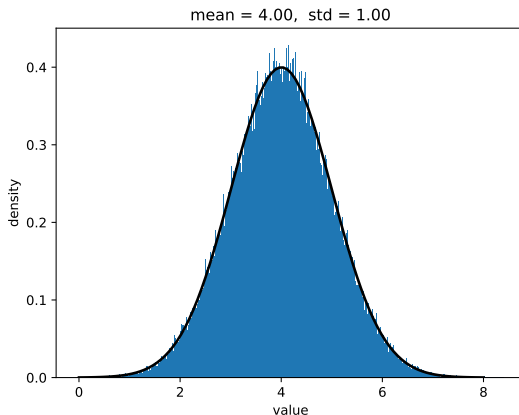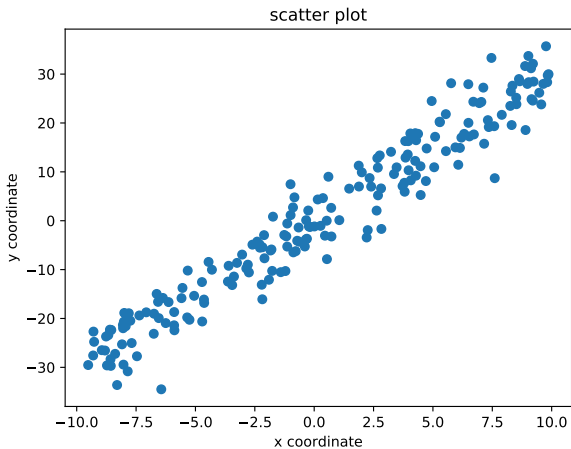# Machine learning I, supervised learning: reminders on probabilities and statistics
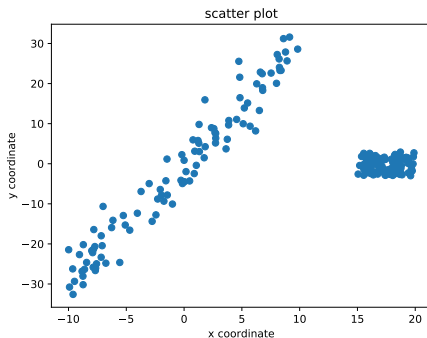
To have a solid understanding of machine learning, it is necessary to be familiar with elementary probabilities and statistics.

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

Random variables

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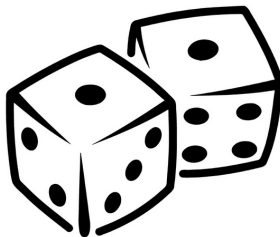▶ For instance :
  ▶ the result of a dice throw

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 periphérique at the same time

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

- ▶ Number of cars on the periphérique : 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 probably of being between x and $x + dx$ is $p(x)dx$.
- ▶ https://en.wikipedia.org/wiki/Probability_density_function
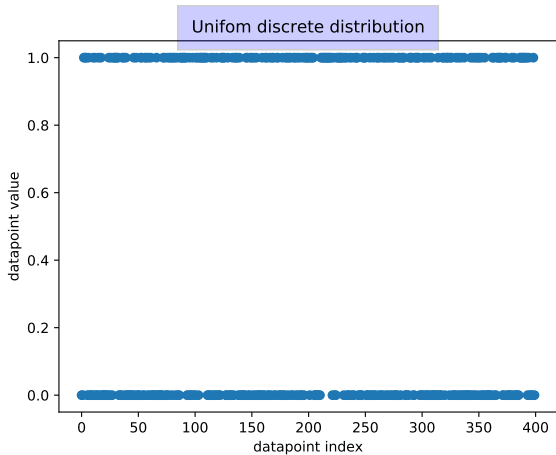- ▶ Not that some variables are neither discrete nor continuous.

