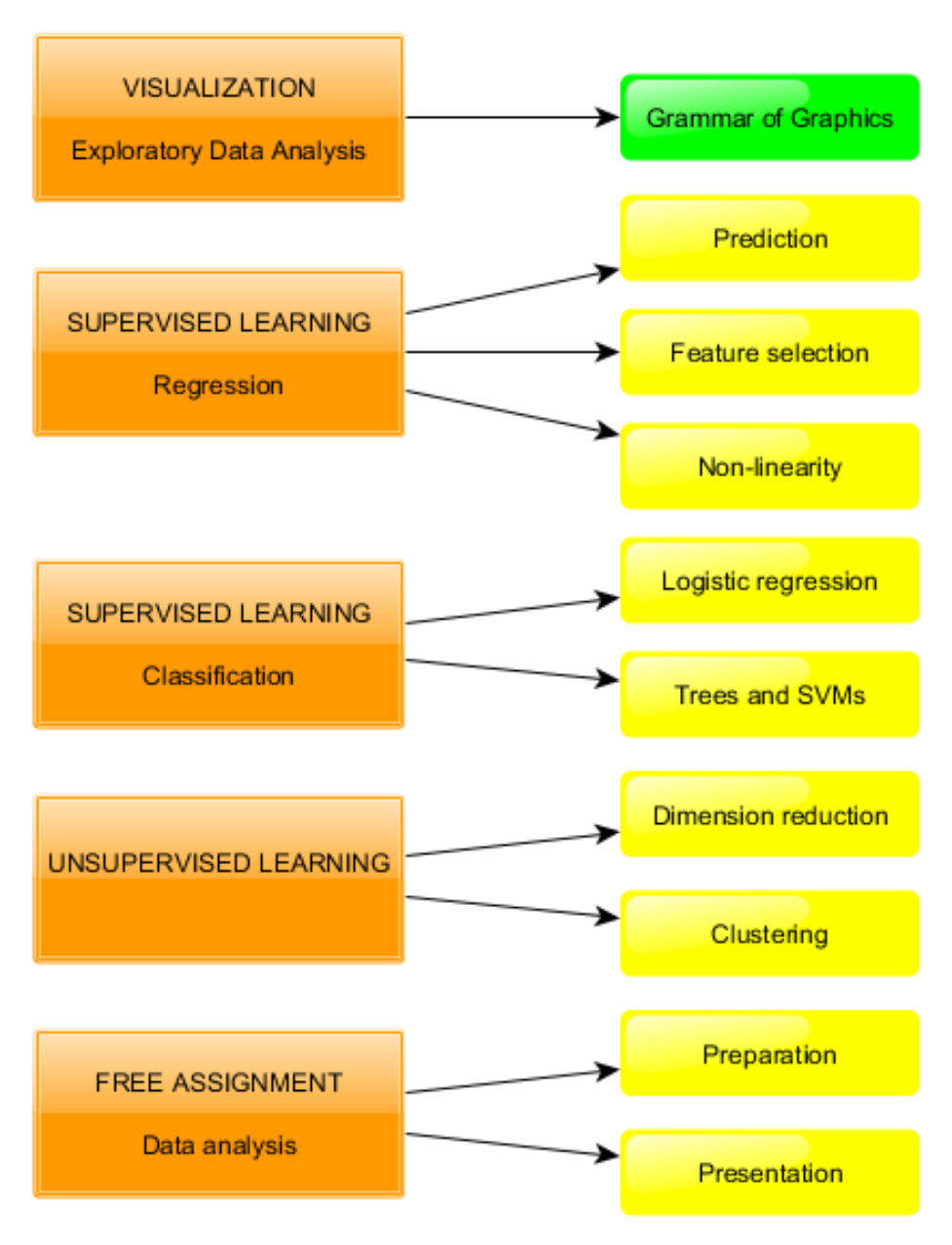
Visualization

Grammar of Graphics

Maarten Cruyff

Program


VISUALIZATION
Exploratory Data Analysis
Grammar of Graphics
SUPERVISED LEARNING
Regression
Prediction
Feature selection
Non-linearity
SUPERVISED LEARNING
Classification
Logistic regression
Trees and SVMs
UNSUPERVISED LEARNING
Dimension reduction
Clustering
FREE ASSIGNMENT
Data analysis
Preparation
Presentation

Content

1. Data visualization
2. Grammar of Graphics
3. Grammar of Data Manipulation
4. Tufte's Principles of Graphical Excellence

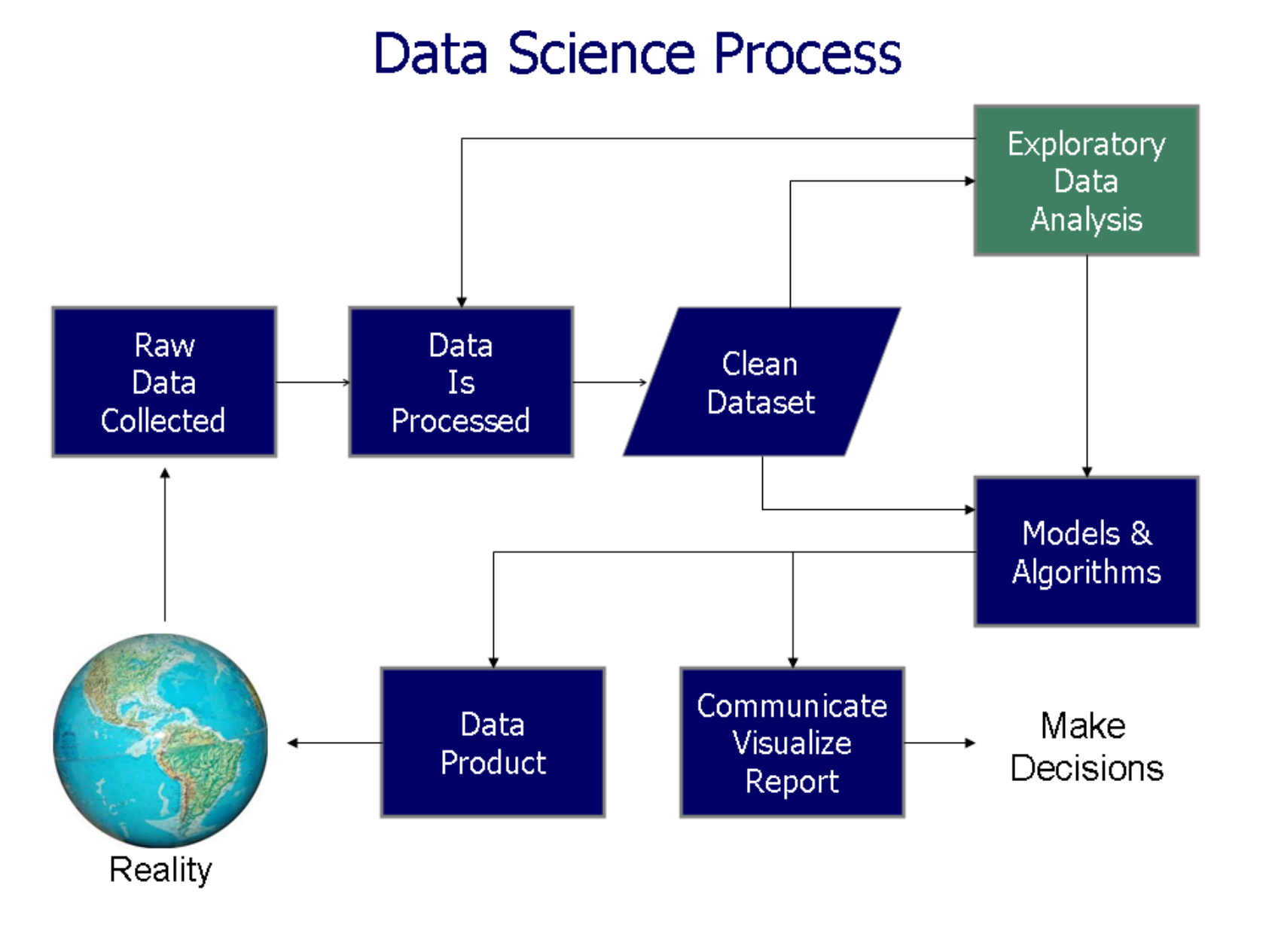
What's data visualization?

