Reminders on probabilities

July 2, 2019

Overview

- This afternoon we will discuss two topics.

Overview

- This afternoon we will discuss two topics.
- **Probabilities and statistics :** they are at the core of modern Machine Learning, so it is nice to have some intruition on them.

Overview

- This afternoon we will discuss two topics.
- **Probabilities and statistics :** they are at the core of modern Machine Learning, so it is nice to have some intruition on them.
- **Decision trees :** they will be our first Machine Learning tool.

Random variables

- A **random variable** is a quantity that can take several values

Random variables

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

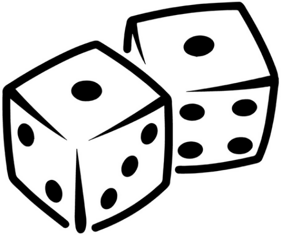


Figure: Dice

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

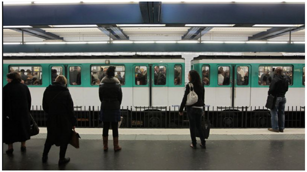


Figure: Some metro station

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 periphrique at the same time

Random variables

What are the differences between these random variables ?

Random variables

What are the differences between these random variables ?

- Some are **continuous**, others **discrete**
- **continuous :**

Random variables

What are the differences between these random variables ?

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

Random variables

What are the differences between these 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**.

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

- Periphrique :

Probability distributions

- Periphrique : probably a time-dependent very complicated distribution

Continuous variables

- How would you model a continuous variable ? Can you assign a number to a waiting time or a weather ?

Continuous variables

- How would you model a continuous variable ? Can you assign a number to a waiting time or a weather ?
- One needs to use **probability densities**. Formally, the probably of being between x and $x + dx$ is $p(x)dx$.

Continuous variables

- How would you model a continuous variable ? Can you assign a number to a waiting time or a weather ?
- One needs to use **probability densities**. Formally, the probably of being between x and $x + dx$ is $p(x)dx$.

Elementary rules on probabilities

- to be valid, a probability distribution must abide by some rules.

Elementary rules on probabilities

- to be valid, a probability distribution P must abide by some rules.
- $P(\text{"any outcome happens"}) = 1$

Elementary rules on probabilities

- to be valid, a probability distribution P must abide by some rules.
- $P(\text{"any outcome happens"}) = 1$
- if A and B are two **incompatible** possible outcomes, then

Elementary rules on probabilities

- to be valid, a probability distribution P must abide by some rules.
- $P(\text{"any outcome happens"}) = 1$
- if A and B are two **incompatible** possible outcomes, then

$$P(A \cup B) = P(A) + P(B) \tag{1}$$

Elementary rules on probabilities

- to be valid, a probability distribution P must abide by some rules.
- $P(\text{"any outcome happens"}) = 1$
- if A and B are two **incompatible** possible outcomes, then

$$P(A \cup B) = P(A) + P(B) \tag{2}$$

- The probability of "A or B" is the same as the **sum** of the probabilities $P(A)$ and $P(B)$.

Example

- If we throw a dice, are the outcomes 1 and 2 compatible ?

Example

- If we throw a dice, are the outcomes 1 and 2 compatible ?
- They are **not** , so we have :

$$P(1 \cup 2) = P(1) + P(2) \tag{3}$$

Consequence :

- We note $\bar{A}$ the **complementary** event of A.

Consequence :

- We note $\bar{A}$ the **complementary** event of A.
- A and $\bar{A}$ are incompatible.

