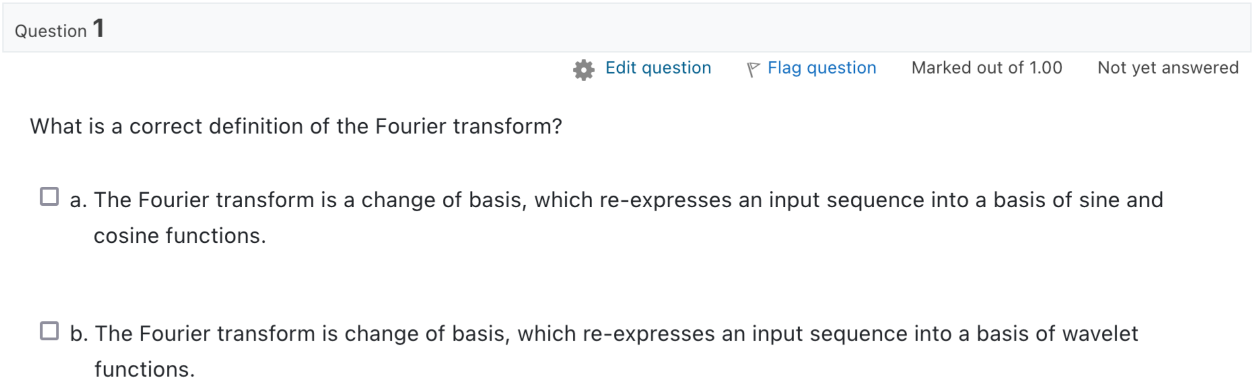
NBE-4070 : Basics of Biomedical Data Analysis

Stéphane Deny
Prof. in Neuroscience and Biomedial Engineering and Computer Science
Aalto University

Lecture 3: Principal Component Analysis (PCA)

Quiz 2



Question **1**

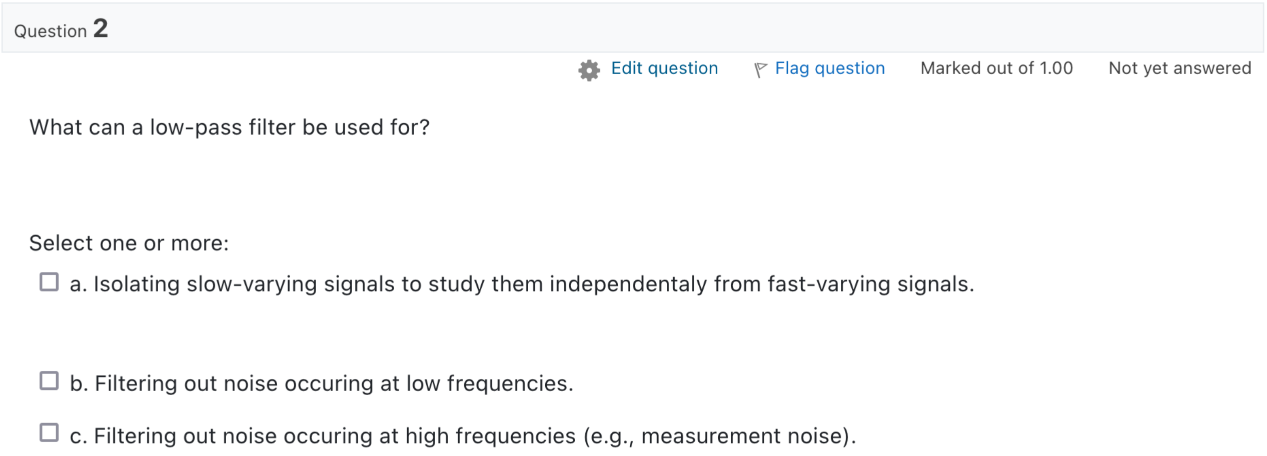
Marked out of 1.00

Not yet answered

What is a correct definition of the Fourier transform?

□ a. The Fourier transform is a change of basis, which re-expresses an input sequence into a basis of sine and cosine functions.

□ b. The Fourier transform is change of basis, which re-expresses an input sequence into a basis of wavelet functions.