Communication of data by encoding it as visual objects, i.e.

- dots, lines, bars, etc.

to make data more accessible, understandable and usable.

. . . an integral part of data science process



. . . an art

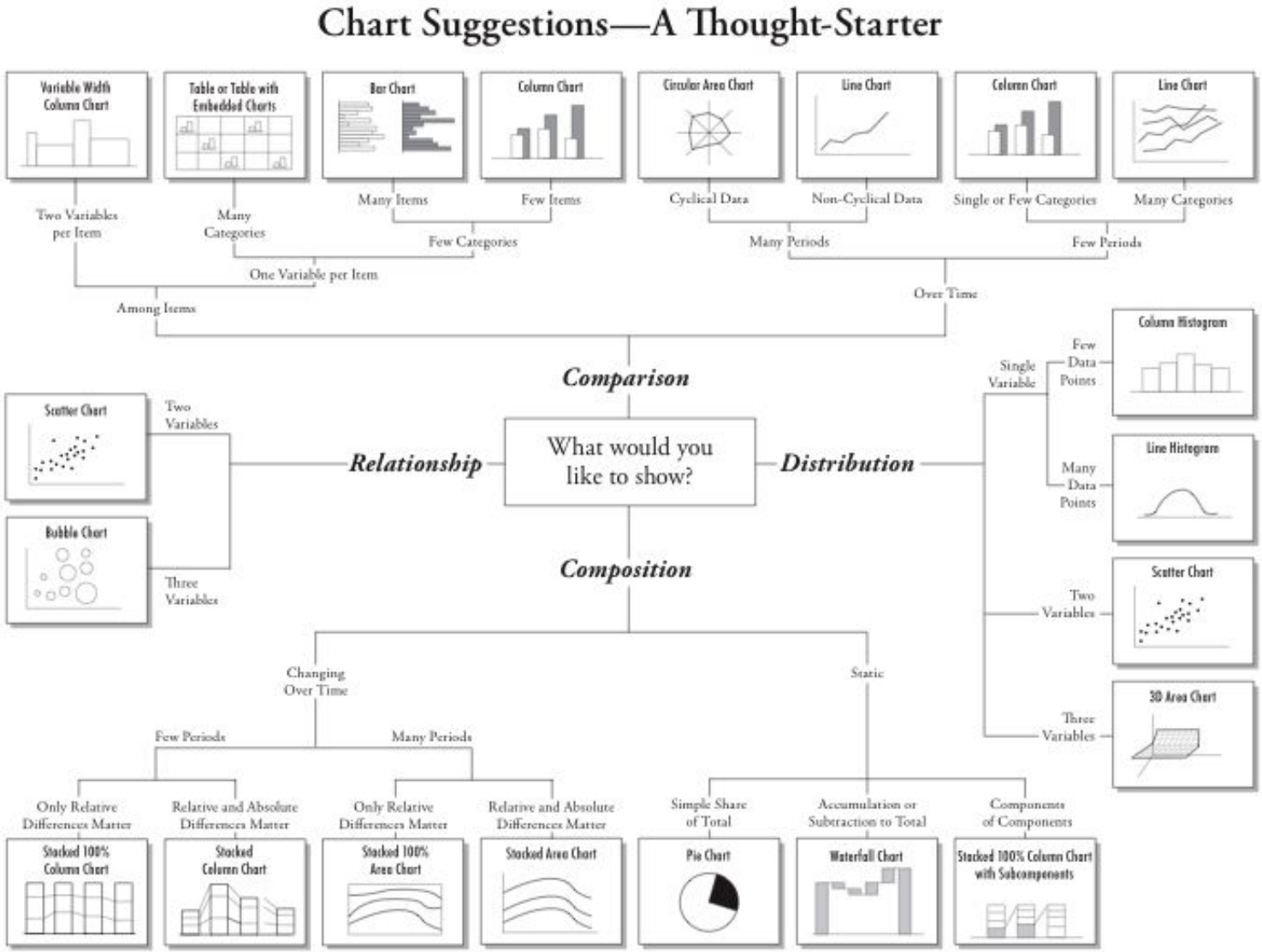


Chart Suggestions—A Thought-Starter
Variable Width Column Chart
Table or Table with Embedded Charts
Bar Chart
Column Chart
Circular Area Chart
Line Chart
Column Chart
Line Chart
Two Variables per Item
Many Categories
Many Items
Few Items
Cyclical Data
Non-Cyclical Data
Single or Few Categories
Many Categories
Few Categories
Many Periods
Few Periods
One Variable per Item
Over Time
Among Items
Comparison
Scatter Chart
Two Variables
Bubble Chart
Three Variables
Relationship
What would you like to show?
Distribution
Composition
Single Variable
Few Data Points
Many Data Points
Two Variables
Three Variables
Column Histogram
Line Histogram
Scatter Chart
3D Area Chart
Changing Over Time
Static
Few Periods
Many Periods
Only Relative Differences Matter
Relative and Absolute Differences Matter
Only Relative Differences Matter
Relative and Absolute Differences Matter
Simple Share of Total
Accumulation or Subtraction to Total
Components of Components
Stacked 100% Column Chart
Stacked Column Chart
Stacked 100% Area Chart
Stacked Area Chart
Pie Chart
Waterfall Chart
Stacked 100% Column Chart with Subcomponents
© 2006 A. Abela — a.v.abela@gmail.com

Why data visualization?

