Data Handling: Import, Cleaning and Visualisation

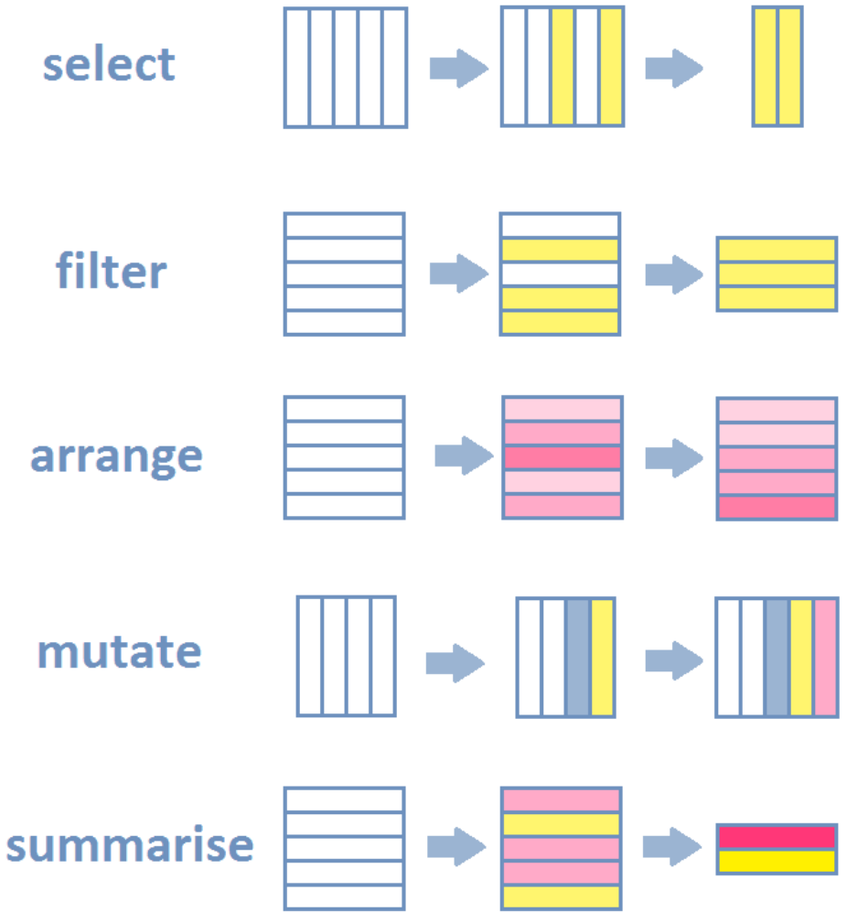
Lecture 10:
Exploratory Data Analysis and Visualization, Part II

Dr. Aurélien Sallin

2024-12-12

Recap: Data cleaning and Data Visualization

Use the five building block from `dplyr()`



Source: Intro to R for Social Scientists

Important additional tools

- `ifelse(test, yes, no)` returns a value with the same shape as the logical test. Filled with elements selected from either `yes` or `no` depending on whether the element of `test` is `TRUE` or `FALSE`

```
df |>
  mutate(gender = ifelse(gender == "m", 1, 0))
```

- `case_when` vectorises multiple `ifelse()` statements. It is the `dplyr` equivalent of `if...else`

```
df |>
  mutate(
    agegroup = case_when(
      age >= 0 & age < 18 ~ "0-18",
      age >= 18 & age < 64 ~ "19-64",
      age >= 65 & age < 100 ~ ">64",
      .default = "999"
    )
  )
```

Data visualization through tables and graphs

A chart typically contains at least one axis, the values are represented in terms of visual objects (dots, lines, bars) and axes typically have scales or labels.

- If we are interested in exploring, analyzing or communicating **patterns** in the data, charts are more useful than tables.

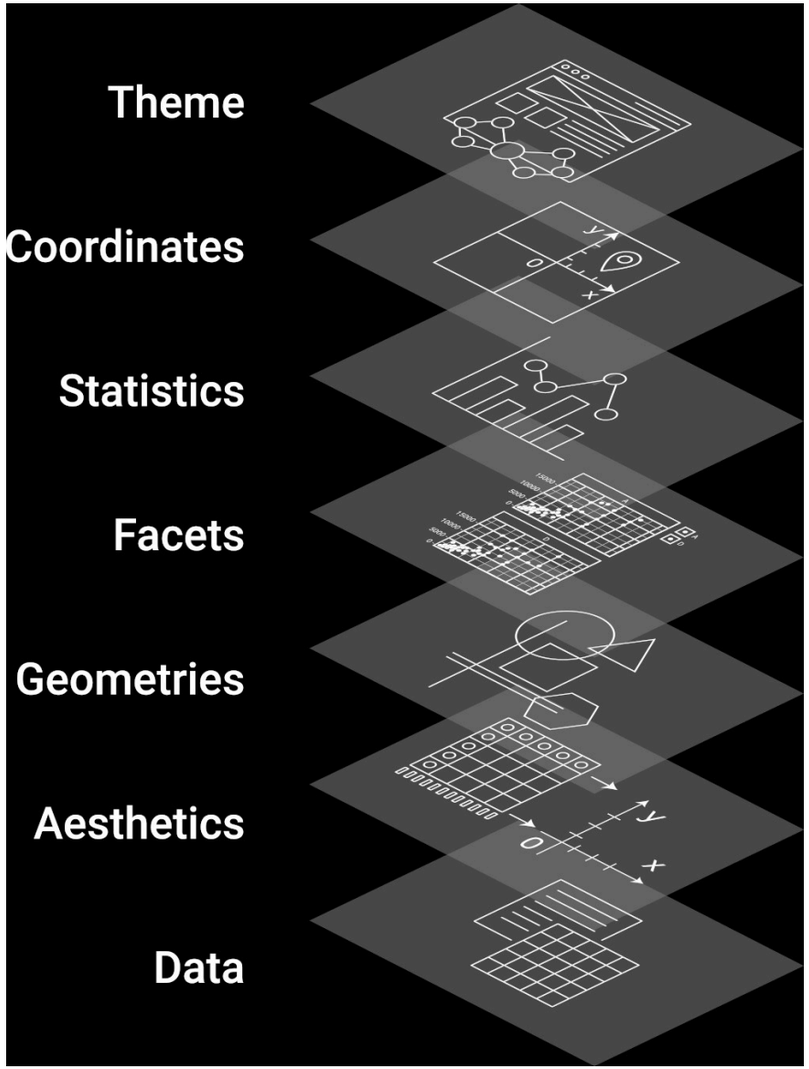
A table typically contains rows and columns, and the values are represented by text.

- If we are interested in exploring, analyzing or communicating **specific numbers** in the data, tables are more useful than graphs.

The grammar of graphics

- The *`ggplot2`* package is an implementation of Leland Wilkinson's 'Grammar of Graphics'.
- `ggplot2` is so good that it has become *THE* reference [In python, use `plotnine` to apply the grammar of graphics.]

Grammar of graphics



The grammar of graphics in action

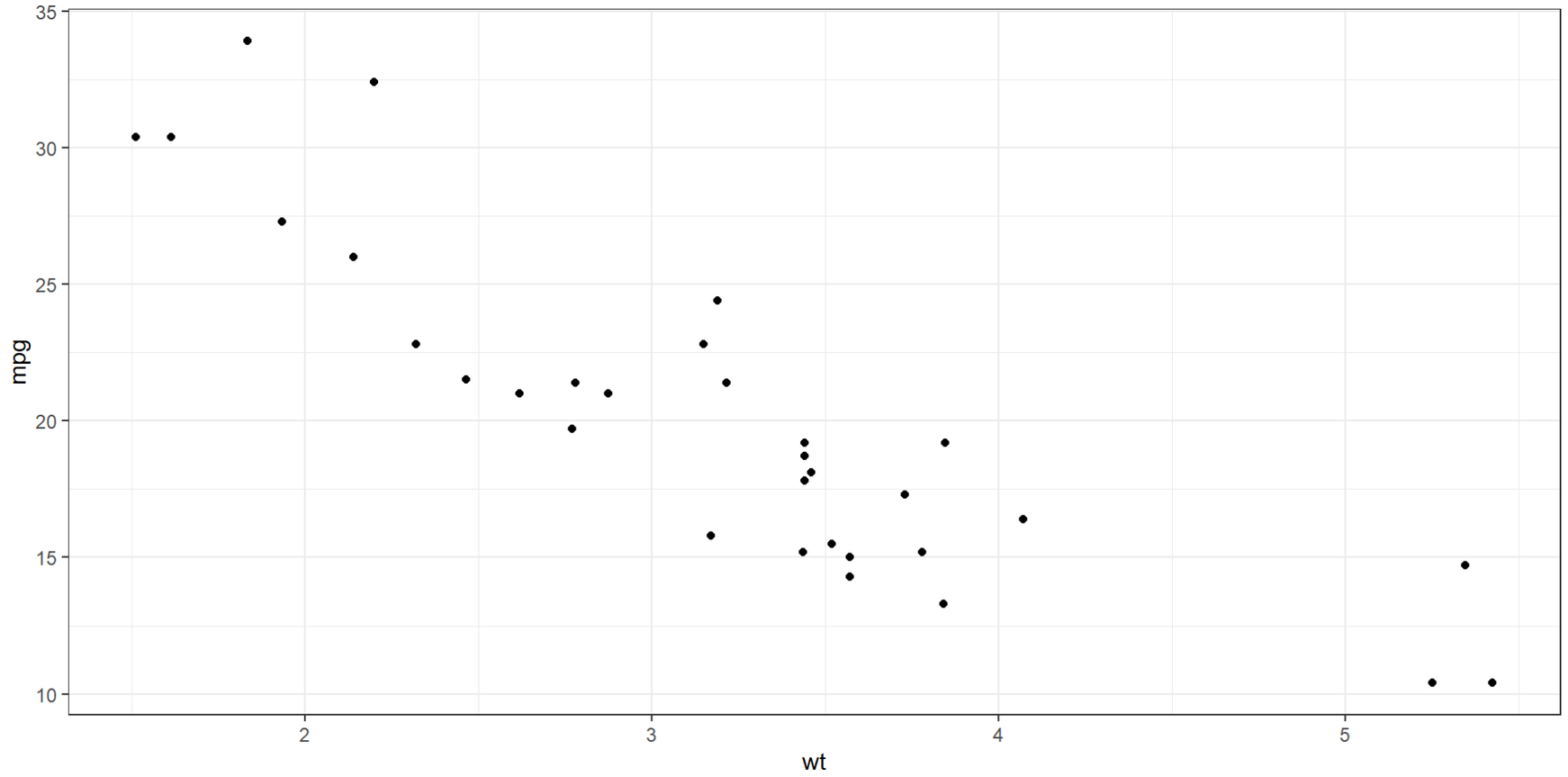
Example from A Comprehensive Guide to the Grammar of Graphics for Effective Visualization of Multi-dimensional Data using the built-in `mtcars` dataset in R.

```
mtcars # mtcars is a built-in dataset in R
```

```
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic       30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla    33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Dodge Challenger  15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
```

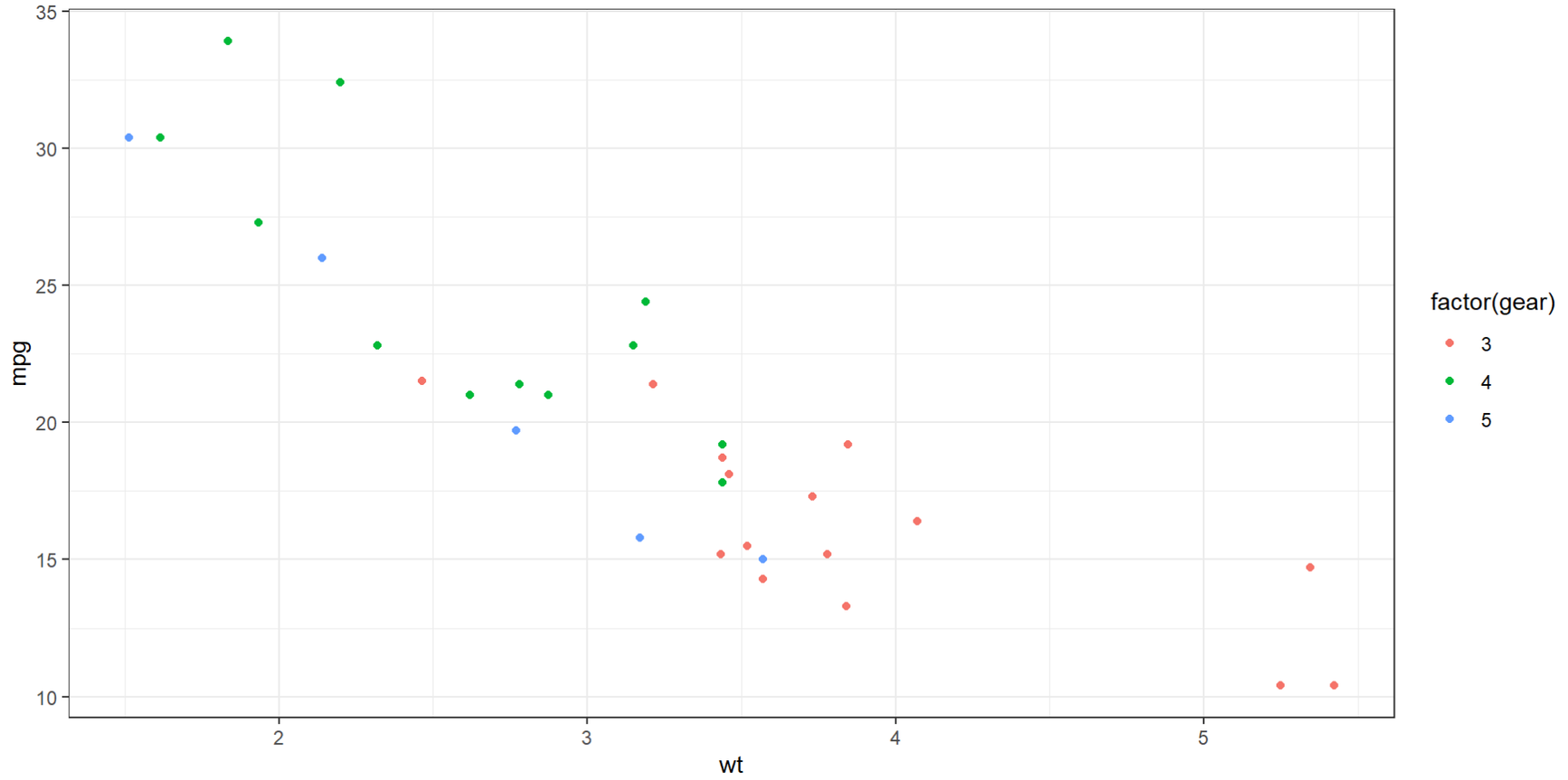
From two dimensions...