Consequence :

- We note $\bar{A}$ the **complementary** event of A.
- A and $\bar{A}$ are incompatible.
- So $P(A \cup \bar{A}) = P(A) + P(\bar{A})$

Consequence :

- We note $\bar{A}$ the **complementary** event of A.
- A and $\bar{A}$ are incompatible.
- So $P(A \cup \bar{A}) = P(A) + P(\bar{A})$
- But $P(A \cup \bar{A}) = 1$

Consequence :

- We note $\bar{A}$ the **complementary** event of A.
- so we have :

$$P(\bar{A}) = 1 - P(A) \tag{4}$$

Consequence :

- We note $\bar{A}$ the **complementary** event of A.
- Taking again the exemple of the dice throw, for instance, what is the complementary event of the event

$$A = \text{"outcome is 1 or 3"} \tag{5}$$

Example :

- We note $\bar{A}$ the **complementary** event of A.
- Taking again the exemple of the dice throw, for instance, what is the complementary event of the event

$$A = \text{"outcome is 1 or 3"} \tag{6}$$

Example :

- We note $\bar{A}$ the **complementary** event of A.
- Taking again the exemple of the dice throw, for instance, what is the complementary event of the event

$$A = \text{"outcome is 1 or 3"} \tag{7}$$

- $$\bar{A} = \text{"outcome is 2 or 4 or 5 or 6"} \tag{8}$$

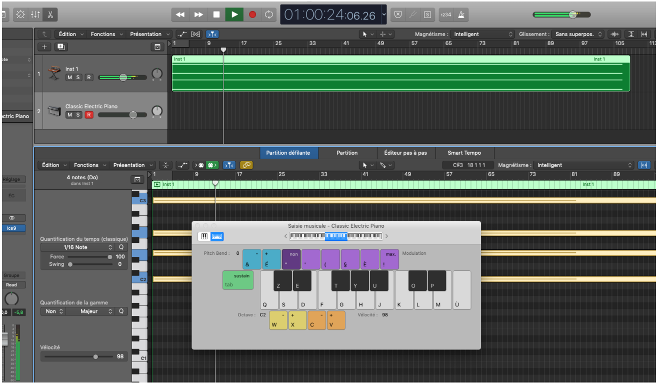
Example :

- $$A = \text{"outcome is 1 or 3"} \tag{9}$$
- $$\bar{A} = \text{"outcome is 2 or 4 or 5 or 6"} \tag{10}$$
- $$P(A) = ? \tag{11}$$
- $$P(\bar{A}) = ? \tag{12}$$

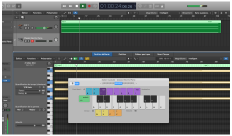
Example :

- $$A = \text{"outcome is 1 or 3"} \tag{13}$$
- $$\bar{A} = \text{"outcome is 2 or 4 or 5 or 6"} \tag{14}$$
- $$P(A) = \frac{1}{3} \tag{15}$$
- $$P(\bar{A}) = \frac{2}{3} \tag{16}$$

Exercise 1 : probabilities and tones

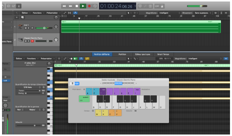


Exercise 1 : probabilities and tones

- 

- A major scale has **7** tones.
- A keyboard contains **12** tones.

Exercise 1 : probabilities and tones

- 
- A major scale has **7** tones.
- A keyboard contains **12** tones.
- If I play random notes on the keyboard, after how many notes do I have 9 chances out of 10 to play a note that is out of the major scale ? (in that case C Major).

Solution Exercise 1

- Now that we have written the formula to find the solution, let us find it with python !

Exercise 2 : probabilities and rhythm

- Exercise with eigths.

Examples

- Let us now illustrate and discuss important and famous distributions.

Law 1

- Example : I chose a random number between 0 and 10 with equal probability.

