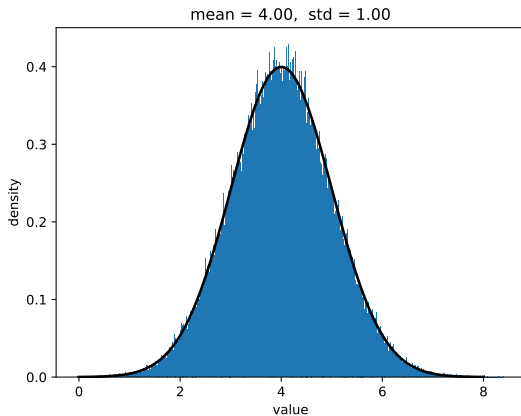


# Machine learning II, unsupervised learning and agents: overview of mathematical tools

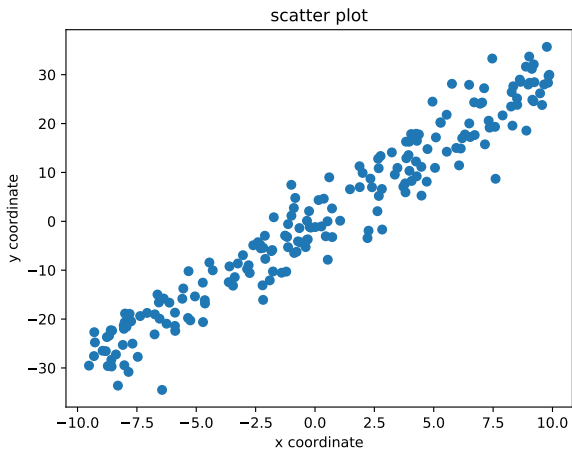


Probabilities and statistics

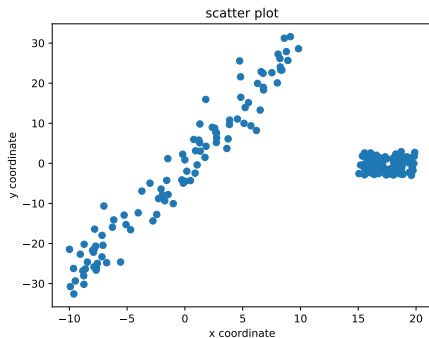
Optimization

To have a solid understanding of modern machine learning, it is necessary to be familiar with elementary probabilities and statistics.

## Random variables



## Random variables



We want to analyse how the data are **distributed**. For instance the  $x$  coordinate, the  $y$  coordinate.

## Random variables

- ▶ (informal definition) A **random variable** is a quantity that can take several values, with some randomness.
- ▶ [https://en.wikipedia.org/wiki/Random\\_variable](https://en.wikipedia.org/wiki/Random_variable)

## Random variables

- ▶ A **random variable** is a quantity that can take several values
- ▶ For instance :
  - ▶ the result of a dice throw

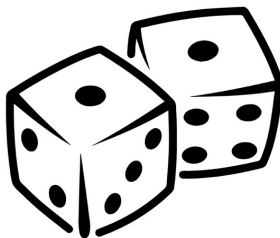


Figure – Dice

## Random variables

- ▶ A **random variable** is a quantity that can take several values
- ▶ For instance :
  - ▶ the result of a dice throw
  - ▶ waiting time with RATP



Figure – Some metro station



## Random variables

- ▶ A **random variable** is a quantity that can take several values
- ▶ For instance :
  - ▶ the result of a dice throw
  - ▶ waiting time with RATP
  - ▶ weather



Figure – Weather in November

# Random variables

- ▶ A **random variable** is a quantity that can take several values
- ▶ For instance :
  - ▶ the result of a dice throw
  - ▶ waiting time with RATP
  - ▶ weather
  - ▶ number of cars taking the périphérique at the same time

## Why are random variables important ?

- ▶ most datasets encountered in machine learning can be considered as sampled from random variables.
- ▶ this is important for theoretical studies, and hence for applications : a better theoretical understanding of a problem allows to choose the best algorithm to solve it.
- ▶ theoretical results are sometimes precise in the sense that they allow to estimate the order of magnitude of the statistical error (e.g. the prediction error) as a function of  $d$  (dimension of the samples) and  $n$  (number of samples)
- ▶ a subdomain of machine learning is "statistical learning"

## Random variables

- ▶ Some random variables are **continuous**, others **discrete**

## Random variables

- ▶ Some are **continuous**, others **discrete**
- ▶ **continuous** : weather, RATP

## Random variables

- ▶ Some are **continuous**, others **discrete**
- ▶ **continuous** : weather, RATP
- ▶ **discrete** : dice (6 possibilities), number of cars ( $> 10000$ )

# Probability distributions

- ▶ A random variable is linked to a **probability distribution**.
- ▶ It quantifies the probability of observing one outcome.

