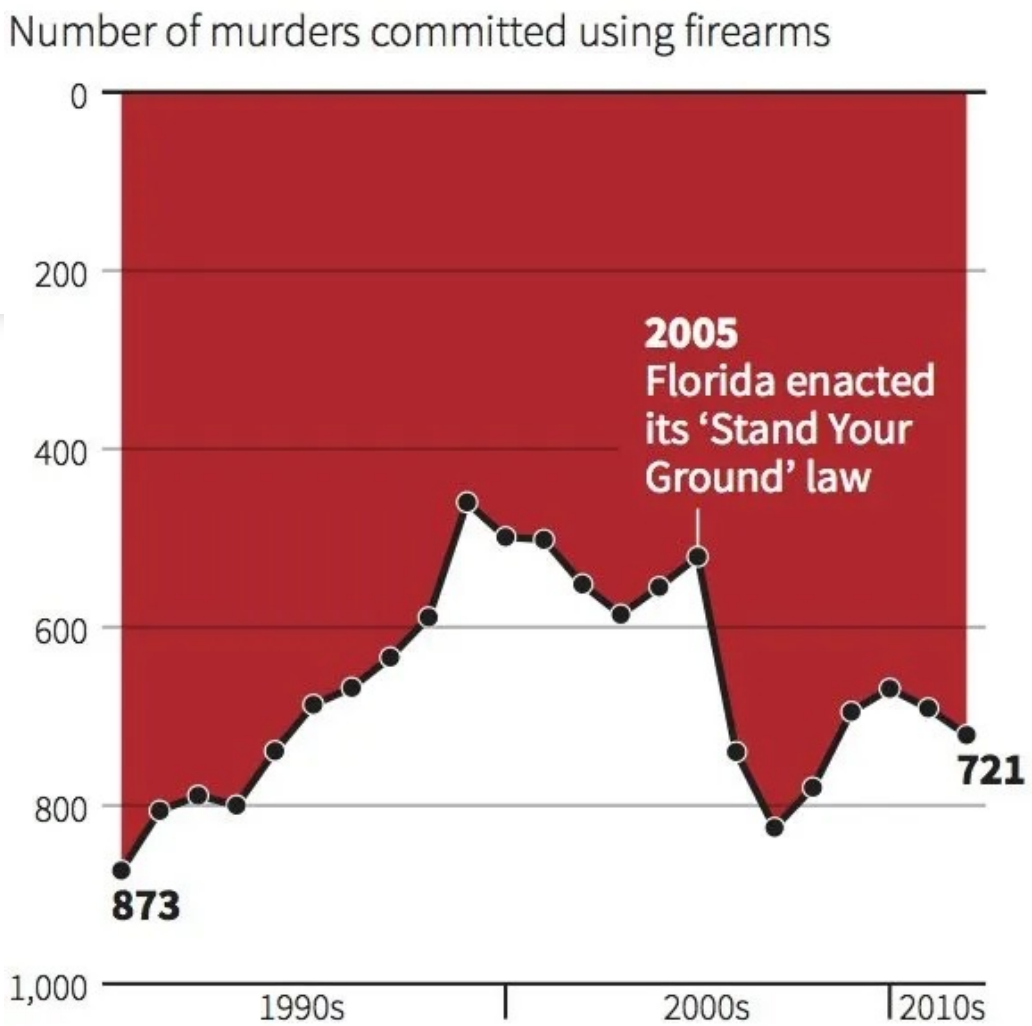


Source: When Data Visualization Really Is not Useful (and When It Is)

Principles of Data Visualization

1. Tell the Truth
2. Show as much data as possible
3. Keep it simple
4. Do not inverse charts

Gun deaths in Florida



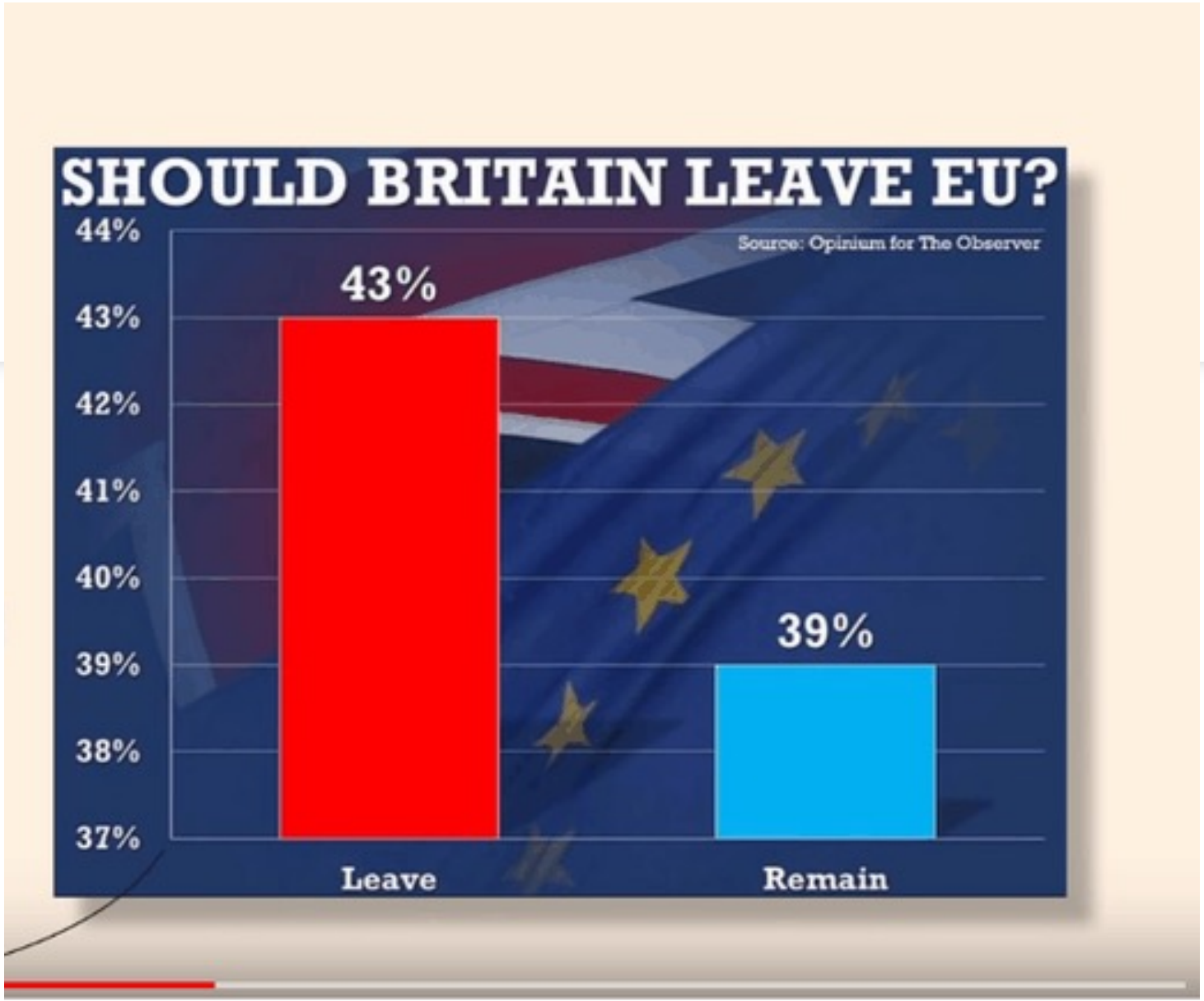
Source: Florida Department of Law Enforcement

C. Chan 16/02/2014

REUTERS

Principles of Data Visualization

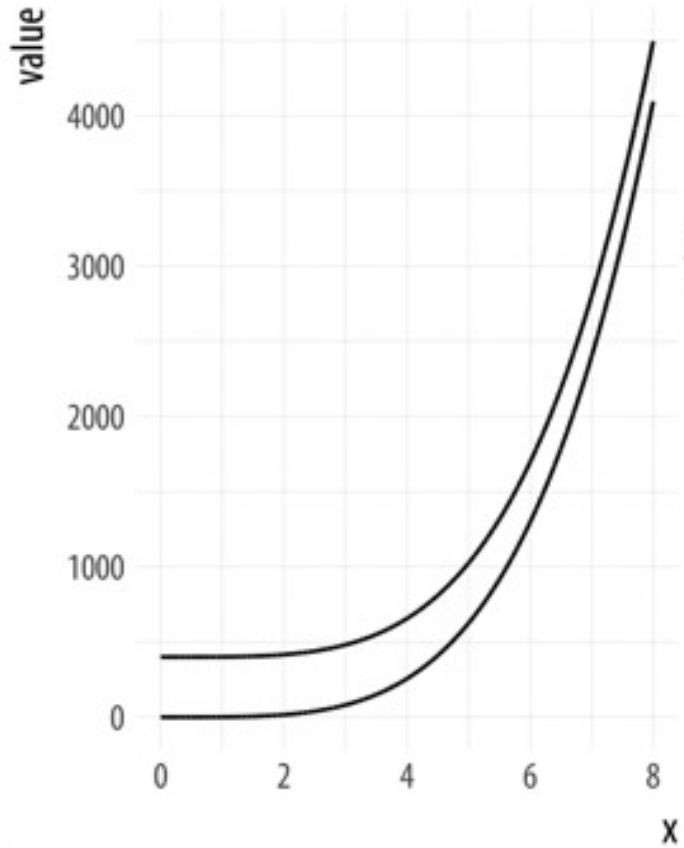
1. Tell the Truth
2. Show as much data as possible
3. Keep it simple
4. Do not inverse charts
5. Do not cut chart axes short

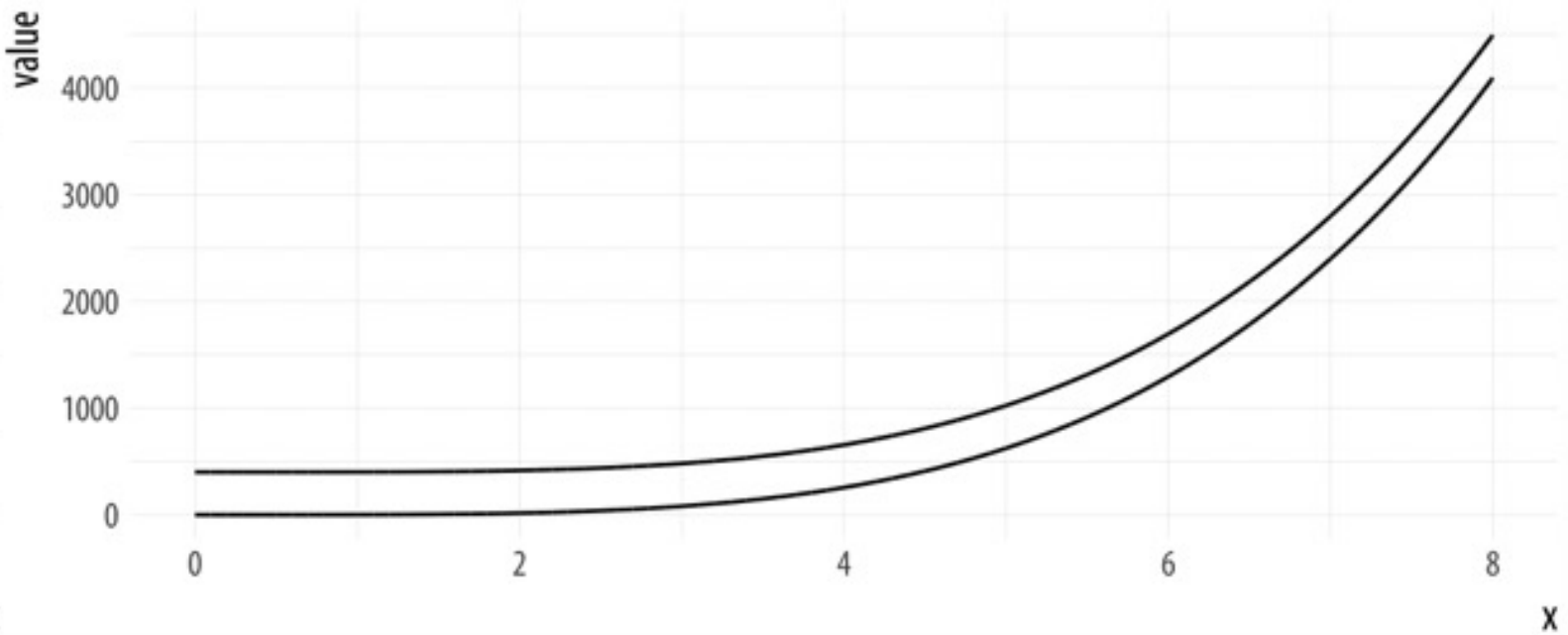


Source: When Data Visualization Really Is not Useful (and When It Is)