Uniform discrete

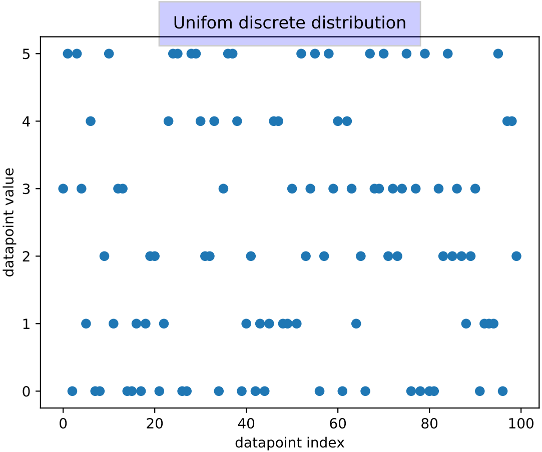


Figure: Uniform discrete distribution

Bernoulli

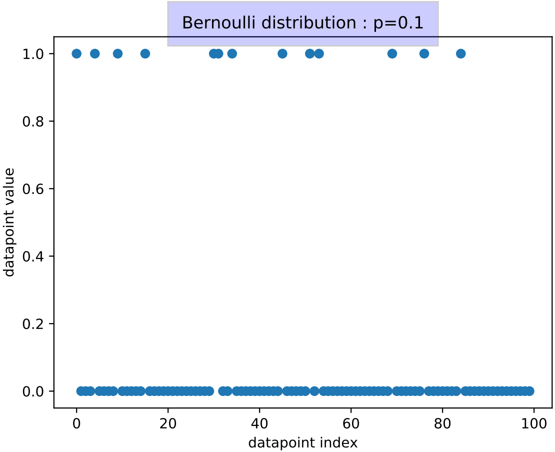


Figure: Bernoulli distribution

Bernoulli

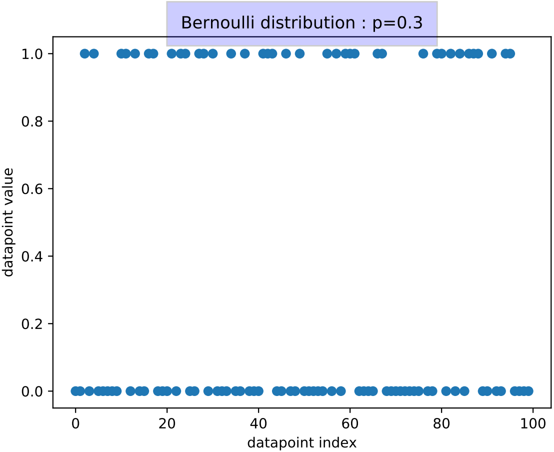


Figure: Bernoulli Distribution

Bernoulli

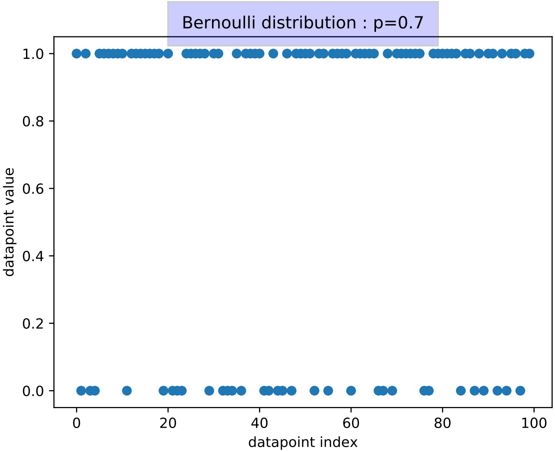


Figure: Bernoulli Distribution

Uniform continuous

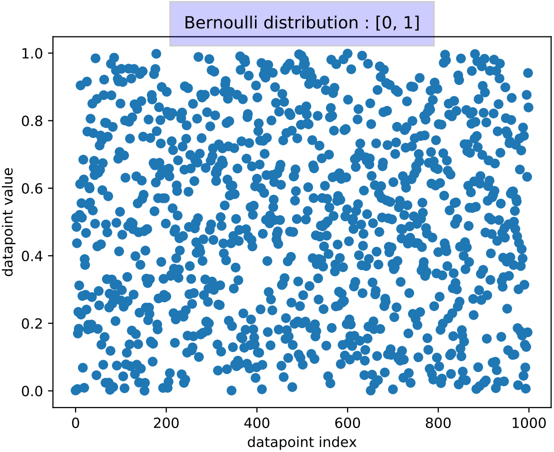


Figure: Uniform continuous distribution

Uniform continuous

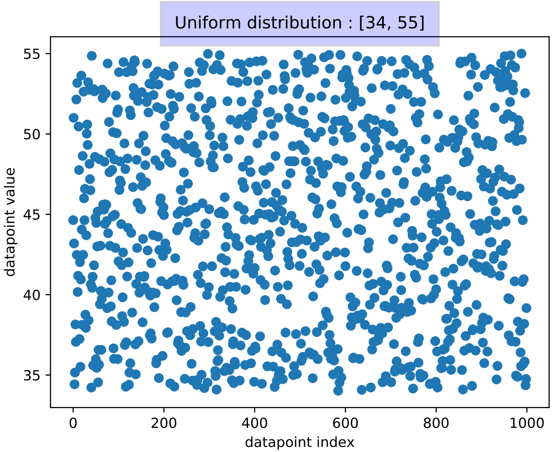


Figure: Uniform continuous distribution