## Probability distributions

- ▶ A random variable is linked to a **probability distribution**, which is a function  $P$
- ▶ It quantifies the probability of observing one outcome.
- ▶ For a discrete variable : each possible outcome is associated with a number between 0 and 1



## Probability distributions

- ▶ For a dice game, the possible outcomes are in the set  $\{1, 2, 3, 4, 5, 6\}$
- ▶ For a dice game :  $P(1) = ?$   $P(2) = ?$   $P(3) = ?$   $P(4) = ?$   
 $P(5) = ?$   $P(6) = ?$

## Probability distributions

- ▶ For a dice game, the possible outcomes are in the set  $\{1, 2, 3, 4, 5, 6\}$
- ▶ For a dice game :  $P(1) = \frac{1}{6}$ ,  $P(2) = \frac{1}{6}$ ,  $P(3) = \frac{1}{6}$ ,  $P(4) = \frac{1}{6}$ ,  $P(5) = \frac{1}{6}$ ,  $P(6) = \frac{1}{6}$
- ▶ This is called a **uniform distribution**

## Probability distributions

- ▶ Périphérique : probably a time-dependent very complicated distribution

## Continuous variables

- ▶ The situation is different for continuous random variables.
- ▶ The distribution is given by a **probability density function**. Informally, the probability of being between  $x$  and  $x + dx$  is  $p(x)dx$ .
- ▶ [https://en.wikipedia.org/wiki/Probability\\_density\\_function](https://en.wikipedia.org/wiki/Probability_density_function)
- ▶ Note that some variables are neither discrete nor continuous.