Question **2**

Marked out of 1.00

Not yet answered

What can a low-pass filter be used for?

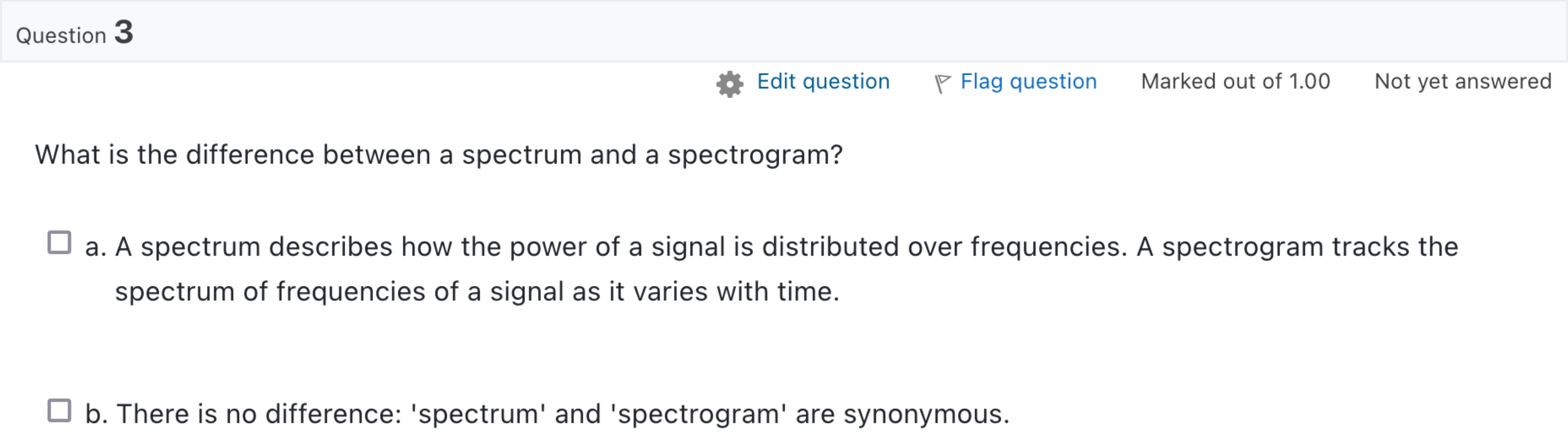
Select one or more:

□ a. Isolating slow-varying signals to study them independentaly from fast-varying signals.

□ b. Filtering out noise occuring at low frequencies.

□ c. Filtering out noise occuring at high frequencies (e.g., measurement noise).

Quiz 2



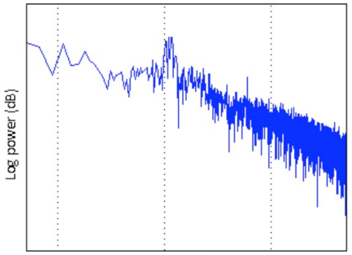
Question **3**

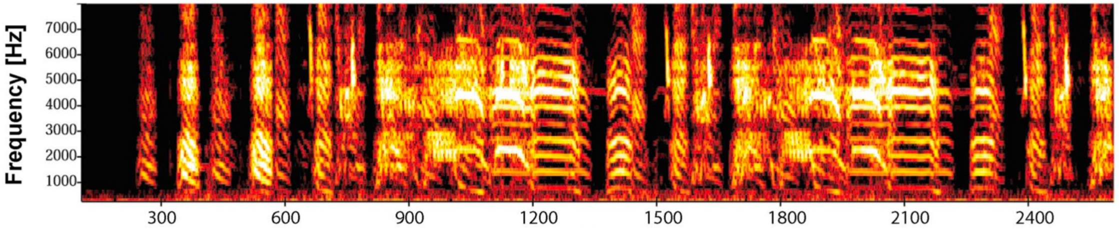
Edit question Flag question Marked out of 1.00 Not yet answered

What is the difference between a spectrum and a spectrogram?