Normal

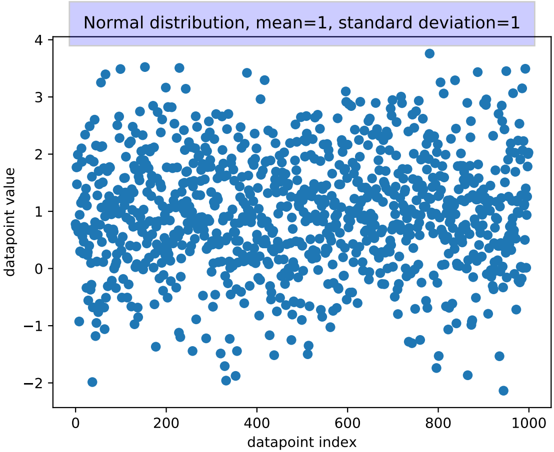


Figure: Normal distribution

Normal

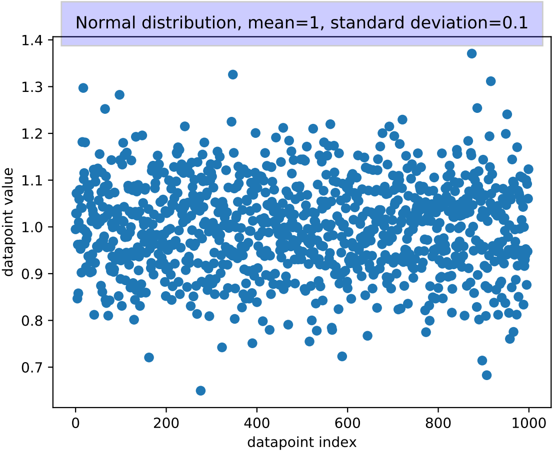


Figure: Normal distribution

Normal

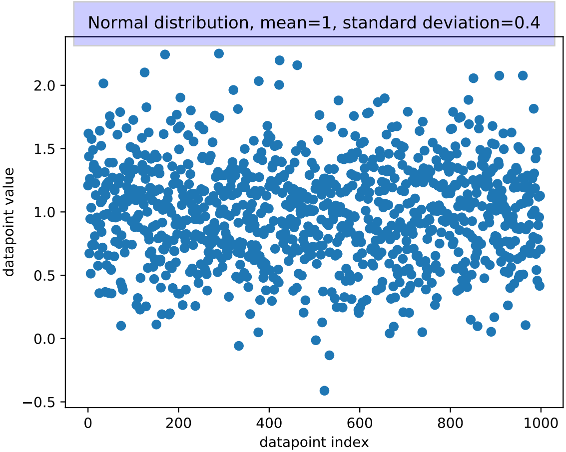


Figure: Normal distribution

White noise

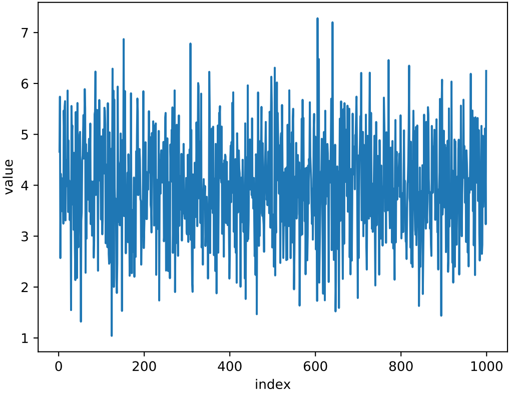


Figure: White noise

Histograms

Is looking at the raw dataset really **informative ?**

Histograms

Is looking at the raw dataset really **informative ?**
It is informative, but often a **histogram** tells more.

Uniform discrete

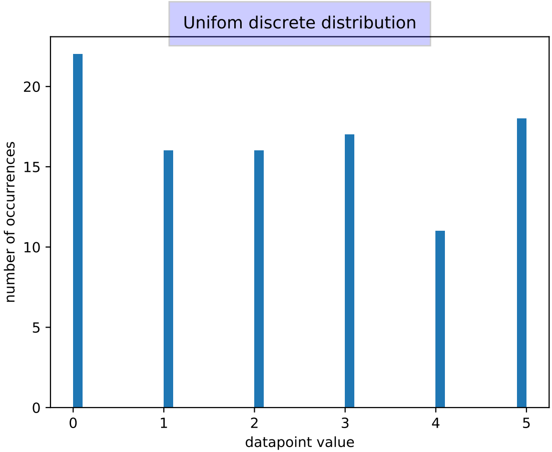