## Uniform discrete

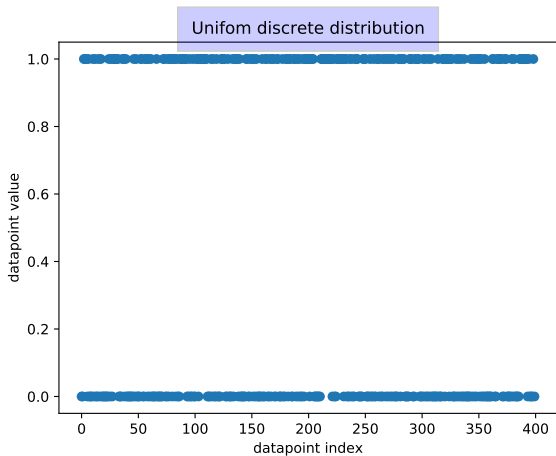


Figure – Uniform discrete distribution with 2 values

## Uniform discrete

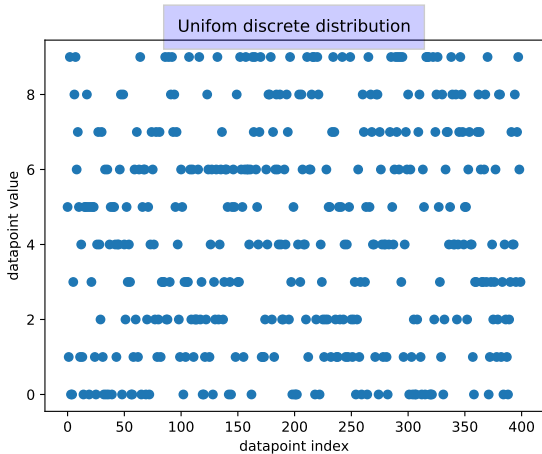


Figure – Uniform discrete distribution with 10 values

# Bernoulli

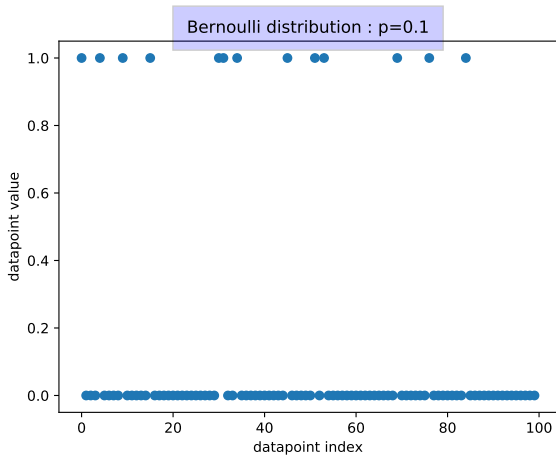


Figure – Bernoulli distribution

## Bernoulli $p$

- ▶ With probability  $p$ ,  $X = 1$
- ▶ With probability  $1 - p$ ,  $X = 0$



# Bernoulli

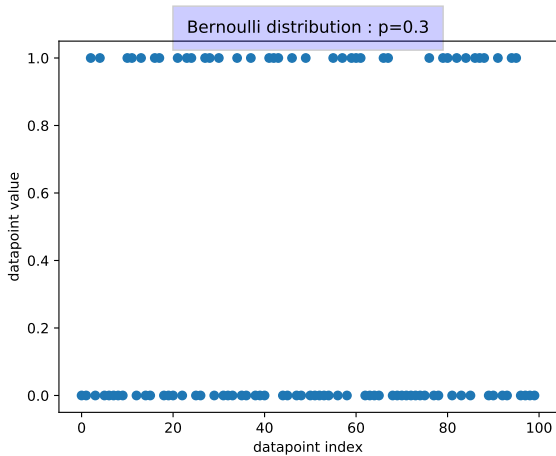


Figure – Bernoulli Distribution

# Bernoulli

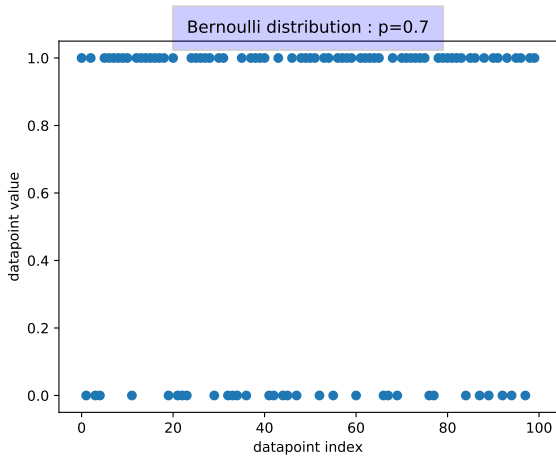


Figure – Bernoulli Distribution

