Probabilities and statistics for machine learning and data science

Tiphaine Viard

2021 – 2022

TELECOM Paris

IP PARIS

Goals and outline

Brush up on the parts of probabilities and statistics you will need

Probabilities: Random variables (and their types), conditional probabilities, Bayes' theorem, "naive" Bayes

Statistics: Distributions, common pitfalls, paradoxes

Machine learning: Prepare your data, common pitfalls, bias-variance trade-off...

Technical: Quick introduction to the popular python tools

Give technical and conceptual tools to use in your daily practice

Random variables

Quantify some uncertainty on a population

- **Discrete**: dice roll, coin flip, number of people, etc.
- **Continuous**: height, waiting time, etc.

Random variable $\neq$ realization

Random variables' realizations follow a **distribution**, $X \sim P(X)$

- Cumulative ($P(X \geq x)$) or not ($P(X = x)$)
- Mass function vs density functions
- $\sum_X P(X) = 1$, $P(X) > 0$
- Independence: $A \perp\!\!\!\perp B \Leftrightarrow P(A \cap B) = P(A)P(B)$

Conditional probability

The probability that A happens knowing B has happened is:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

e.g. Prob. that it rains given that the ground is wet

Be careful, in general, $P(B|A) \neq P(A|B)$

Bayes' theorem

So, what is the link between $P(A|B)$ and $P(B|A)$?

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

This is one of the key theorems of machine learning and data science

The Naive Bayes classifier

Suppose we want to classify email as **legitimate** or **spam**. Given an email, described by features $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ we want to **classify** it into K classes $\{c_1, \dots, c_K\}$.

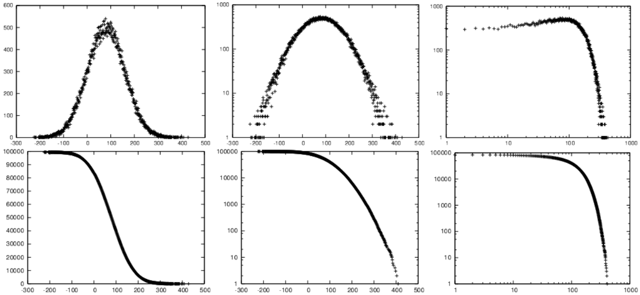
$$p(\boldsymbol{x}|c_k) = \frac{p(c_k|\boldsymbol{x})p(c_k)}{p(\boldsymbol{x})} = \frac{1}{p(\boldsymbol{x})} p(c_k) \prod_i p(x_i|c_k)$$

Trick: assume $\forall i, j < n, x_i$ independent x_j

$$\hat{y} = \arg\max_k p(\boldsymbol{x}|c_k)$$

Easy to implement, scalable, performs surprisingly well

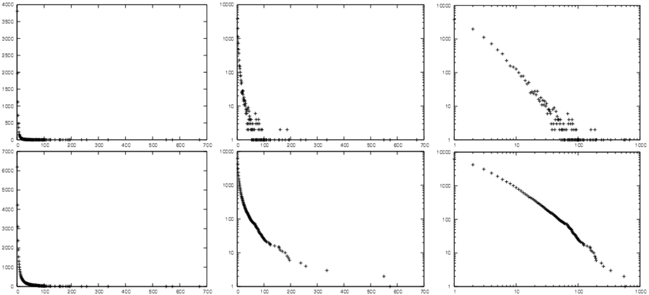
Common data distributions



Be careful: many tools have assumptions on the underlying distribution!

Source: *T. Viard*

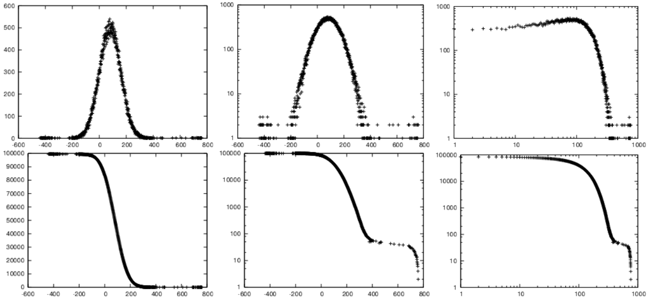
Common data distributions



Be careful: many tools have assumptions on the underlying distribution!

Source: *T. Viard*

Common data distributions



Be careful: many tools have assumptions on the underlying distribution!