Figure: Historgram 1

Bernoulli

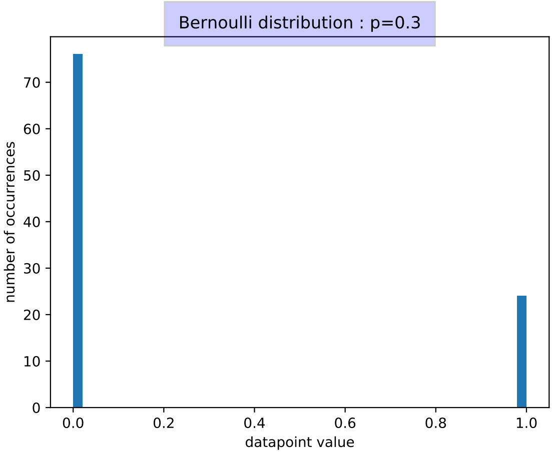


Figure: Historgram 2

Uniform continuous

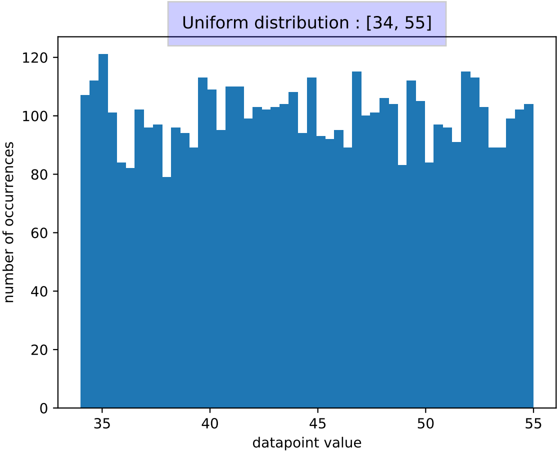


Figure: Historgram 3

Normal

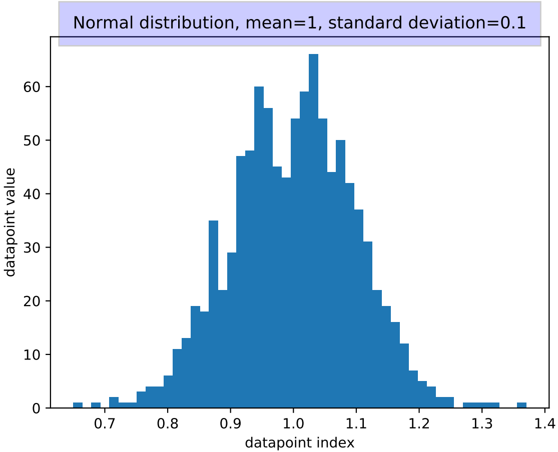


Figure: Historgram 4

Exercise

I put values in the file **mysterious_distro_1.csv**

Exercise

I put values in the file **mysterious_distro_1.csv**
Can you analyze these values in terms of a **distribution ?**
Use **read_myst_1** to analyze the distribution (suggestion : change the number of bins used)