Principles of Data Visualization

1. Tell the Truth
2. Show as much data as possible
3. Keep it simple
4. Do not inverse charts
5. Do not cut chart axes short
6. Find the right aspect-ratio

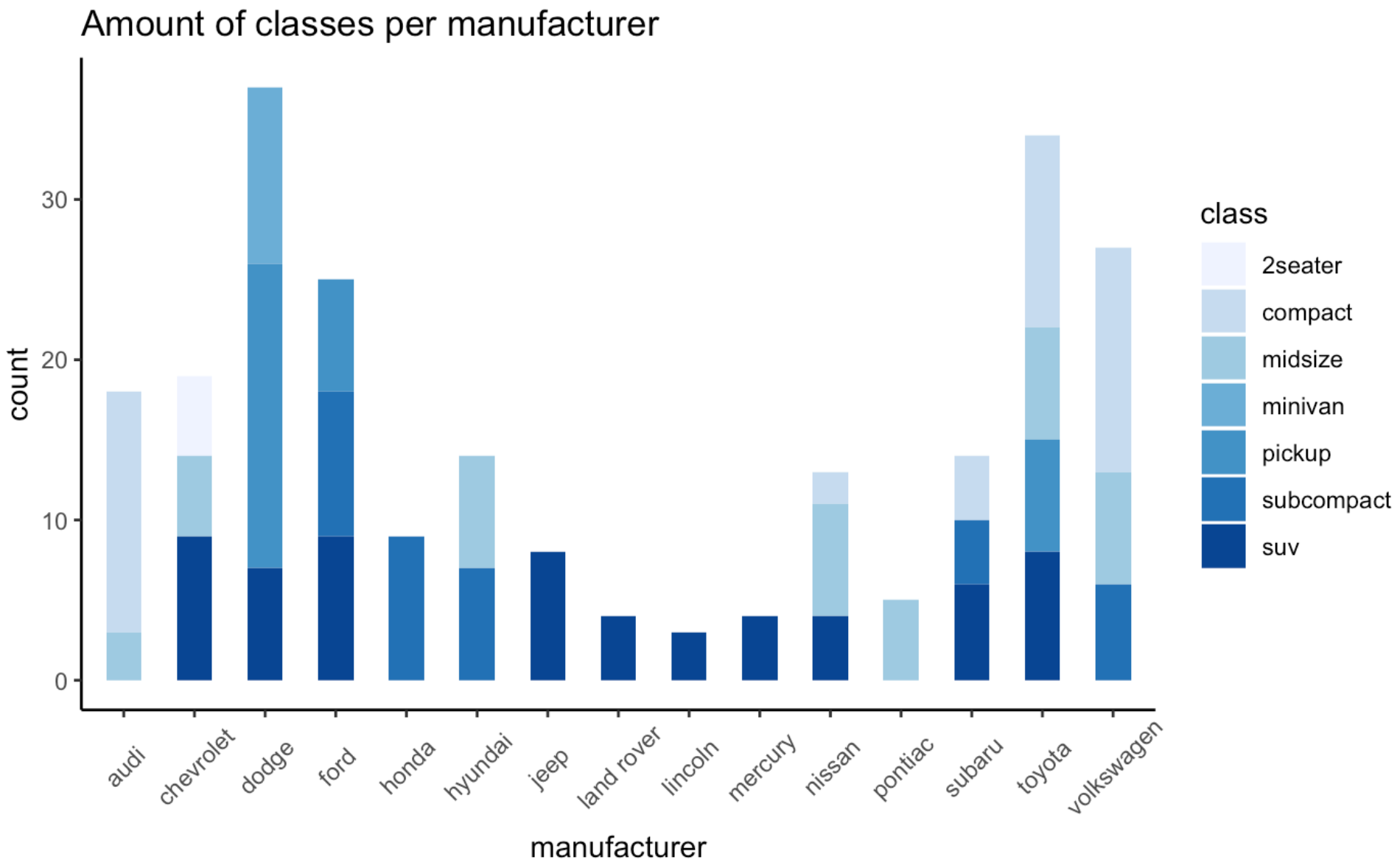




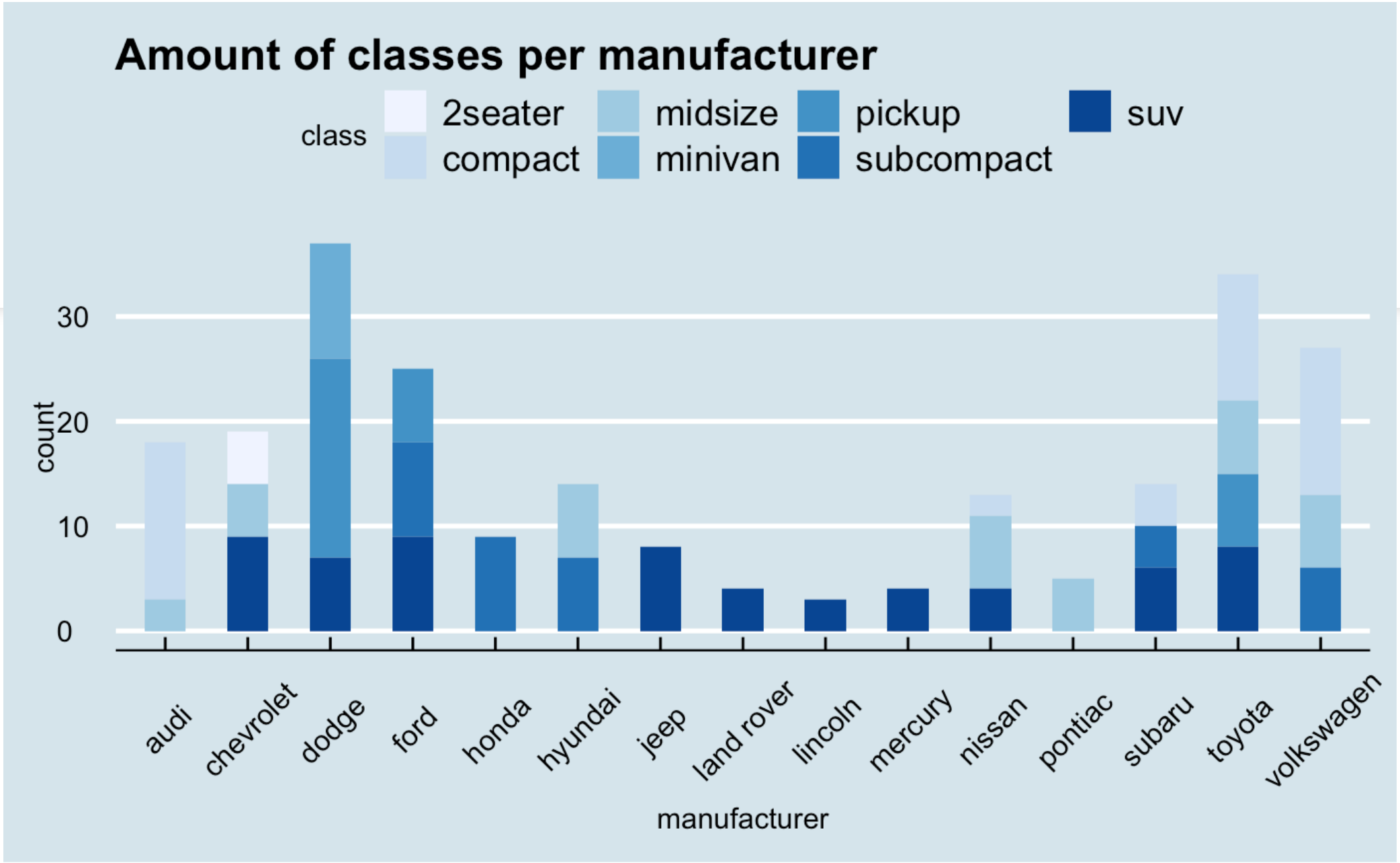
Source: https://socviz.co/lookatdata.html

Principles of Data Visualization

1. Tell the Truth
2. Show as much data as possible
3. Keep it simple
4. Do not inverse charts
5. Do not cut chart axes short
6. Find the right aspect-ratio
7. Maximize data-ink ration (but not too much)