Source: *T. Viard*

Distributions meet reality

The **power-law** is the most commonly seen distribution in real-world data

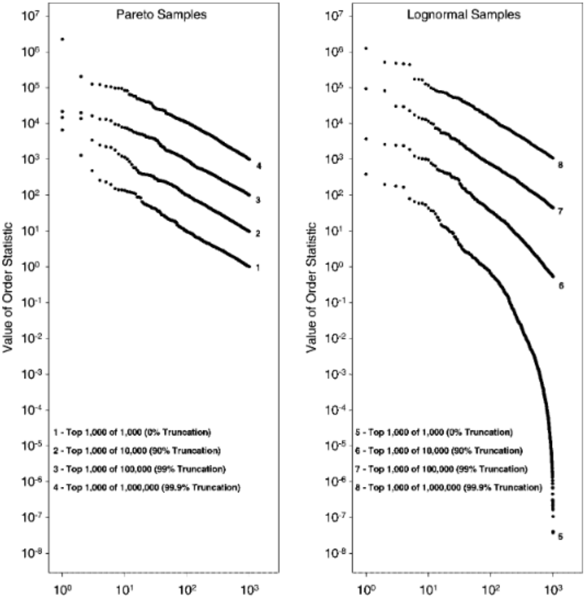
$$f(x) = ax^{-\alpha}$$

Countless examples :

- Pareto law
- Zipf law
- Scaling law
- "heavy-tail" distributions
- 80-20 principle
- etc.

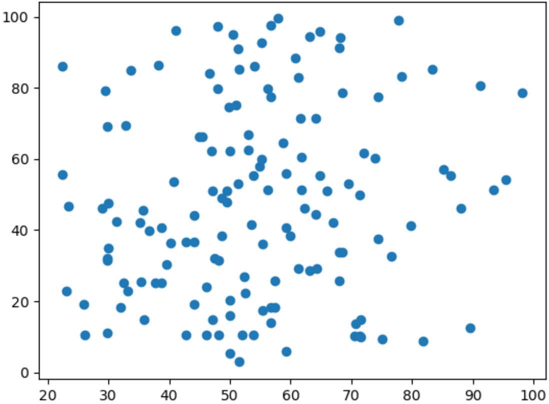
Be careful : data actually rarely follows a power law !

False, weak and inverse power laws



Always plot your data

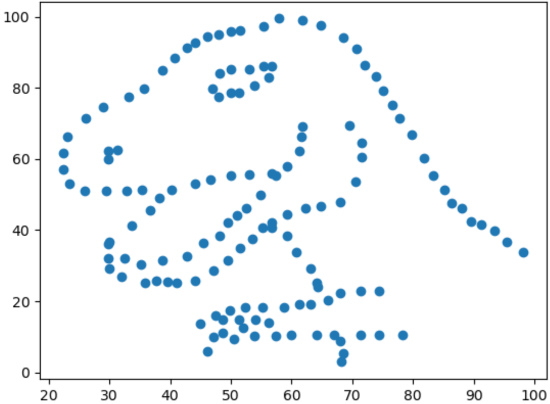
Dataset: $|X| = 142$ points, $\mu_1(X) = 54.22(16.76)$, $\mu_2(X) = 47.83(26.93)$



Source: *The Datasaurus dataset, A. Cairo, 2016*

Always plot your data

Dataset: $|X| = 142$ points, $\mu_1(X) = 54.22(16.76)$, $\mu_2(X) = 47.83(26.93)$



Source: *The Datasaurus dataset, A. Cairo, 2016*

What is machine learning?

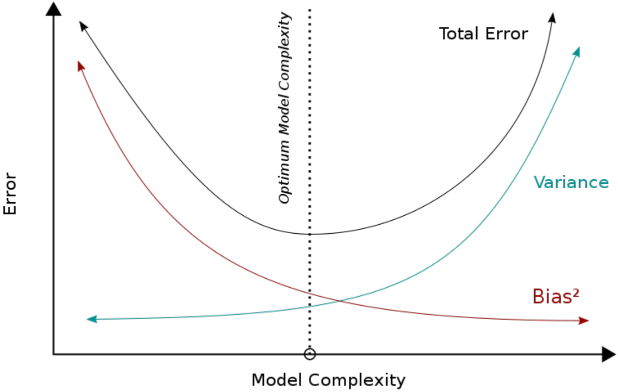
Perform a **task** with a clear (=formalized) objective

Supervised vs **unsupervised**

- (U) Find subgroups of interest
- (U) Detect anomalies
- (S) Predict a price, the weather, autocomplete text
- (S) Classify documents into an ontology

Underfitting and overfitting

The **bias-variance** trade off: how to learn well, but also generalize?