## Uniform continuous

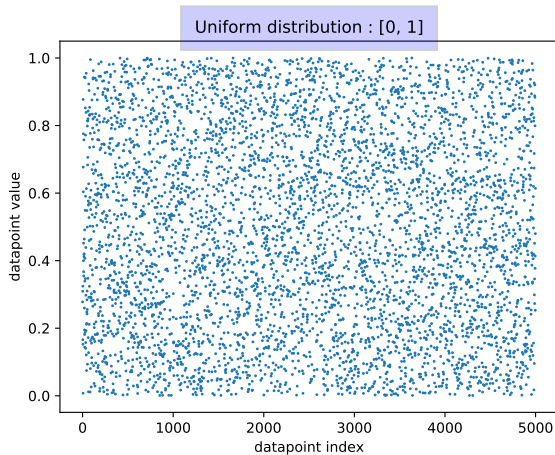


Figure – Uniform continuous distribution

## Uniform continuous

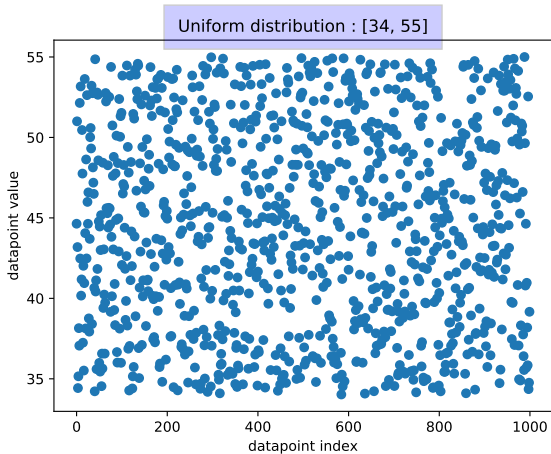


Figure – Uniform continuous distribution

## Uniform continuous

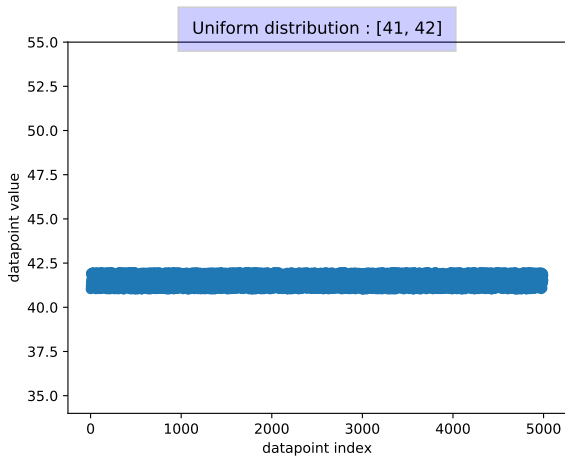


Figure – Uniform continuous distribution

# Normal

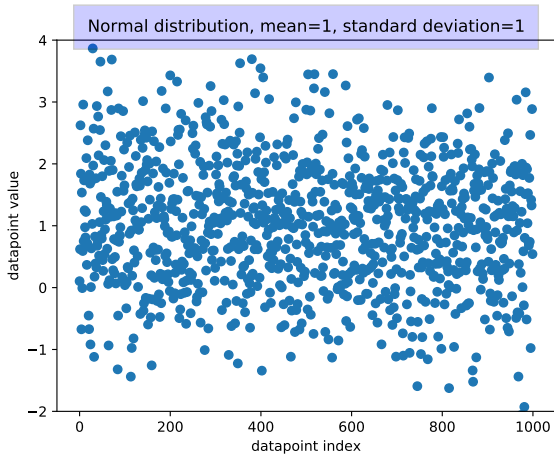


Figure – Normal distribution

# Normal

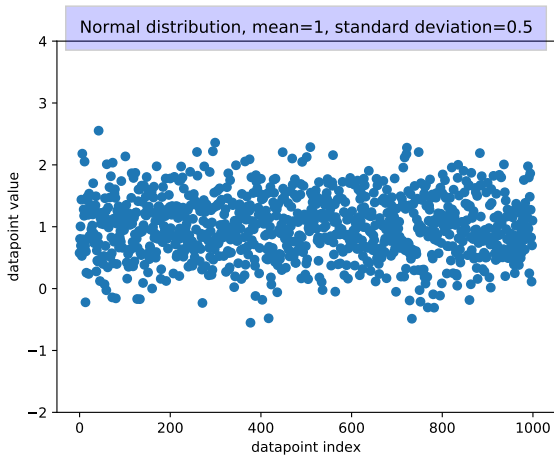


Figure – Normal distribution

# Normal

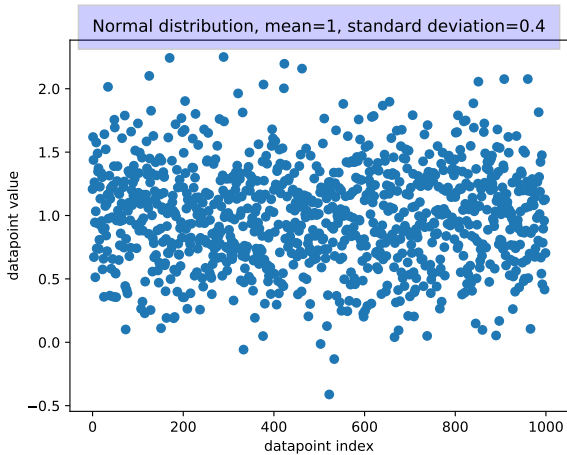


Figure – Normal distribution