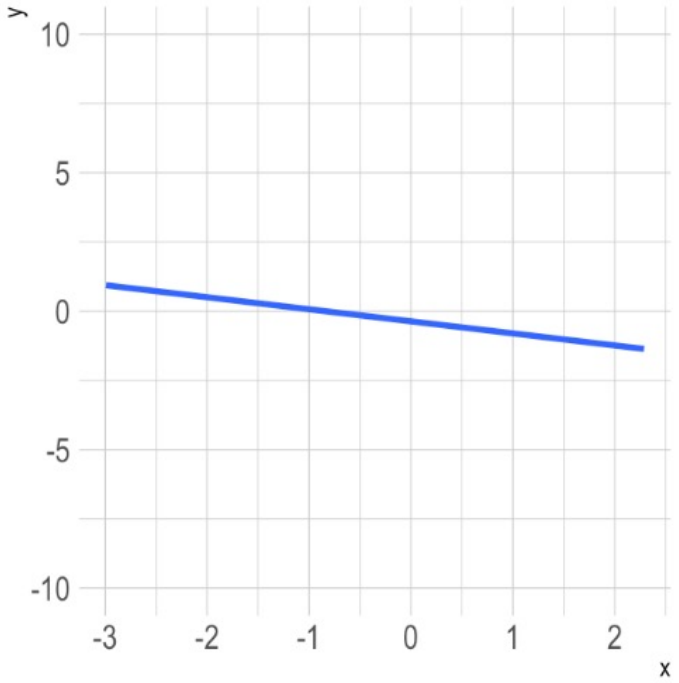
Source: Philipp Mansur

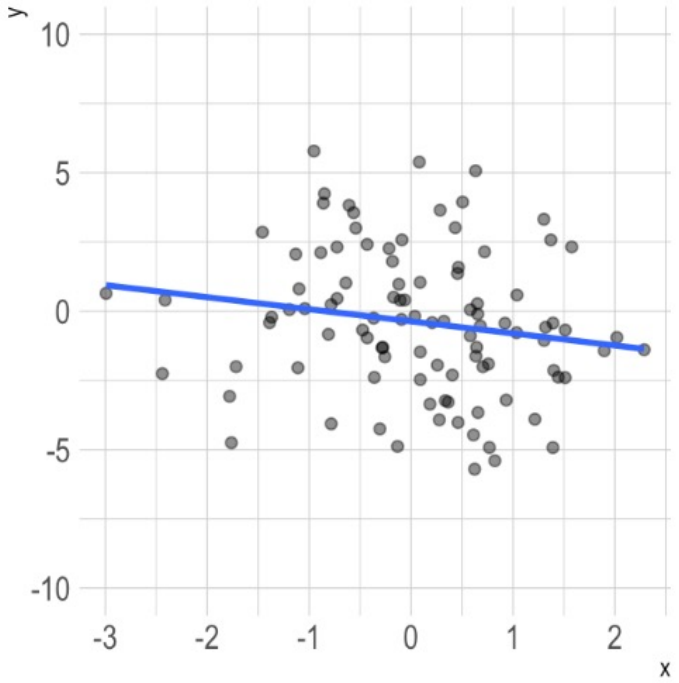


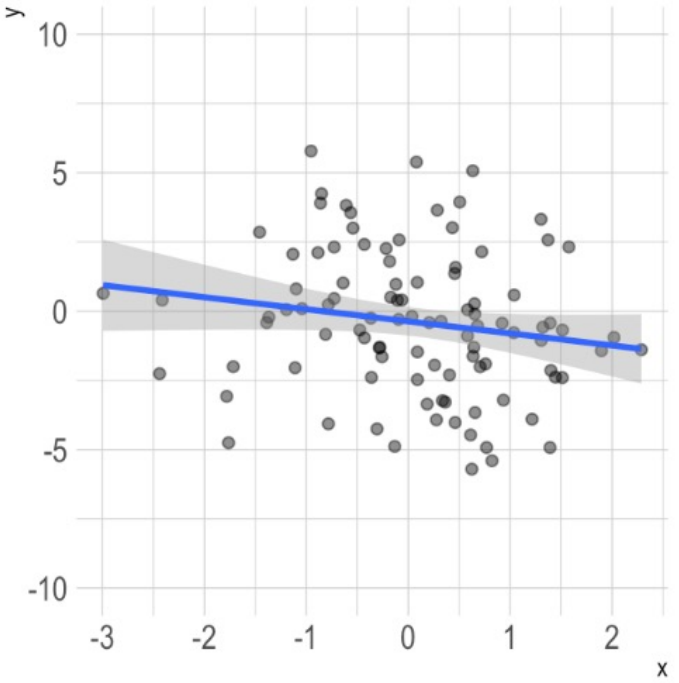
Source: Philipp Mansur

Principles of Data Visualization

1. Tell the Truth
2. Show as much data as possible
3. Keep it simple
4. Do not inverse charts
5. Do not cut chart axes short
6. Find the right aspect-ratio
7. Maximize data-ink ration (but not too much)
8. Be clear about uncertainty







Source: Philipp Mansur

Principles of Data Visualization

1. Tell the Truth
2. Show as much data as possible
3. Keep it simple
4. Do not inverse charts
5. Do not cut chart axes short
6. Find the right aspect-ratio
7. Maximize data-ink ration (but not too much)
8. Be clear about uncertainty
9. Be careful with embellishment

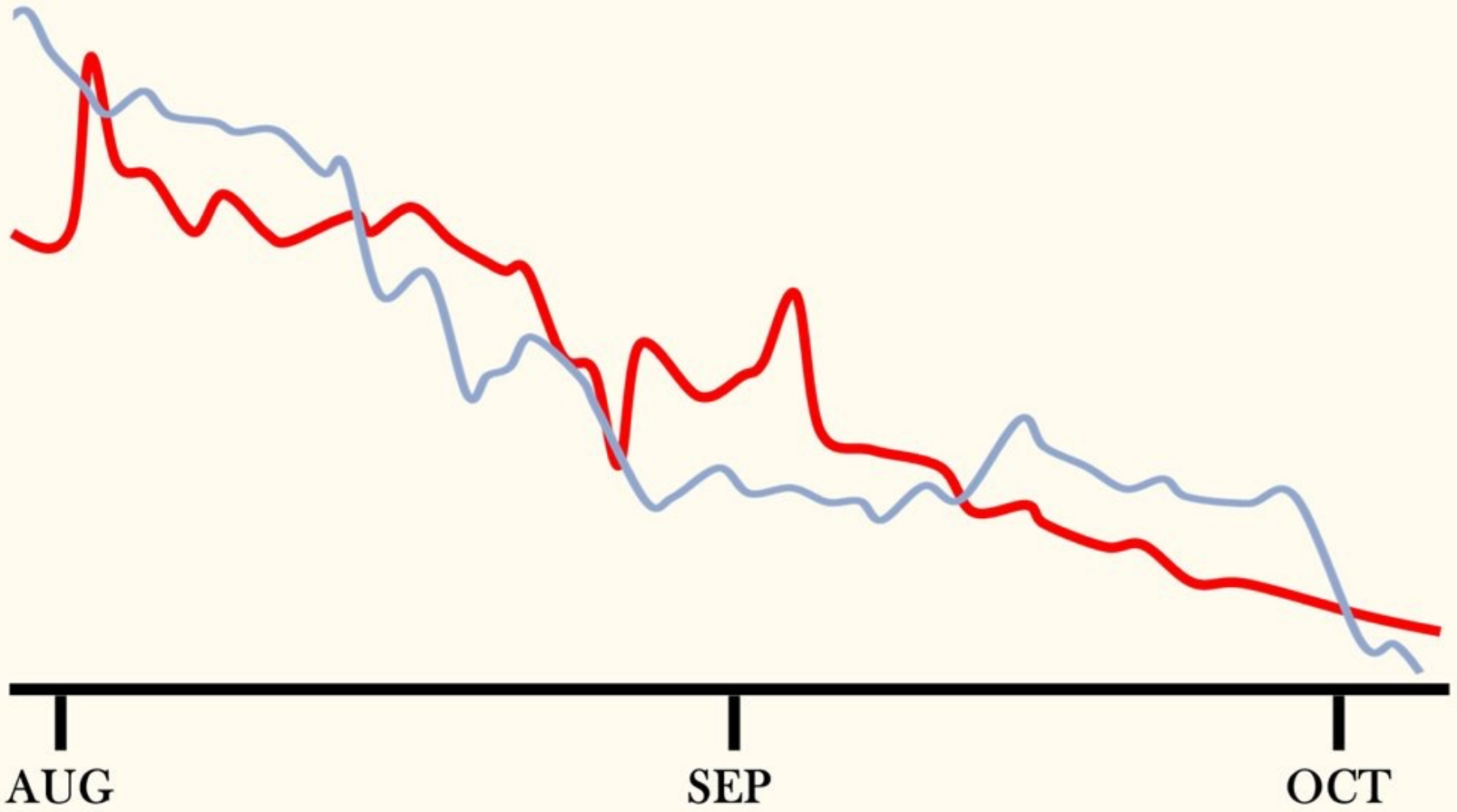
Figure 1.6: `Monstrous Costs' by Nigel Holmes. Also a classic of its kind.

Source: https://socviz.co/lookatdata.html

Principles of Data Visualization

1. Tell the Truth
2. Show as much data as possible
3. Keep it simple
4. Do not inverse charts
5. Do not cut chart axes short
6. Find the right aspect-ratio
7. Maximize data-ink ration (but not too much)
8. Be clear about uncertainty
9. Be careful with embellishment
10. Label your axes and figure

Joe Biden's Approval Rating and Covid Cases/100k in Florida



Source: Philipp Mansur

Ggplot2 basics

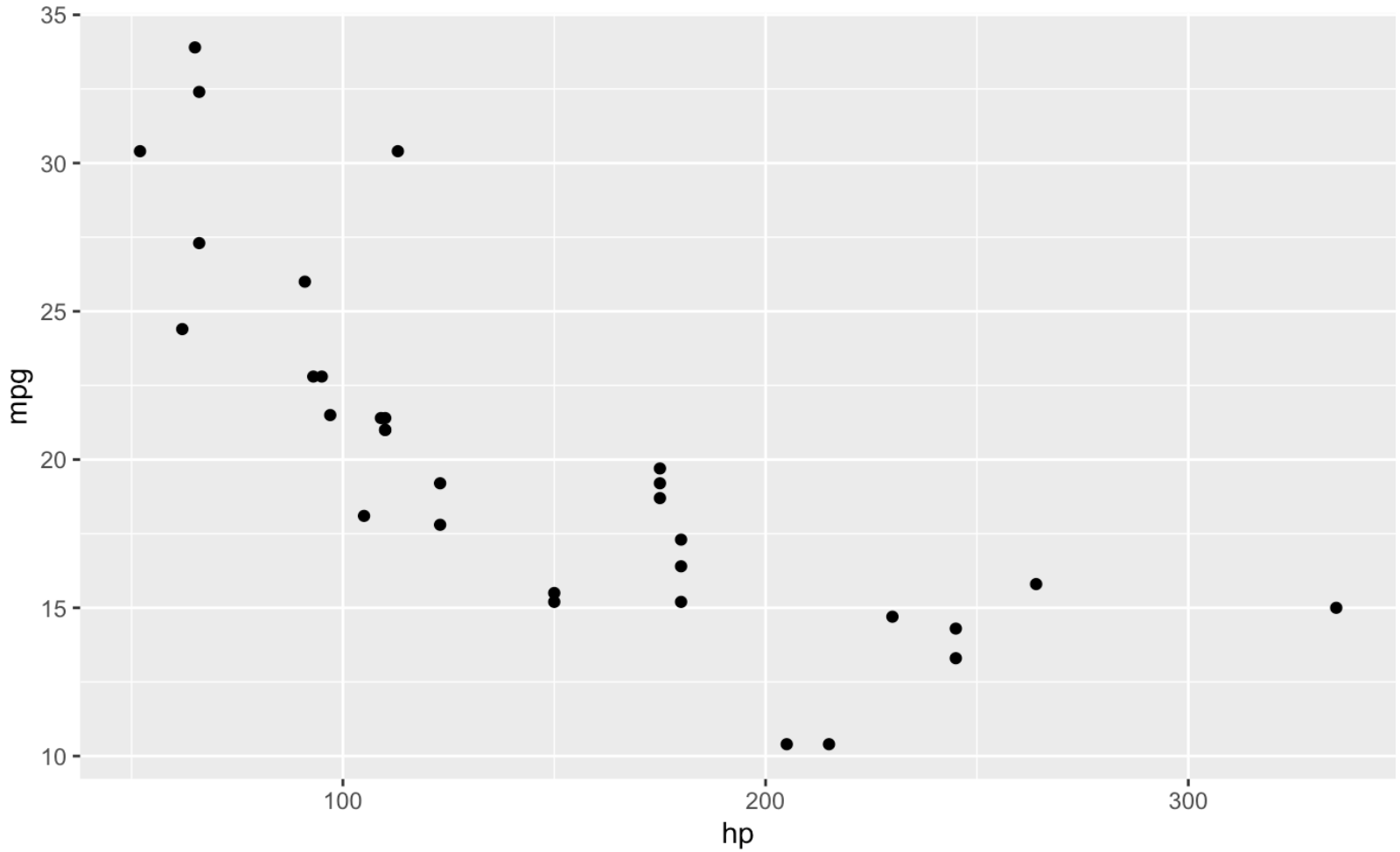
```
library(tidyverse)
data(mtcars)
mtcars
```

	mpg <dbl>	cyl <dbl>	disp <dbl>	hp <dbl>	drat <dbl>	wt <dbl>	qsec <dbl>	vs <dbl>	am <dbl>
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0

