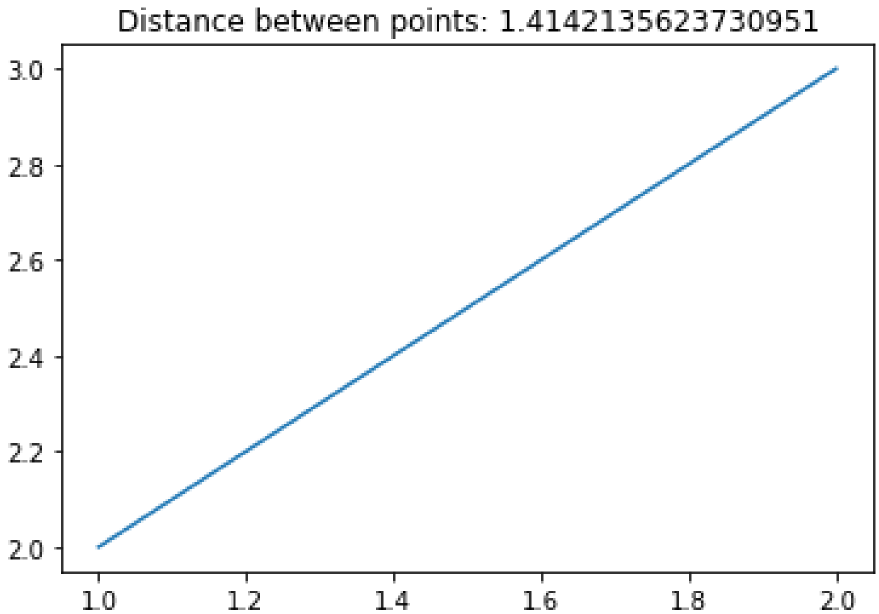
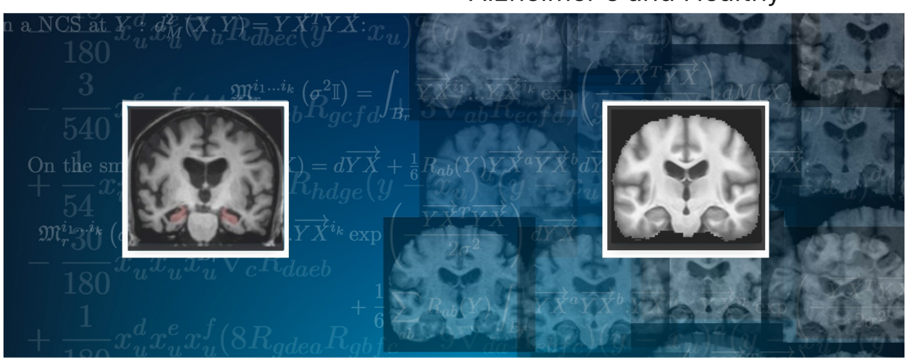
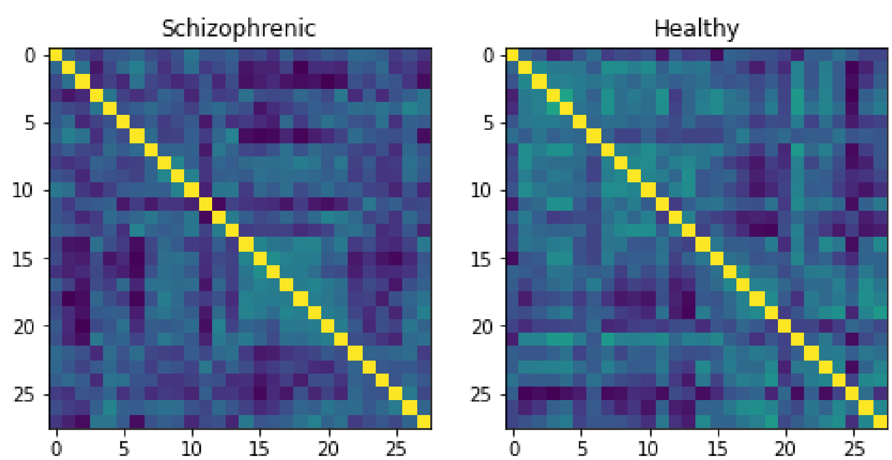
03 Norm and distance