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



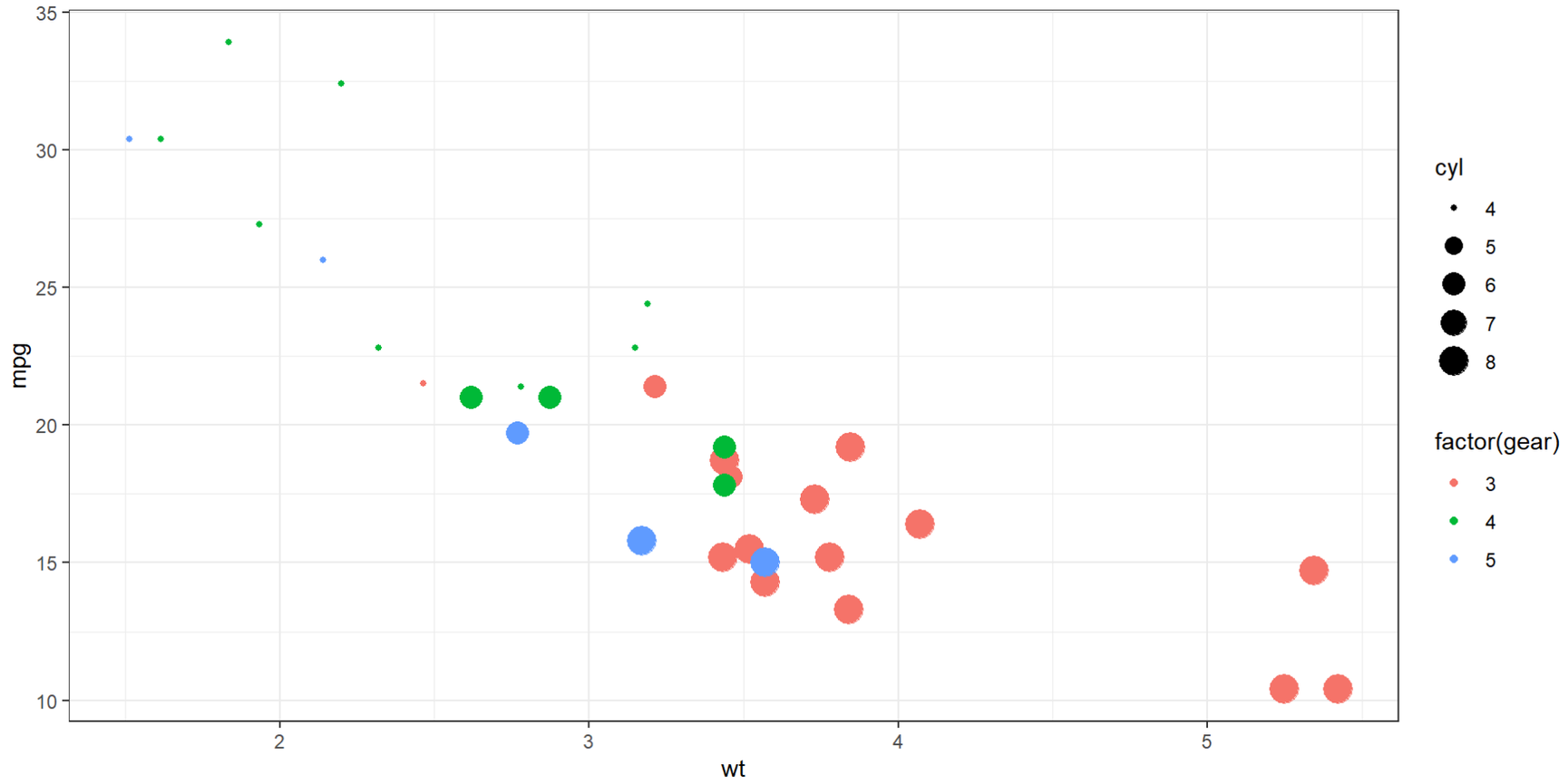
To three dimensions...

```
ggplot(mtcars, aes(x = wt, y = mpg, color=factor(gear))) +
  geom_point() +
  theme_bw()
```



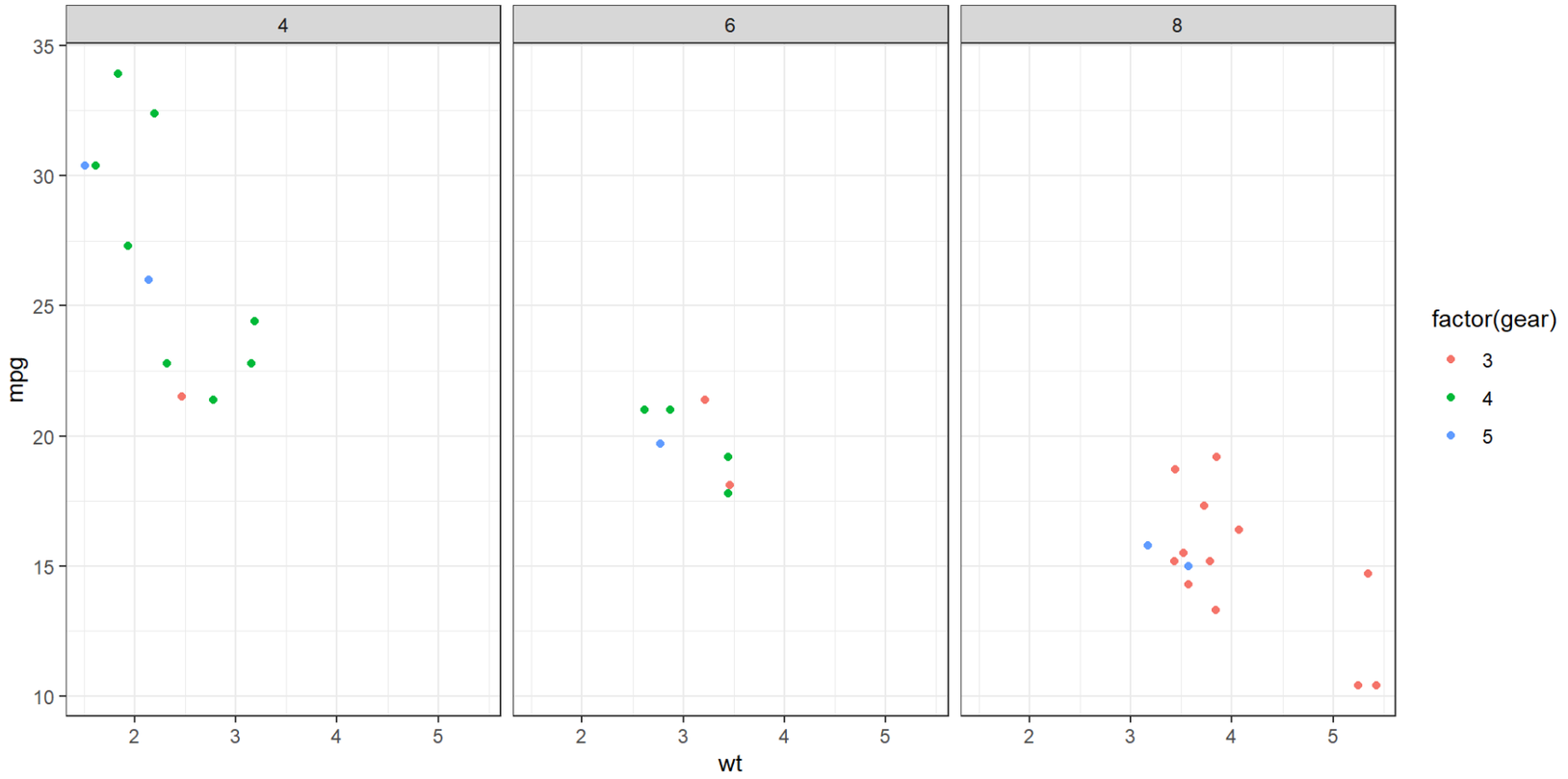
To four dimensions...

```
ggplot(mtcars, aes(x = wt, y = mpg, color=factor(gear), size = cyl)) +
  geom_point() +
  theme_bw()
```



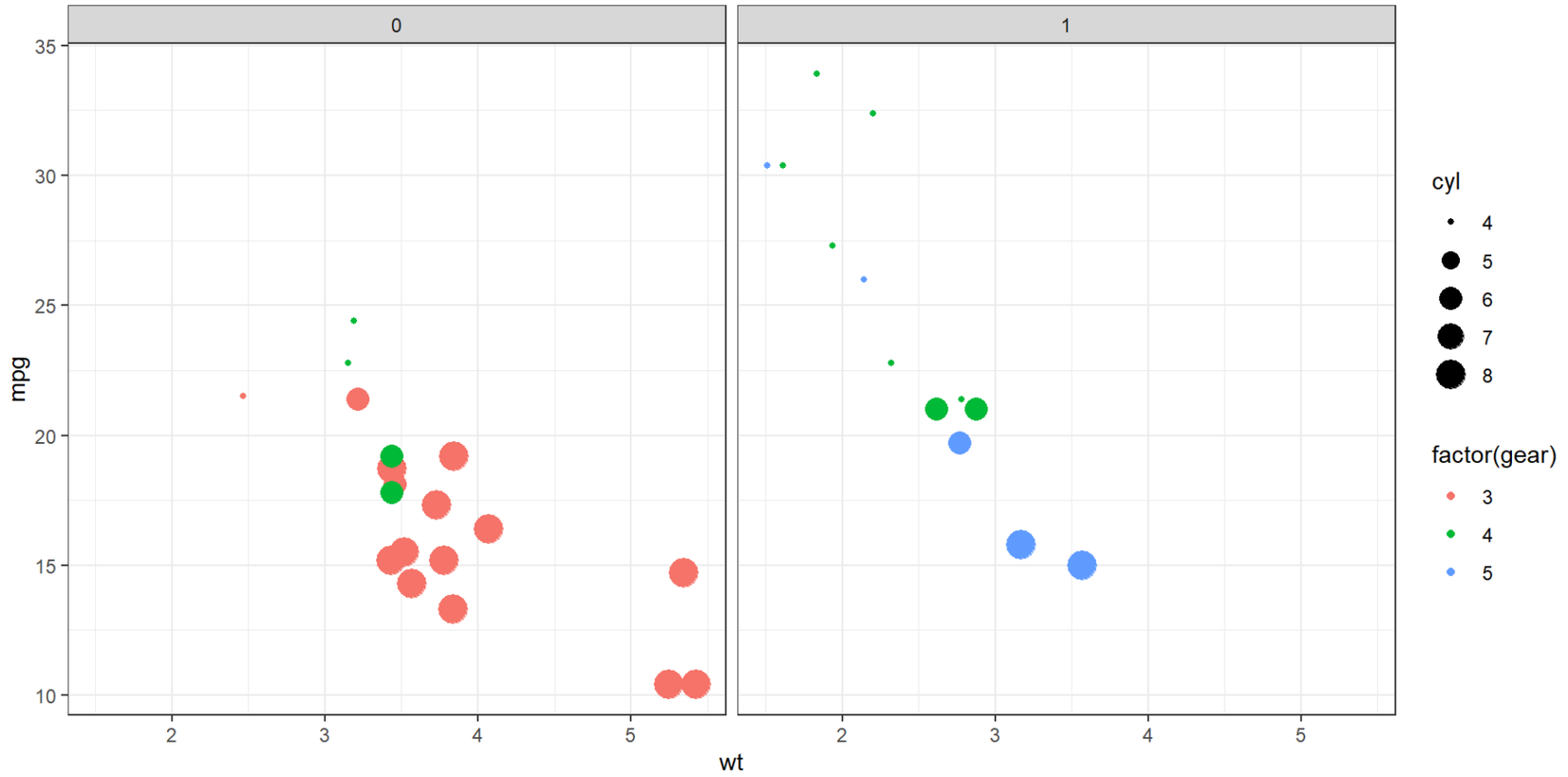
To four dimensions (with `facets`)...

```
ggplot(mtcars, aes(x = wt, y = mpg, color=factor(gear))) +
  geom_point() +
  facet_wrap(~cyl) +
  theme_bw()
```