Uniform discrete

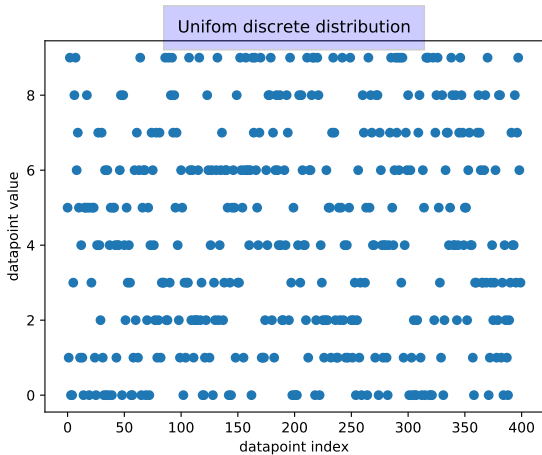Figure – Uniform discrete distribution with 2 values

Uniform discrete

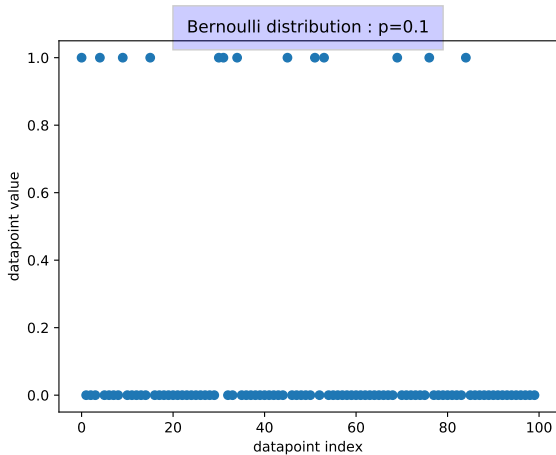Figure – Uniform discrete distribution with 10 values

Bernoulli

Figure – Bernoulli distribution

Bernoulli p

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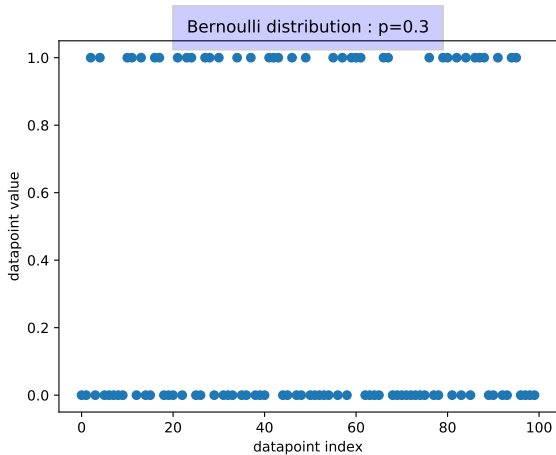- ▶ With probability $1 - p$, $X = 0$