Alzheimer's and Healthy





Unit 1: Vectors, Book ILA Ch. 1-5

- 01 Vectors
- 02 Linear Functions
- ***03 Norms and Distances***
- 04 Clustering

- 05 Linear Independence

Unit 2: Matrices, Book ILA Ch. 6-11 + Book IMC Ch. 2

Unit 3: Least Squares, Book ILA Ch. 12-14 + Book IMC Ch. 8

Unit 4: Eigen-decomposition, Book IMC Ch. 10, 12, 19

Outline: 03 Norms and Distances

- Norm
- Distance
- Standard deviation
- Angle

Outline: 03 Norms and Distances

- **Norm**
- Distance
- Standard deviation
- Angle

Norm

Definition: The Euclidean norm, or just norm, of an n-vector x is:

$$||x|| = \sqrt{x_1^2 + \ldots + x_n^2} = \sqrt{x^T x}$$

Remarks: The norm:

- is used to measure the size of a vector
- reduces to the absolute value for scalar, i.e. for $n = 1$.

Norm

Properties: For any scalar β and any n-vectors x, y:

1. Homogeneity: $||\beta x|| = |\beta| ||x||$
2. Triangle inequality: $||x + y|| \leq ||x|| + ||y||$
3. Nonnegative: $||x|| \geq 0$
4. Definite: $||x|| = 0$ if and only if $x = 0$

$\text{Exercise (at home)}$: Show Prop. 1, 3, 4. Verify Prop. 2 in Python.

Norm

In Python, the module `linalg` from `numpy` has a function computing the norm.

In [8]:
```
import numpy as np

x = np.array([2, -1, 2])

print(np.sqrt(np.sum(x ** 2)))
print(np.linalg.norm(x))
print(np.sqrt((np.inner(x, x))))
```

```
3.0
3.0
3.0
```

Root Mean Square (RMS) value

Definition: The mean-square value of an n-vector x is:

$$\frac{x_1^2 + \ldots + x_n^2}{n} = \frac{||x||^2}{n}.$$

Definition: The root-mean-square (RMS) value of an n-vector x is:

$$rms(x) = \sqrt{\frac{x_1^2 + \ldots + x_n^2}{n}} = \frac{||x||}{\sqrt{n}}.$$

Remarks: rms(x) gives "typical" values of $|x_i|$: e.g. rms(1_n) = 1

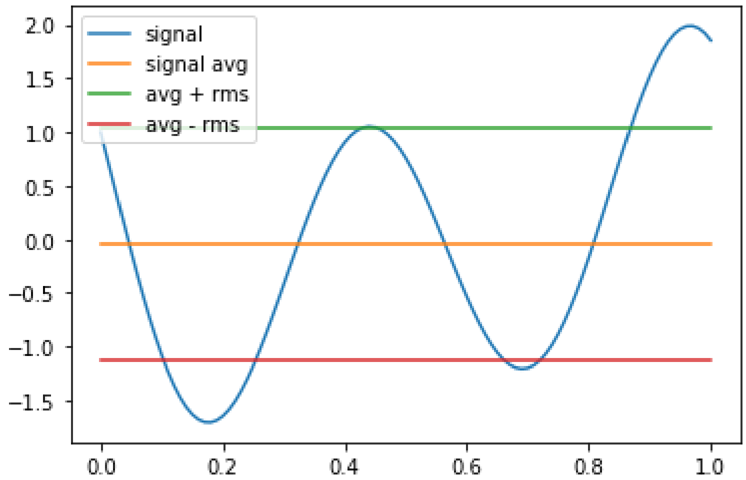
Exercise: Write a function computing the root-mean-square value in Python.

In [10]:
```
def rms(x):
    return np.linalg.norm(x) / np.sqrt(len(x))

ones = np.ones(4)
rms(ones)
```

Out[10]:
```
1.0
```

In Python, we can visualize the RMS value using `matplotlib` .