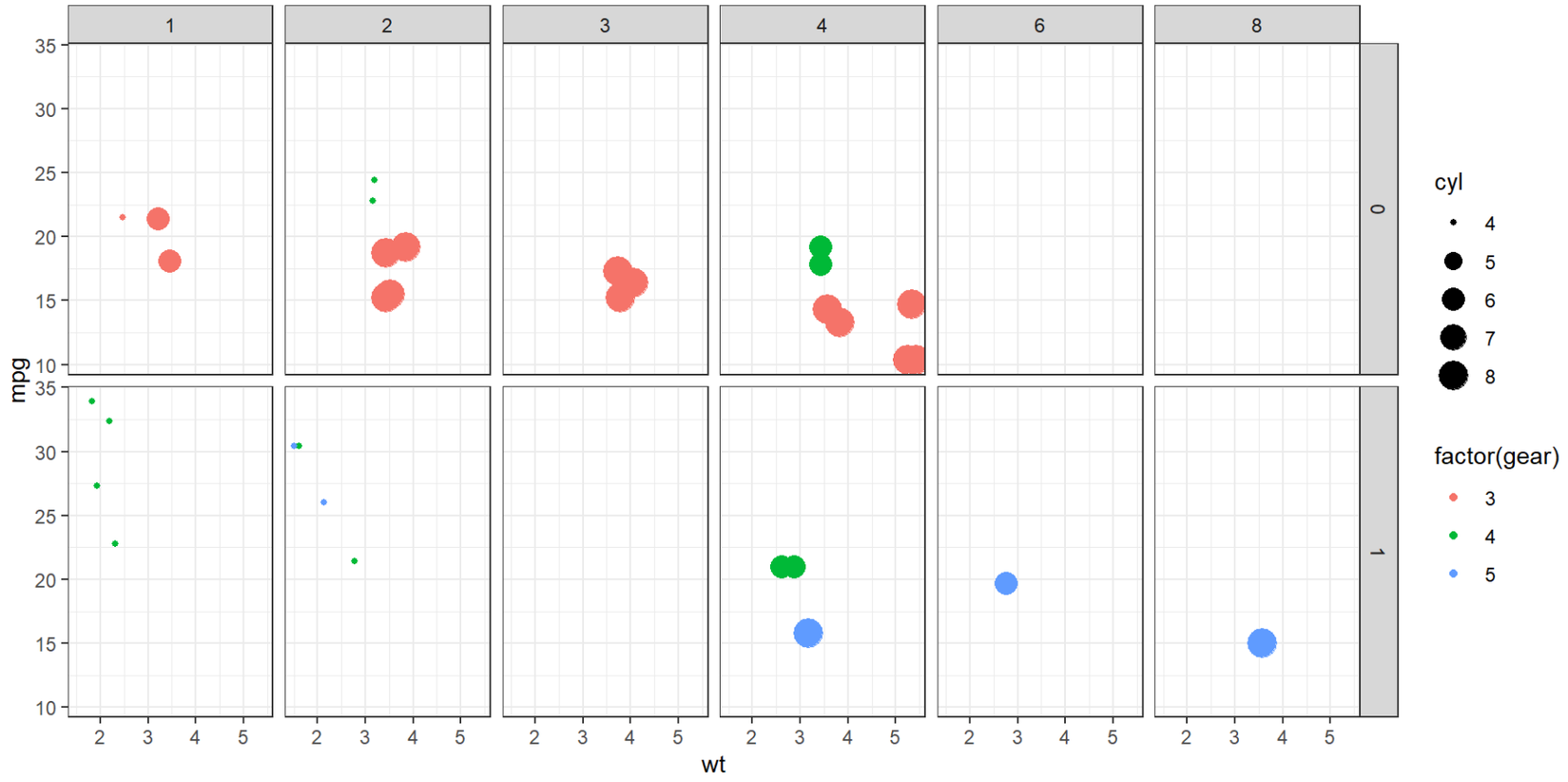
To five dimensions

```
ggplot(mtcars, aes(x = wt, y = mpg, color=factor(gear), size = cyl)) +
  geom_point() +
  facet_wrap(~am) +
  theme_bw()
```



To six dimensions

```
ggplot(mtcars, aes(x = wt, y = mpg, color=factor(gear), size = cyl)) +
  geom_point() +
  facet_grid(am ~ carb) +
  theme_bw()
```



Quiz on `geom_bar()`

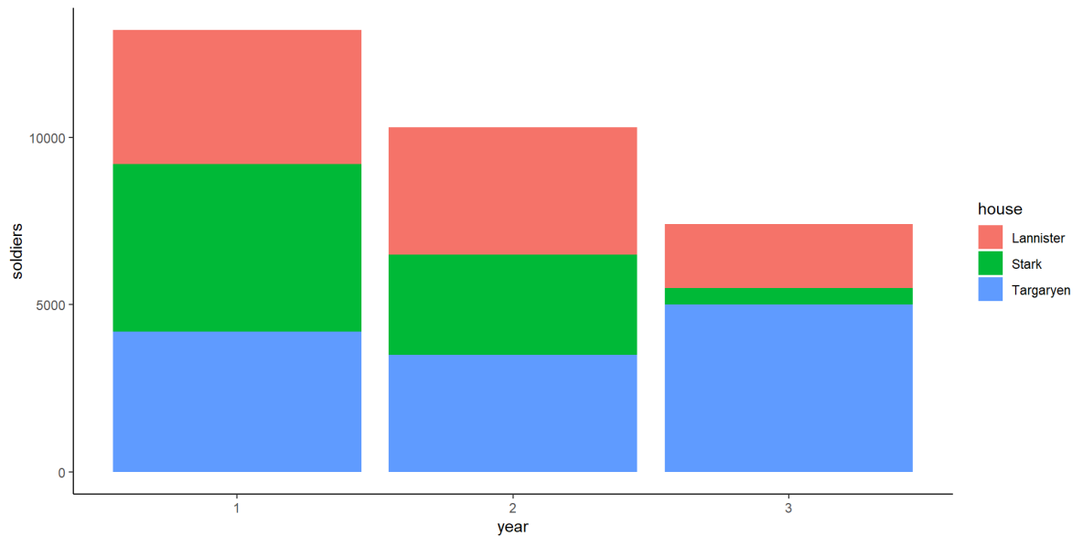
Which code produced the figure? (This question would not be an exam question, as it requires specific knowledge of `geom_bar()`. Solve it with R.)

```
got_data <- tibble(
  house = c("Stark", "Stark", "Stark", "Lannister", "Lannister", "Lannister",
            "Targaryen", "Targaryen", "Targaryen"),
  soldiers = c(5000, 3000, 500, 4000, 3800, 1900, 4200, 3500, 5000),
  year = c(1,2,3,1,2,3,1,2,3)
)
```

```r
ggplot(got_data, aes(x = year, y = soldiers, fill = house)) +
  geom_bar(stat = "identity") +
  theme_classic()
```

```r
ggplot(got_data, aes(x = year, y = soldiers, fill = house)) +
  geom_histogram() +
  theme_classic()
```

```r
ggplot(got_data, aes(x = year, fill = house)) +
  geom_bar() +
  theme_classic()
```



Today

📢 Announcements: about the exam

Exam for exchange students

- 🎁 19.12.2024 at 16:15 in room 01-207.

Lockdown browser

- Exam and LockDown Browser: check Sharepoint on StudentWeb and test on Canvas.
  Password: DataHandling2023

Announcements: about the exam

Expectations for the exam:

- Same format as quizzes and mock exam, including True/False questions, multiple-choice, and multiple-correct options. These are designed to test your understanding of the material.
- There will also be 3-4 essay-style questions aimed at evaluating your ability to apply your knowledge to new situations.
    - E.g., you might be asked to explain particular steps of the data analysis process in a given situation. You can use code, R concepts, or you can explain in plain English. The more precise the better.
- **You will not be required to write exact R code**, but you should be able to interpret and understand the code provided in the exam.
- I expect you to be familiar with all R commands and concepts covered in the lectures, exercises, in-class code, and additional practice exercises.
- The readings are not mandatory for the exam. The focus will be on the material discussed in class and during the exercises.

Today

Goals for today

Building on what we covered last week:

1. Know how to conduct exploratory data analysis (EDA).
2. Visualize data using tables.
3. Visualize data using the grammar of graphics.
4. Produce effective data visualization.

Today and next time (first hour)

5. Work with text data
6. Dashboard with Shiny

From graphs to effective data visualization

Data visualization: some principles

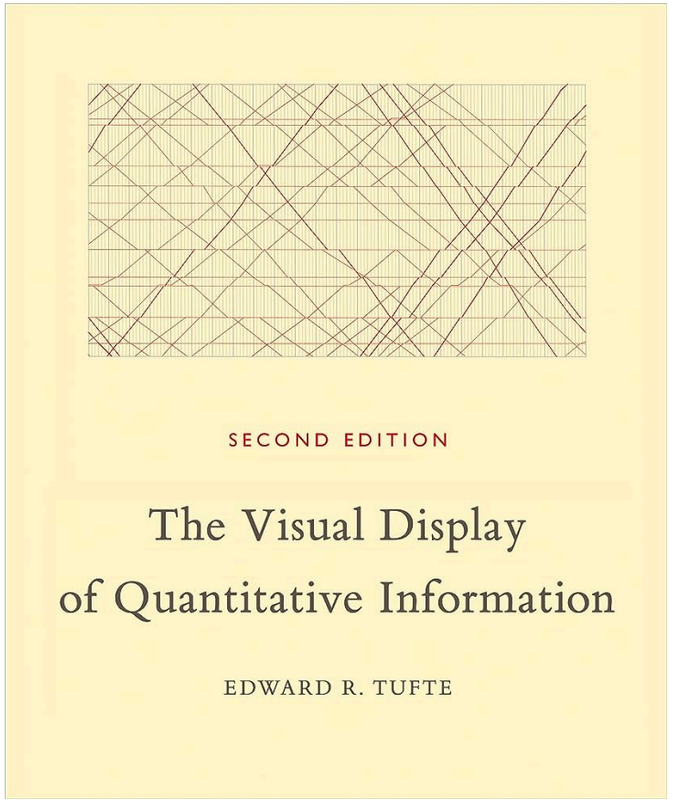
- Values are represented by their **position relative to the axes**: line charts and scatterplots.
- Values are represented by the **size of an area**: bar charts and area charts.
- Values are **continuous**: use chart type that visually connects elements (line chart).
- Values are **categorical**: use chart type that visually separates elements (bar chart).

(Source: Data Visualization Basics for Economists)

“Greatest number of ideas in
the shortest time with the
least ink in the smallest space”
(Edward Tufte, 1983)

Data visualization: some principles

Recommendations from Edward Tufte's "The Visual Display of Quantitative Information" (1983)

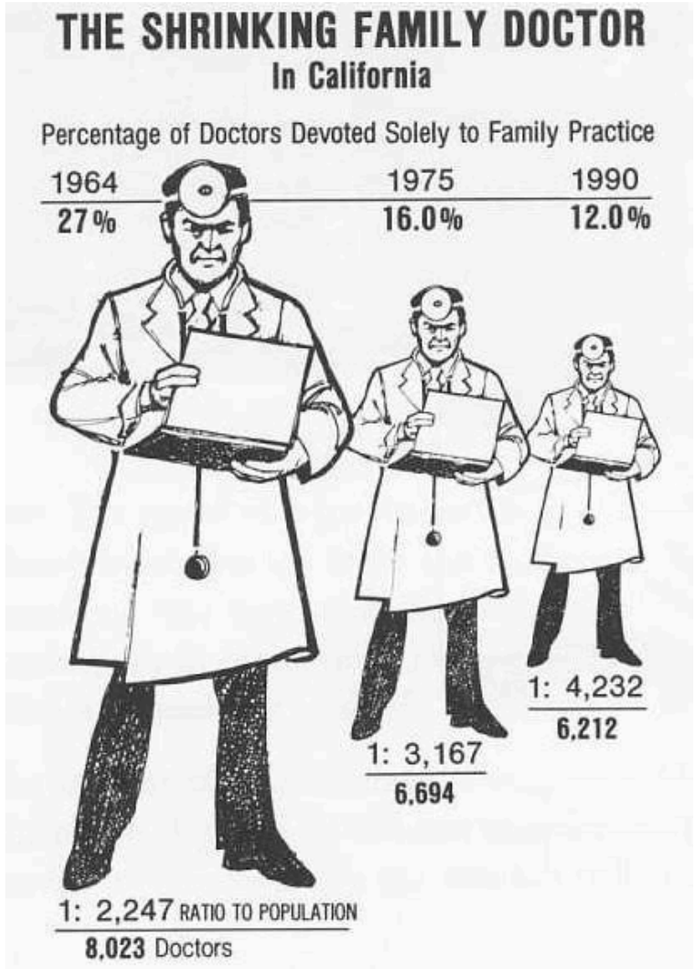


Lie Factor, or strive for graphical integrity

We can quantify the **Lie Factor** of a graph as a measure of how much the graphic distorts the data.

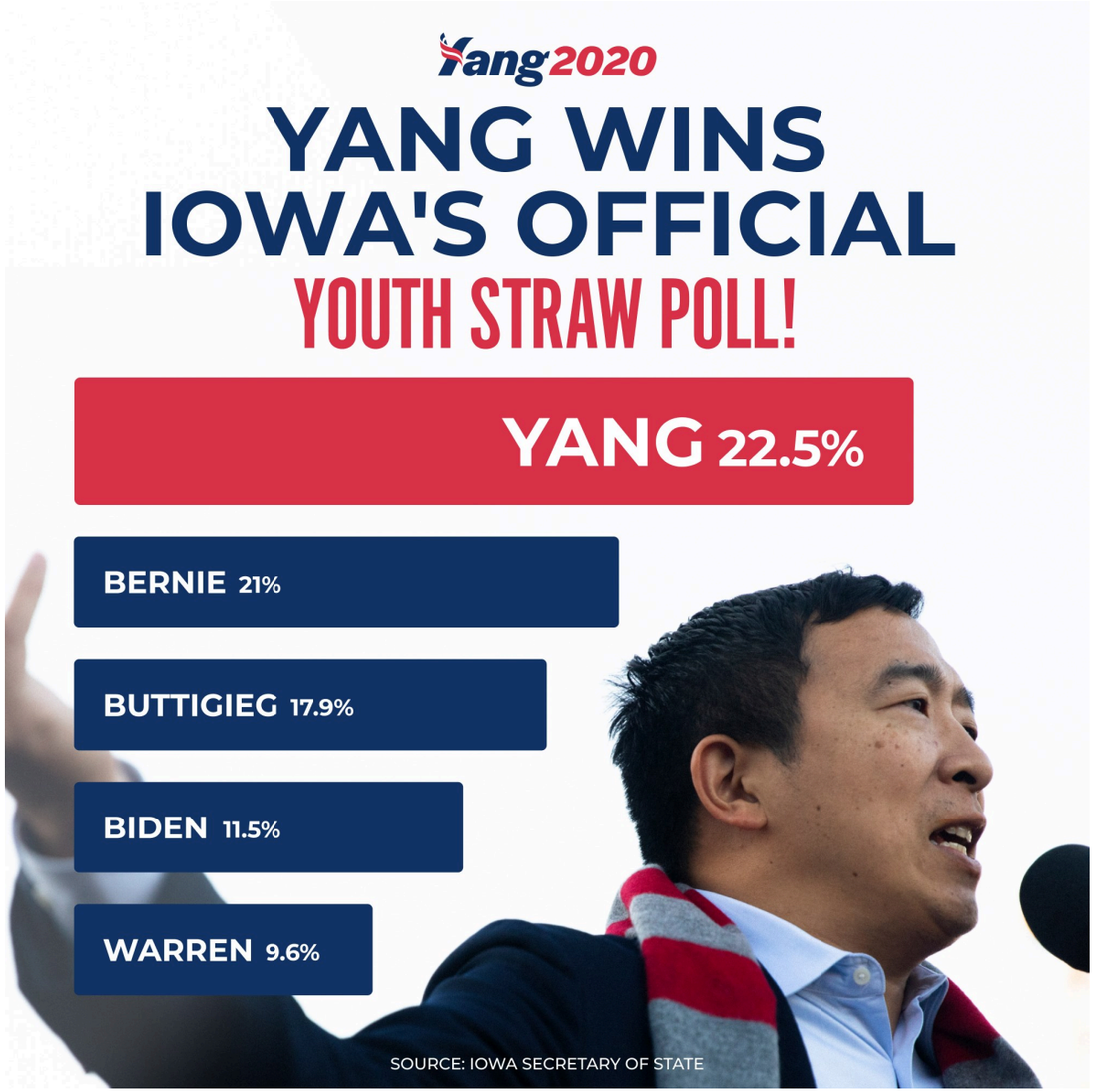
“The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the quantities represented.” (Tufte, 1983)

$$\text{Lie Factor} = \frac{\text{size of effect shown in graphic}}{\text{size of effect in data}}$$



Lie Factor, or strive for graphical integrity

$$\text{Lie Factor} = \frac{\text{Yang had } 39.1\% \text{ of total ink}}{\text{Yang had } 22.5\%} = 1.74$$



Thou shalt not truncate the Y axis.