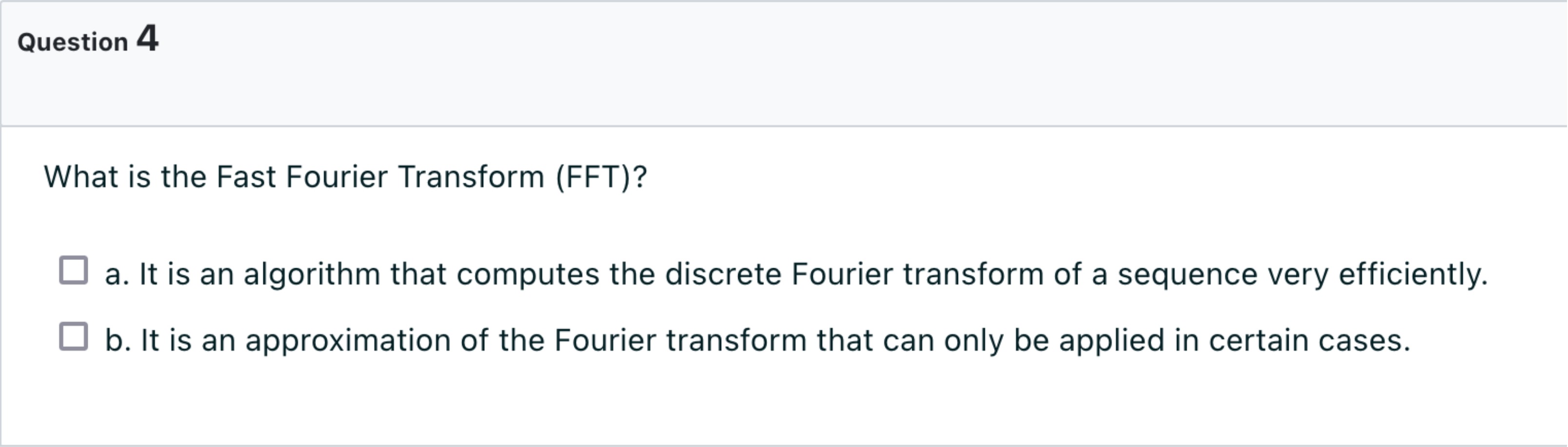
☐ a. A spectrum describes how the power of a signal is distributed over frequencies. A spectrogram tracks the spectrum of frequencies of a signal as it varies with time.

☐ b. There is no difference: 'spectrum' and 'spectrogram' are synonymous.





Quiz 2



Question 4

What is the Fast Fourier Transform (FFT)?

□ a. It is an algorithm that computes the discrete Fourier transform of a sequence very efficiently.

□ b. It is an approximation of the Fourier transform that can only be applied in certain cases.

Outline of the course

1. Mean, Standard Deviation, Standard Error, Confidence Intervals, T-test
2. Fourier Transform, Wavelet Transforms, Spectrograms, High-pass, Low-pass filters
3. Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)
4. Clustering Methods
5. Linear Regression / Logistic Regression
6. Non-linear Methods: Independant Component Analysis, t-Stochastic Neighbour Embedding, Random Forests, Deep Networks
7. Invited lectures from the biomedical industry

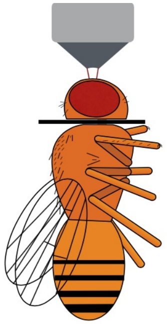
Explain to your neighbor for 5 minutes

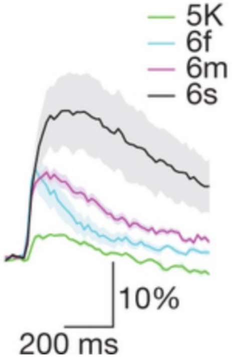
- What is PCA?

(switch roles)

- What can it be used for?