Human brain excels in distinguishing differences in

- length
- shape
- orientation
- hue

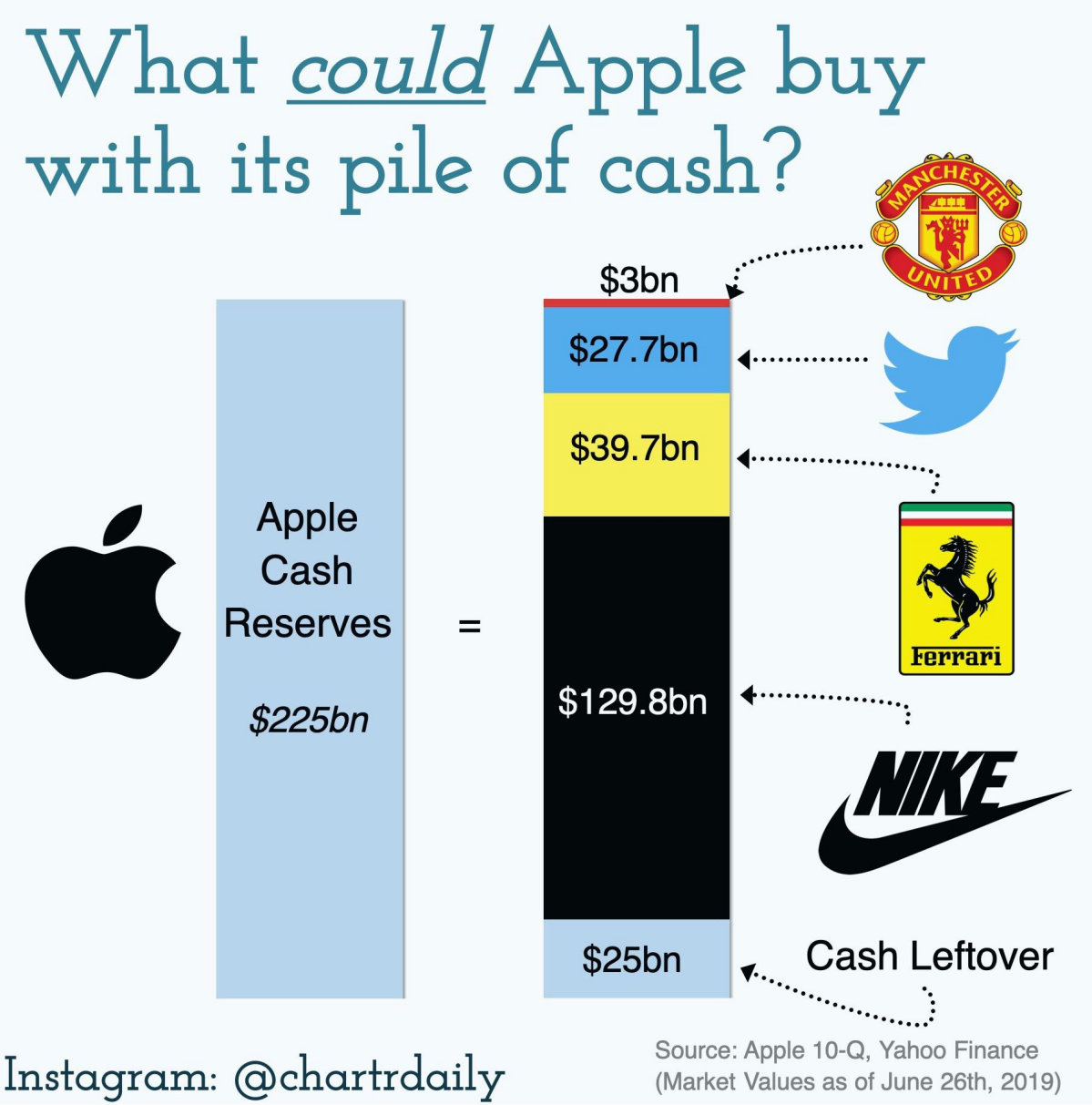
Brain processes graphical information better than text:

- Retina: 10^6 or 10^7 bits/sec
- Reading: 10^2 to 10^3 bits/sec

Early example

John Snow finds source of 1854 Soho cholera outbreak (not miasma)