Tucker Carlson is guilty of committing chart sins

Thou shalt not truncate the Y axis.

JEMIMA KELLY + Add to myFT

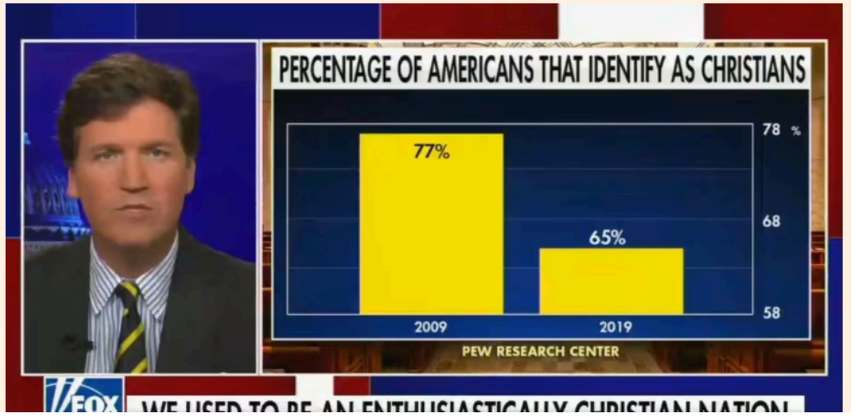
Jemima Kelly SEPTEMBER 28 2021 48

Unlock the Editor's Digest for free

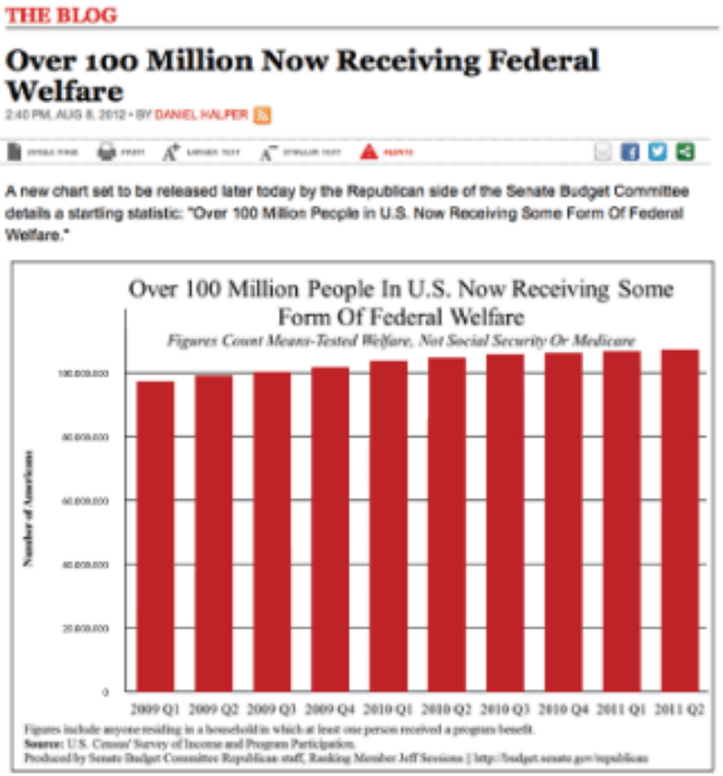
Roula Khalaf, Editor of the FT, selects her favourite stories in this weekly newsletter.

Enter your email address Sign up

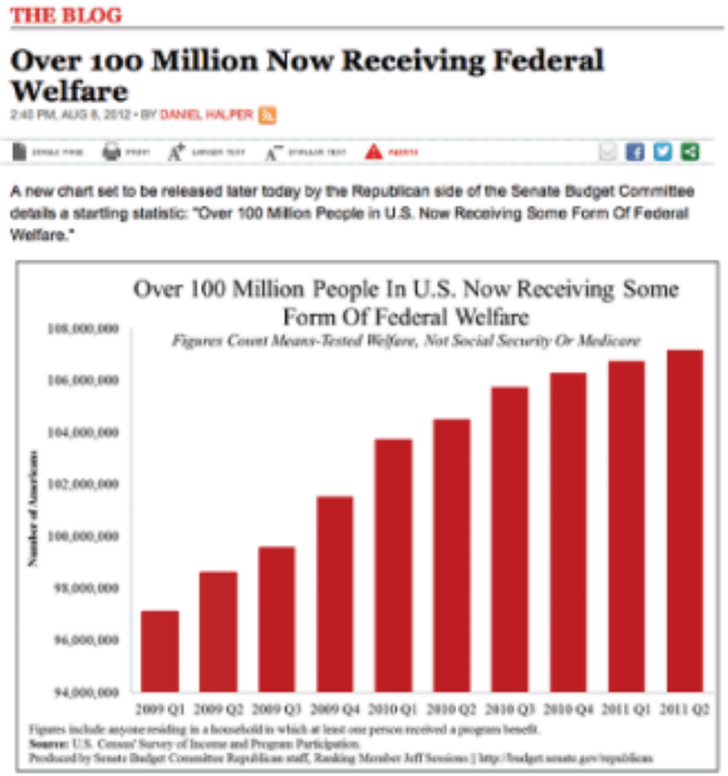
True chart crime fans, rejoice! For unto you a new atrocity is born:



Thou shalt not truncate the Y axis.



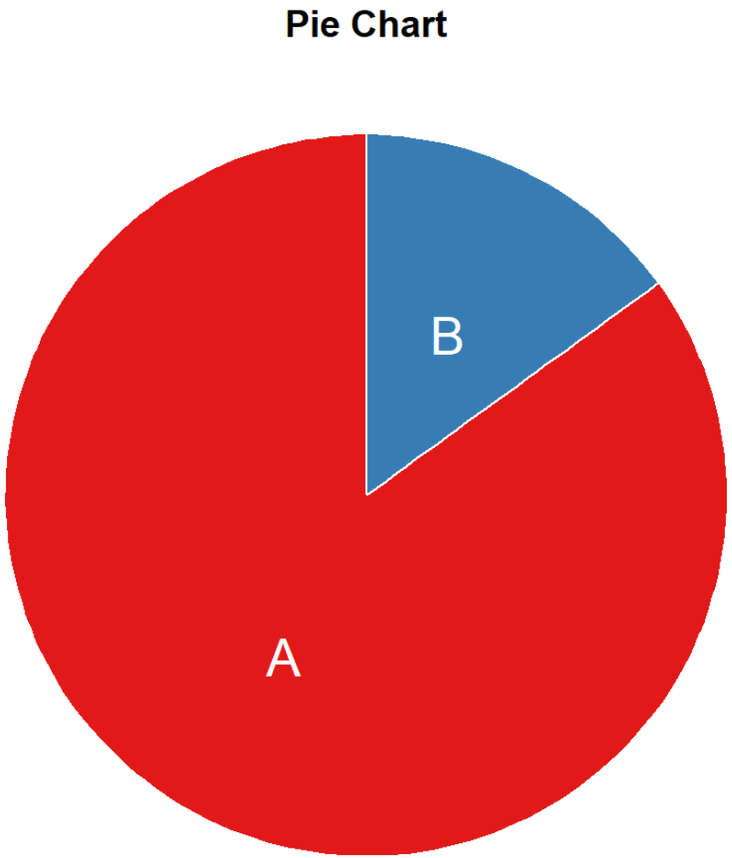
lie factor: 1

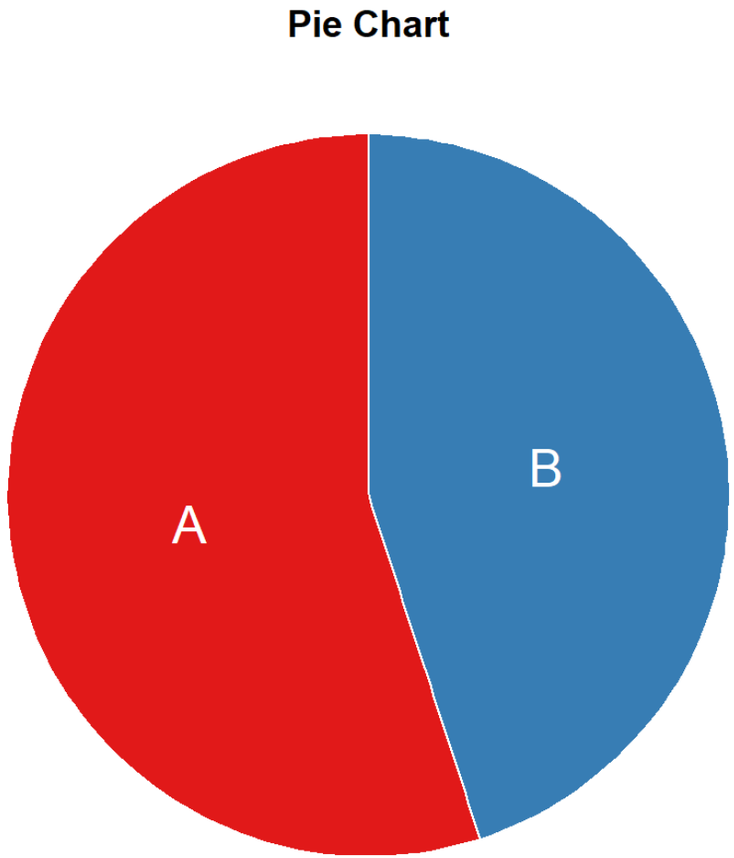


lie factor: 16,08

Source: The lie factor and the baseline paradox

Avoid pie charts



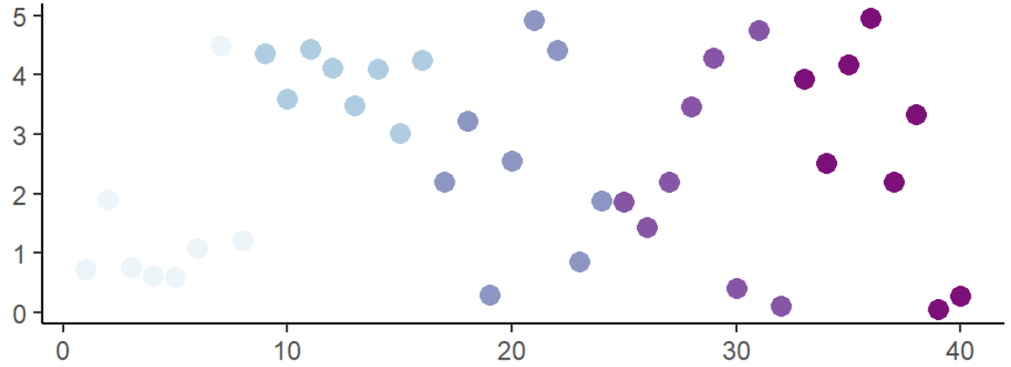


“All variations lead to overestimation of small values and underestimation of large ones.” Kosara et al, 2018

Different types of colors for different types of data

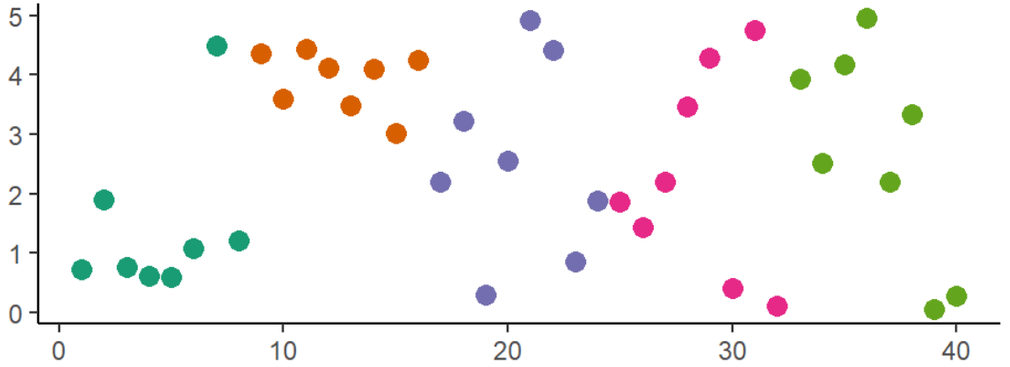
Palette: Sequential

scale_color_brewer(palette = "BuPu")



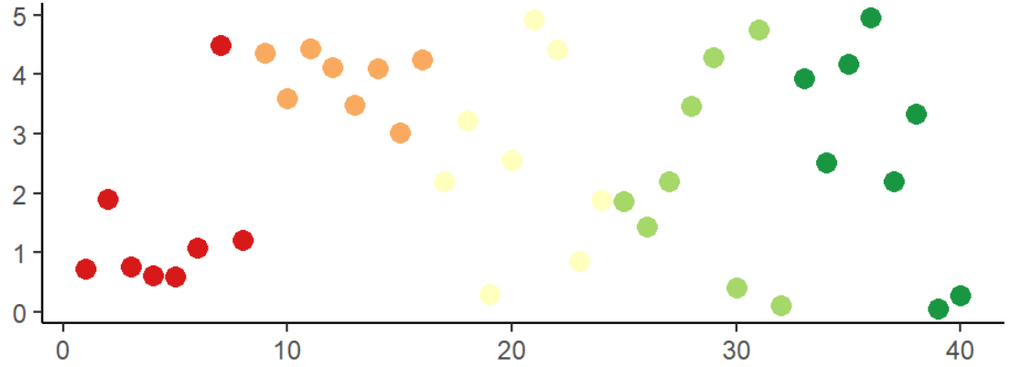
Palette: Qualitative

scale_color_brewer(palette = "Dark2")