Source: *Wikipedia*

The bias-variance trade-off

Source: *Wikimedia commons*
More "complex" models result in higher variance

This is **not a theorem** !

$f_{a,b}(x) = a\sin(bx)$ can interpolate any number of points, and has **high bias, high variance**

Source: *https://arxiv.org/pdf/1912.08286.pdf*

Splitting your dataset

Splitting is a way to use all your data for a machine learning task

Three different splits:

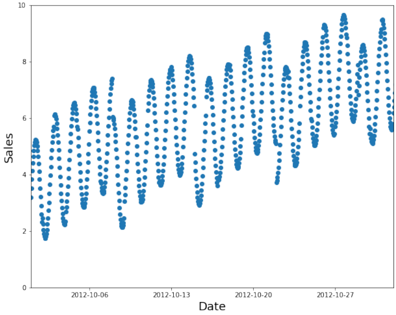
- Train: to understand the data;
- Validation: to see if you're doing well;
- Test: to see how you truly generalize

Is chronological order important? Is categorical representativity important? Which properties will you be breaking?

Cross-validation: swap the splits and apply your algorithm

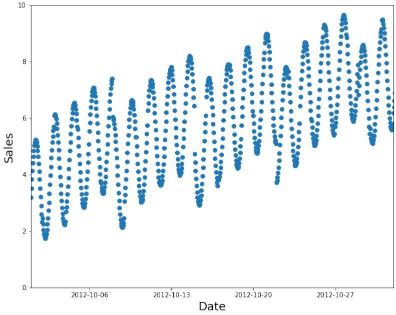
Evaluate performance on average (please report deviations!)

Splitting your data : some examples



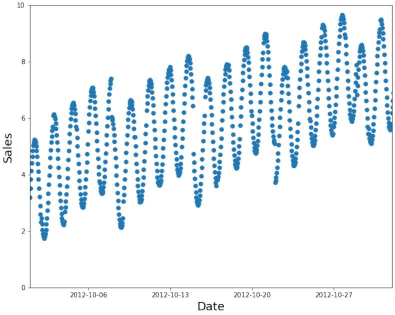
- Goal : predict sales over time

Splitting your data : some examples



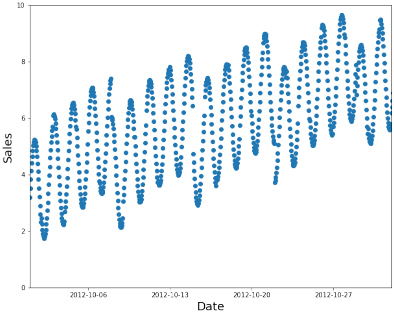
- Time series : random selection is too easy and inaccurate. Split temporally instead

Splitting your data : some examples



- Time series : random selection is too easy and inaccurate. Split temporally instead
- Goal : identify dangerous driving situations

Splitting your data : some examples



- Time series : random selection is too easy and inaccurate. Split temporally instead

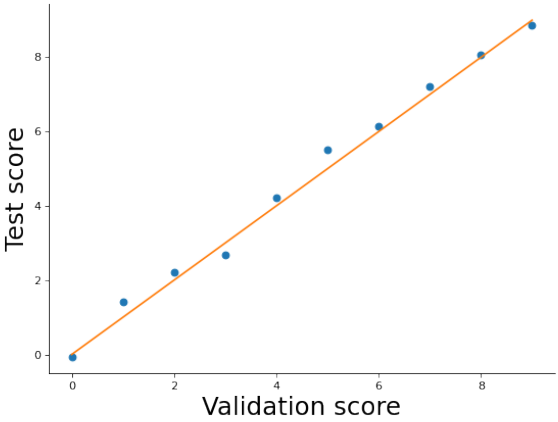
- You might recognize the person, rather than the situation during testing; test dataset should have only unseen persons

Is my validation dataset good?

How can we choose a **good validation** dataset ?