A case study: whole brain calcium imaging in the fruit fly





Experiment

- Calcium imaging with a two-photon scanner (indicator: GCaMP6m)

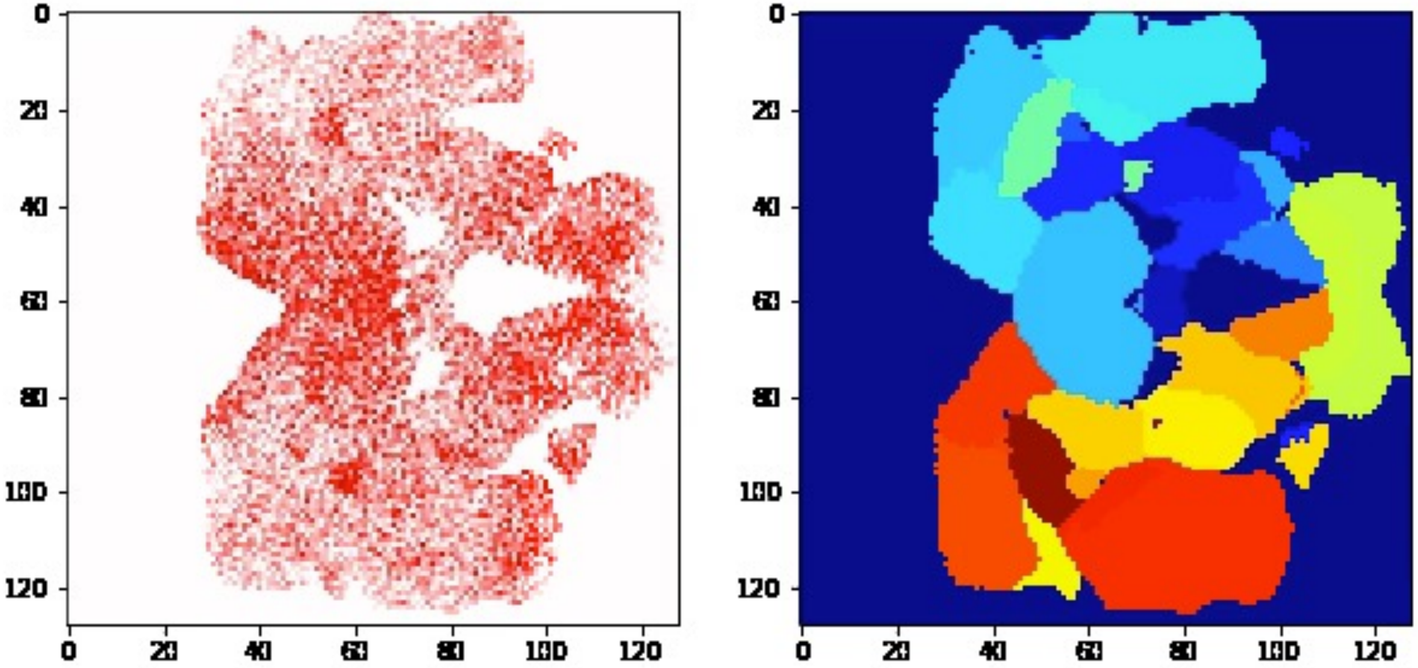
- 250,000 voxels (200,000 neurons)
2.6 micron*2.6 micron*7.5micron

- acquisition frequency: 2Hz

- session duration: 33 minutes (4000 points)