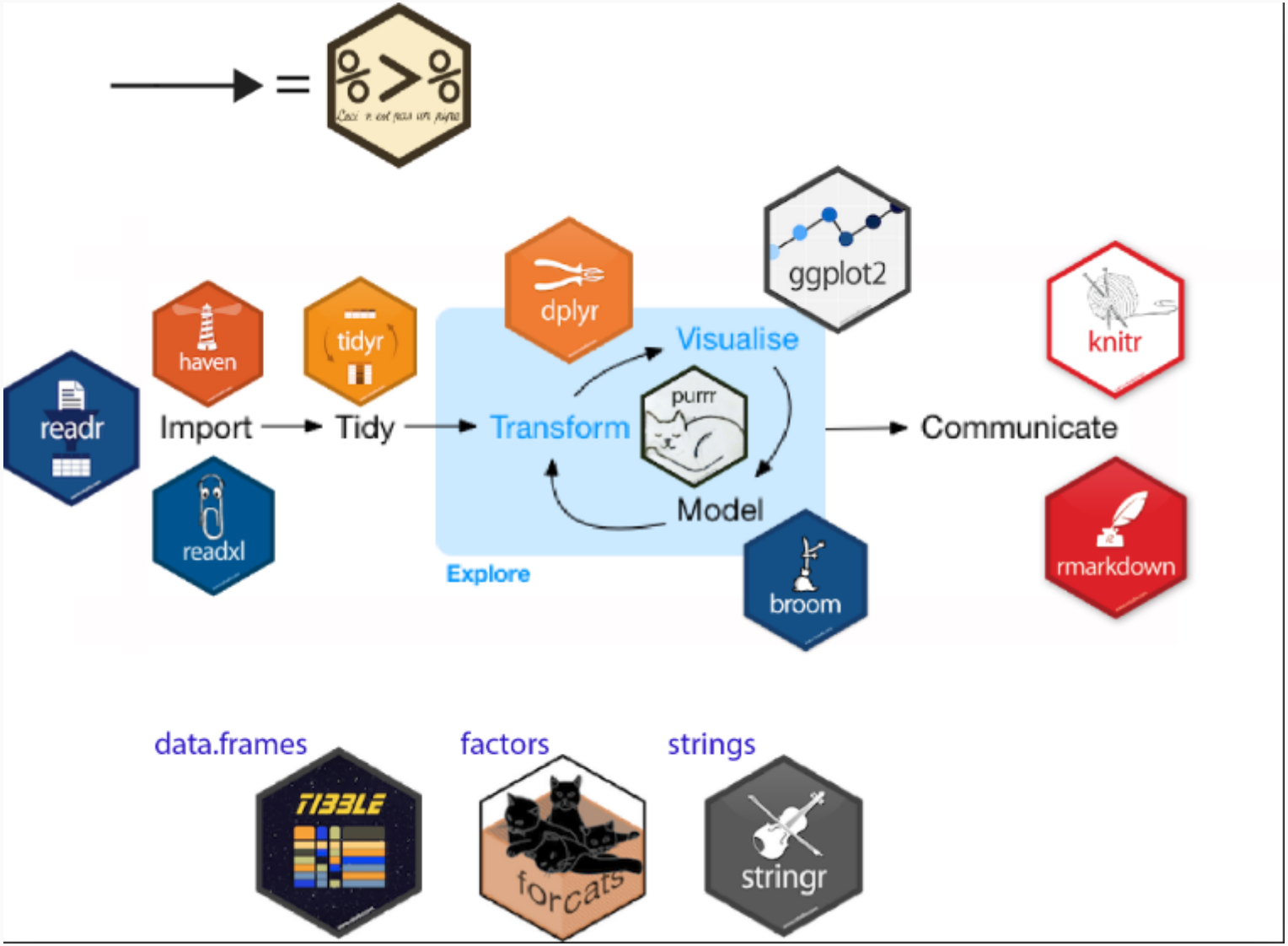
Recent example



Grammar of Graphics

Data science in R

Packages and pipes



Packages for data visualization

`ggplot2`

- package for making plots
- based on the *grammar of graphics*

`dplyr`

- package for data preparation
- based on the *grammar of data manipulation*

Both packages work beautifully together

- with pipe operator %>%

Grammar of Graphics

Buidling plots layer-by-layer

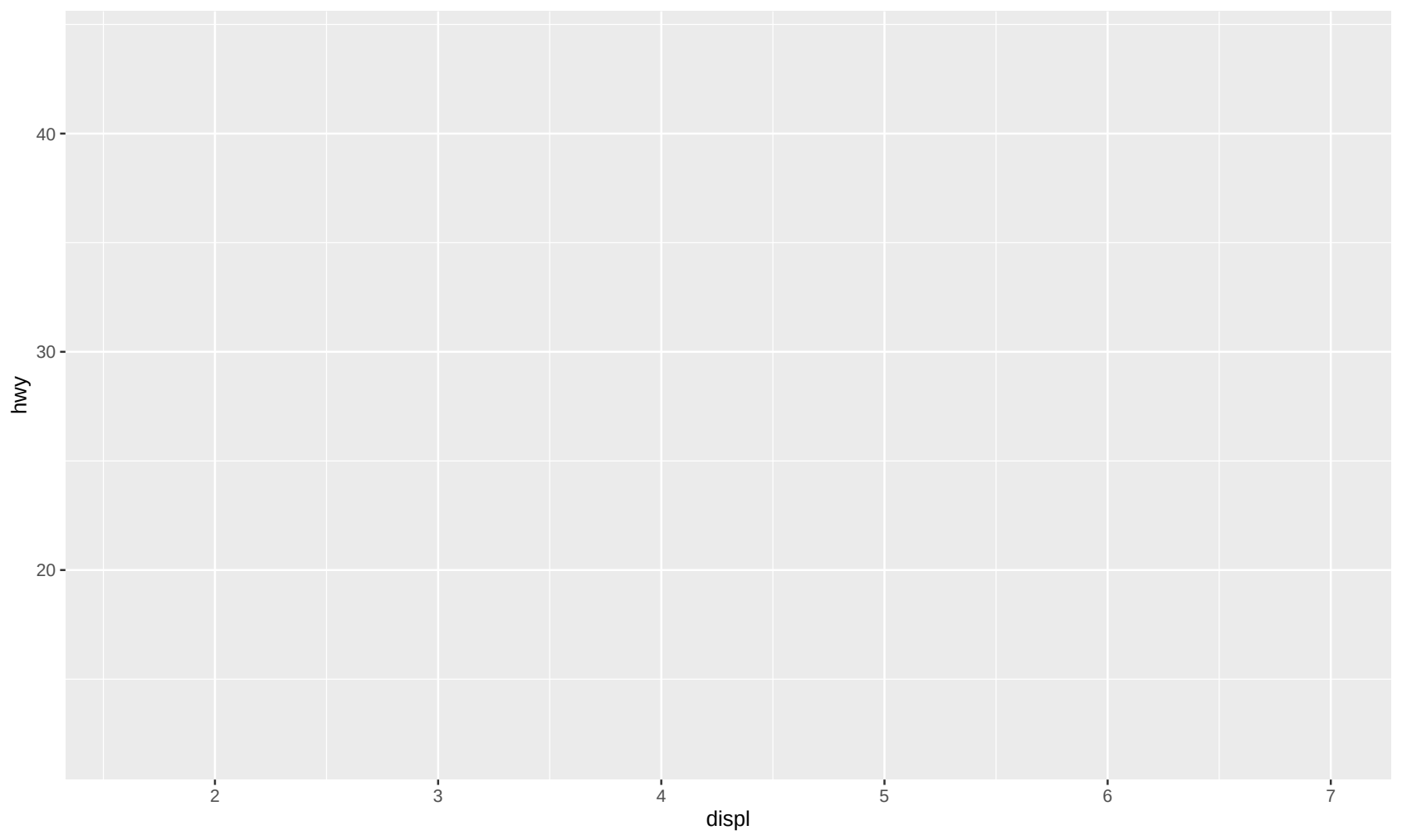
Describes all the non-data ink	Theme
Plotting space for the data	Coordinates
Statistical models & summaries	Statistics
Rows and columns of sub-plots	Facets
Shapes used to represent the data	Geometries
Scales onto which data is mapped	Aesthetics
The actual variables to be plotted	Data

Example of ggplot2 code

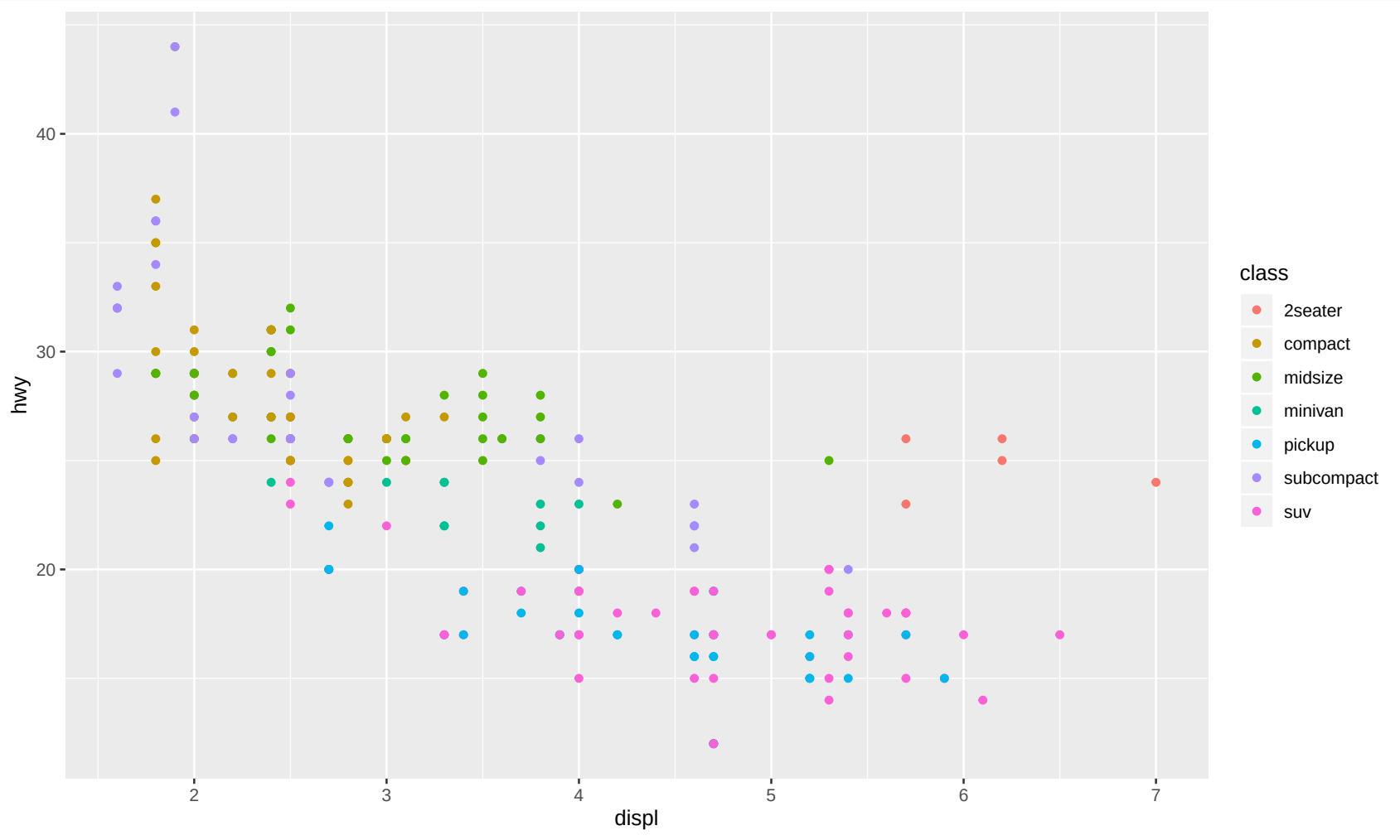
```
ggplot(data    = <data> ,
       mapping = aes(x = <variable>, y = <variable>)) +
  geom_point(aes(col = <variable>)) +
  geom_smooth() +
  facet_grid(rows = vars(<variable>)) +
  theme_minimal()
```

- aes variable to x- and y-axis
- geom_... type of plot (point, density, bar, histogram, etc.)
- aes(col = <variable>) color points by scores/levels of
- facet_grid produce plot for each level vars(<variable>)
- theme_... changes appearance of plot