Bernoulli
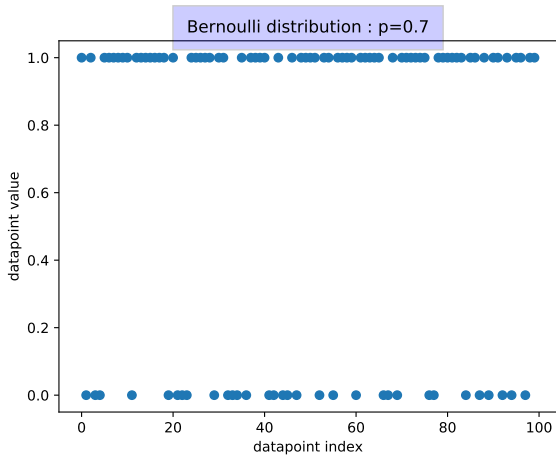

Figure – Bernoulli Distribution

Bernoulli

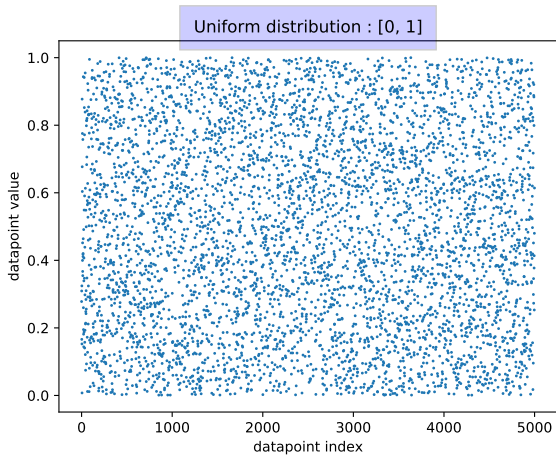Figure – Bernoulli Distribution

Uniform continuous

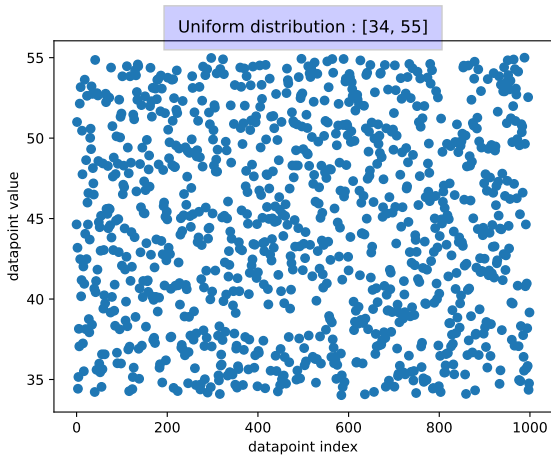Figure – Uniform continuous distribution

Uniform continuous

Figure – Uniform continuous distribution
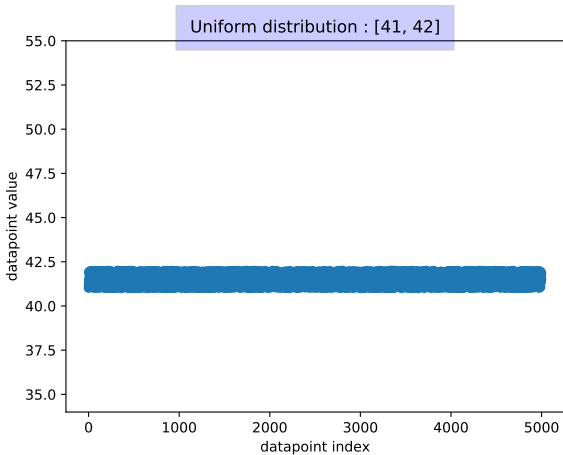
Uniform continuous

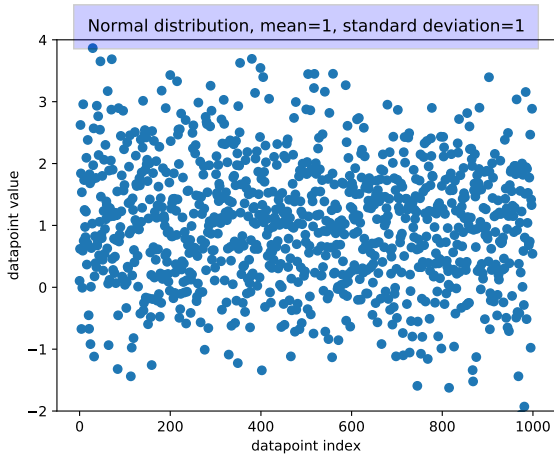Figure – Uniform continuous distribution
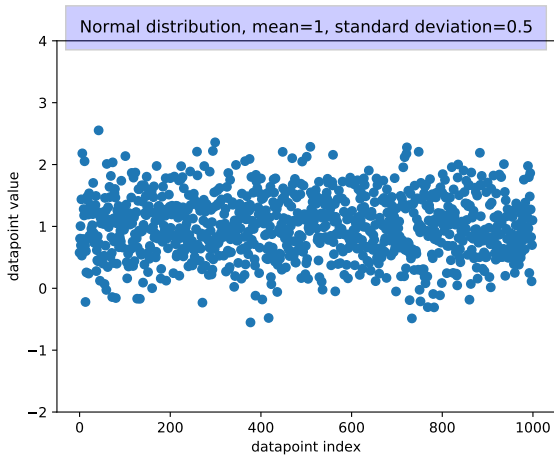
Normal