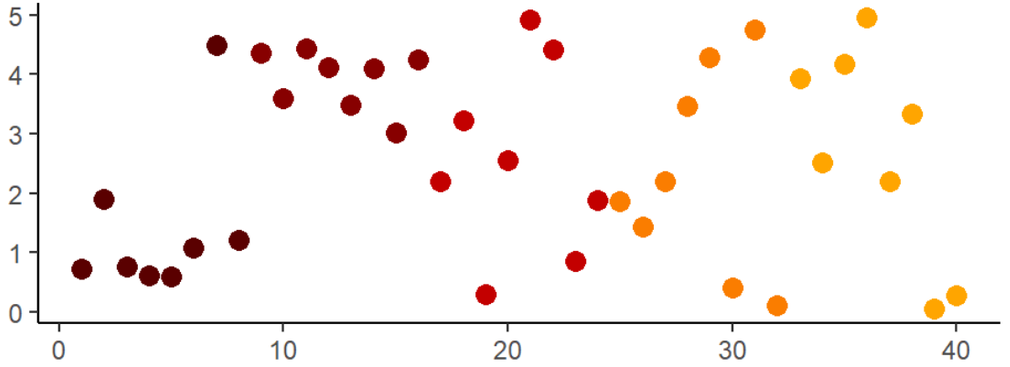
Palette: R Color brewer: Diverging

scale_color_brewer(palette = "RdYlGn")



Palette: Gryffindor

scale_color_hp(house = "Gryffindor", discrete = TRUE)

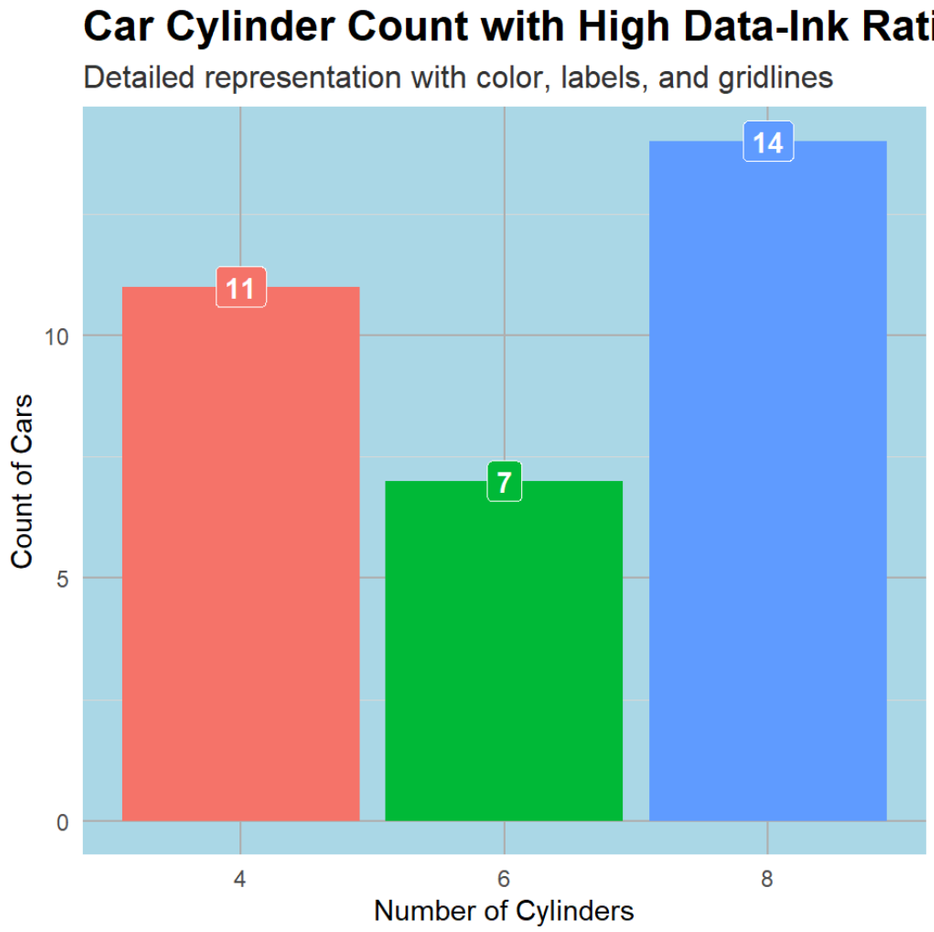


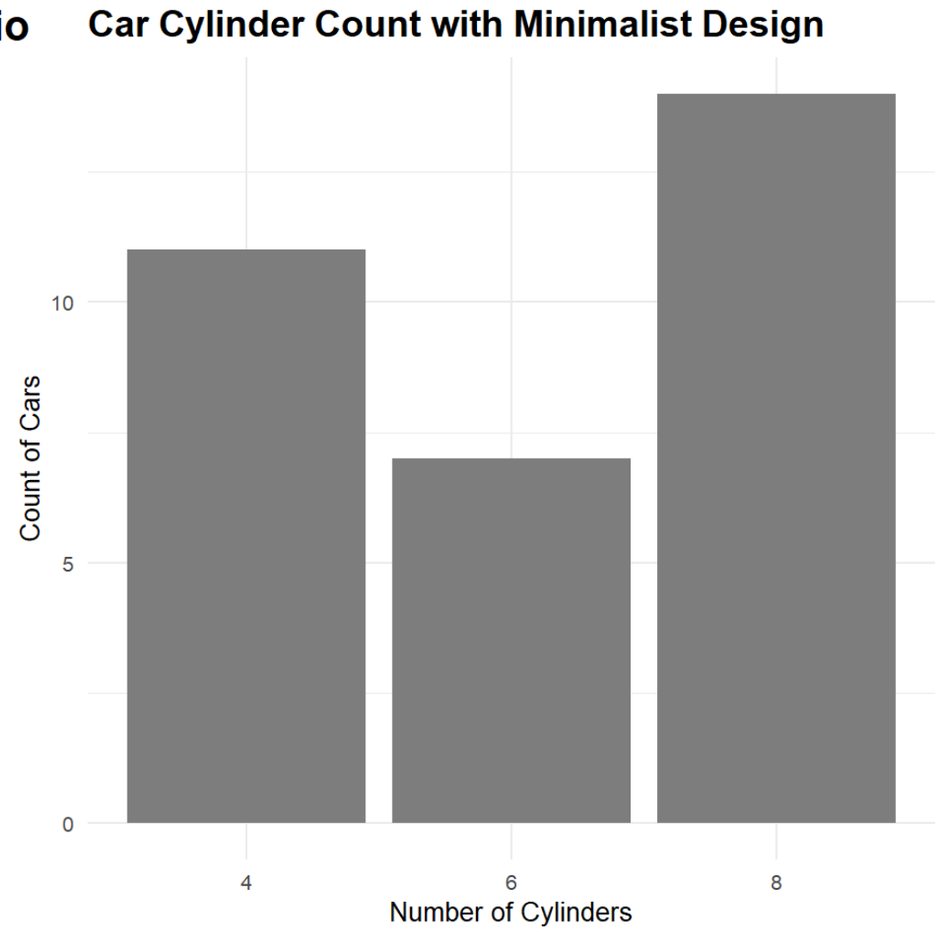
Only what matters should be reported

- Data-ink Ratio = $\frac{\text{ink used for data points}}{\text{total ink used to print the graphic}}$
  - Data ink: data points and measured quantities, such as the dots in a scatter plot
  - Non-data ink: functional marks such as titles, labels, axes, gridlines and tick points or decorative marks

Limits to this approach: we still need some ink to interpret and understand the data.