## White noise

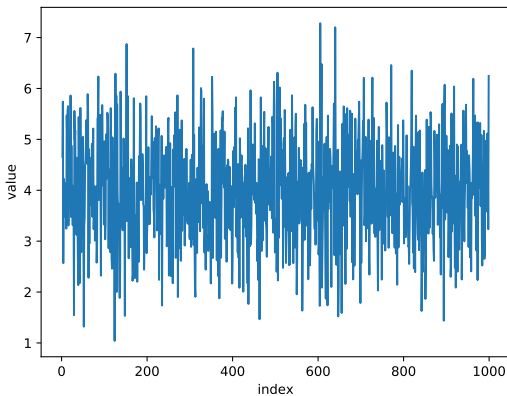


Figure – White noise

# Histograms

**Histograms** are an alternative representation of the results of a (one-dimensional) random variable.

## Uniform discrete

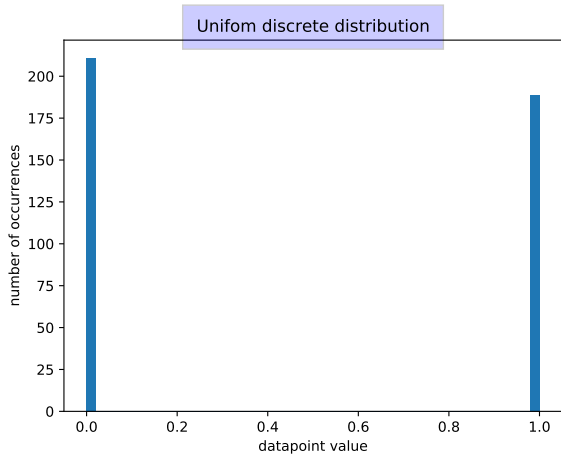


Figure – Histogram 1

## Uniform discrete

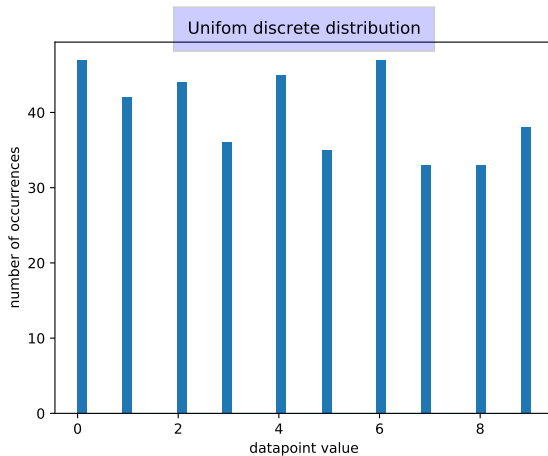


Figure – Histogram 1

# Bernoulli

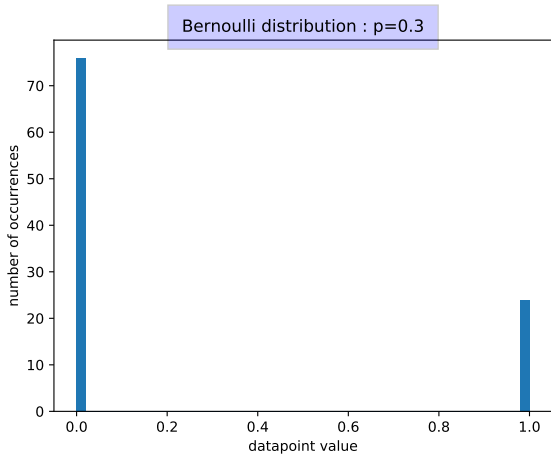


Figure – Histogram 2

## Uniform continuous

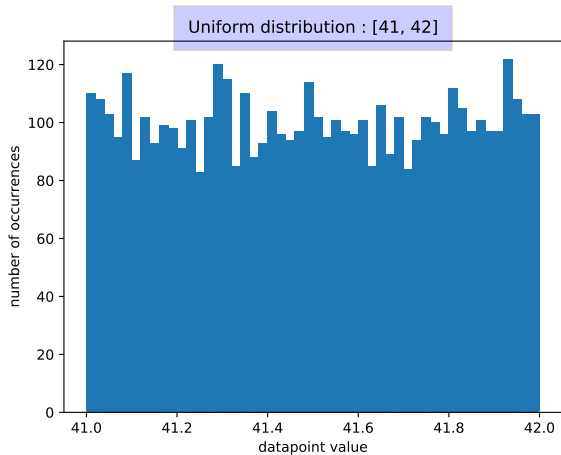


Figure – Histogram 3

# Normal

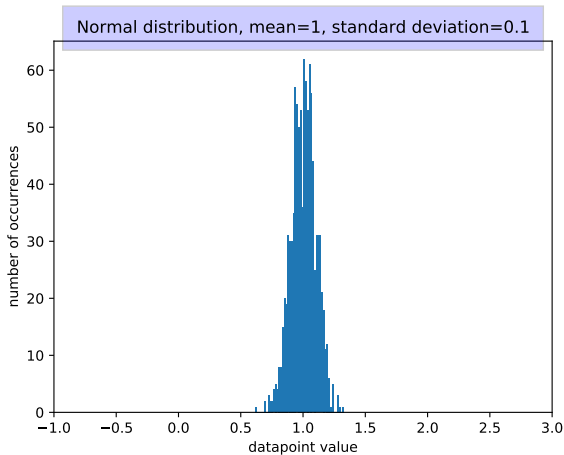


Figure – Histogram 4