Figure – Normal distribution

Normal

Figure – Normal distribution

Normal

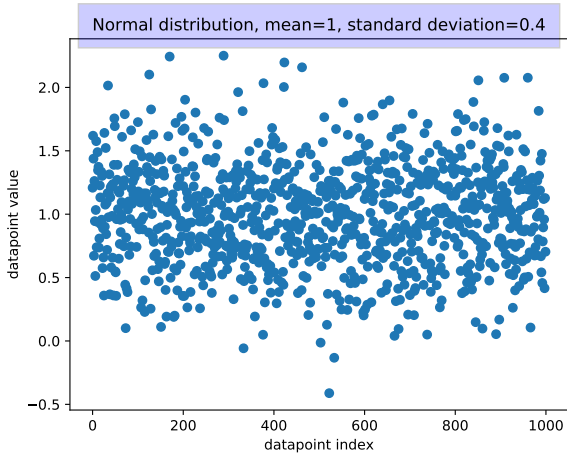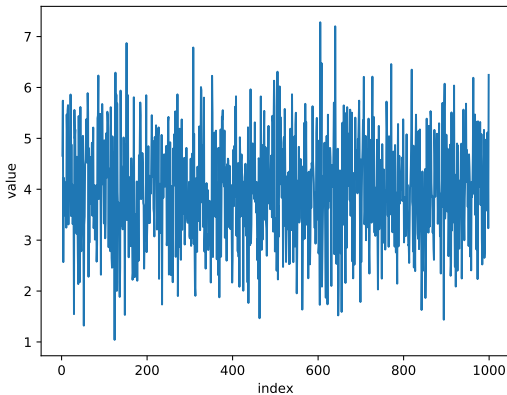
Figure – Normal distribution