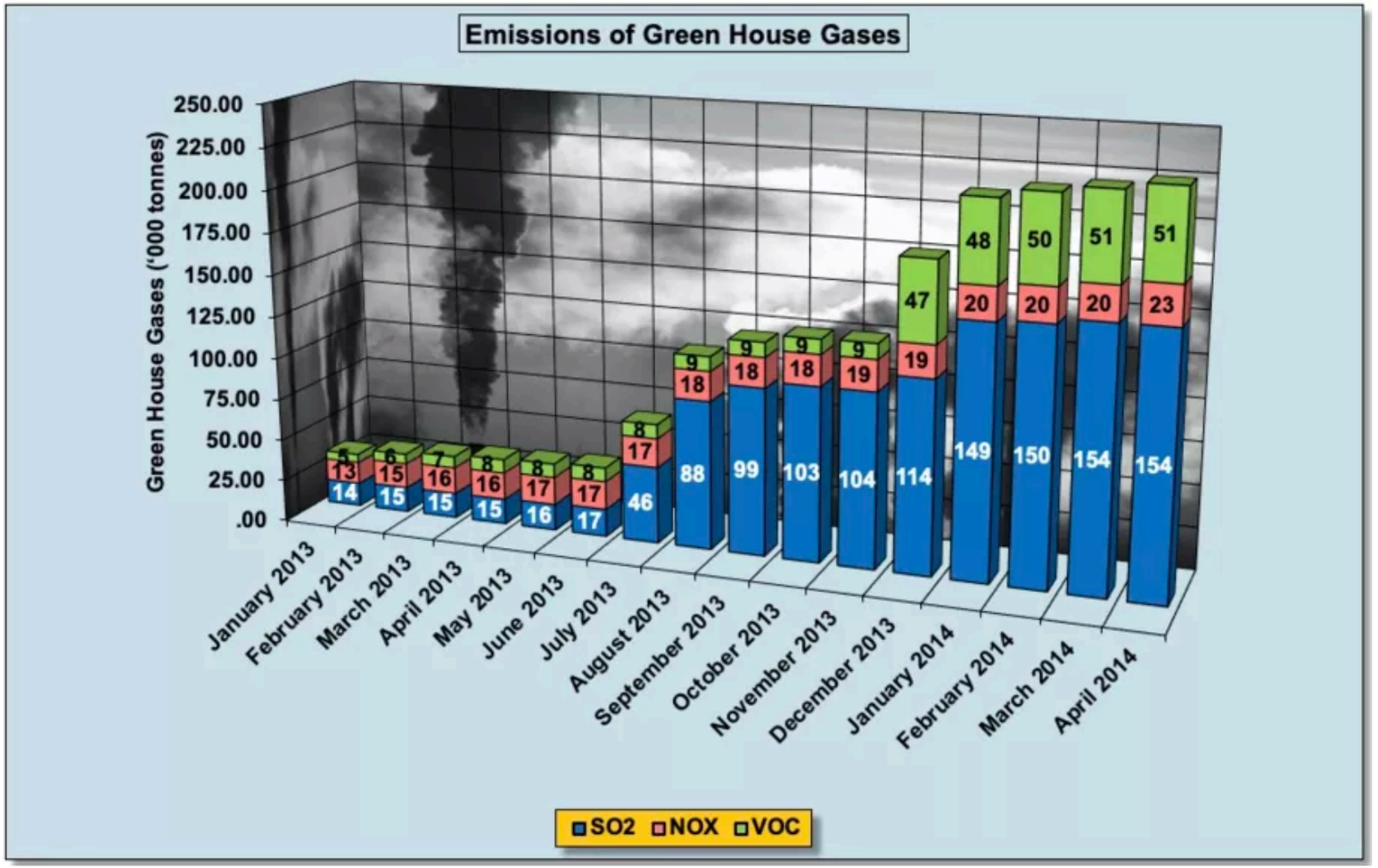
Only what matters should be reported





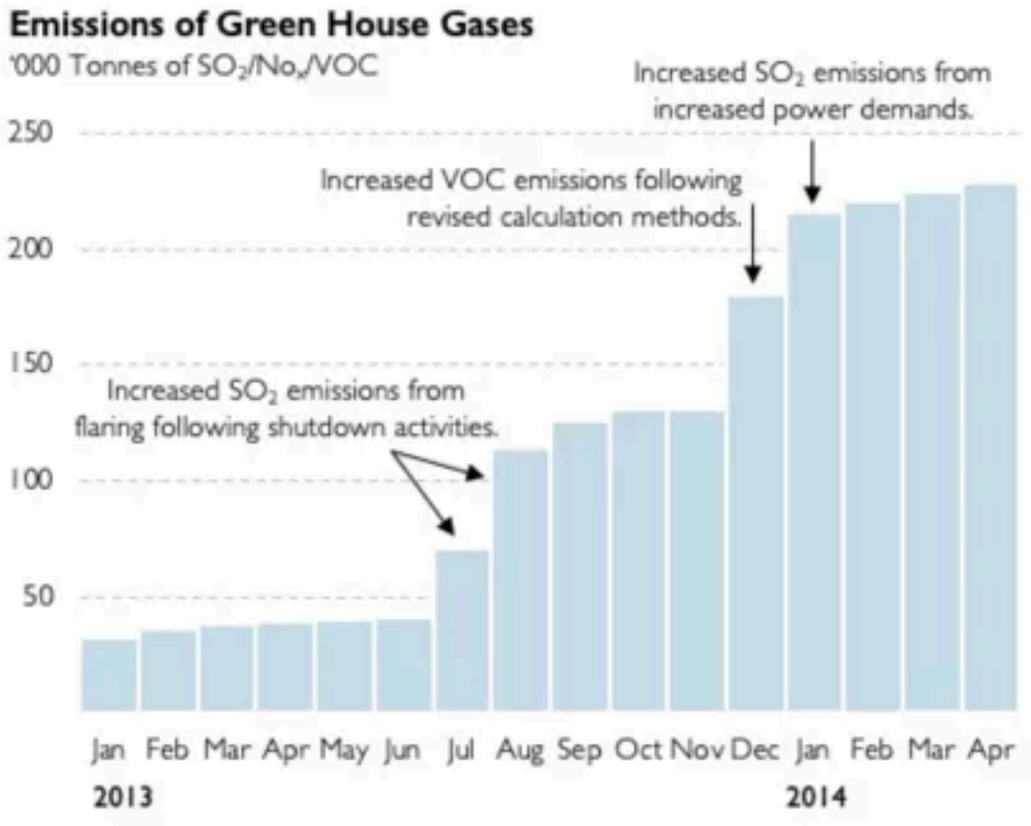
- Show code for the graphs

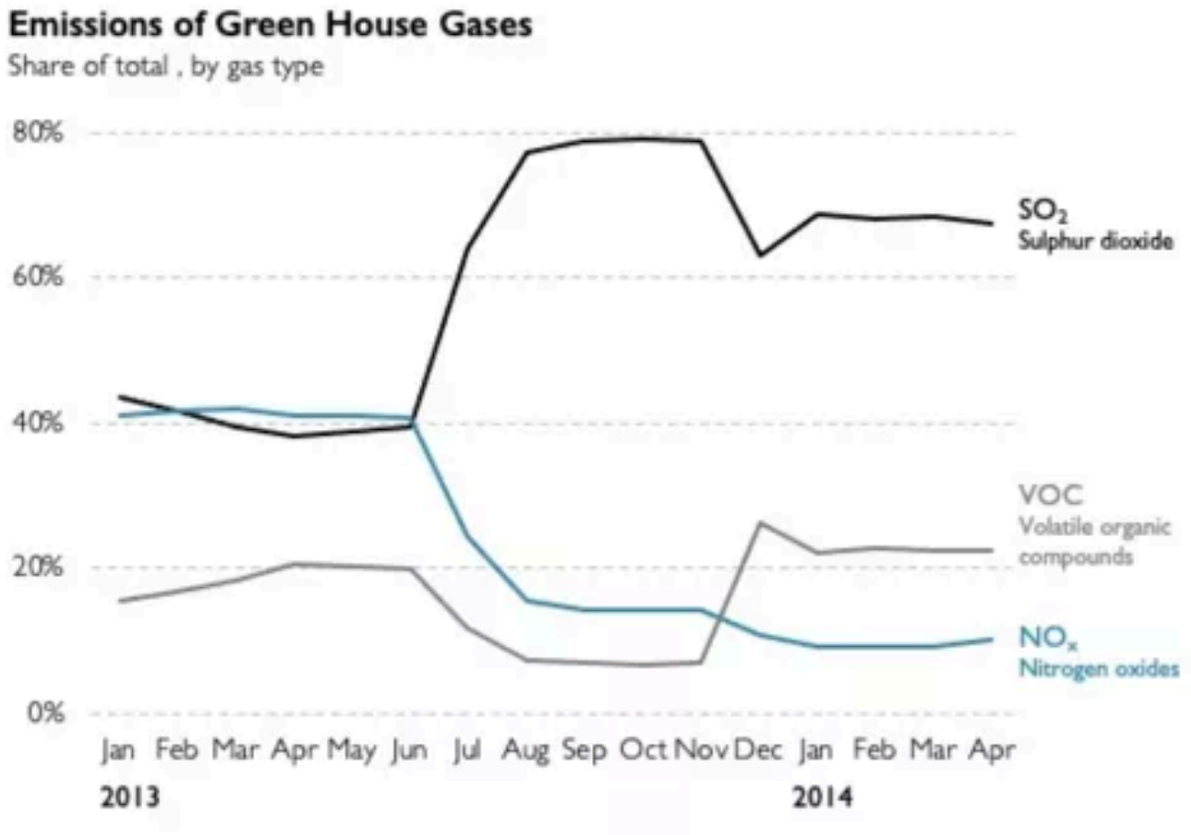
Only what matters should be reported



Source: simplexct.com

Only what matters should be reported





Source: simplexct.com

Only what matters should be reported

Works for tables as well…

Short-term credit lines by bank

June 30, 1980 (millions USD)

	Finance Companies					Manufacturing Companies							
Bank	**Canada**	**Germany**	**UK**	**USA**	***Total***	**Argentina**	**Australia**	**Canada**	**France**	**UK**	**USA**	***Total***	***Grand Total***
Allied & Associates	0.0	0.0	18.6	0.0	***18.6***	0.0	0.0	0.0	0.0	0.0	0.0	***0.0***	***18.6***
Bank of America	0.0	0.0	0.0	17.5	***17.5***	0.0	3.3	0.0	0.0	0.0	17.5	***20.8***	***38.3***
Bankers Trust	0.0	0.0	0.0	13.0	***13.0***	0.0	0.0	0.0	0.0	0.0	13.0	***13.0***	***26.0***
Banque National de Paris	0.0	0.0	11.3	0.0	***11.3***	0.0	0.0	15.0	34.6	0.0	0.0	***49.6***	***60.9***
Barclays	0.0	0.0	42.5	0.0	***42.5***	0.0	0.0	0.0	0.0	133.4	0.0	***133.4***	***175.9***
Chase Manhattan	0.0	0.0	0.0	15.2	***15.2***	0.0	0.0	0.0	0.0	0.0	15.0	***15.0***	***30.2***
Chemical	0.0	0.0	0.0	13.3	***13.3***	0.0	0.0	0.0	0.0	0.0	13.0	***13.0***	***26.3***
CIBC	37.8	0.0	27.1	0.0	***64.9***	0.0	0.0	222.9	3.6	3.4	0.0	***229.9***	***294.8***
Citibank	0.0	0.0	0.0	17.5	***17.5***	0.5	0.0	0.0	0.0	1.1	17.5	***19.1***	***36.6***
Commerzbank	0.0	3.3	0.0	0.0	***3.3***	0.0	0.0	0.0	0.0	0.0	0.0	***0.0***	***3.3***
Continental Illinois	0.0	0.0	4.6	21.8	***26.4***	0.0	0.0	0.0	0.0	0.0	21.1	***21.1***	***47.5***
Crédit Lyonnais	0.0	0.0	9.0	0.0	***9.0***	0.0	0.0	0.0	33.0	0.0	3.0	***36.0***	***45.0***
Deutsche Bank	0.0	3.3	0.0	0.0	***3.3***	0.0	0.0	0.0	0.0	0.0	0.0	***0.0***	***3.3***
Dresdner	0.0	3.3	0.0	0.0	***3.3***	0.0	0.0	0.0	0.0	0.0	0.0	***0.0***	***3.3***
FNB Chicago	0.0	0.0	0.0	15.0	***15.0***	0.0	0.0	0.0	0.0	0.0	15.0	***15.0***	***30.0***
Lloyds	0.0	0.0	42.5	0.0	***42.5***	0.0	0.0	0.0	0.0	36.2	0.0	***36.2***	***78.7***
Midland	0.0	0.0	54.3	0.0	***54.3***	0.0	0.0	0.0	0.0	36.2	0.0	***36.2***	***90.5***
Royal Bank of Canada	0.0	0.0	11.3	0.0	***11.3***	0.0	0.0	0.0	0.0	0.0	15.0	***15.0***	***26.3***
Société General	0.0	0.0	9.0	0.0	***9.0***	0.0	0.0	0.0	50.1	0.0	0.0	***50.1***	***59.1***
Toronto Dominion	0.0	0.0	6.8	0.0	***6.8***	0.0	0.0	0.0	0.0	0.0	15.2	***15.2***	***22.0***
Others	0.0	7.8	50.8	78.1	***136.7***	40.1	25.5	0.0	16.1	19.2	48.2	***149.1***	***285.8***
Total	***37.8***	***17.7***	***287.8***	***191.4***	***534.7***	***40.6***	***28.8***	***237.9***	***137.4***	***229.5***	***193.5***	***867.7***	***1,402.4***

Short-term credit lines by bank

June 30, 1980 (millions USD)

		MANUFACTURING COMPANIES							FINANCE COMPANIES				
Bank	Grand Total	Canada	UK	USA	France	Argentina	Australia	Total	UK	USA	Canada	Germany	Total
Total	**1,402.4**	**237.9**	**229.5**	**193.5**	**137.4**	**40.6**	**28.8**	**867.7**	**287.8**	**191.4**	**37.8**	**17.7**	**534.7**
CIBC	294.8	222.9	3.4	-	3.6	-	-	229.9	27.1	-	37.8	-	64.9
Others	285.8	-	19.2	48.2	16.1	40.1	25.5	149.1	50.8	78.1	-	7.8	136.7
Barclays	175.9	-	133.4	-	-	-	-	133.4	42.5	-	-	-	42.5
Midland	90.5	-	36.2	-	-	-	-	36.2	54.3	-	-	-	54.3
Lloyds	78.7	-	36.2	-	-	-	-	36.2	42.5	-	-	-	42.5
Banque National de Paris	60.9	15.0	-	-	34.6	-	-	49.6	11.3	-	-	-	11.3
Société General	59.1	-	-	-	50.1	-	-	50.1	9.0	-	-	-	9.0
Continental Illinois	47.5	-	-	21.1	-	-	-	21.1	4.6	21.8	-	-	26.4
Crédit Lyonnais	45.0	-	-	3.0	33.0	-	-	36.0	9.0	-	-	-	9.0
Bank of America	38.3	-	-	17.5	-	-	3.3	20.8	-	17.5	-	-	17.5
Citibank	36.6	-	1.1	17.5	-	0.5	-	19.1	-	17.5	-	-	17.5
Chase Manhattan	30.2	-	-	15.0	-	-	-	15.0	-	15.2	-	-	15.2
FNB Chicago	30.0	-	-	15.0	-	-	-	15.0	-	15.0	-	-	15.0
Royal Bank of Canada	26.3	-	-	15.0	-	-	-	15.0	11.3	-	-	-	11.3
Chemical	26.3	-	-	13.0	-	-	-	13.0	-	13.3	-	-	13.3
Bankers Trust	26.0	-	-	13.0	-	-	-	13.0	-	13.0	-	-	13.0
Toronto Dominion	22.0	-	-	15.2	-	-	-	15.2	6.8	-	-	-	6.8
Allied & Associates	18.6	-	-	-	-	-	-	-	18.6	-	-	-	18.6
Commerzbank	3.3	-	-	-	-	-	-	-	-	-	-	3.3	3.3
Deutsche Bank	3.3	-	-	-	-	-	-	-	-	-	-	3.3	3.3
Dresdner	3.3	-	-	-	-	-	-	-	-	-	-	3.3	3.3

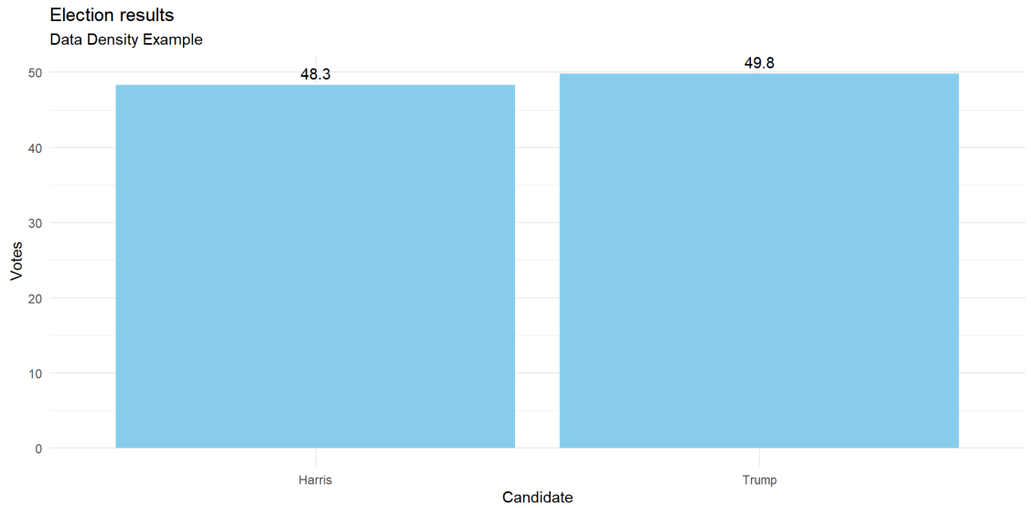
Source: simplexct.com

Data density

The data density takes the number of data points that are being graphed relative to the physical size of the graphic to capture the principle of aiming to present many numbers in a small space:

- Data density = $\frac{\text{number of entries in data matrix}}{\text{area of data graphic}}$

Data density



1. Graph

Election results	
Data Density Example	
candidate	votes
Trump	49.8
Harris	48.3

2. Table

49.8% voted for Trump, 48.3% for Harris.

3. Text

Data visualization: from a graph to a story

- Two pieces of advice I personally received:

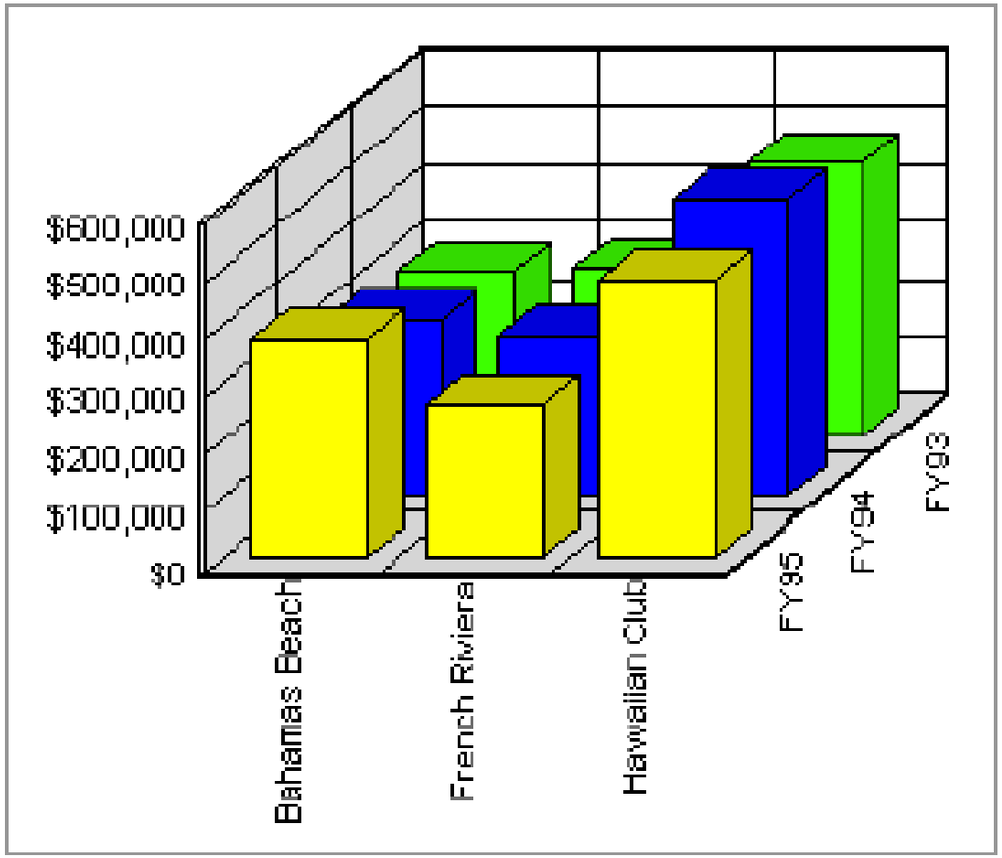
  1. If possible, **fit your whole story in one graph**.
  2. Your audience should understand your graph **without the need of listening to you** or **reading your text.**

- Be simple and avoid unnecessary fanciness.
- Avoid pie charts and 3D charts.