# Normal

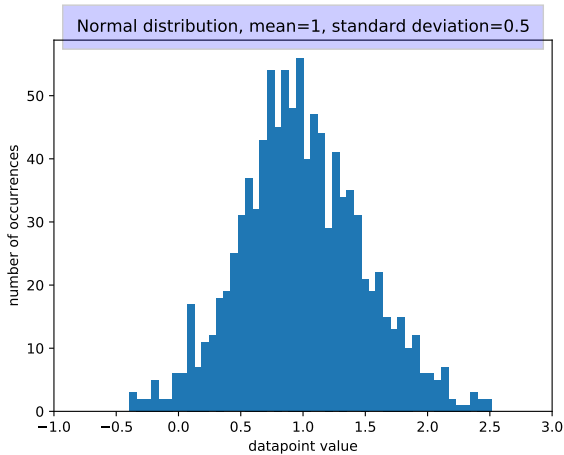


Figure – Histogram 4



**cd distributions/**

We can use the files **analyze\_distribution\_1.py** and **analyze\_distribution\_2.py** to analyze and plot some simple datasets, stored in **csv\_files/**

## Distribution 1

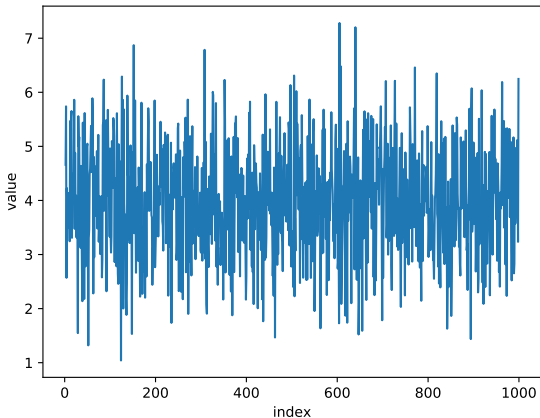


Figure – The data we analyze

# histograms

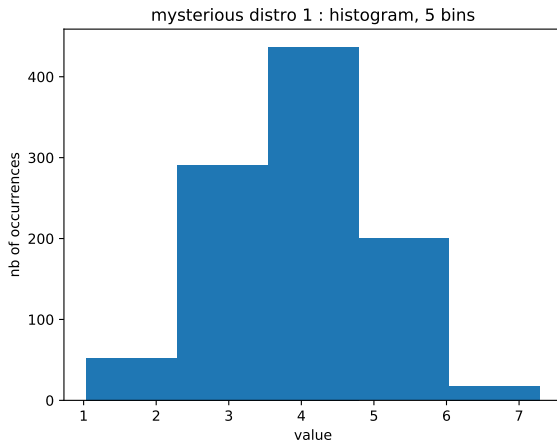


Figure – 5 bins

# histograms

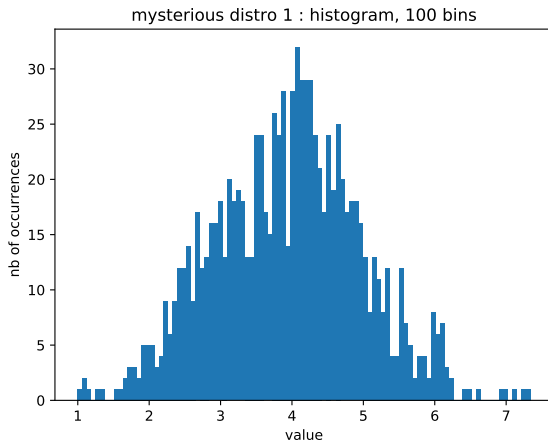


Figure – 100 bins

# histograms

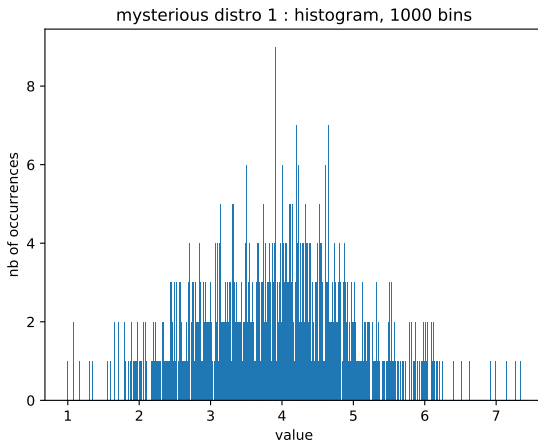


Figure – 1000 bins (too many)