White noise

Figure – White noise

Histograms

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

Uniform discrete
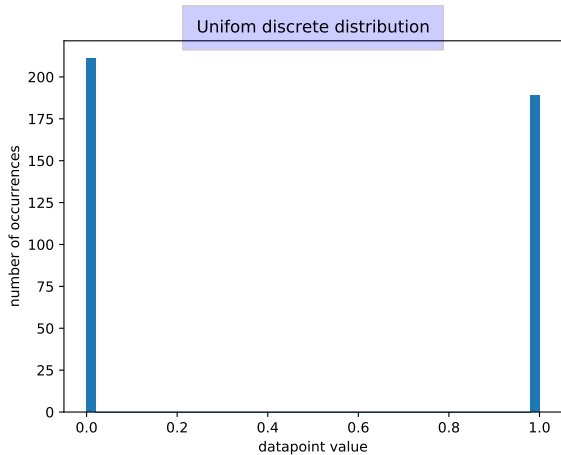

Figure – Histogram 1

Uniform discrete
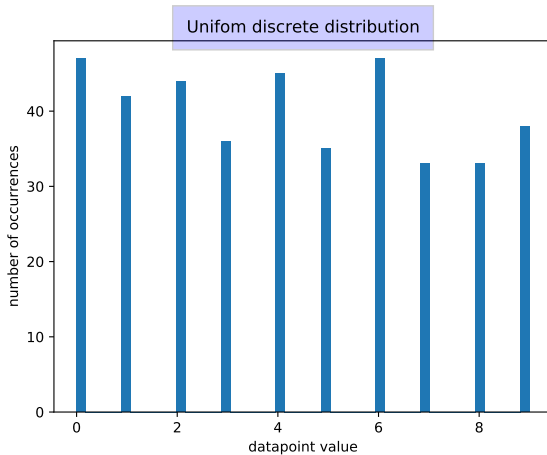
Figure – Histogram 1

Bernoulli
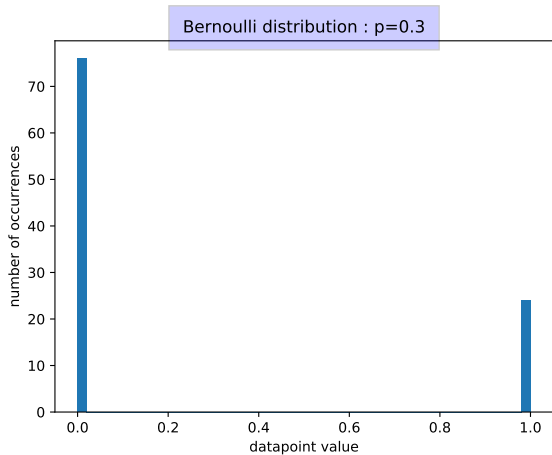
Uniform continuous
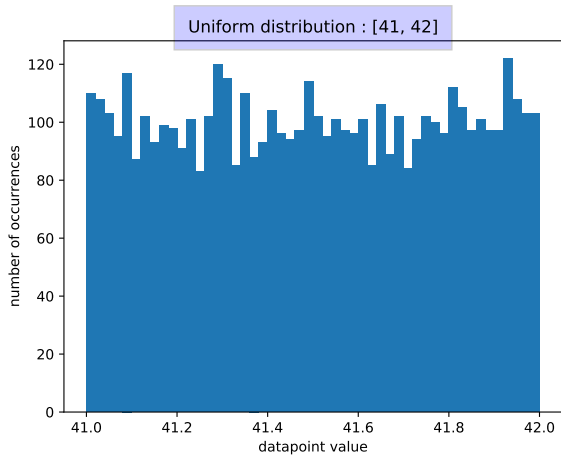
Figure – Histogram 3

Normal

Normal distribution, mean=1, standard deviation=0.1
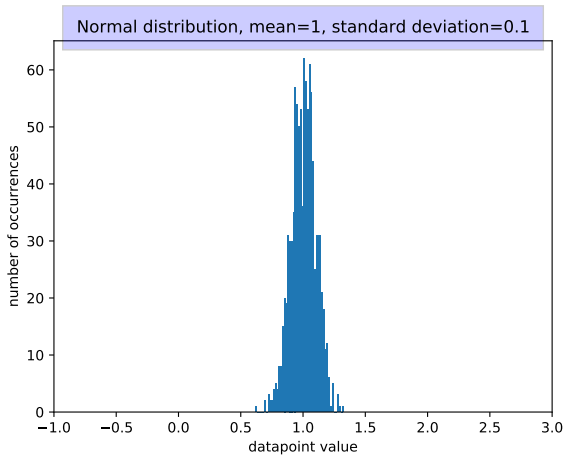
Figure – Histogram 4

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
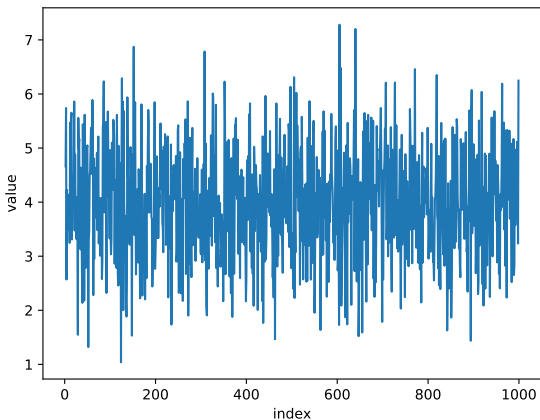
Figure – The data we analyze