A Design problem

A Design Problem

What is wrong with the graph below? Create your own version of the graph.



Source: perceptual edge

A Design Problem

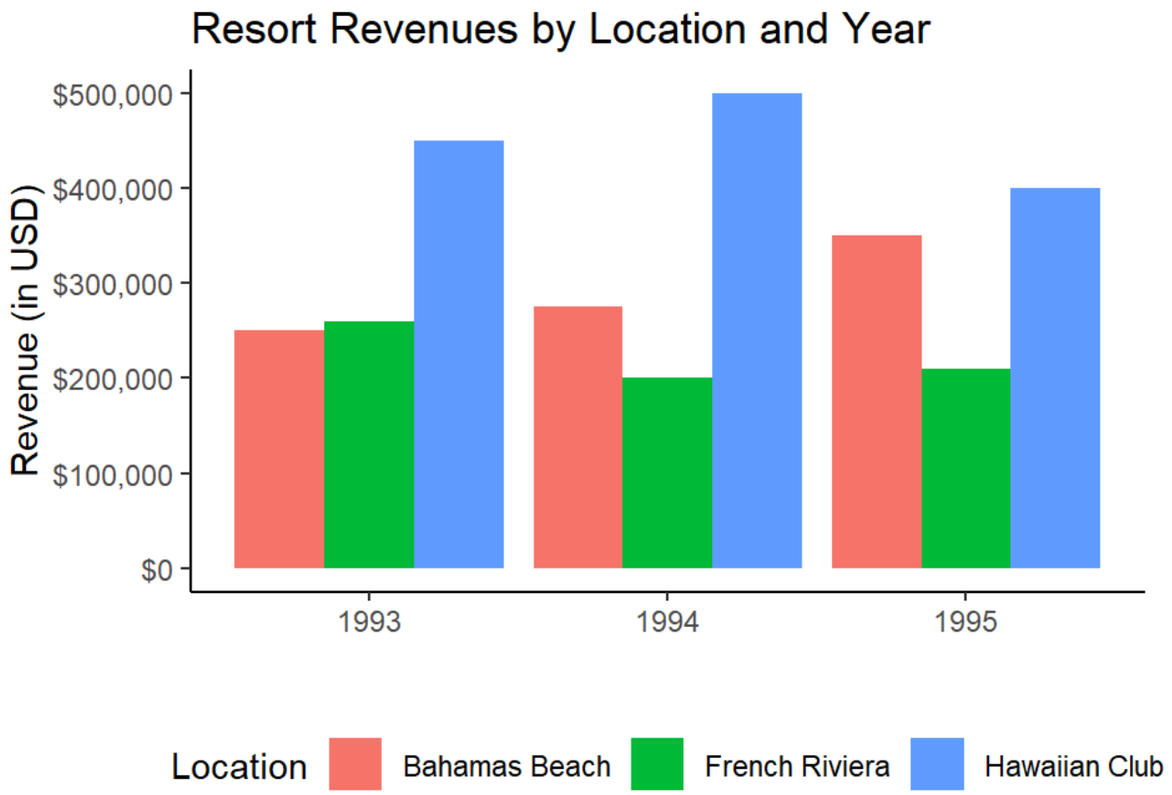
Use the following data to create your own version of the graph:

```
dataChallenge <- data.frame(
  Location = rep(c("Bahamas Beach", "French Riviera", "Hawaiian Club"), each = 3),
  Fiscal_Year = rep(c("FY93", "FY94", "FY95"), times = 3),
  Revenue = c(
    250000, 275000, 350000,  # Bahamas Beach (FY93, FY94, FY95)
    260000, 200000, 210000,  # French Riviera (FY93, FY94, FY95)
    450000, 500000, 400000   # Hawaiian Club (FY93, FY94, FY95)
  )
)
```

The problem with this graph

- The 3-D bars are impossible to read.
- The heavy grid lines offer nothing but distraction.
- The vertically-oriented labels (i.e., the resort names and years) are difficult to read.
- The years run from back to front, which is counter-intuitive.

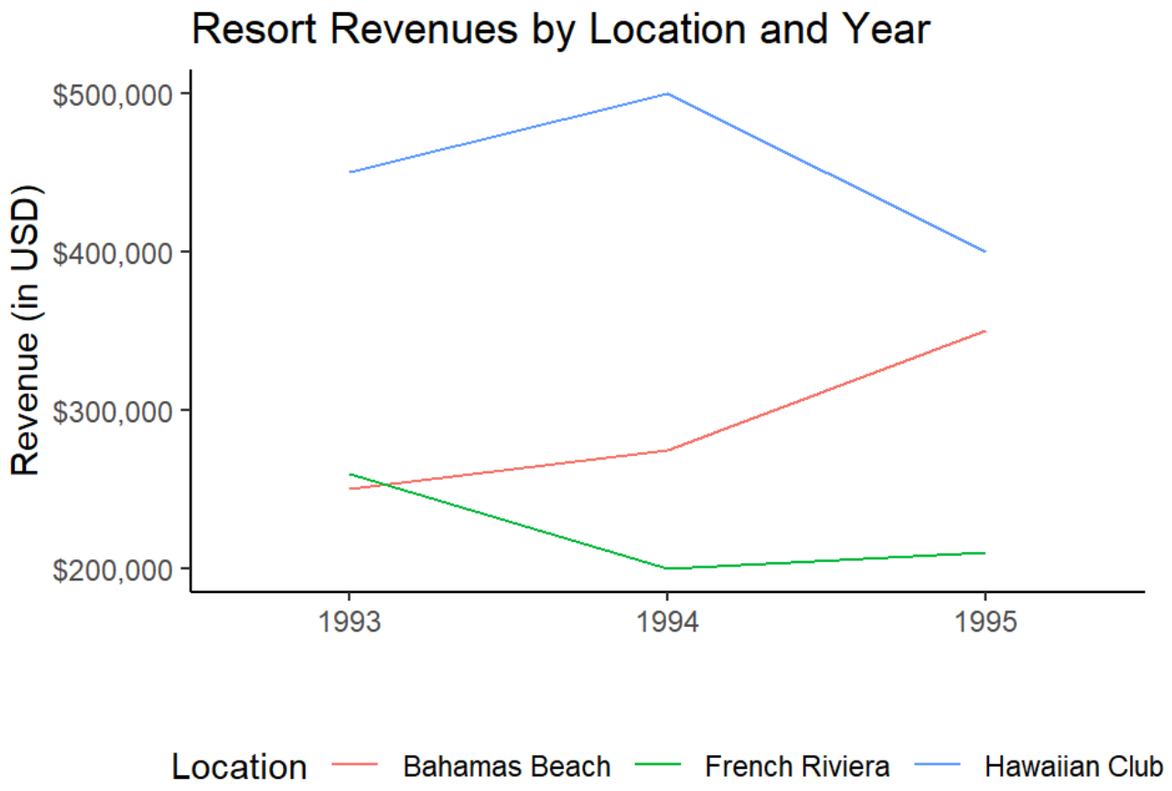
A first solution: comparative performance



- The three resorts have been arranged in order of rank, based on revenue, to highlight their comparative performance.
- The years have been arranged from left to right, which is intuitive.
- The legend has been placed below the bars.

▶ Show code

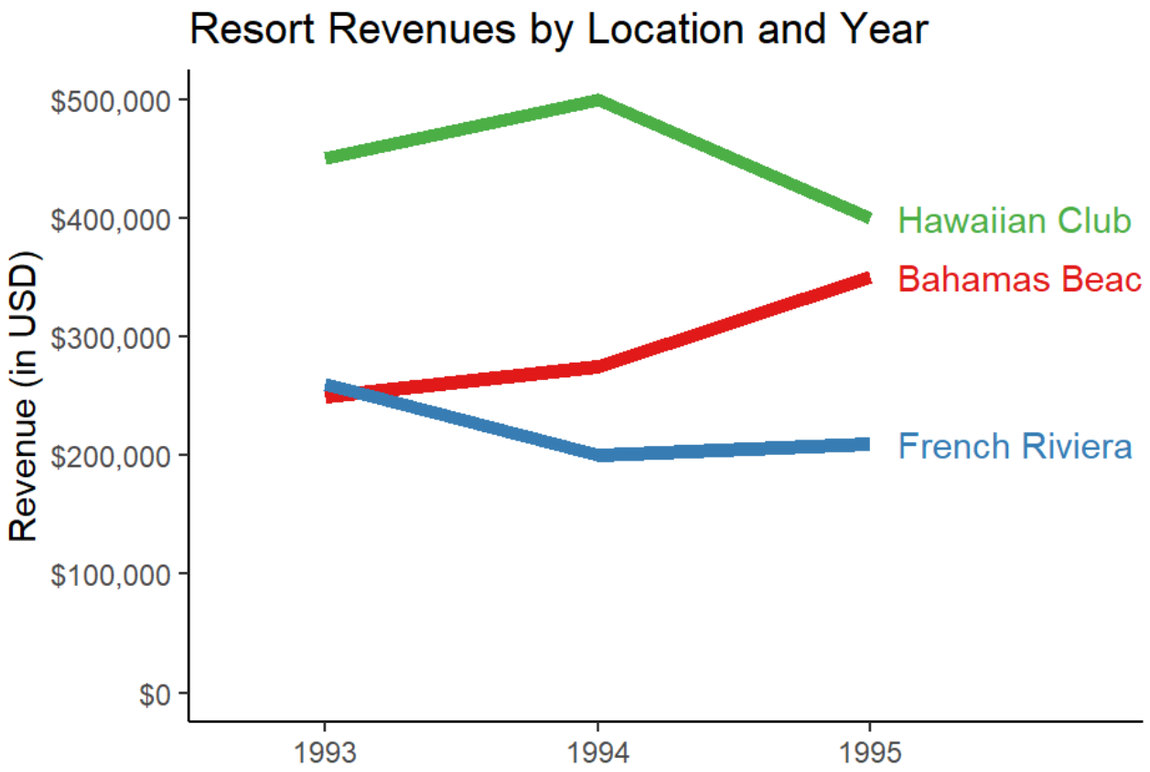
A second solution: change of revenue over time



- This design makes it easy to see how revenue has changed from year to year at each of these resorts.
- However, the magnitudes are difficult to read because the y-axis does not start at 0!
- The eye is still going back and forth between the lines and the legend.

▶ Show code

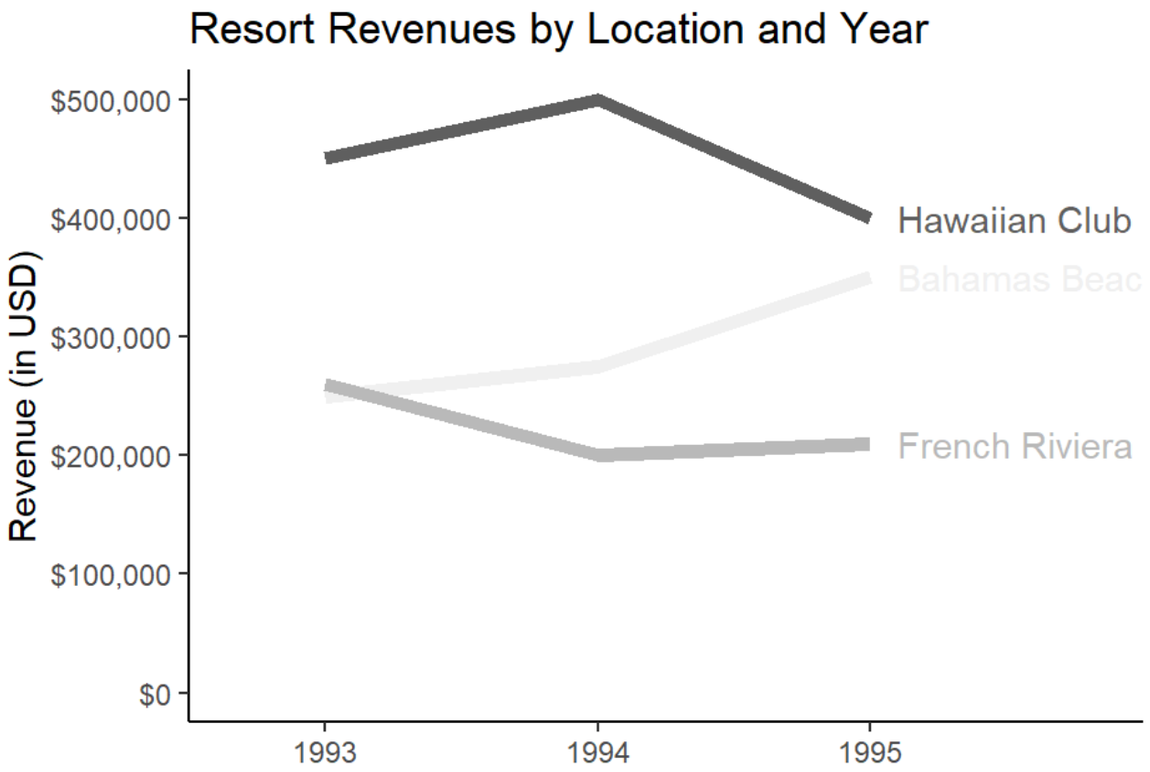
A second solution: change of revenue over time



- This design makes it easy to see how revenue has changed from year to year at each of these resorts.

▶ Show code

A second solution: change of revenue over time



- Different color palette.

▶ Show code

Conclusion

Data visualization is an art of story-telling, deception, and scientific exactitude 🤓.