In [11]:
```
import matplotlib.pyplot as plt

t = np.linspace(0, 1, 101); x = np.cos(8 * t) - 2 * np.sin(11 * t)
plt.plot(t, x); plt.plot(t, np.average(x) * np.ones(len(x)))
plt.plot(t, (np.average(x) + rms(x)) * np.ones(len(x)))
plt.plot(t, (np.average(x) - rms(x)) * np.ones(len(x)))
plt.legend(('signal', 'signal avg', 'avg + rms', 'avg - rms'),
           loc='upper left');
```



Chebyshev inequality

Proposition: Consider an n-vector x. The Chebyshev inequality states:

$$\#\{|x_i| \geq a\} \leq \left(\frac{||x||}{a}\right)^2,$$

i.e. the number of entries x_i such that $|x_i| \geq a$ is no more than $\left(\frac{||x||}{a}\right)^2$, i.e.:

$$\frac{\#\{|x_i| \geq a\}}{n} \leq \left(\frac{rms(x)}{a}\right)^2,$$

i.e. the fraction of entries x_i such that $|x_i| \geq a$ is no more than $\left(\frac{rms(x)}{a}\right)^2$.

Example: With $a = 5rms(x)$, we see that in any vector x, no more than 4\% of entries can satisfy $|x_i| \geq 5rms(x)$.

Outline: 03 Norms and Distances

- Norm
- **Distance**
- Standard deviation
- Angle

Distance

Definition: The Euclidean distance, or just distance, between n-vectors a and b is:

$$dist(a, b) = ||a - b||.$$

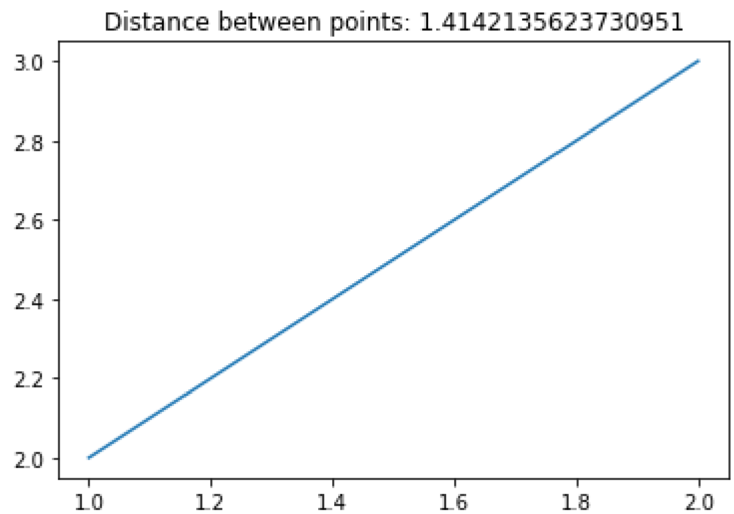
This definition agrees with ordinary distance for $n = 1, 2, 3$.

Definition: The RMS deviation between the n-vectors a and b is defined as:

$$rms(a - b).$$

In Python, we can compute distances using the `norm` function.

In [15]:
```
a = np.array([1, 2])
b = np.array([2, 3])

plt.plot([a[0], b[0]], [a[1], b[1]])
plt.title(f"Distance between points: {np.linalg.norm(a - b)}");
```

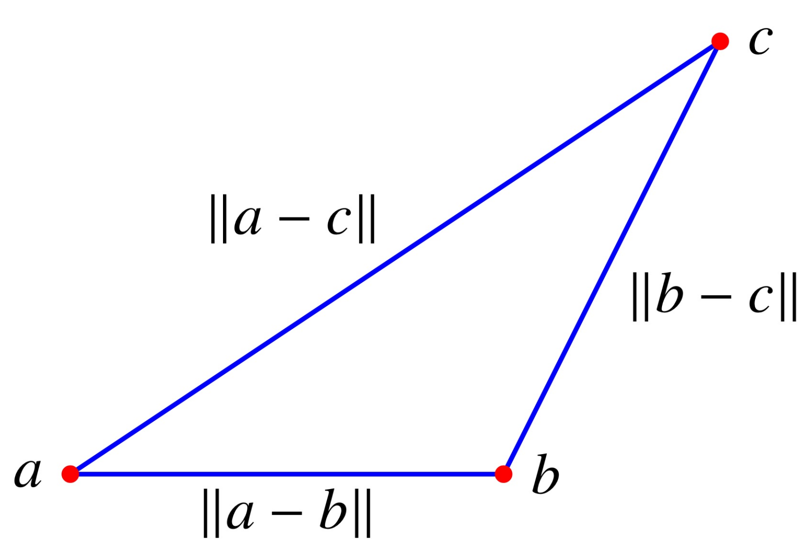


Triangle inequality

- Remember the triangle inequality: $||x + y|| \leq ||x|| + ||y||$
- Apply with: $x = a - b$ and $y = b - c$ and get:

$$||a - c|| \leq ||a - b|| + ||b - c||.$$

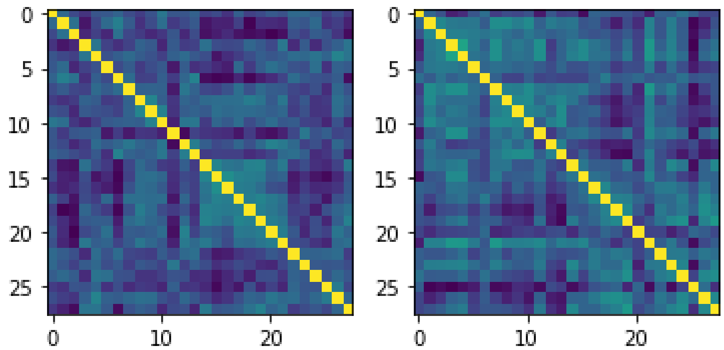
  i.e. the third edge is not longer than the sum of the other two.



In Python, we can use a notion of distance to compute differences between more complex data.

In [14]:

```
import matplotlib.pyplot as plt
import numpy as np
import geomstats.datasets.utils as ds
data, patient_ids, labels = ds.load_connectomes()
fig = plt.figure(figsize=(6, 3))
ax = fig.add_subplot(121); imgplot = ax.imshow(data[0])
ax = fig.add_subplot(122); imgplot = ax.imshow(data[1])
```



We verify that two schizophrenic subjects are "closer" than a schizophrenic subject and a healthy control.

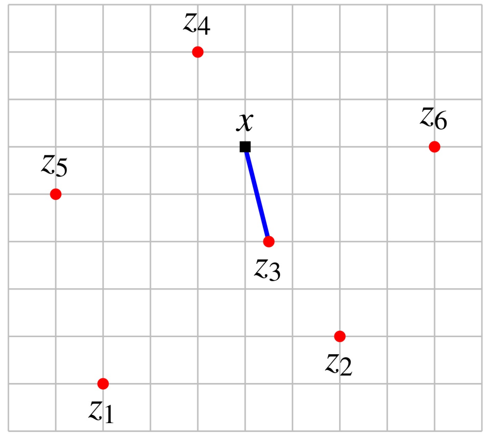
In [17]:

```
print(labels)
healthy = data[0]; schiz1 = data[1]; schiz2 = data[2]
print(f"Dist healthy-schizophrenic: {np.linalg.norm(healthy - schiz1):1.2}")
print(f"Dist between 2 schizophrenics: {np.linalg.norm(schiz1 - schiz2):1.2}")
```

```
[1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
 0 0 1 0 0 1 0 0 1 0 1 0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 1 1 1 1 1 0 1 0 1
 0 0 0 1 1 1 0 0 0 1 0 1]
Dist healthy-schizophrenic: 5.3
Dist between 2 schizophrenics: 4.5
```

Nearest Neighbor

Definition: If $z_1, \ldots, z_m$ is a list of n-vectors, z_j is the nearest neighbor of the n-vector x if $\|x - z_j\| \leq \|x - z_i\|$, for all $i = 1, \ldots, m$.



Exercise: Design an algorithm that can predict if a subject is schizophrenic or not.

Outline: 03 Norms and Distances

- Norm
- Distance
- **Standard deviation**
- Angle

Standard Deviation

Definition: The standard deviation of the n-vector x is:

$$std(x) = rms(x - \bar{x}1) = \frac{||x - \bar{x}1.||}{\sqrt{n}}.$$

The standard deviation gives the typical amount that x_i varies around $\bar{x}$.