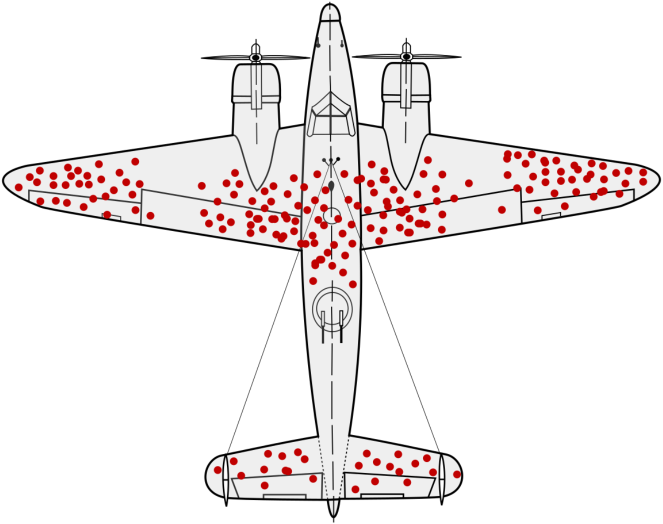
Is my validation dataset good?

How can we choose a **good validation** dataset ?



Survivorship bias

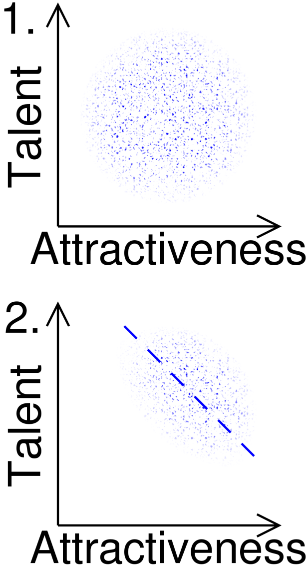
The observation sometimes implies a condition



Source: *Wikimedia commons*

The Berkson paradox

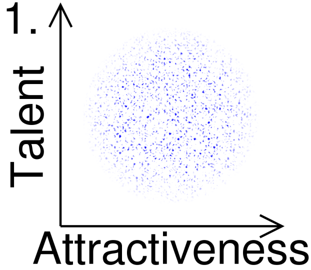
- Talent and attractiveness are **uncorrelated** in the general population
- However, looking at talent vs attractiveness among celebrities yields a **negative correlation**
- Why?

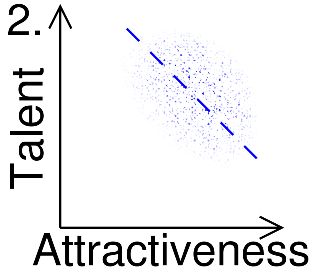


This is a **sampling/selection** bias

The Berkson paradox

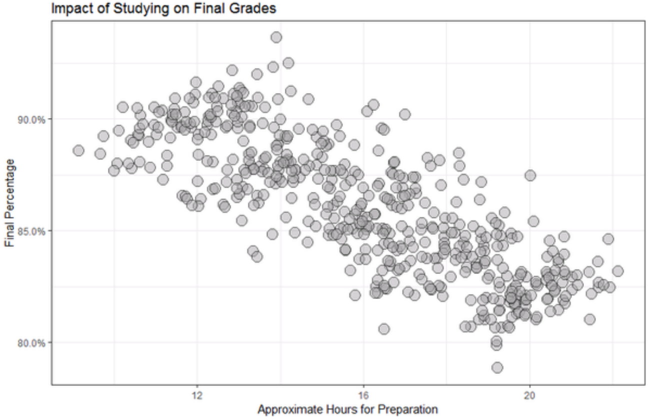
- Talent and attractiveness are **uncorrelated** in the general population
- However, looking at talent vs attractiveness among celebrities yields a **negative correlation**
- Why?
- The second plot ignores that part of the population is **nor talented nor attractive**