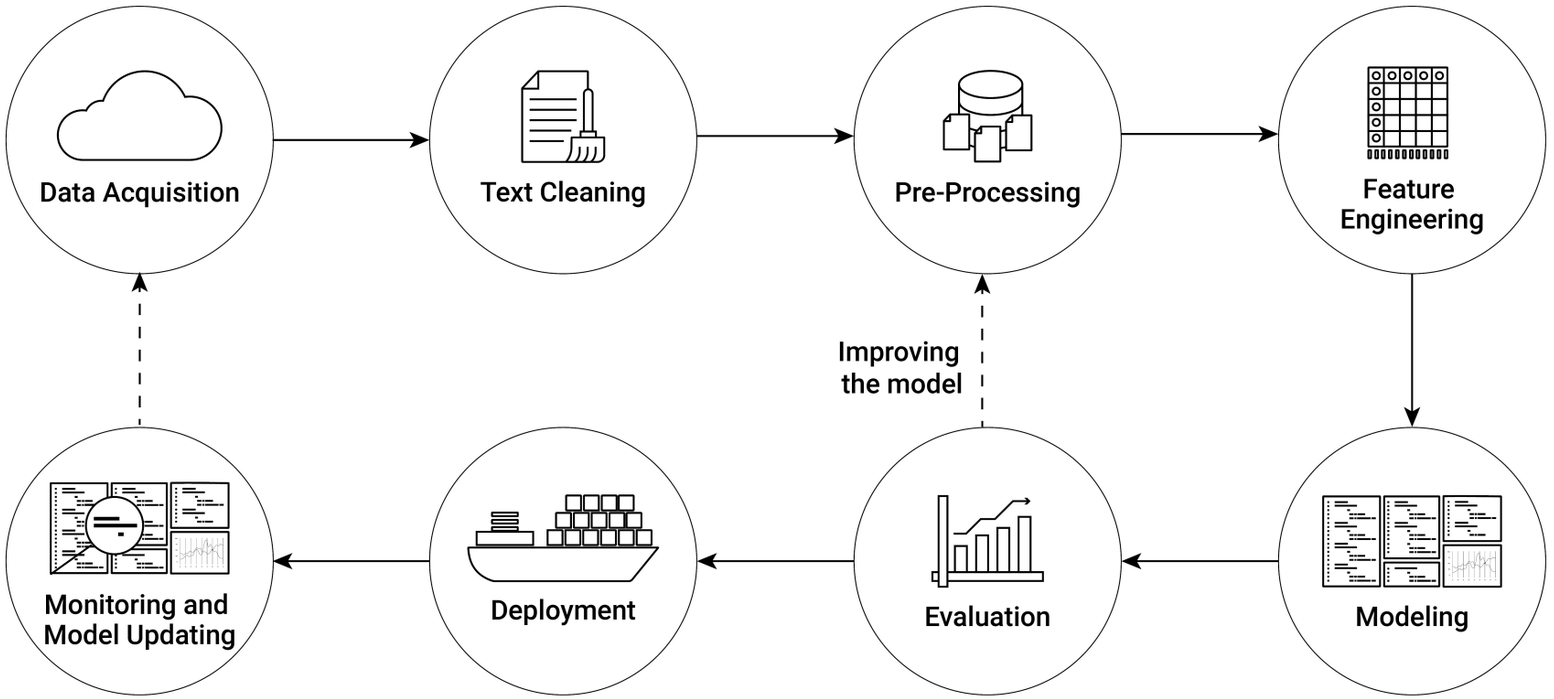
Text data

Text data is increasedly used

- Text as data has become increasingly available due to the Internet and text digitization.
- Examples: literary texts, financial analyses, social media reactions, political discourses, etc.
- Main challenge: **Text is unstructured**.

Eight Steps in Text Analysis

Focus on steps 1-4 for this course.



Key R Packages for Text Analysis

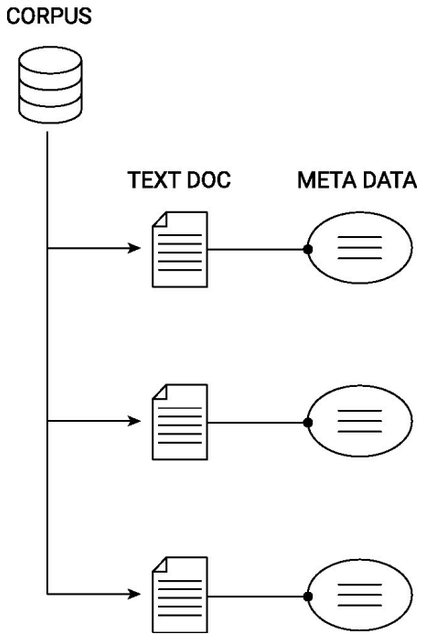
`tidytext`: Converts text to/from tidy formats. Works well with tidyverse.

`quanteda`: Comprehensive package for preprocessing, visualization, and statistical analysis.

From Raw Text to Corpus

The base, raw material, of quantitative text analysis is a **corpus**. A corpus is, in NLP, *a collection of authentic text organized into datasets*.

- Example: Newspapers, novels, tweets, etc.
- In `quanteda`: A data frame with a character vector for documents and additional metadata columns.



Parse text data

- Text in a raw form is often found in a `.json` format (after web scraping), in a `.csv` format, or in simple `.txt` files.
- The first task is then to import the text data in R and transform it as a corpus.
- We will use the `inauguration` corpus from `quanteda`, which is a standard corpus used in introductory text analysis. It contains the inauguration discourses of the five first US presidents.
- This text data can be loaded from the `readtext` package. The text is contained in a csv file, and is loaded with the `read.csv()` function. The metadata of this corpus is the year of the inauguration and the name of the president taking office.

```
# set path to the package folder
path_data <- system.file("extdata/", package = "readtext")

# import csv file
dat_inaug <- read.csv(paste0(path_data, "/csv/inaugCorpus.csv"))
names(dat_inaug)
```

```
[1] "texts"     "Year"      "President" "FirstName"
```

Create a corpus

```
# Create a corpus
corp <- corpus(dat_inaug, text_field = "texts")
print(corp)
```

```
Corpus consisting of 5 documents and 3 docvars.
text1 :
"Fellow-Citizens of the Senate and of the House of Representa..."

text2 :
"Fellow citizens, I am again called upon by the voice of my c..."

text3 :
"When it was first perceived, in early times, that no middle ..."

text4 :
"Friends and Fellow Citizens: Called upon to undertake the du..."

text5 :
"Proceeding, fellow citizens, to that qualification which the..."
```

```
# Look at the metadata in the corpus using `docvars`
docvars(corp)
```

```
  Year  President FirstName
1 1789 Washington    George
2 1793 Washington    George
3 1797      Adams      John
4 1801  Jefferson    Thomas
5 1805  Jefferson    Thomas
```

```r
# In quanteda, the metadata in a corpus can be handled like data frames.
docvars(corp, field = "Century") <- floor(docvars(corp, field = "Year") / 100) + 1
```

Regular Expressions

Used to detect patterns in strings, replace parts of text, extract information from text.

- The use of the `stringr()` package has made regular expressions easier to deal with.
  - `str_count()`

```
# Count occurences of the word "peace"
str_count(corp, "[Pp]eace")
```

```
[1] 0 0 5 7 4
```

```
# Count occurences of the words "peace" OR "war"
str_count(corp, "[Pp]eace|[Ww]ar")
```

```
[1]  1  0 10 10  8
```

Regular Expressions

Used to detect patterns in strings, replace parts of text, extract information from text.

- The use of the `stringr()` package has made regular expressions easier to deal with.
  - `str_count()`

```
# Count occurences of the mention of the first person pronoun "I"
str_count(corp, "I") # counts the number of "I" occurences. This is not what we want.
```

```
[1] 30  6 24 23 28
```

```
str_count(corp, "[I][[:space:]]") # counts the number of "I" followed by a space.
```

```
[1] 23  6 13 21 18
```

```
# Extract the first five words of each discourse
str_extract(corp, "^(\\S+\\s|[[:punct:]]|\\n){5}") # ^serves to anchor at the beginning of the string, (){5} shows the group
```

```
[1] "Fellow-Citizens of the Senate and "    "Fellow citizens, I am again "
[3] "When it was first perceived, "         "Friends and Fellow Citizens:\n\n"
[5] "Proceeding, fellow citizens, to that "
```

From Corpus to Tokens

Tokens: Building blocks of text (words, punctuation, etc.).

- LLMs operate on tokenized text as input. The tokenization process converts raw text into numerical representations that the model can process.

```
toks <- tokens(corp)
head(toks[[1]], 20)
```

```
 [1] "Fellow-Citizens" "of"             "the"            "Senate"         "and"
 [6] "of"              "the"            "House"          "of"             "Representatives"
[11] ":"               "Among"          "the"            "vicissitudes"   "incident"
[16] "to"              "life"           "no"             "event"          "could"
```

From Corpus to Tokens

Tokens: Building blocks of text (words, punctuation, etc.).

- Remove punctuation and stopwords.
- Create N-grams (e.g., “not friendly”).

```
# Remove punctuation
toks <- tokens(corp, remove_punct = TRUE)
head(toks[[1]], 20)
```

```
 [1] "Fellow-Citizens" "of"             "the"            "Senate"         "and"
 [6] "of"              "the"            "House"          "of"             "Representatives"
[11] "Among"           "the"            "vicissitudes"   "incident"       "to"
[16] "life"            "no"             "event"          "could"          "have"
```

```
# Remove stopwords
stopwords("en")
```

```
  [1] "i"          "me"         "my"         "myself"     "we"         "our"        "ours"
  [8] "ourselves"  "you"        "your"       "yours"      "yourself"   "yourselves" "he"
 [15] "him"        "his"        "himself"    "she"        "her"        "hers"       "herself"
 [22] "it"         "its"        "itself"     "they"       "them"       "their"      "theirs"
 [29] "themselves" "what"       "which"      "who"        "whom"       "this"       "that"
 [36] "these"      "those"      "am"         "is"         "are"        "was"        "were"
 [43] "be"         "been"       "being"      "have"       "has"        "had"        "having"
 [50] "do"         "does"       "did"        "doing"      "would"      "should"     "could"
```

```
  [57] "ought"      "i'm"       "you're"    "he's"      "she's"     "it's"      "we're"
  [64] "they're"    "i've"      "you've"    "we've"     "they've"   "i'd"       "you'd"
  [71] "he'd"       "she'd"     "we'd"      "they'd"    "i'll"      "you'll"    "he'll"
  [78] "she'll"     "we'll"     "they'll"   "isn't"     "aren't"    "wasn't"    "weren't"
  [85] "hasn't"     "haven't"   "hadn't"    "doesn't"   "don't"     "didn't"    "won't"
  [92] "wouldn't"   "shan't"    "shouldn't" "can't"     "cannot"    "couldn't"  "mustn't"
  [99] "let's"      "that's"    "who's"     "what's"    "here's"    "there's"   "when's"
 [106] "where's"    "why's"     "how's"     "a"         "an"        "the"       "and"
 [113] "but"        "if"        "or"        "because"   "as"        "until"     "while"
 [120] "of"         "at"        "by"        "for"       "with"      "about"     "against"
 [127] "between"    "into"      "through"   "during"    "before"    "after"     "above"
 [134] "below"      "to"        "from"      "up"        "down"      "in"        "out"
 [141] "on"         "off"       "over"      "under"     "again"     "further"   "then"
 [148] "once"       "here"      "there"     "when"      "where"     "why"       "how"
```

```
toks <- tokens_remove(toks, pattern = stopwords("en"))
head(toks[[1]], 20)
```

```
 [1] "Fellow-Citizens" "Senate"          "House"           "Representatives" "Among"
 [6] "vicissitudes"    "incident"        "life"            "event"           "filled"
[11] "greater"         "anxieties"       "notification"    "transmitted"     "order"
[16] "received"        "14th"            "day"             "present"         "month"
```

From Corpus to Tokens

```r
# We can keep words we are interested in
tokens_select(toks, pattern = c("peace", "war", "great*", "unit*"))
```

```
Tokens consisting of 5 documents and 4 docvars.
text1 :
[1] "greater" "United"  "Great"   "United"  "united"  "great"   "great"   "united"

text2 :
[1] "united"

text3 :
 [1] "war"    "great"  "United" "great"  "great"  "peace"  "great"  "peace"  "peace"  "United"
[11] "peace"  "peace"
[ ... and 2 more ]

text4 :
 [1] "greatness" "unite"     "unite"     "greater"   "peace"     "peace"     "peace"     "war"
 [9] "peace"     "greatest"  "greatest"  "great"
[ ... and 1 more ]

text5 :
[1] "United" "peace"  "great"  "war"    "war"    "War"    "peace"  "peace"  "peace"
```

From Corpus to Tokens

```
# Remove "fellow" and "citizen"
toks <- tokens_remove(toks, pattern = c(
    "fellow*",
    "citizen*",
    "senate",
    "house",
    "representative*",
    "constitution"
))
```

From Corpus to Tokens

```
# Build N-grams (onegrams, bigrams, and 3-grams)
toks_ngrams <- tokens_ngrams(toks, n = 2:3)

# Build N-grams based on a structure: keep n-grams that containt a "never"
toks_neg_bigram_select <- tokens_select(toks_ngrams, pattern = phrase("never_*"))
head(toks_neg_bigram_select[[1]], 30)
```

```
 [1] "never_hear"            "never_expected"        "never_hear_veneration" "never_expected_nation"
```

From Tokens to Document-Term Matrix

- **DTM**: Rows = documents, Columns = tokens.
- Contains count frequencies or indicators.
- Use domain knowledge to reduce DTM dimensions.

Code Example:

```
dfmat <- dfm(toks)
print(dfmat)
```

```
Document-feature matrix of: 5 documents, 1,818 features (72.28% sparse) and 4 docvars.
       features
docs    among vicissitudes incident life event filled greater anxieties notification transmitted
  text1     1            1        1    1     2      1       1         1            1           1
  text2     0            0        0    0     0      0       0         0            0           0
  text3     4            0        0    2     0      0       0         0            0           0
  text4     1            0        0    1     0      0       1         0            0           0
  text5     7            0        0    2     0      0       0         0            0           0
[ reached max_nfeat ... 1,808 more features ]
```

```
dfmat <- dfm(toks)
dfmat <- dfm_trim(dfmat, min_termfreq = 2) # remove tokens that appear less than 1 times
```

Analyzing DTMs

Use DTMs for:

- Machine learning models
- Document classification
- Predicting authorship

Statistics

Very basic statistics about documents are the **top features** of each document, the frequency of expressions in the corpus.

```
library(quanteda.textstats)

tstat_freq <- textstat_frequency(dfmat, n = 5)

topfeatures(dfmat, 10)
```

```
government        may     public        can     people      shall    country      every         us
        40         38         30         27         27         23         22         20         20
   nations
        18
```

Statistics

The frequency of tokens can be represented in a text plot.

```
library(quanteda.textplots)
quanteda.textplots::textplot_wordcloud(dfmat, max_words = 100)
```

Conclusion

- **Tokens**: Absolutely crucial for LLMs. They determine how the model interprets text, manage context, and enable learning.
- **DTMs**: Indirectly useful for preprocessing, exploratory analysis, or hybrid systems but less central to modern LLMs.

DTMs are still used in business cases for description of text input.