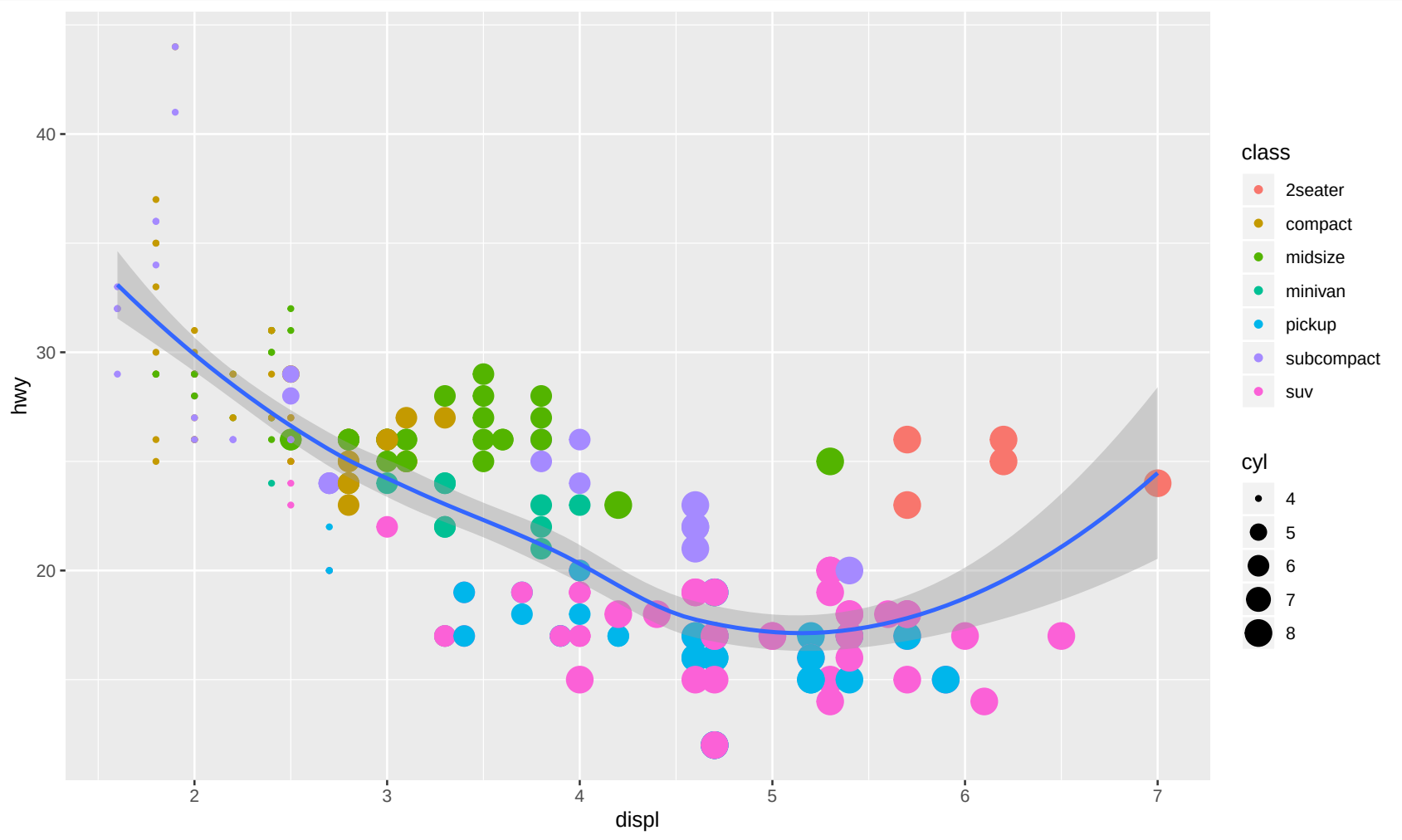
```
ggplot(data = mpg, aes(x = displ, y = hwy))
```



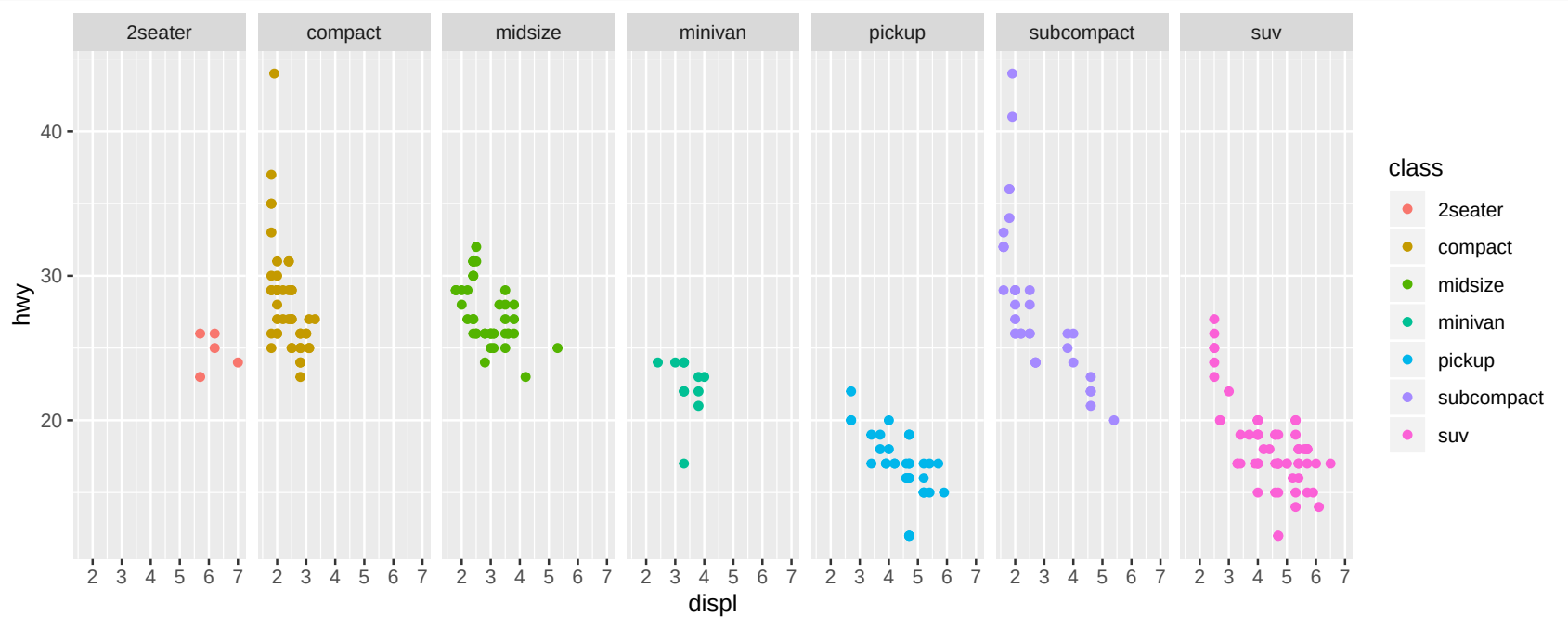
```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class))
```