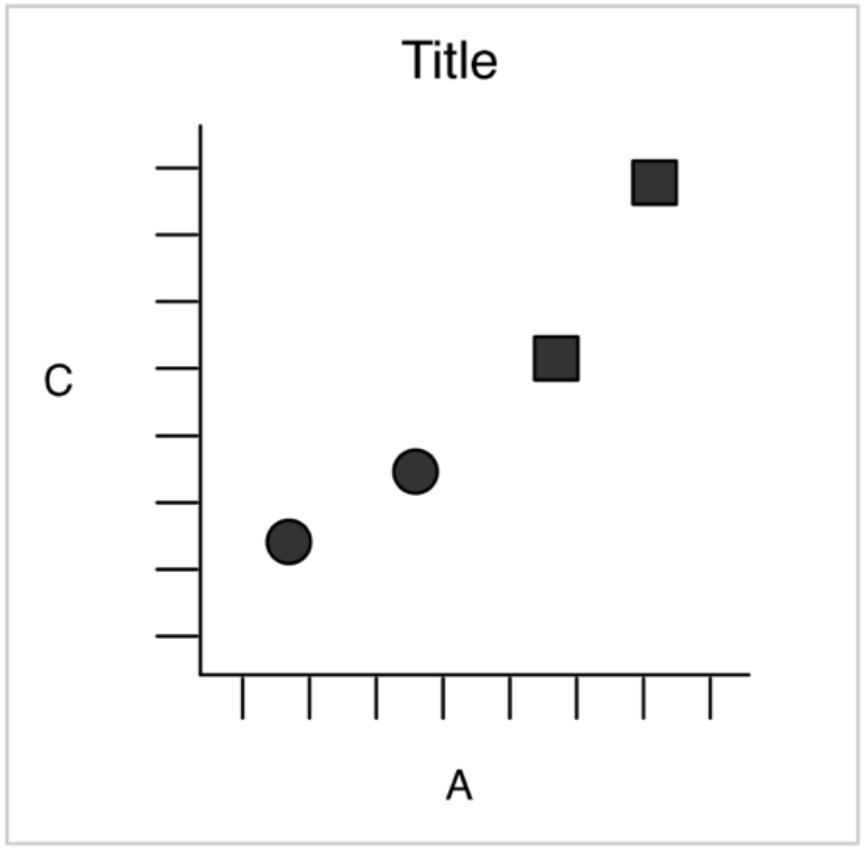
1-10 of 32 rows | 1-10 of 12 columns

Previous 1 2 3 4 Next

```
ggplot(mtcars,
       aes(x = hp, y = mpg)) +
  geom_point()
```