## Normal distribution

```
import csv
import numpy as np

file_name = 'mysterious_distro_1.csv'

mean = 4
std_dev = 1
nb_point = 1000

with open('csv_files/' + file_name, 'w') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=',')
    for point in range(1, nb_point):
        random_variable = np.random.normal(loc=mean, scale=std_dev)
        filewriter.writerow([str(point), str(random_variable)])
```

Figure – `create_normal.py` : Creation of the distribution

## Second example

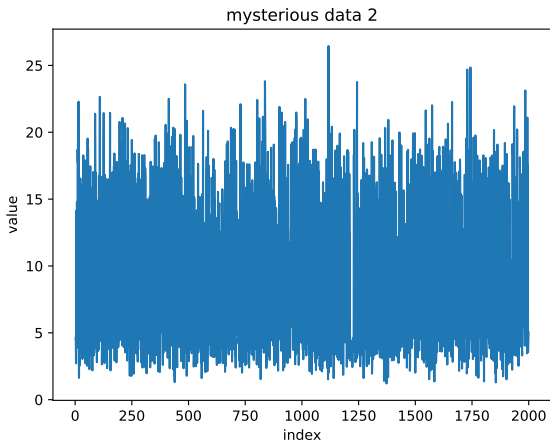


Figure – Second distribution

# Fitting

In most cases, it won't be that straightforward to fit a distribution :

- ▶ the random variable may be multidimensional
- ▶ need to choose a family of distributions (parametric vs non-parametric)
- ▶ an optimization might be needed in order to find good parameters.



## Multidimensional vectors

We often consider random variables and data that live in spaces with a higher dimension than 2 (random vectors).

- ▶ images
- ▶ sensor that receives **multimodal information**

## Correlation

Random vectors with correlated components are common statistical objects.

- ▶ In physics, temperature and pressure, measured by some sensors are correlated.
- ▶ In a dataset of customers of a company, some dimensions are likely to be correlated.

To study the statistical relationship between components, we can compute the **covariance** of the two components, or the **correlation**, (normalized covariance (see below)).

<https://en.wikipedia.org/wiki/Correlation>

## Expected value (espérance)

- ▶ For a discrete random variable  $X$  that takes the values  $x_i$  with probability  $p_i$  :

$$E(X) = \sum_{i=1}^n p_i x_i \quad (1)$$

- ▶ For a continuous random variable  $X$  with density  $p$  :

$$E(X) = \int x p(x) dx \quad (2)$$

Note that  $X$  may have values in  $\mathbb{R}^d$ , with  $d \geq 1$ .

## Expected value (espérance)

### Exercice 1 : Computing an expected value

- For a discrete random variable  $X$  that takes the values  $x_i$  with probability  $p_i$  :

$$E(X) = \sum_{i=1}^n p_i x_i \quad (3)$$

- For a continuous random variable  $X$  with density :

$$E(X) = \int x p(x) dx \quad (4)$$

Compute the expected value of the dice game.

## Variance

The variance is a measure of the dispersion of a random real variable.

<https://en.wikipedia.org/wiki/Variance>

$$\text{var}(X) = E\left((X - E(X))^2\right) \quad (5)$$

Note that we can also define the variance of a multidimensional random variable (which means a random vector). In that case, it is a matrix.

# Covariance

The covariance is a measure of the relationship between the variations of two random variables.

$$\text{cov}(X, Y) = E\left((X - E(X))(Y - E(Y))\right) \quad (6)$$

## Correlation

The correlation is the covariance divided by the square roots of the variances.

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{Var}(X)}\sqrt{\text{Var}(Y)}} \quad (7)$$

## Example

The data in `csv_files/distribution_3.csv` contain samples of a random variable with 5 dimensions (random vector). Some of these dimensions are correlated.



# Covariance

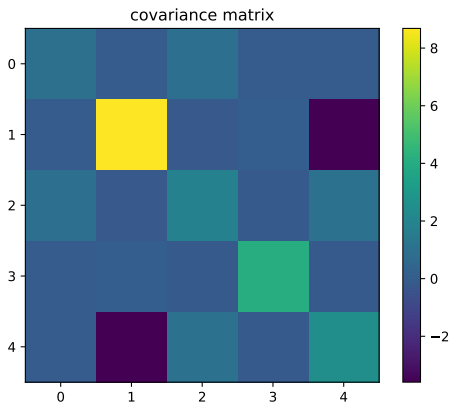


Figure – Covariance matrix of the random vector.