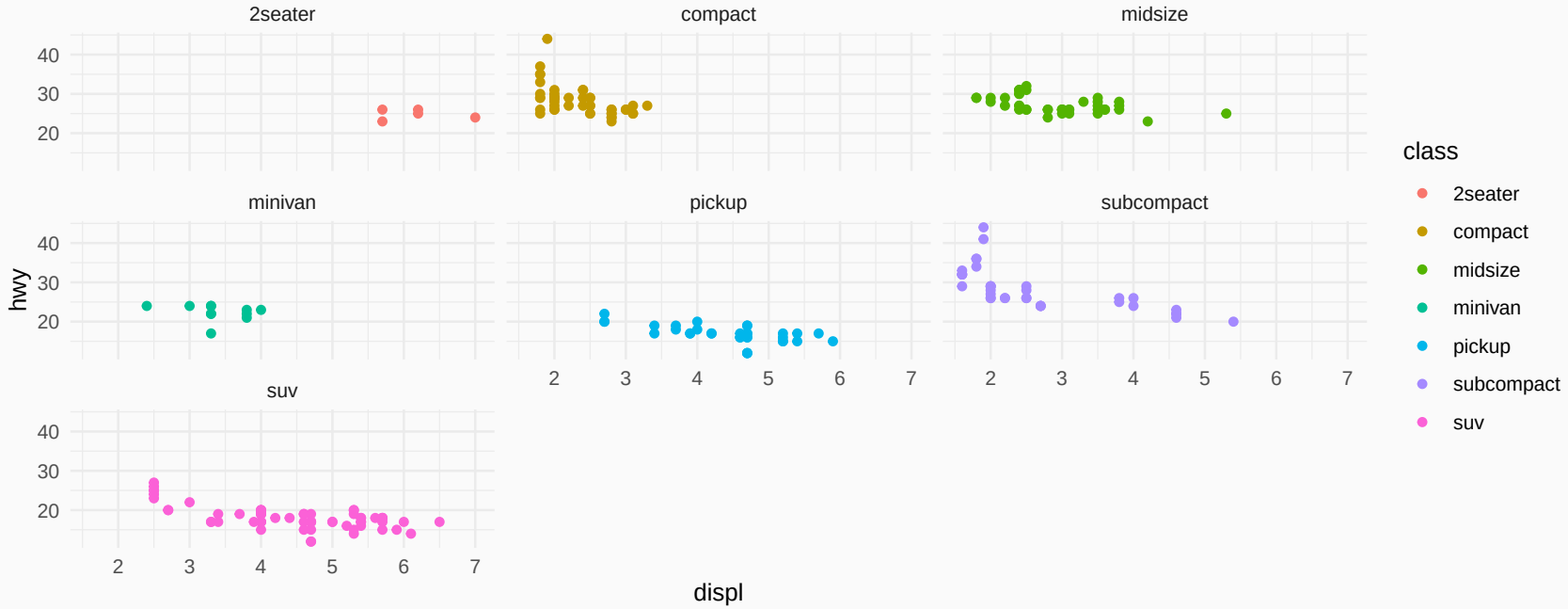
```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class, size = cyl)) +
  geom_smooth()
```



```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  facet_grid(cols = vars(class))
```



```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  facet_wrap(vars(class)) +
  theme_minimal()
```



Grammar of data manipulation

dplyr

Prepare data for plotting with `dplyr` functions

- `filter()` select subgroup(s) of cases
- `select()` select variables
- `mutate()` make new variables
- `summarise()` derive summary statistics
- `group_by()` stratify on the levels of one or more variables
- `arrange()` sort the outcome

Example

Dsiplay mean of log of `Sepal.Width` for selection plant species excluding `setosa` of the data set `iris`.

1. Deselect `setosa`
2. Compute log of `Sepal.Width`
3. Compute group means
4. Display means ordered from low to high

This can be done in many ways, e.g.

a. with traditional R functions
b. with `dplyr` functions without the pipe operator
c. with `dplyr` and pipe operator

Traditional R functions

```
newiris <- subset(iris, subset = Species !="setosa")
lg      <- log(newiris$Sepal.Width)
mean_lg <- aggregate(lg,
                     by = list(newiris$Species),
                     FUN = mean)
mean_lg[order(mean_lg$x), ]
```

```
##      Group.1        x
## 1 versicolor 1.012282
## 2  virginica 1.084180
```

- Not easy to oversee what is going on
- Pollution workspace with byproducts `newiris` and `sq`

With dplyr

```
arrange(
  summarize(
    mutate(
      group_by(
        filter(iris, Species !="setosa"),
        Species),
      lg = log(Sepal.Width)),
    mean_lg = mean(lg)),
  mean_lg)
```

```
## # A tibble: 2 x 2
##   Species    mean_lg
##   <fct>        <dbl>
## 1 versicolor    1.01
## 2 virginica     1.08
```

- Code has to be read from the inside out, but no workspace pollution

Using %>%

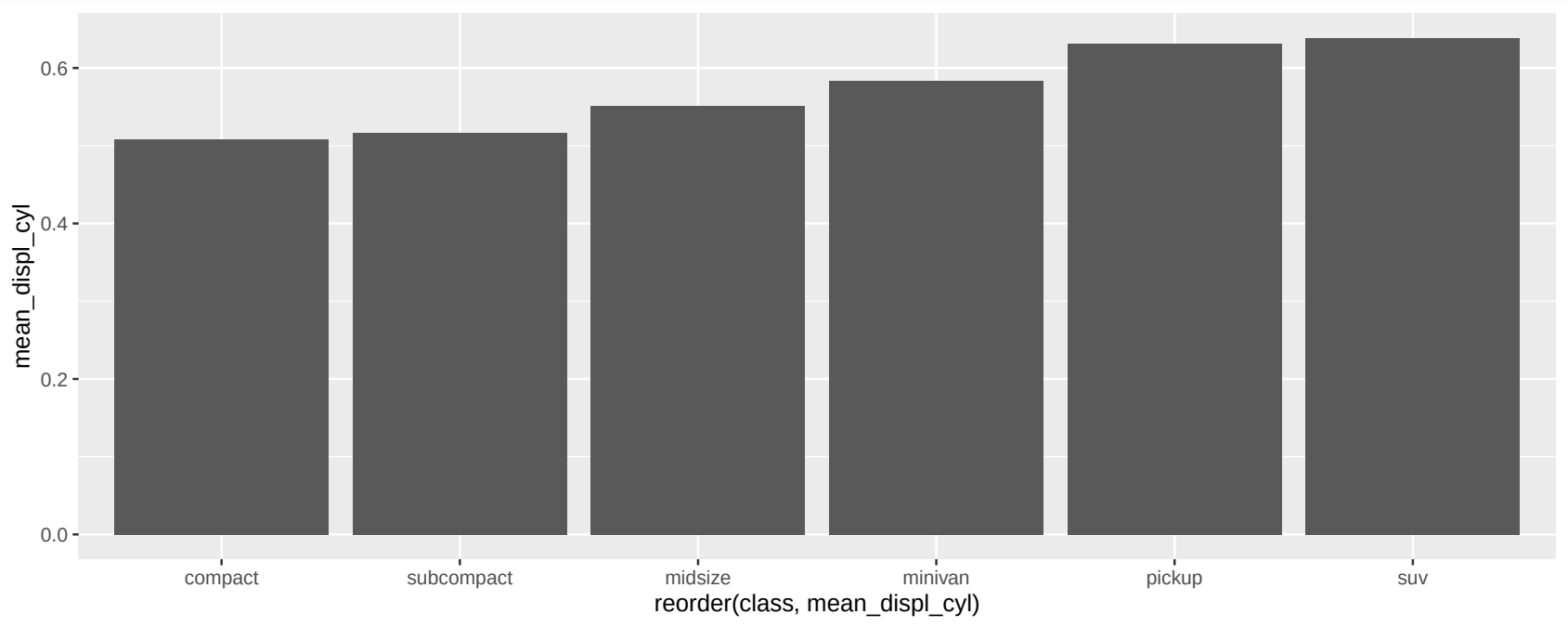
```
iris %>%
  filter(Species !="setosa") %>%
  mutate(lg = log(Sepal.Width)) %>%
  group_by(Species) %>%
  summarise(mean_lg = mean(lg)) %>%
  arrange(mean_lg)
```

```
## # A tibble: 2 x 2
##   Species    mean_lg
##   <fct>        <dbl>
## 1 versicolor    1.01
## 2 virginica     1.08
```