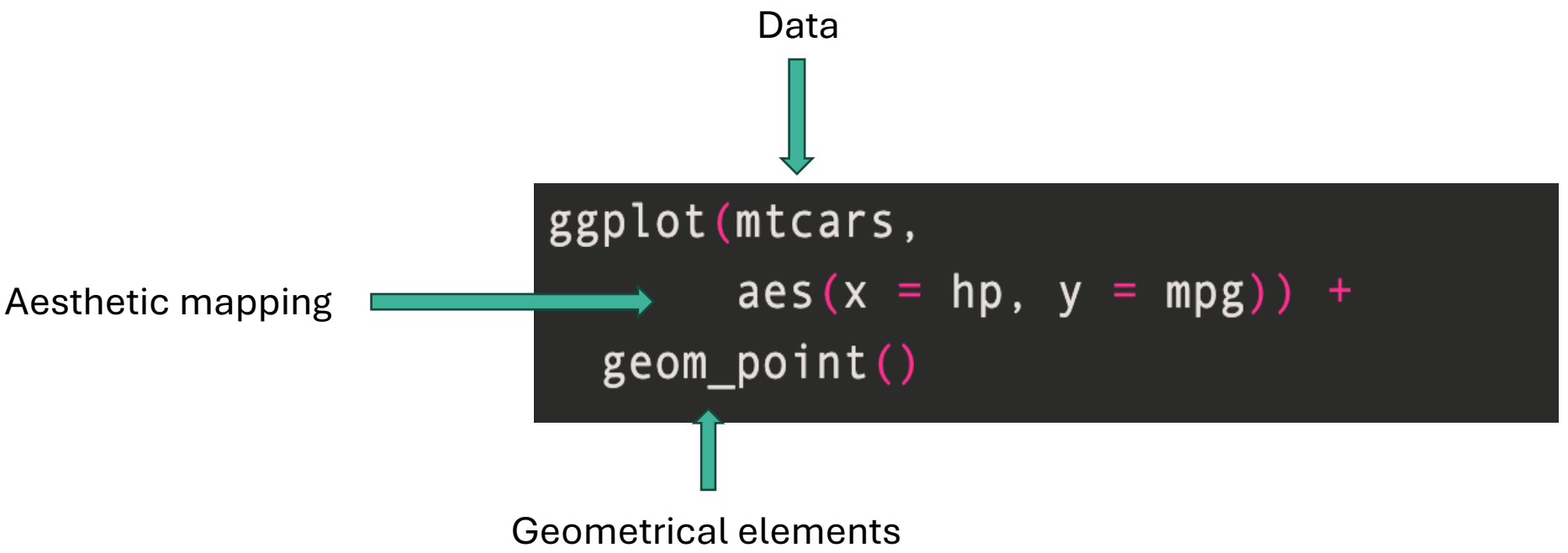


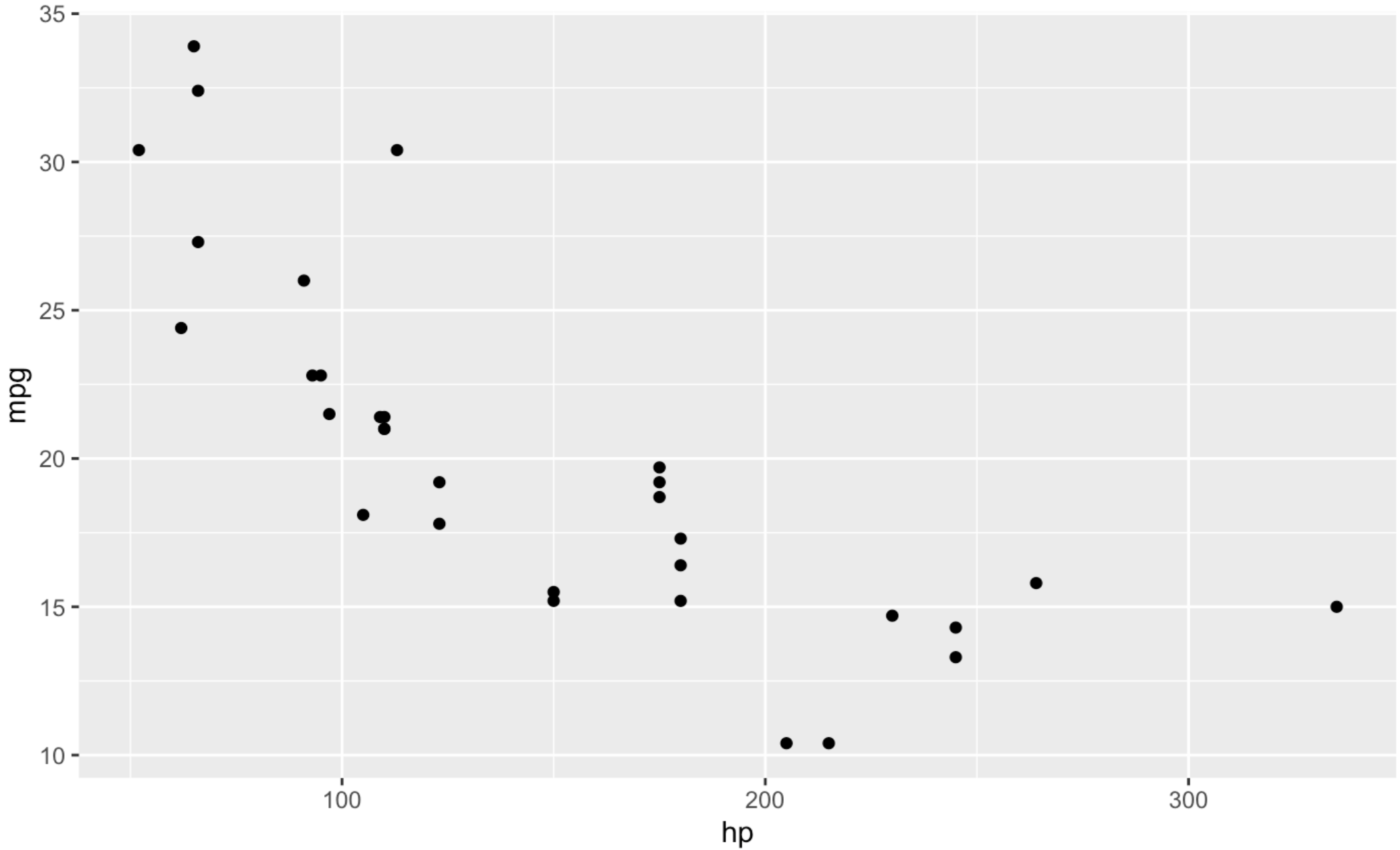
Grammar of Graphics

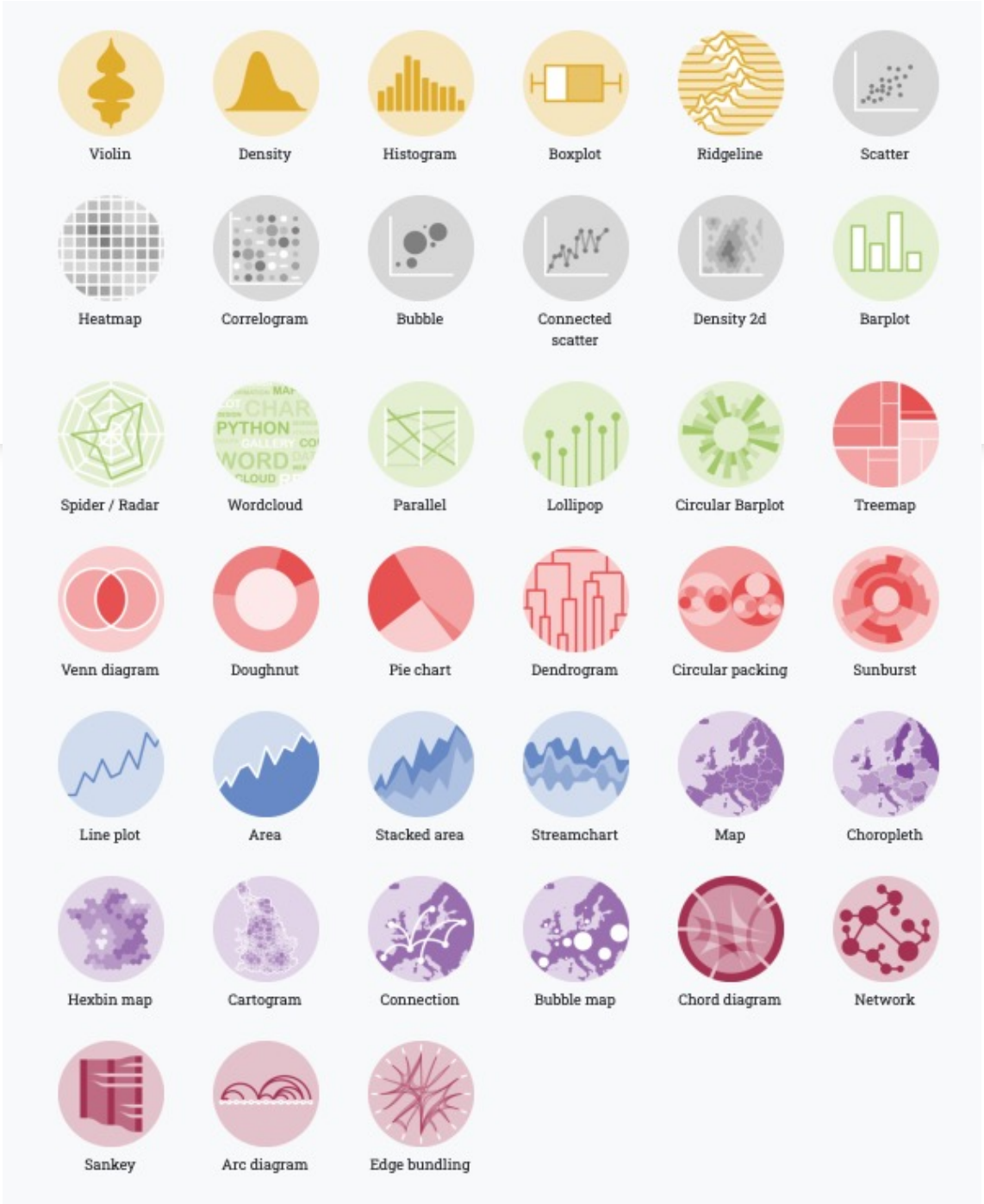


Source: Philipp Mansur

Grammar of Graphics

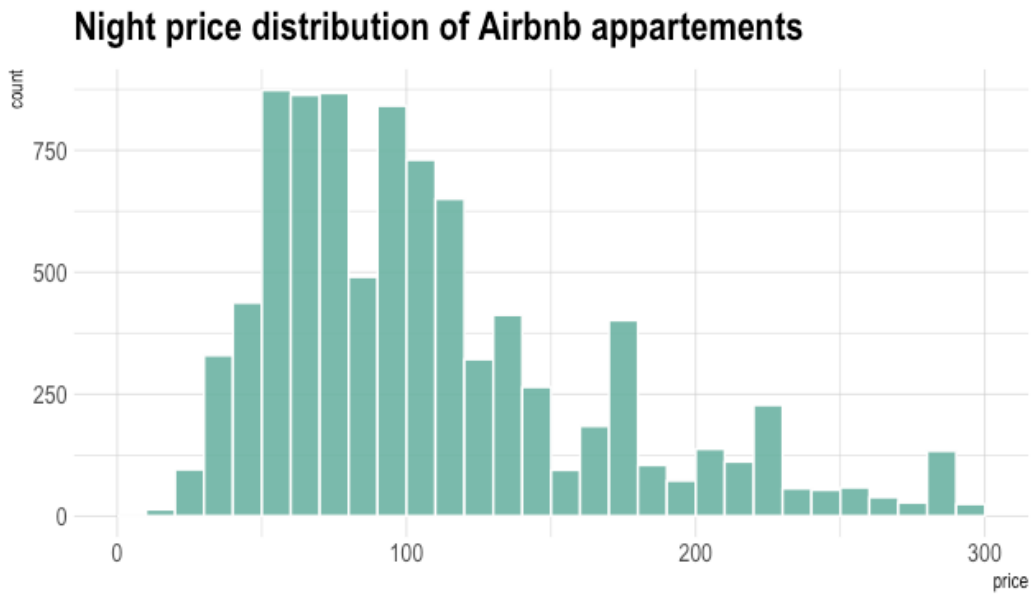


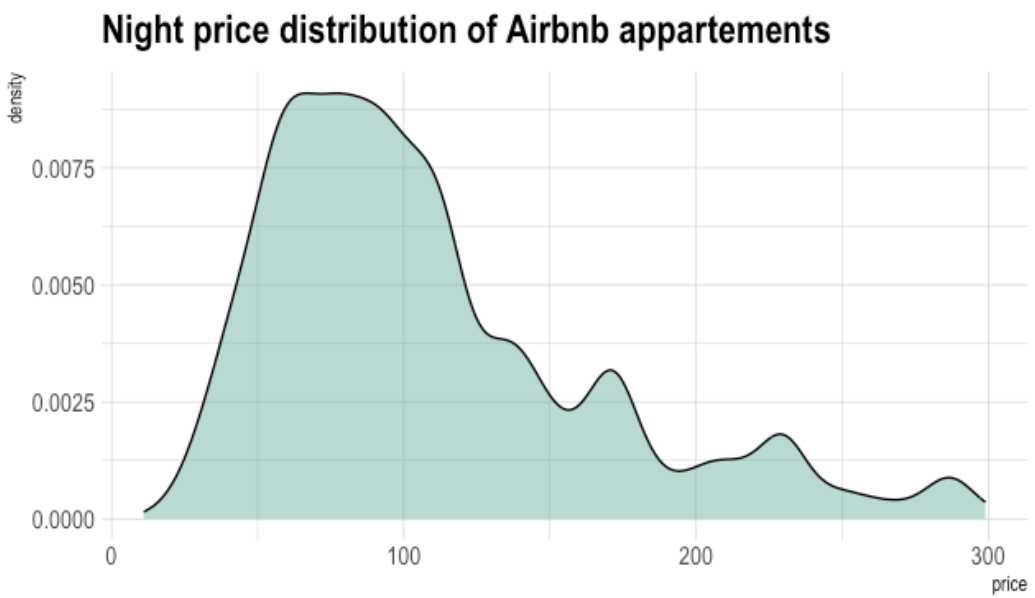

35
30
25
20
15
10
mpg
100
200
300
hp



Source: data-to-vis.com

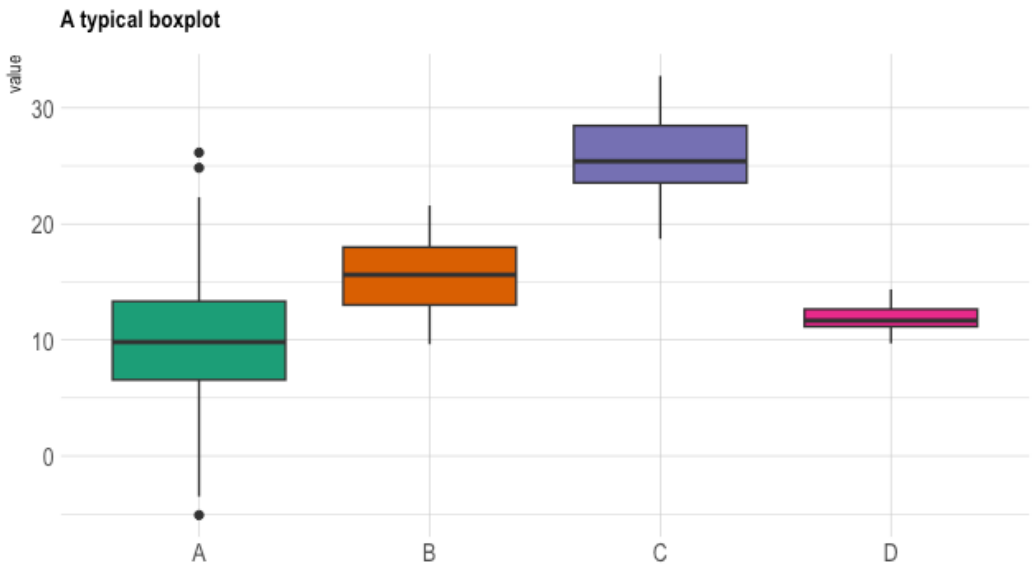
Histograms or Density Plots

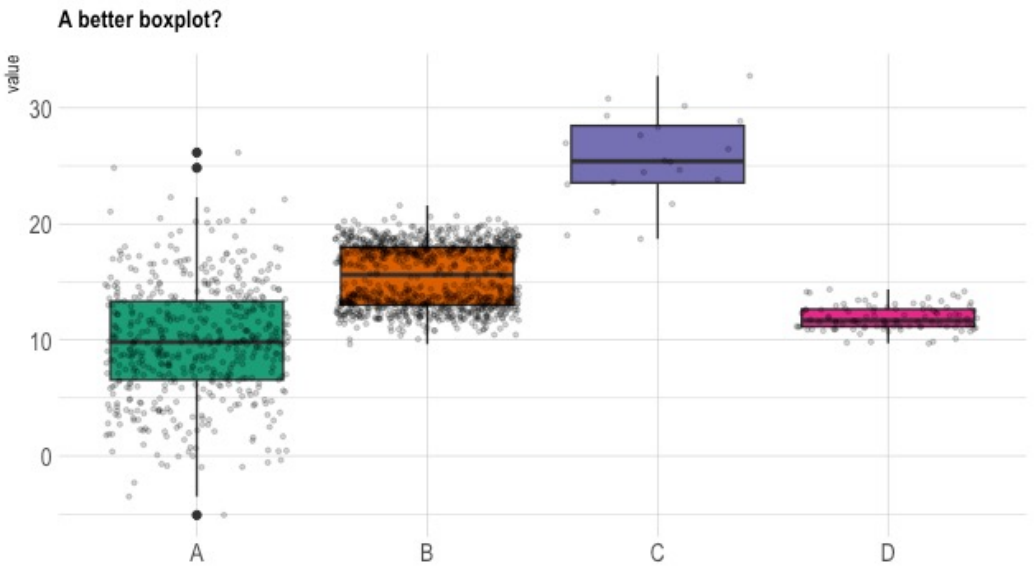




Source: Philipp Mansur

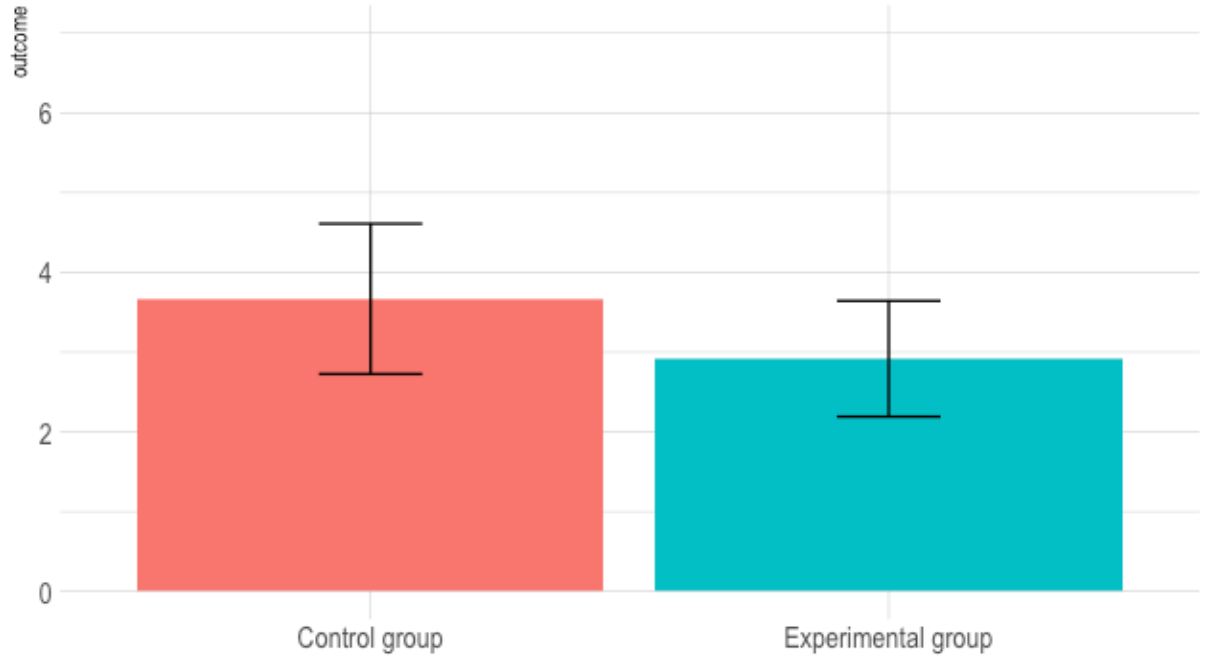
Box Plots





Source: Philipp Mansur

Bar Charts with error-bars

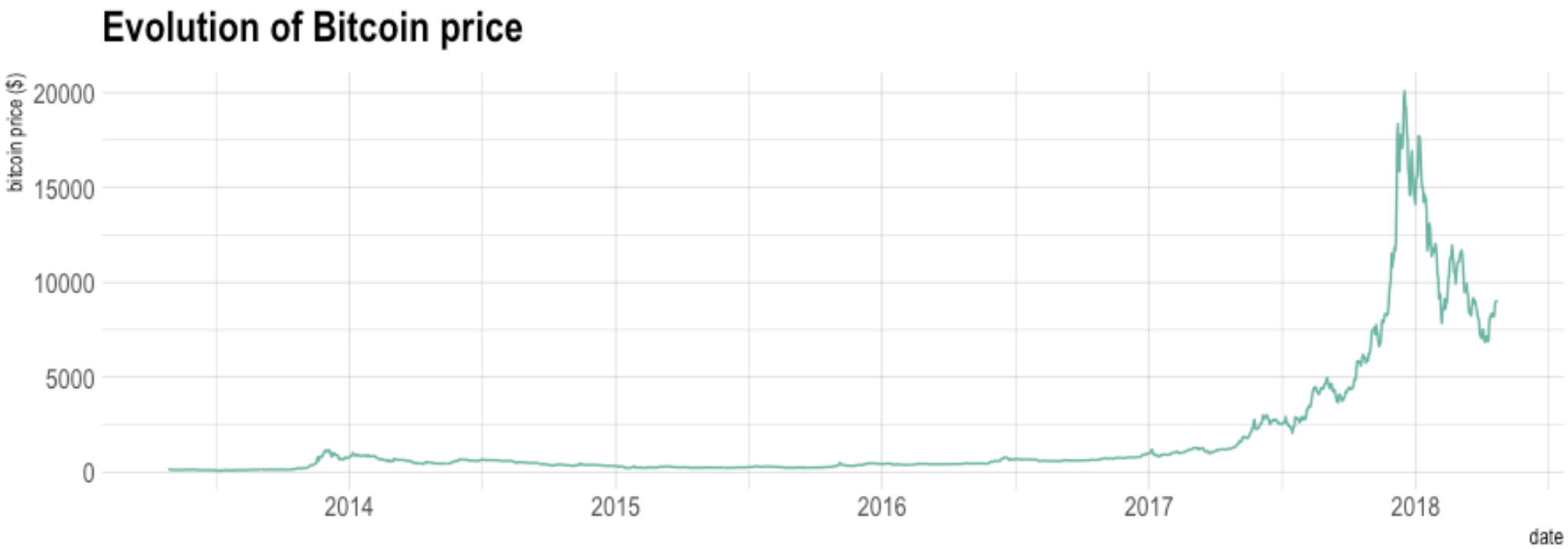


Source: Philipp Mansur

Scatter Plots

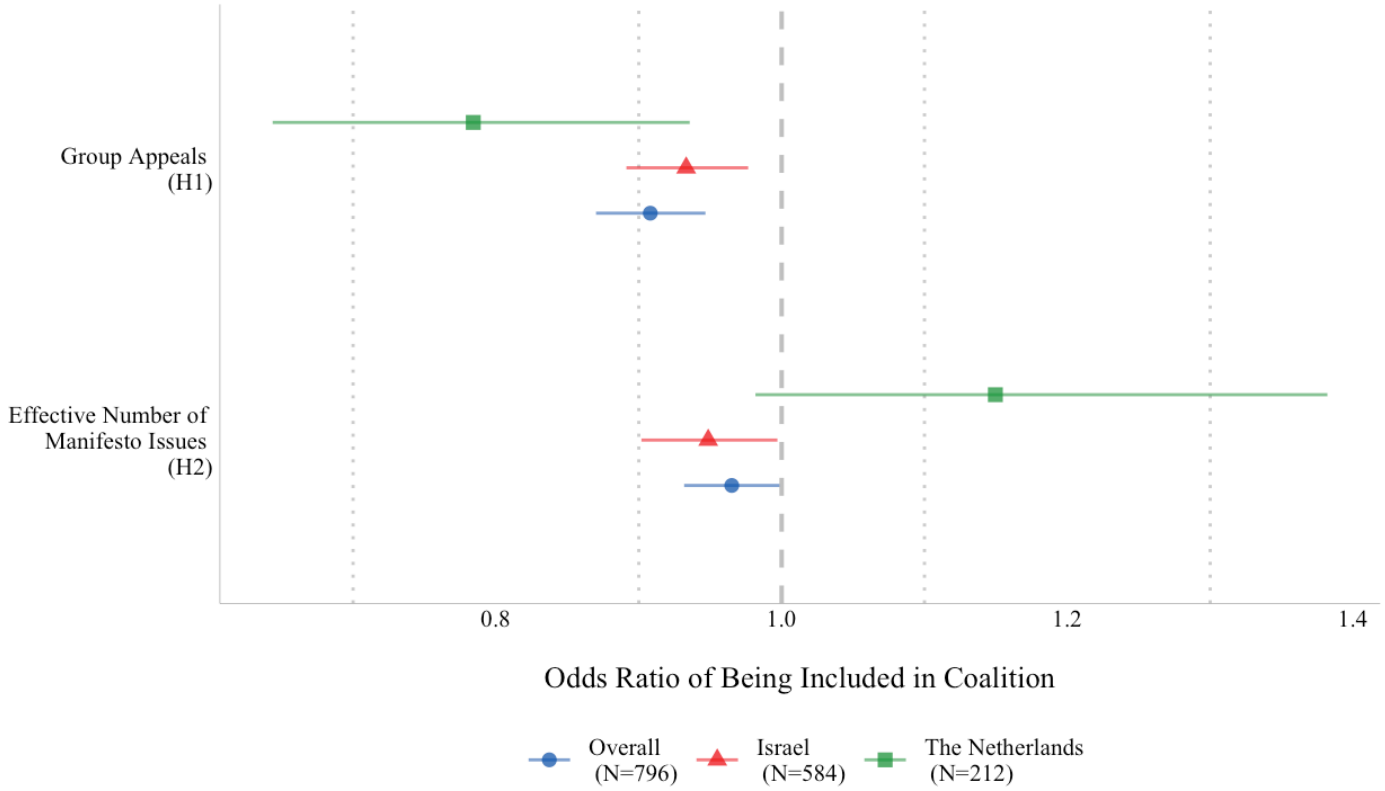
Source: Philipp Mansur

Line Plots



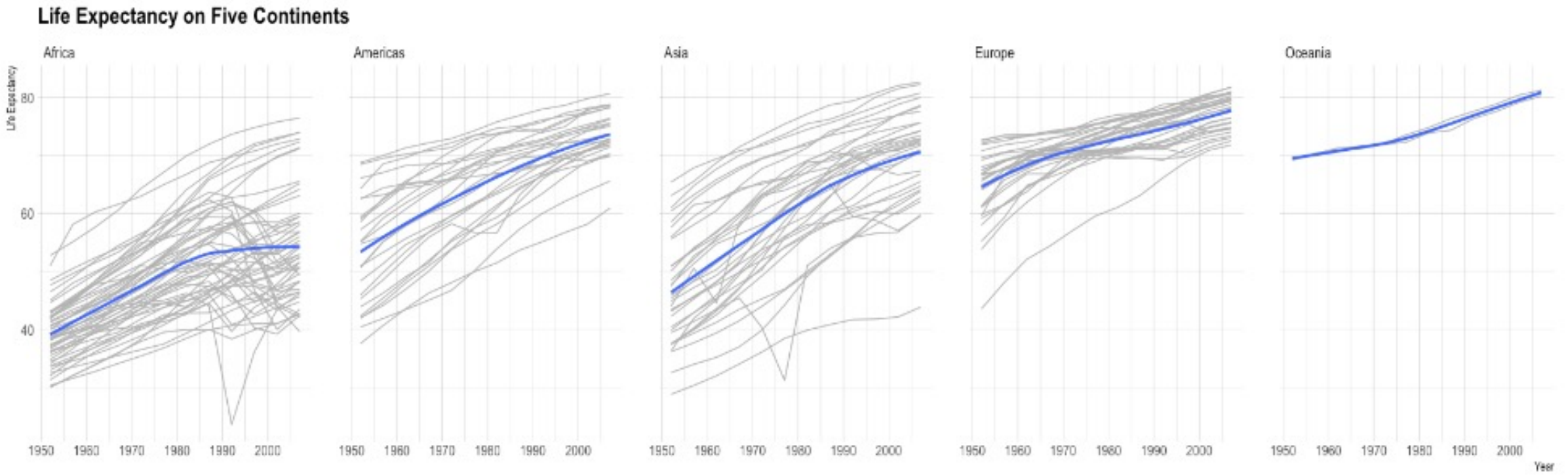