Properties:

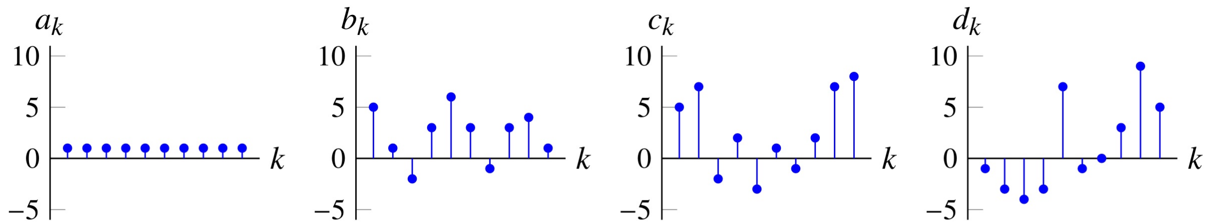
1. $std(x) = 0$ if and only if $x = \alpha 1$ for some α.
2. We have: $rms(x)^2 = \bar{x}^2 + std(x)^2$

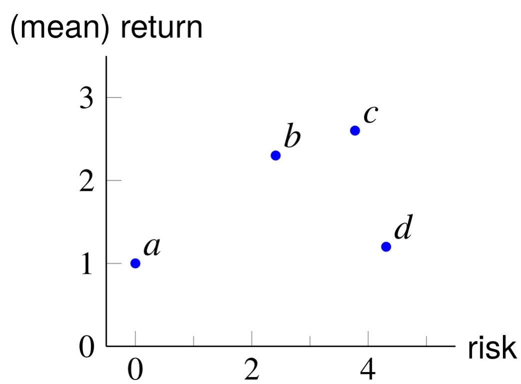
Example: Finance

- x is time series of returns (say, in %) on some investment or asset over some period,
- $\bar{x}$ is the mean return over the period,
- $std(x)$ measures how variable the return is over the period, and is called the risk.

Multiple investments (with different return time series) are:

- compared in terms of return and risk,
- and plotted on a risk-return plot.





Exercise: Show, for return time series x with mean return 8% and risk 3%, a loss ($x_i \leq 0$) can occur in no more than 14.1% of the time. Hint: Use Chebychev inequality.

Outline: 03 Norms and Distances

- Norm
- Distance
- Standard deviation
- **Angle**

Cauchy-Schwarz Inequality

Theorem: For any two n-vectors a and b, we have the Cauchy-Schwarz inequality:

$$|a^Tb| \leq ||a||||b||.$$

Exercise (at home): Use the Cauchy-Schwarz inequality to prove the triangle inequality.

Angle

Definition: The angle between two non-zeros n-vectors a and b is:

$$\angle(a, b) = arccos(\frac{a^Tb}{||a||||b||}).$$

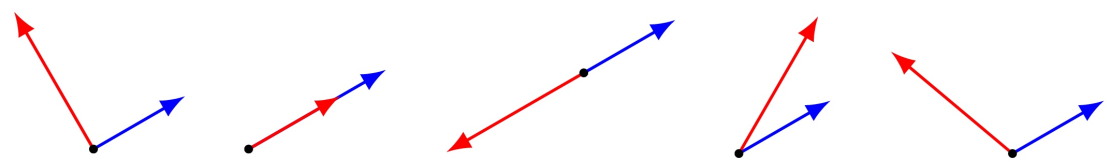
It coincides with the ordinary angle in 2D and 3D.

Properties: The angle $\angle(a, b)$ is the number in $[0, \pi]$ that satisfies:
$a^Tb = ||a||||b|| \cos(\angle(a, b))$

Classification of angles

Write: $\theta = \angle(a, b)$

- $\theta = \pi/2 = 90°$: a and b are orthogonal, written $a \perp b$, ($a^T b = 0$)
- $\theta = 0$: a and b are aligned ($a^T b = ||a|| ||b||$)
- $\theta = \pi = 180°$: a and b are anti-aligned ($a^T b = -||a|| ||b||$)
- $\theta \leq \pi/2 = 90°$: a and b make an acute angle ($a^T b \geq 0$)
- $\theta \geq \pi/2 = 90°$: a and b make an obtuse angle ($a^T b \leq 0$)



Outline: 03 Norms and Distances

- Norm
- Distance
- Standard deviation
- Angle

Resources

- Book ILA Ch. 3