## Correlation matrix

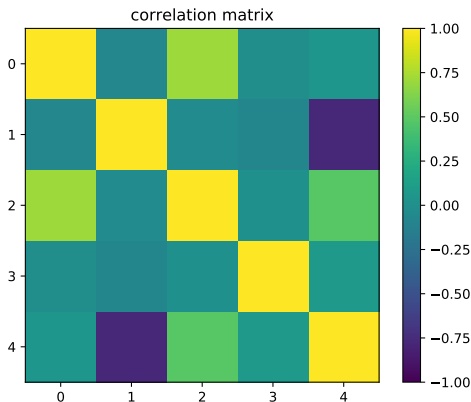


Figure – Correlation matrix for the distribution, note the difference in the scale.

## Generation of the data

```
mean_1 = 4
std_dev_1 = 1

mean_2 = 15
std_dev_2 = 3

mean_3 = -5
std_dev_3 = 2

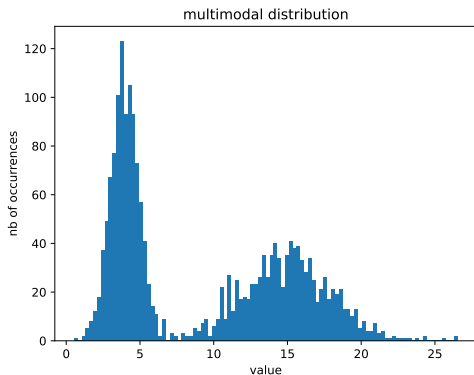
mean_noise = 0
noise_std_dev = 1

nb_point = 1000

with open('csv_files/' + file_name, 'w') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=',')
    for point in range(1, nb_point):
        noise = np.random.normal(loc=mean_noise, scale=noise_std_dev)
        random_variable_1 = np.random.normal(loc=mean_1, scale=std_dev_1)
        random_variable_2 = np.random.normal(loc=mean_2, scale=std_dev_2)
        random_variable_3 = random_variable_1 + noise
        random_variable_4 = np.random.normal(loc=mean_3, scale=std_dev_3)
        random_variable_5 = -0.4 * random_variable_2 + noise
        filewriter.writerow([str(point),
                             str(random_variable_1),
                             str(random_variable_2),
                             str(random_variable_3),
                             str(random_variable_4),
                             str(random_variable_5)])
```

Figure – Multidimensional random variable

**Exercise 2:** Create a one-dimensional dataset with a histogram that looks like this one!



**Figure** – This distribution has several **modes**

## Minimization of a function

Optimization is another core aspect of machine learning.

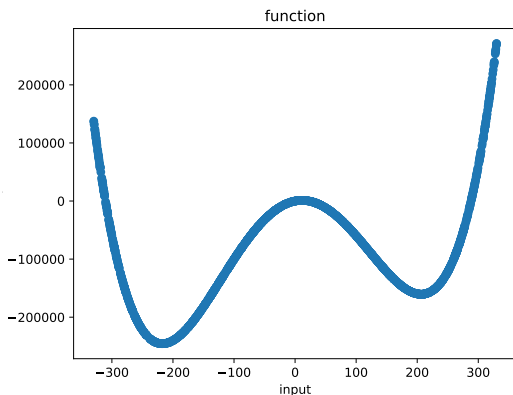


Figure – Loss function

## Optimization in machine learning

The loss function typically represents the quality of a set of parameters to solve a problem.

- ▶ in supervised learning, typically a measure of the prediction error on the dataset
- ▶ in clustering, typically a distortion
- ▶ in density estimation, a likelihood

## Analytic minimization

What is the minimum of the function

$$f : x \rightarrow (x - 1)^2 + 3.5 \quad (8)$$

And for what value  $x$  is it obtained ?

## Iterative algorithms

However, in most applications of machine learning, it is not possible to use an analytical solution, either because :

- ▶ we do not know the analytical solution
- ▶ we know how to compute it, but the computation is too costly for practical use.

Instead, we use **iterative algorithms** (gradient descent, coordinate descent, etc.)



## Gradient algorithms

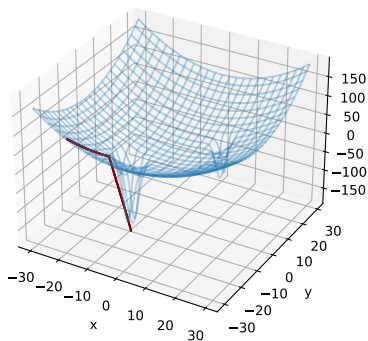


Figure – Optimization trajectory.