Source: Philipp Mansur

Coefficient Plots



Source: Alona Dolinsky

Facetted Plots

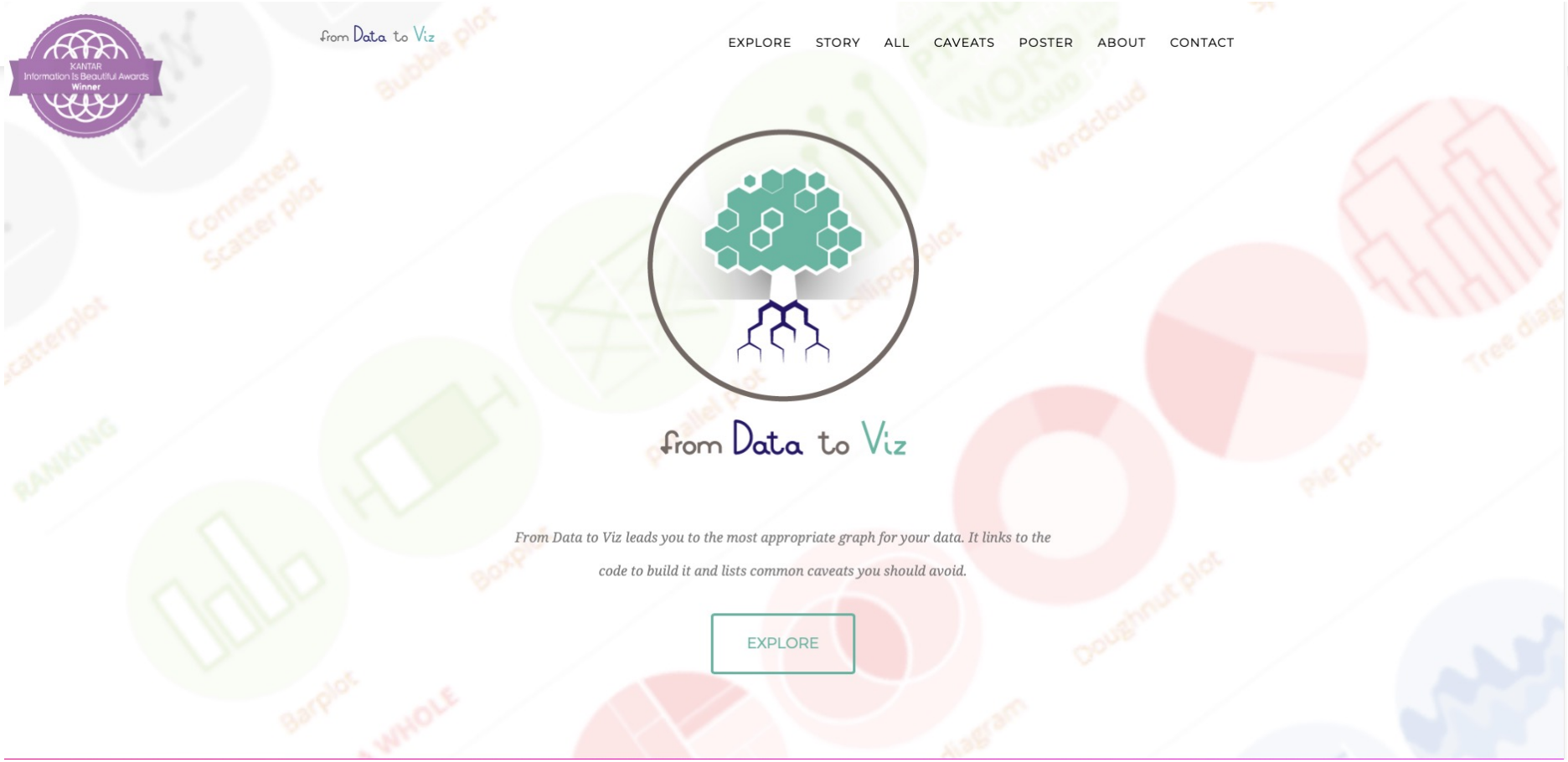


Source: Philipp Mansur

So how do you choose the right graph?

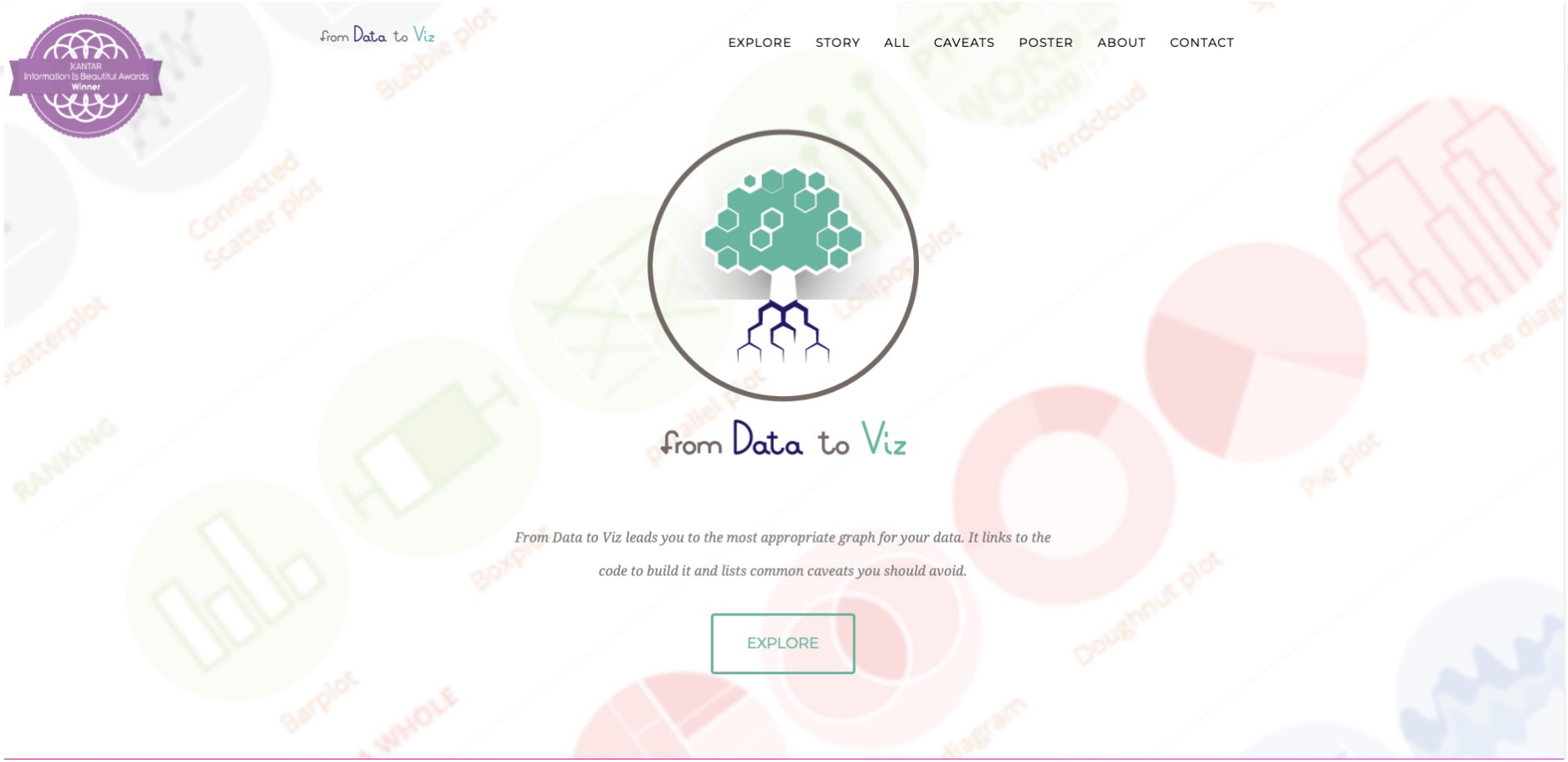
- Some visualizations are better than others
- Making a really good or really useful figure is not just about following the guidelines
- Some visualizations work well not because they are necessarily better looking but because of the way humans process visual information
- Key:
  - Pair appropriate data type to graph type
  - Experiment!
  - Don’t use Pie Charts ☺

Which visualization works with what data?




Break

Which visualization works with what data?



from Data to Viz

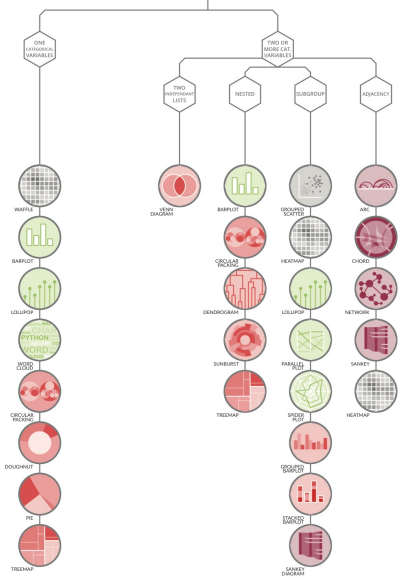