Exercise

When you have guessed the kind of distribution it is, you need to finds its **parameters**.

- its mean
- its standard deviation

This is called **fitting** a distribution to a dataset : it's a classical machine learning problem.
To do so, uncomment the last section of the script **read_myst_1**

Distribution 1

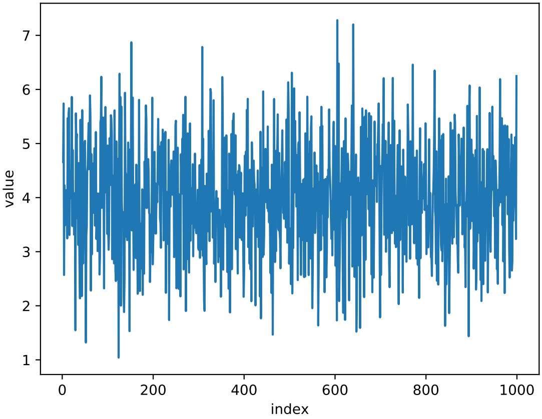


Figure: The data we analyze

histograms

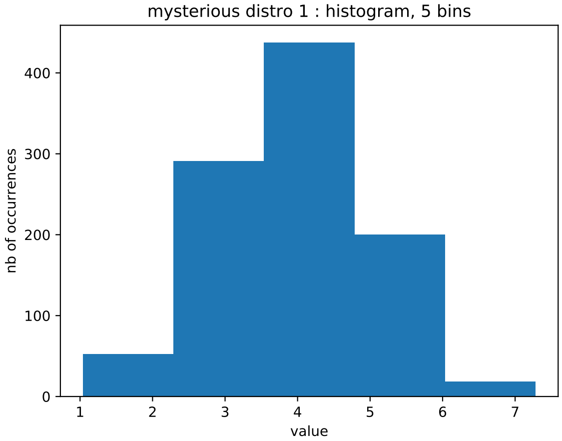


Figure: 5 bins

histograms

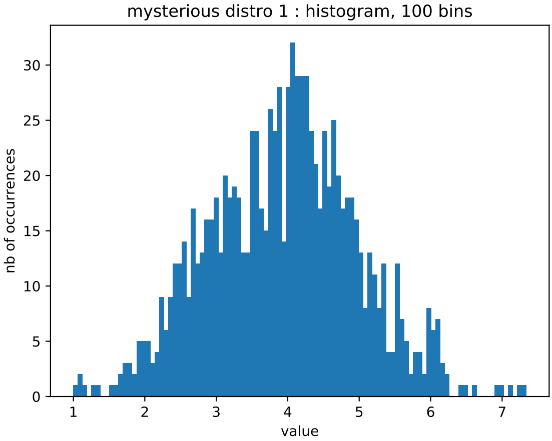


Figure: 100 bins

histograms

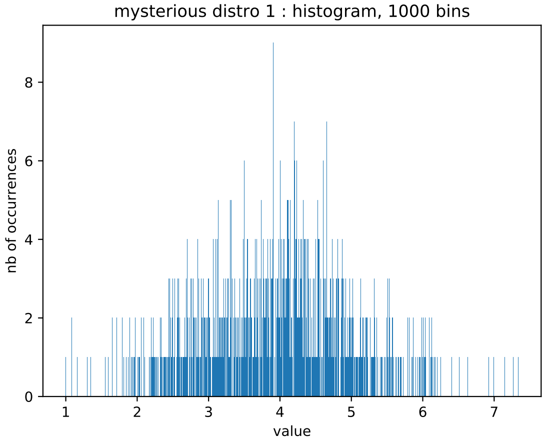


Figure: 1000 bins (too many)

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

Let's try to perform the same analysis on the file **mysterious_distro_2.csv** using **read_myst_2**.