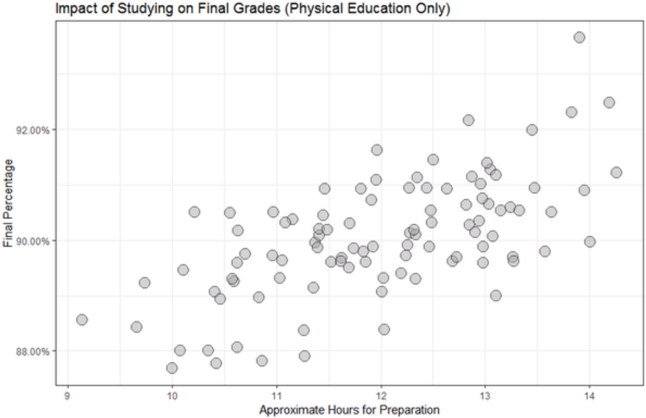


This is a **sampling/selection** bias

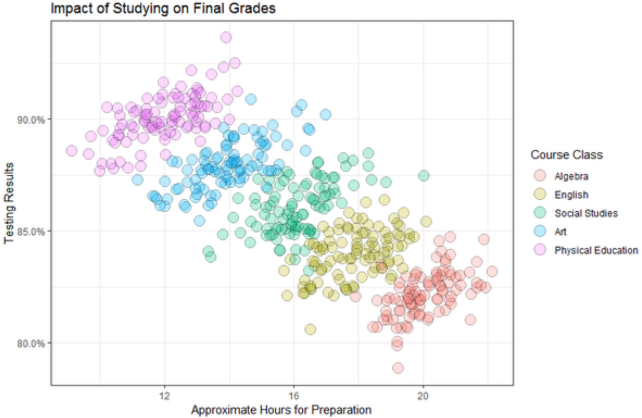
The Yule-Simpson paradox



The Yule-Simpson paradox



The Yule-Simpson paradox

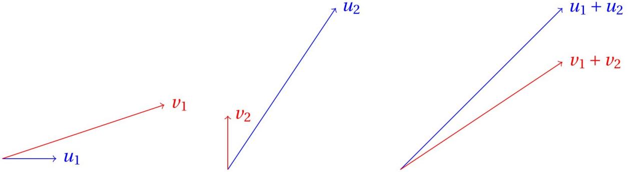


The Yule-Simpson paradox, formalized

D groups of data, such that D_1 has A_i trials and a_i successes, and D_2 has B_i trials and b_i successes.

$$\forall i, \frac{a_i}{A_i} \geq \frac{b_i}{B_i}, \text{ but } \frac{\sum_i a_i}{\sum_i A_i} \leq \frac{\sum_i b_i}{\sum_i B_i}$$

A geometric interpretation : suppose $u_1 < v_1$, $u_2 < v_2$, but $u_1 + u_2 > v_1 + v_2$



Faster python code: `python` vs `pypy`

`python` (CPython) compiles your code into bytecode

Reliant on the python virtual machine

`pypy` uses **JIT compilation**: static bytecode compilation + dynamic machine-dependent compilation (at runtime)

7.5 times faster on average, up to 50 times

Usage is **simple**: `pypy3 yourcode.py`

So, should we use `pypy` all the time?

Limitations of `pypy`

`pypy` has **significant limitations**:

- It **does not support C bindings** : `numpy`, `scipy` etc. will not be fastened,
- **Static compilation** has a cost: not great for once-run scripts,
- Garbage collection is run differently: usually implies larger memory footprint
- It is not **up-to-date**: `pypy3` runs `python 3.7`

`pypy`: an easy solution to fasten pure python scripts that will be run often

Faster python code: `cython`

What if we could compile python to C/C++?

A superset of Python that compiles to C/C++

- Wrap existing C libraries
- Fasten current code by static typing
- Fasten current code through bindings (e.g. numpy ndarrays)

Example :
`https://github.com/sknetwork-team/scikit-network/blob/master/sknetwork/ranking/betweenness.pyx`

Faster python code: profiling your code

Obtain a rundown of functions, bottlenecks...

```
$ python -m [-s ncalls] cProfile <your_script>
```

```
Array created successfully
         400039 function calls in 0.088 seconds

   Ordered by: cumulative time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.004    0.004    0.088    0.088 <ipython-input-1-66b56f7cc511>:10(main)
        1    0.057    0.057    0.083    0.083 <ipython-input-1-66b56f7cc511>:1(create_ar
   400000    0.026    0.000    0.026    0.000 {method 'append' of 'list' objects}
        1    0.000    0.000    0.000    0.000 <ipython-input-1-66b56f7cc511>:6(print_sta
        1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
        2    0.000    0.000    0.000    0.000 /usr/local/lib/python3.6/dist-packages/ipy
        3    0.000    0.000    0.000    0.000 /usr/local/lib/python3.6/dist-packages/ipy
        3    0.000    0.000    0.000    0.000 /usr/local/lib/python3.6/dist-packages/zmq
        3    0.000    0.000    0.000    0.000 /usr/lib/python3.6/threading.py:1104(is_al
        2    0.000    0.000    0.000    0.000 /usr/local/lib/python3.6/dist-packages/ipy
        3    0.000    0.000    0.000    0.000 /usr/lib/python3.6/threading.py:1062(_wait
        2    0.000    0.000    0.000    0.000 /usr/local/lib/python3.6/dist-packages/ipy
        3    0.000    0.000    0.000    0.000 {method 'acquire' of '_thread.lock' object
        3    0.000    0.000    0.000    0.000 /usr/local/lib/python3.6/dist-packages/ipy
        2    0.000    0.000    0.000    0.000 {built-in method posix.getpid}
```