'From Data to Viz' is a classification of chart types based on input data format. It will help you find the perfect chart in three simple steps :

1. Identify what type of data you have.
2. Go to the corresponding decision tree and follow it down to a set of possible charts.
3. Choose the chart from the set that will suit your data and your needs best.

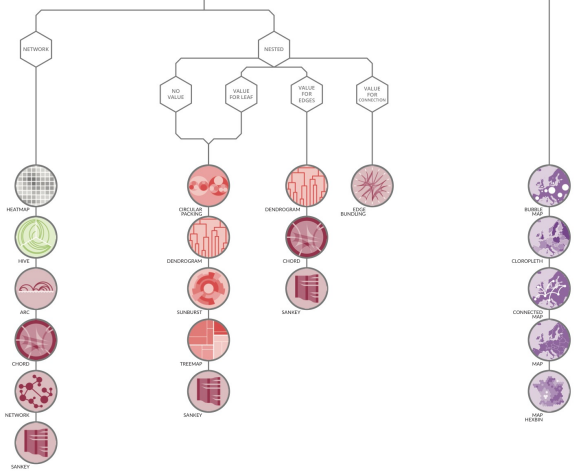
Dataviz is a world with endless possibilities and this project does not claim to be exhaustive. However it should provide you with a good starting point. For an interactive version and much more, visit:

data-to-viz.com

CATEGORIC

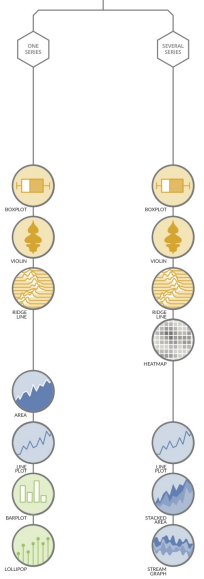


RELATIONAL



MAP

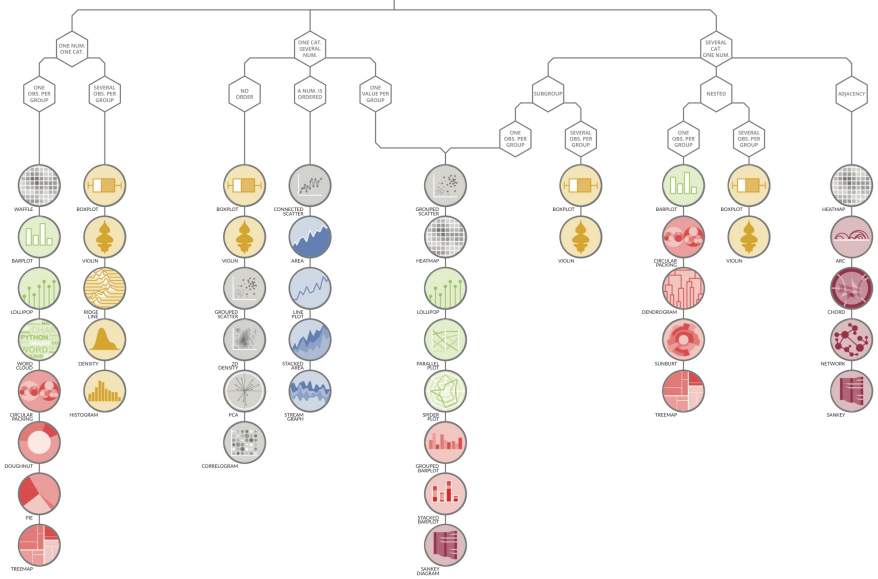
TIME SERIES



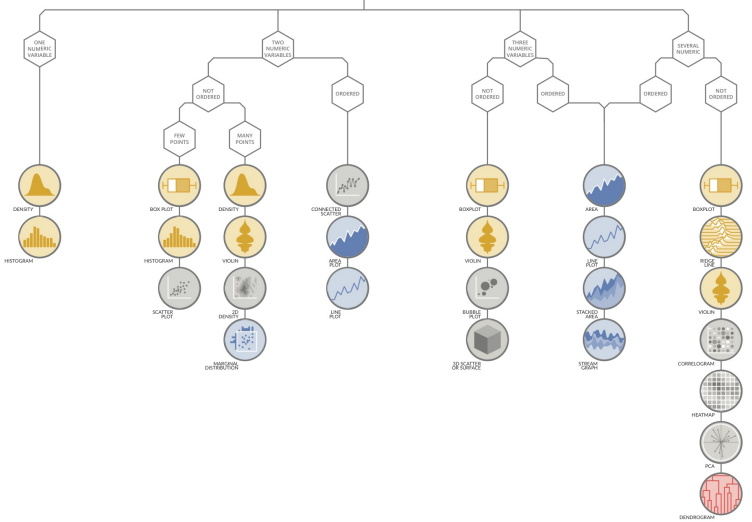
WHAT DO YOU WANT TO SHOW ?