Second example

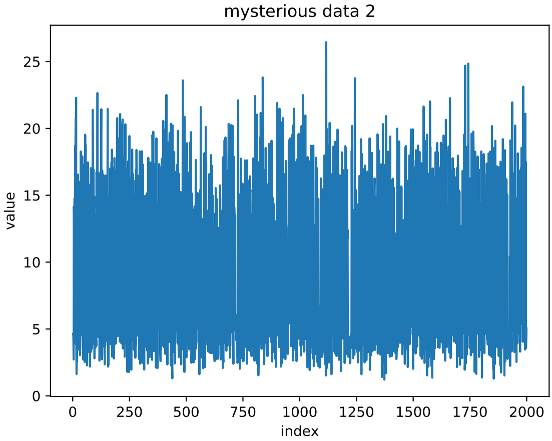


Figure: Second distribution

Multimodal distribution

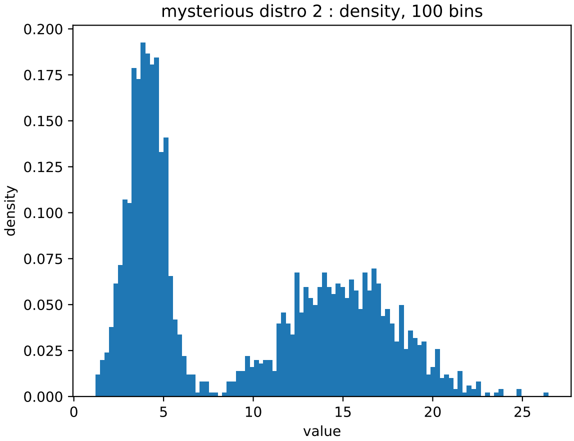


Figure: This distribution has several **modes**

Multimodal distribution

```
mean_1 = 4
std_dev_1 = 1
nb_point_1 = 1000

mean_2 = 15
std_dev_2 = 3
nb_point_2 = 1000

nb_point = nb_point_1 + nb_point_2

with open('csv_files/' + file_name, 'w') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=',')
    for point in range(1, nb_point):
        if random.randint(1, 2) == 1:
            random_variable = np.random.normal(loc=mean_1, scale=std_dev_1)
            filewriter.writerow([str(point), str(random_variable)])
        else:
            random_variable = np.random.normal(loc=mean_2, scale=std_dev_2)
            filewriter.writerow([str(point), str(random_variable)])
```

Figure: **create_bimodal.py** : Generation of multimodal distribution

Fitting

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

Fitting

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

- what distribution do we want to use ?
- even if we know the right shape of the distribution, how to choose the parameters ?

Maximum Likelihood

The **Maximum Likelihood** method is one example method used in Machine Learning.
Say you have a dataset $(x_1, ..., x_n)$.

Maximum Likelihood

The **Maximum Likelihood** method is one example method used in Machine Learning.
Say you have a dataset $(x_1, ..., x_n)$.
You first need to choose a **model** (which is the distribution) of your dataset, p.

Maximum Likelihood

The **Maximum Likelihood** method is the one used in Machine Learning.
Say you have a dataset $(x_1, ..., x_n)$.
You first need to choose a **model** (which is the distribution) of your dataset, p.
Then, you must optimize the **parameters of this model**, noted θ.

Maximum Likelihood

The Likelihood of your model is

$$L(\theta) = \prod_{i=1}^{n} p(x_i|\theta) \tag{17}$$

Maximum Likelihood

The Likelihood of your model is

$$L(\theta) = \prod_{i=1}^{n} p(x_i|\theta) \tag{18}$$

This is the function that you want to **maximise**.