Running code

- Jupyter notebooks/labs or nbdev2
- Google Colab
- Scripts and `gitlab`
- Telecom servers

From home :

`$ ssh <login>@ssh.enst.fr`

Then bounce :

`$ ssh lame14` or `$ ssh gpu1`

Copy files (if not using git) :

`$ scp <file1> ... <fileN> <login>@lame14:<dst>`

The many shapes of bias

Social vs **statistical**, **Model** vs **data**

- “The data isn’t good enough!”
- Good average performance $\neq$ good local performance
- Data can be noisy, statistically biased
- But also socially biased
- Generalization vs stochastic parrotting
- What is lost when fitting?
- How do we adapt to error?

Datasheets for datasets

A coffee machine comes with a specification. Why not datasets ?

- Built over 4 years with an interdisciplinary focus
- Write down the **motivation**, **collection process**, **intended use**
- Gaining traction among academics and big tech companies

If the dataset does not relate to people, you may skip the remaining questions in this section.

27. **Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (for example, websites)?**
28. **Were the individuals in question notified about the data collection?** If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.
29. **Did the individuals in question consent to the collection and use of their data?** If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.
30. **If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?** If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).
31. **Has an analysis of the potential impact of the dataset and its use on data subjects (for example, a data protection impact analysis) been conducted?** If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.
32. **Any other comments?**

Source: *Datasheets for datasets, Communications of the ACM, Gebru et al., 2021*

Bridging data science and the social sciences

Statistical learning often assume **independence** and a **closed world**

Both are wrong in the real world

Inferential statistics are **not good predictors**[1] in social sciences settings

- stochasticity of social phenomena?
- hidden variables?

[1] Abell, P. (2009). History, case studies, statistics, and causal inference. European Sociological Review, 25(5), 561-567.

Understanding framing

Question **the limits** of your data and model[2]

- Who was involved in the collection?
- What is the domain of answerable questions?

Is your scientific question:

- About the nature of things? (**ontological**)
- About the types of knowledge and ways to extract it from things? (**epistemological**)
- About methods to achieve other goals? (**methodological**)

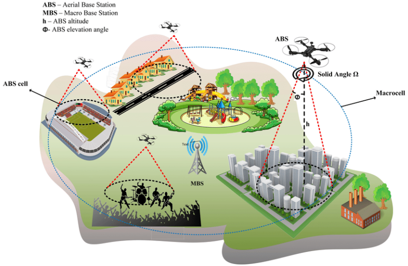
In STEM, objects do not interpret their word: social sciences deal with two levels of theoretical construction

[2]Orlikowski, W. J., & Gash, D. C. (1994). Technological frames: making sense of information technology in organizations. ACM Transactions on Information Systems (TOIS), 12(2), 174-207.

An example : the cellphone

Cellphone: one technological object

- Needs base stations;
- Needs frequency-sharing schemes;
- Needs regulation;
- Needs technical standards;
- ...



It is a technical object in a broad technical, social and regulatory context.

Opening thoughts: what is sociology?

The **systematic** study of society and social interactions

"there are no dancers without the dance, there is no dance without the dancers"

Multiple levels of analysis:

- Micro: individuals, small groups...
- Meso: groups, small institutions...
- Macro: institutions, states...
- Global: larger than a society, a state.

A few questions:

- How do these levels relate to each other?
- How can we study them jointly?
- Can describe effects of *habitualisation* and *institutionalisation*?

Closing thoughts

- **Think** before coding : do not let the data mislead you
- **Show** your data, your model to people with multiple perspectives
- Be wary of data "paradoxes"
- Data science is never "in a vacuum", but raise complicated questions
- Be wary of the "perfect" tool/model
- Seek **simplicity** !
- **Acknowledge** and **understand** the broader context of the technical work

It is OK to make errors, if you **document** them

Leverage **dual expertise**