- easy to follow and no workspace pollution

Combination ggplot and dplyr

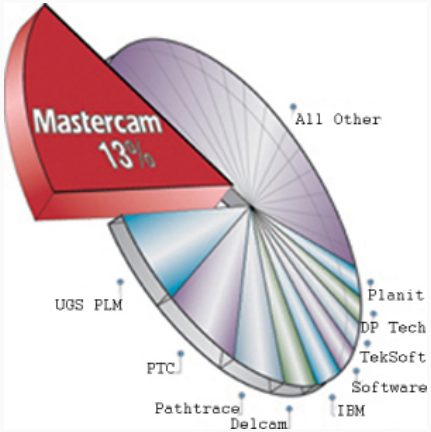
```
mpg %>%
  filter(class != "2seater") %>%
  group_by(class) %>%
  mutate(displ_cyl = displ/cyl) %>%
  summarise(mean_displ_cyl = mean(displ_cyl)) %>%
  ggplot() +
  geom_col(aes(x = reorder(class, mean_displ_cyl), y = mean_displ_cyl))
```



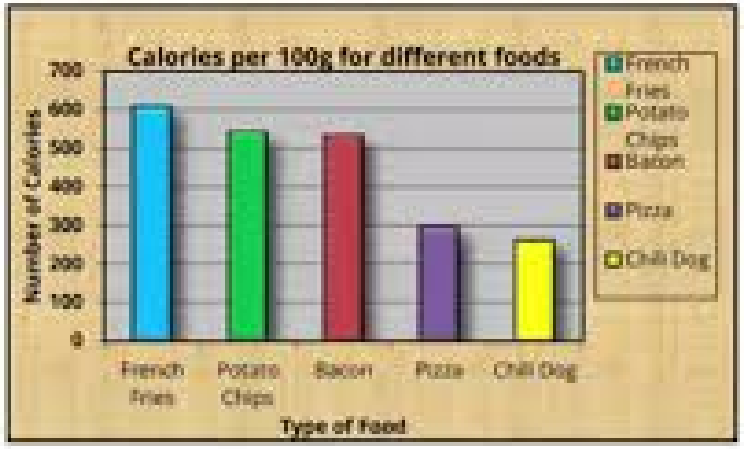
Tufte's Principles of Graphical Excellence

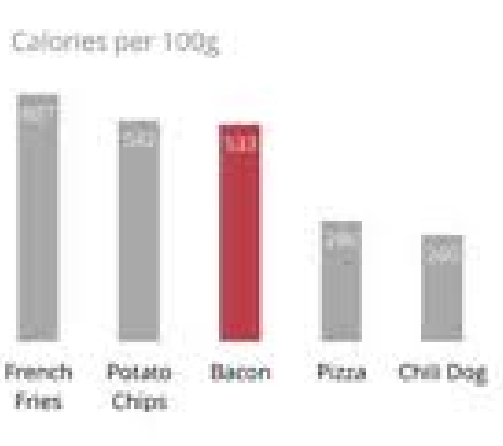
Proportionality, data-to-ink, chartjunk

- proportionality data/shape

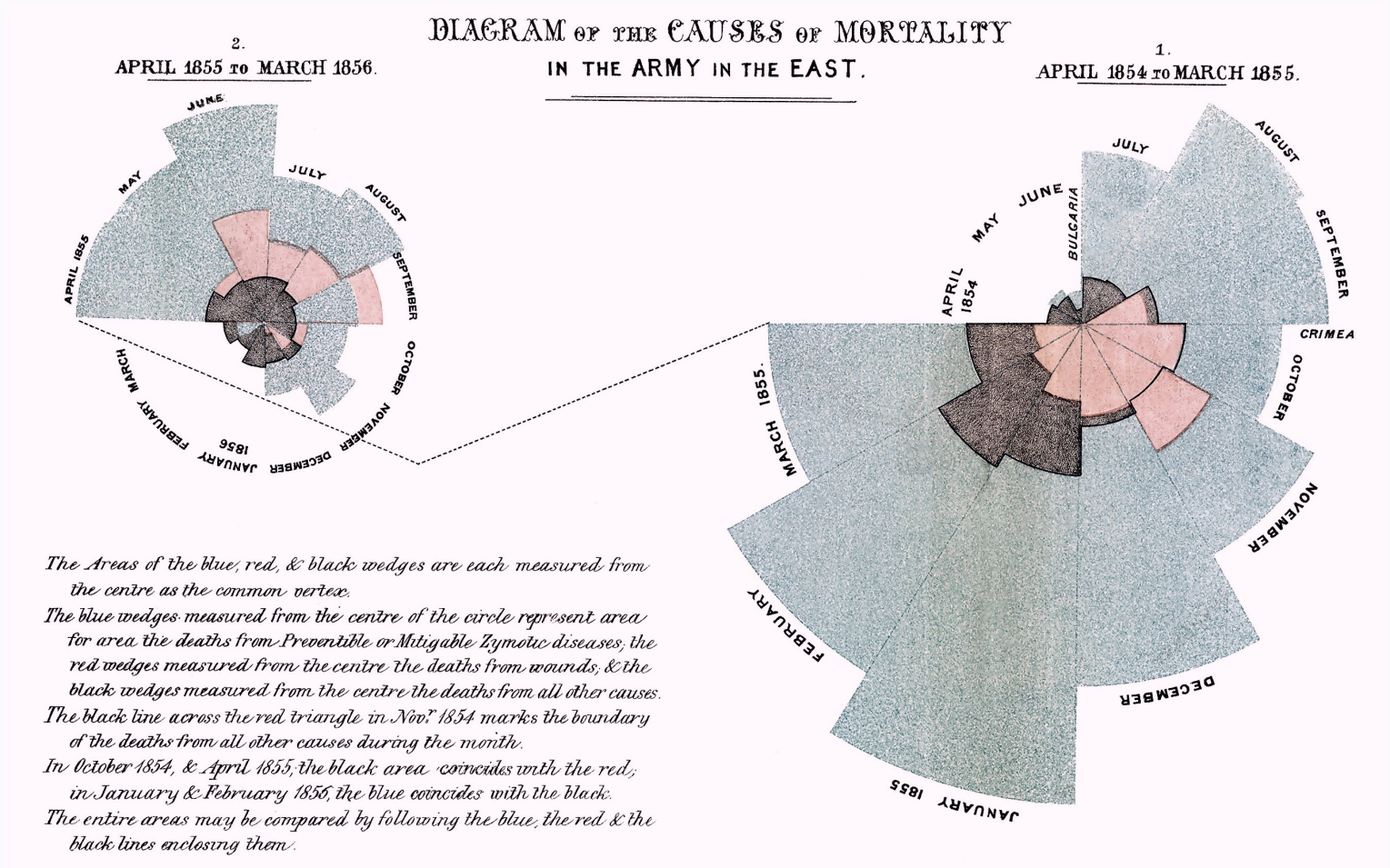


- maximize data-to-ink ratio, omit 'chart junk' (`themes` in `ggplot`)