histograms
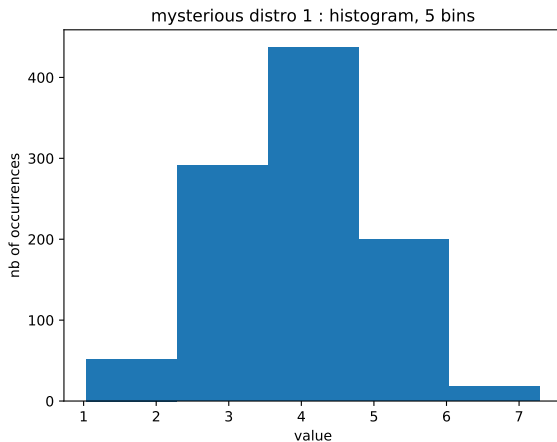
Figure – 5 bins

histograms
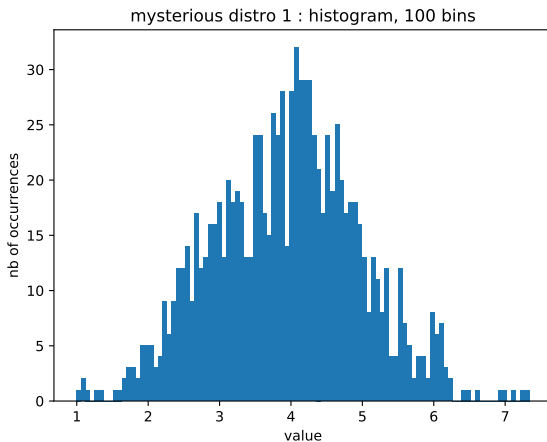
Figure – 100 bins

histograms

mysterious distro 1 : histogram, 1000 bins

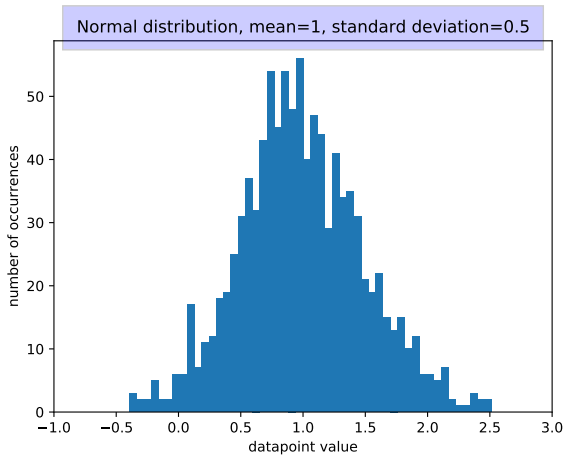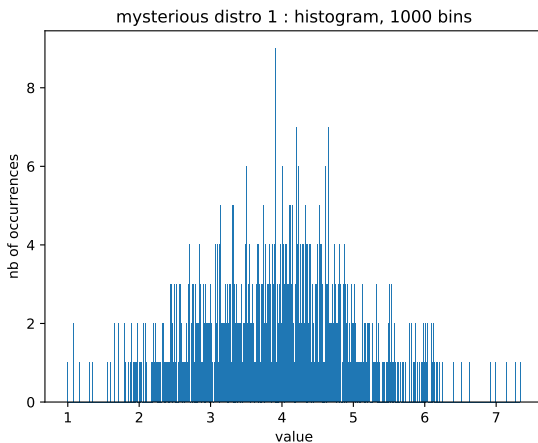Figure – 1000 bins (too many)