- Distribution
- Correlation
- Ranking
- Part of a whole
- Evolution
- Maps
- Flow

CATEGORIC AND NUMERIC



NUMERIC



Data visualization with ggplot2 : : CHEATSHEET

Basics

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

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +                                    required
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
 stat = <STAT>, position = <POSITION>) +                    Not required, sensible defaults supplied
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

ggplot(data = mpg, **aes(**x = cty, y = hwy**))** Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5**)** Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes Common aesthetic values.

color and **fill** - string ("red", "#RRGGBB")

linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

size - integer (line width in mm)

shape - integer/shape name or a single character ("a")

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

GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))
```

a + geom_blank() and **a + expand_limits()** Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1**)** - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

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

a + geom_polygon(aes(alpha = 50)**)** - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)**)** - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)**)** - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1)**)**
b + geom_hline(aes(yintercept = lat)**)**
b + geom_vline(aes(xintercept = long)**)**

b + geom_segment(aes(yend = lat + 1, xend = long + 1)**)**
b + geom_spoke(aes(angle = 1:1155, radius = 1)**)**

ONE VARIABLE continuous

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)
```

c + geom_area(stat = "bin"**)** x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian"**)** x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot() x, y, alpha, color, fill

c + geom_freqpoly() x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5**)** x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)**)** x, y, alpha, color, fill, linetype, size, weight

discrete

```
d <- ggplot(mpg, aes(fl))
```

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

TWO VARIABLES

both continuous

```
e <- ggplot(mpg, aes(cty, hwy))
```

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point() x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile() x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"**)** x, y, alpha, color, linetype, size

e + geom_smooth(method = lm**)** x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

one discrete, one continuous

```
f <- ggplot(mpg, aes(class, hwy))
```

f + geom_col() x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

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

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

both discrete

```
g <- ggplot(diamonds, aes(cut, color))
```

g + geom_count() x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2**)** x, y, alpha, color, fill, shape, size

continuous bivariate distribution

```
h <- ggplot(diamonds, aes(carat, price))
```

h + geom_bin2d(binwidth = c(0.25, 500)**)** x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d() x, y, alpha, color, group, linetype, size

h + geom_hex() x, y, alpha, color, fill, size

continuous function

```
i <- ggplot(economics, aes(date, unemploy))
```

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

i + geom_line() x, y, alpha, color, group, linetype, size

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

visualizing error

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

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

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width Also **geom_errorbarh()**.

j + geom_linerange() x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

```
data <- data.frame(murder = USArrests$Murder,
                   state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))
```

k + geom_map(aes(map_id = state), map = map**)** **+ expand_limits(**x = map$long, y = map$lat**)** map_id, alpha, color, fill, linetype, size

THREE VARIABLES

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))
```

l + geom_contour(aes(z = z)**)** x, y, z, alpha, color, group, linetype, size, weight

l + geom_contour_filled(aes(fill = z)**)** x, y, alpha, color, fill, group, linetype, size, subgroup

l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE**)** x, y, alpha, fill

l + geom_tile(aes(fill = z)**)** x, y, alpha, color, fill, linetype, size, width

What data do you have?

Histogram

```r
```{r, echo = F}
# Load dataset from github
data <- read.table("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/1_OneNum.csv", header=TRUE)

# Make the histogram
data %>%
  filter(price<300 ) %>%
  ggplot(aes(x=price)) +
    geom_histogram(breaks=seq(0,300,10), fill="#69b3a2", color="white", alpha=0.9) +
    ggtitle("Night price distribution of Airbnb appartements") +
    theme_ipsum()
```
```

Source: Philipp Mansur

Density

```r
```{r, echo = F}
data %>%
  filter(price<300) %>%
  ggplot(aes(x=price)) +
    geom_density(breaks=seq(0,300,10), fill="#69b3a2", alpha=0.5) +
    ggtitle("Night price distribution of Airbnb appartements") +
    theme_ipsum()
```
```

Source: Philipp Mansur

Box Plot

```r
```{r, echo = F}
data <- data.frame(
  name=c( rep("A",500), rep("B",500), rep("B",500), rep("C",20), rep('D', 100)  ),
  value=c( rnorm(500, 10, 5), rnorm(500, 13, 1), rnorm(500, 18, 1), rnorm(20, 25, 4), rnorm(100,
12, 1) )
)

# Plot
data %>%
  ggplot( aes(x=name, y=value, fill=name)) +
    geom_boxplot() +
    scale_fill_brewer(palette = "Dark2") +
    theme_ipsum() +
    theme(
      legend.position="none",
      plot.title = element_text(size=11)
    ) +
    ggtitle("A typical boxplot") +
    xlab("")
```
```

Source: Philipp Mansur

Box Plot with gitter

```r
data %>%
  ggplot( aes(x=name, y=value, fill=name)) +
    geom_boxplot() +
  geom_jitter(alpha = .2, size = .75) +
    scale_fill_brewer(palette = "Dark2") +
    theme_ipsum() +
    theme(
      legend.position="none",
      plot.title = element_text(size=11)
    ) +
    ggtitle("A better boxplot?") +
    xlab("")
```

Source: Philipp Mansur

Bar Charts with error-bars

```r
d <- tibble(group = rep(c("Experimental group", "Control group"), 100),
       outcome = rnorm(200, c(3, 4), c(4, 5)))
```

```r
d %>%
  group_by(group) %>%
  summarize(m = mean(outcome),
            se = psych::describe(outcome)$se,
            ll = m - 1.96*se,
            ul = m + 1.96*se) %>%
  ggplot(aes(x = group, y = m, ymin = ll, ymax = ul, fill = group)) +
  geom_bar(stat = "identity") +
  geom_errorbar(width=.2, position=position_dodge(.9)) +
  ylim(0, 7) +
  theme_ipsum() +
  labs(x = "", y = "outcome") +
  theme(legend.position = "none")
```

Source: Philipp Mansur

Scatter Plot

```r
# Load dataset from github
data <- read.table("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/2_TwoNum.csv", header=T, sep=",") %>% dplyr::select(GrLivArea, SalePrice) %>% as_tibble
```

```r
# Linear regression model
m <- summary(lm(SalePrice/1000 ~ GrLivArea, data))
round(m$coef, 2)
```

Source: Philipp Mansur

Scatter Plot

```r
```{r, echo = F}
# plot
data %>%
  ggplot( aes(x=GrLivArea, y=SalePrice/1000)) +
    geom_point(color="#69b3a2", alpha=0.8) +
    ggtitle("Ground living area partially explains sale price of apartments") +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=12)
    ) +
    ylab('Sale price (k$)') +
    xlab('Ground living area')
```
```

Source: Philipp Mansur

Scatter Plot with regression line

```r
```{r, echo = F}
data %>%
  ggplot( aes(x=GrLivArea, y=SalePrice/1000)) +
    geom_point(color="#69b3a2", alpha=0.8, size = .7) +
    geom_smooth(linetype = "dashed", color = "black") +
    geom_smooth(method = "lm", color = "red", se = F) +
    ggtitle("Ground living area partially explains sale price of apartments") +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=12)
    ) +
    ylab('Sale price (k$)') +
    xlab('Ground living area')
```
```

Source: Philipp Mansur

Line Plots

```r
```{r, echo = F, fig.height=6, fig.width = 18}

library(gapminder)
p <- ggplot(data = gapminder, mapping = aes(x = year, y = lifeExp))
p + geom_line(color="gray70", aes(group = country)) +
    geom_smooth(size = 1.1, method = "loess", se = FALSE) +
    facet_wrap(~ continent, ncol = 5) +
    labs(x = "Year",
         y = "Life Expectancy",
         title = "Life Expectancy on Five Continents") +
  theme_ipsum()
```
```

Source: Philipp Mansur

Coefficient Plots

```
(d %>%
    add_case(il) %>%
    add_case(nl) %>%
    mutate(id = factor(id,
                       levels = c("Overall \n (N=796)",
                                  "Israel \n (N=584)",
                                  "The Netherlands \n (N=212)"))) %>%
    filter(term == "Group Appeals \n (H1)" |
             term == "Effective Number of \n Manifesto Issues \n (H2)") %>%
    ggplot(aes(x = term, y = estimate,
               ymax= upper, ymin = lower,
               shape = id, color = id)) +
    geom_point(size = 2, position = position_dodge(.5)) +
    geom_errorbar(width = 0, alpha = .6, position = position_dodge(.5)) +
    labs(x = "",
         y = "Odds Ratio of Being Included in Coalition") +
    theme_ipsum() +
    geom_hline(yintercept = 1, linetype = "dashed",
               color = "gray75", linewidth = .7) +
    theme(axis.title.x = element_text(color = "black", size = 10, hjust = 0.5, margin = margin(t = 12))) +
    theme(axis.title.y = element_text(color = "black", size = 10, hjust = 0.5, margin = margin(t = 12))) +
    theme(axis.text.x = element_text(color = "black", size = 8)) +
    theme(axis.text.y = element_text(color = "black", size = 8)) +
    theme(axis.line.x = element_line(linewidth = 0.1, color = "gray54")) +
    theme(axis.line.y = element_line(linewidth = 0.1, color = "gray54")) +
    theme(panel.grid.minor = element_line(linewidth = 0.5, linetype = "dotted")) +
    theme(panel.grid.major = element_blank()) +
    coord_flip() +
    theme(legend.position = "bottom",
          legend.text = element_text(size = 8),
          legend.title = element_blank()) +
    scale_color_manual(values = fig_cols))
```

Source: Alona Dolinsky

Facetted Plots

```r
```{r, echo = F, fig.height=6, fig.width = 18}

library(gapminder)
p <- ggplot(data = gapminder, mapping = aes(x = year, y = lifeExp))
p + geom_line(color="gray70", aes(group = country)) +
    geom_smooth(size = 1.1, method = "loess", se = FALSE) +
    facet_wrap(~ continent, ncol = 5) +
    labs(x = "Year",
         y = "Life Expectancy",
         title = "Life Expectancy on Five Continents") +
  theme_ipsum()
```
```

Source: Philipp Mansur