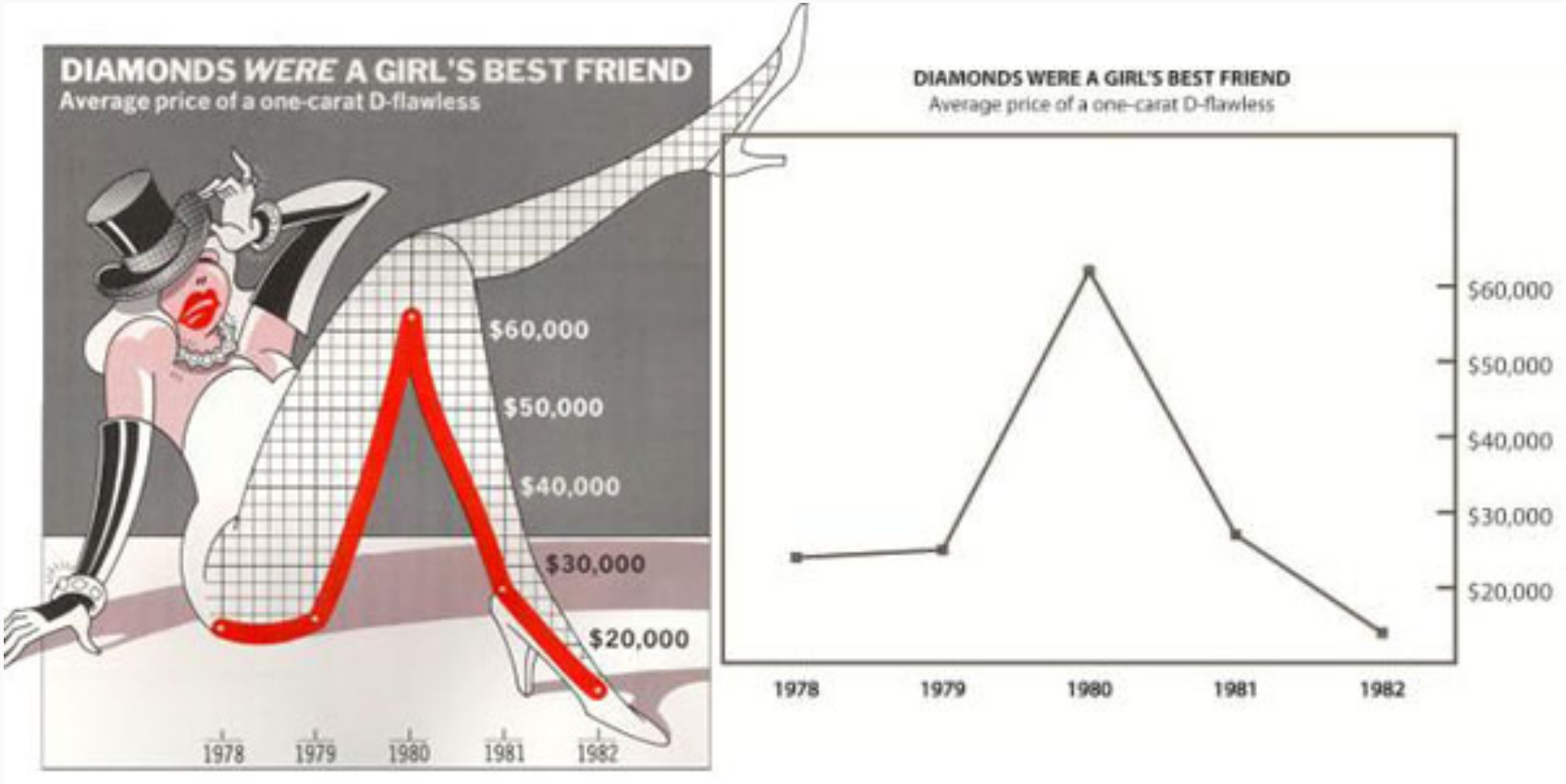




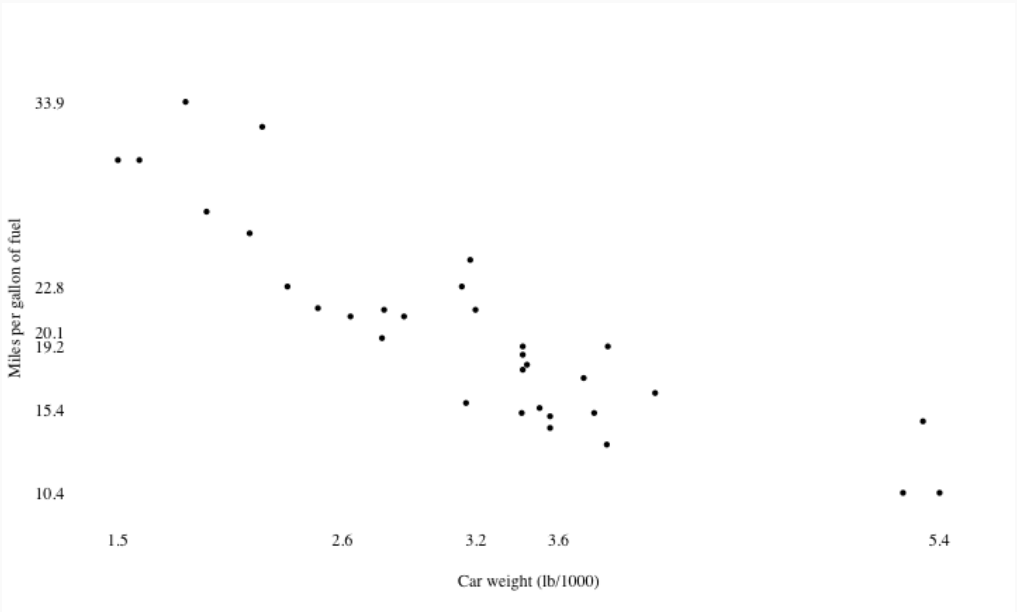
Proportionality principle

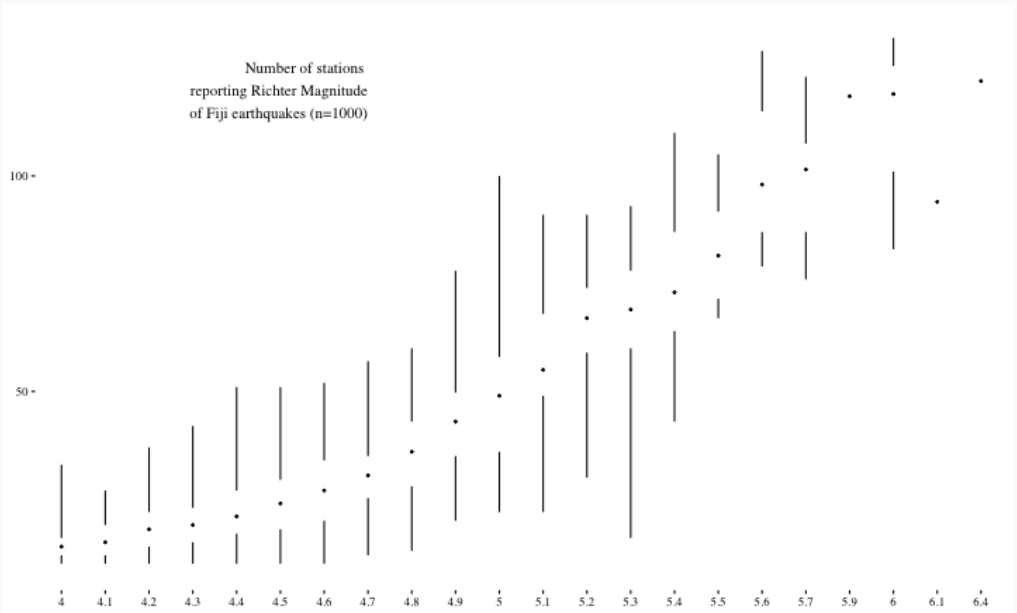


Chartjunk


DIAMONDS *WERE* A GIRL'S BEST FRIEND
Average price of a one-carat D-flawless
$60,000
$50,000
$40,000
$30,000
$20,000
1978
1979
1980
1981
1982
DIAMONDS WERE A GIRL'S BEST FRIEND
Average price of a one-carat D-flawless
$60,000
$50,000
$40,000
$30,000
$20,000
1978
1979
1980
1981
1982

Tufte in practice





Recap

Data visualization is a huge field

Stick to basic principles:

- Map data to aesthetics, geoms, scales, facets
- Which comparisons do I want?
- Maximize data-ink (within reason).

Lab 1A

Get acquainted with `dplyr` and `ggplot2`

1. Make plots with `ggplot2`
2. Combine plotting with `dplyr` functions

An R Markdown template is available

- Open the template in RStudio
- Save the file in an appropriate folder
- Read the instructions
- Fill the R chunks with the appropriate R code
- Test the R code in each chunk, and if it is error free
- Render an HTML file