Maximum Likelihood

Most of the time it's written this way : "minimise $-logL(\theta)$"
Why ?

Maximum Likelihood

Most of the time it's written this way : "minimise $-logL(\theta)$"
Because the log **transforms the product into a sum**, which is easier to **derivate**.

Maximum Likelihood

$$-logL(\theta) = -\sum_{i=1}^{n} \log(p(x_i|\theta)) \tag{19}$$

Max Likelihood

So how can we minimise $-logL(\theta)$? In the case of very large datasets, and large numbers of parameters (tens, hundredths, more), most of the time an **analytic solution** is not available. So people use **gradient descent**.

The gradient descent

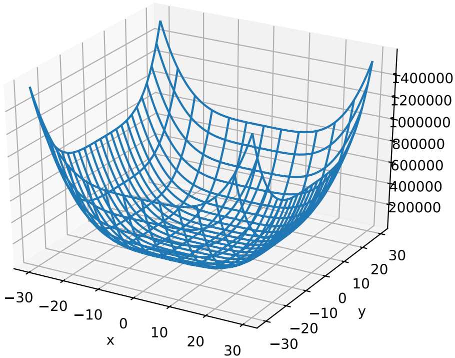
We want x to **minimise** f. We perform, until some criteria is satisfied :

$$x \leftarrow x - \alpha \nabla_f(x) \tag{20}$$

Use the file "gradient_algo.py" and implement the gradient algorithm on a simple example.
I inserted two errors in the code.

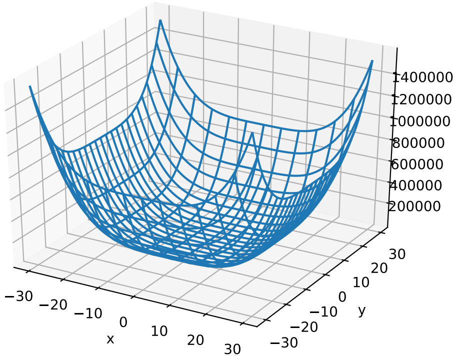
The gradient descent

$$x \leftarrow x - \alpha \nabla_f(x) \tag{21}$$



The gradient descent

Experiment with it, try to change all the parameters and to break it again. Is it stable ?



Multidimensional vectors

We can consider data that live in higher dimensional spaces than 2.

Multidimensional vectors

We can consider data that live in higher dimensional spaces than 2. Examples ?

Multidimensional vectors

We can consider data that live in higher dimensional spaces than 2. Examples ?

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

Correlation

Sometimes the components of a multidimensonial vector $(x_1, ..., x_n)$ are not independent.

Correlation

Sometimes the components of a multidimensonial vector $(x_1, ..., x_n)$ are not independent.
To study this, we can use the **covariance** of the two components, or the **correlation** which is actually clearer.

Correlation, expected value

- Let us introduce these two important quantities (backboard).

Example

Look at the data contained in **mysterious_distro_3.csv**
They contain a random variable with 5 dimensions. Some of these dimensions are correlated.
Think for instance to physics : temperature and pressure, etc. If you have measurements of temperature and pressure, the two would probably be **correlated**.

Correlation matrix

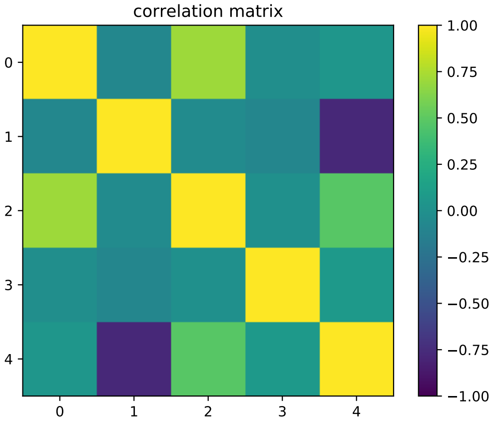


Figure: Correlation matrix for the distribution

Generation of the data

```python
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