Normal distribution

```python
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
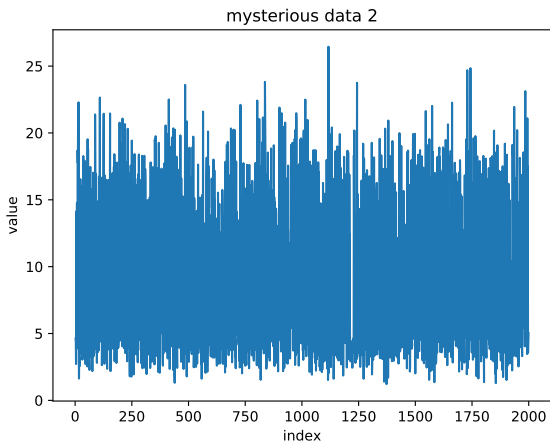
Figure – Second distribution

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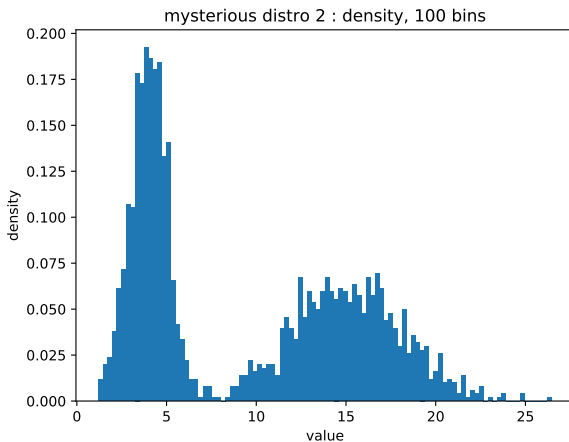
Figure – This distribution has several **modes**

Multimodal distribution

```python
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

Expected value (espérance)

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

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

▶ For a continuous random variable X with density p :

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

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 :

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

▶ For a continuous random variable X with density :

$$E(X) = \int xp(x)dx \qquad (4)$$

Compute the expected value of the dice game.

Multidimensional vectors

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

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

Correlation

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

▶ in statistics and machine learning, this is a very important information !

▶ In physics, variables are often statistically **correlated** : temperature and pressure, etc.

To study this, we can use the **covariance** of the two components, or the correlation, which is a **normalized covariance**.
https://en.wikipedia.org/wiki/Correlation

Variance

The variance is a measure of the dispersion of a random real variable.
https://en.wikipedia.org/wiki/Variance

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

Note that we can also define the variance of a multidimensonial 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.

$$cov(X, Y) = E\Big((X - E(X))(Y - E(Y))\Big) \tag{6}$$

Correlation

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

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

Example

The data **csv_files/distribution_3.csv** contain a random variable
with 5 dimensions. Some of these dimensions are correlated.
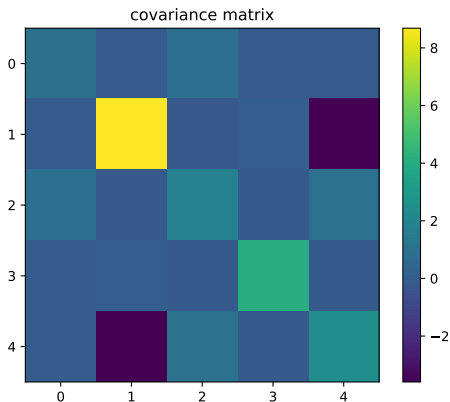
Covariance

Figure – Covariance matrix of the random vector.

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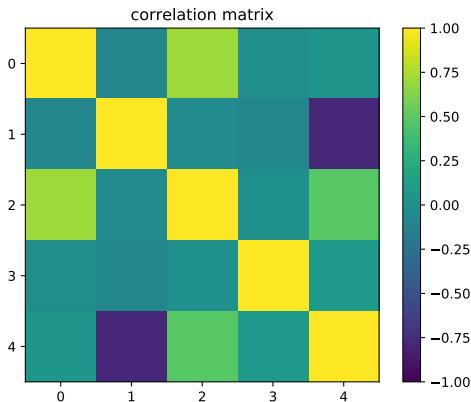
Figure – Correlation matrix for the distribution, note the difference in the scale.

Generation of the data

```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