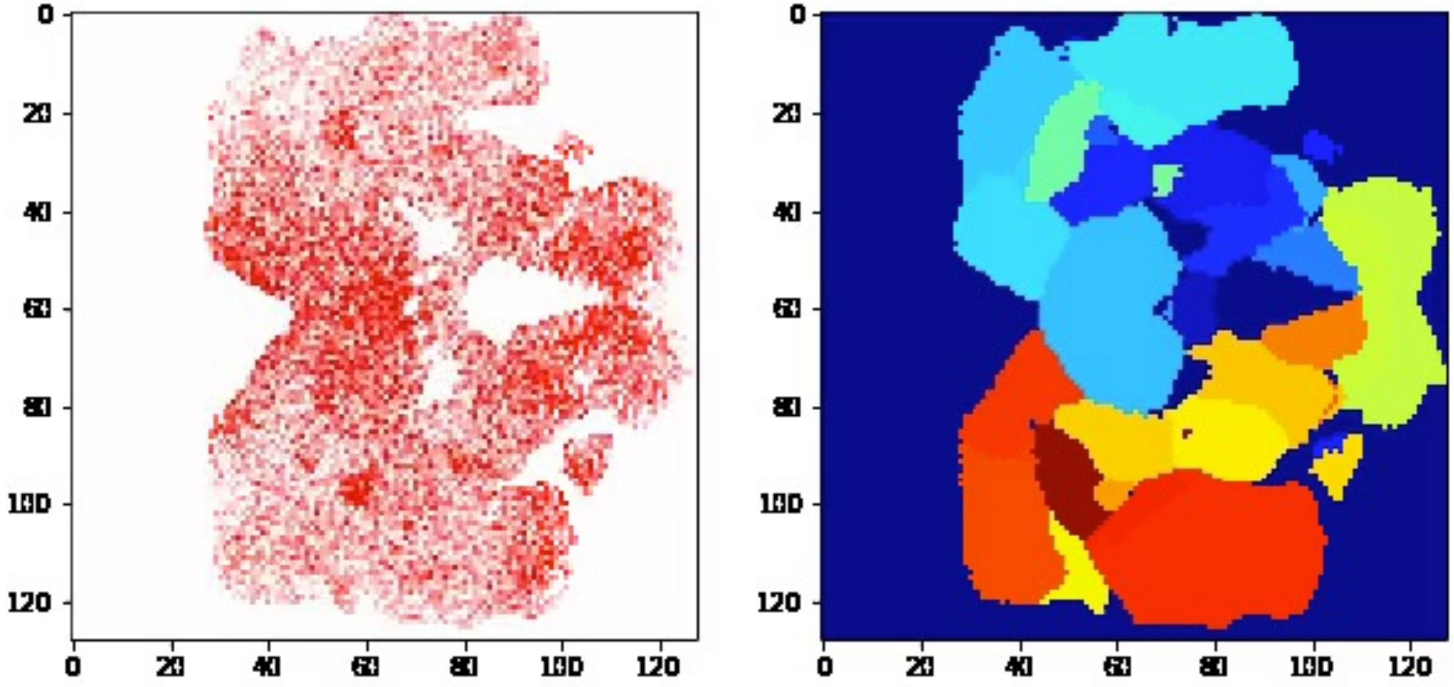
- no stimulus / no behavior

A case study: whole brain calcium imaging in the fruit fly



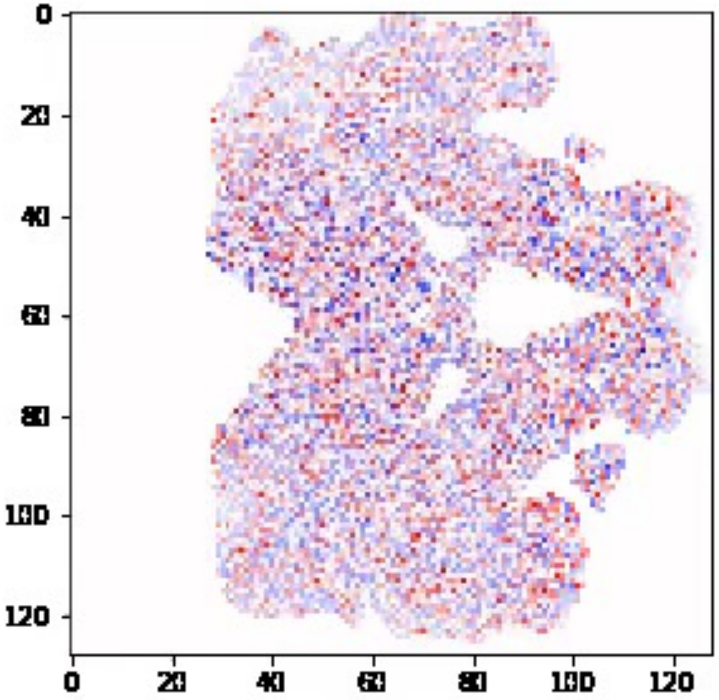
A case study: whole brain calcium imaging in the fruit fly

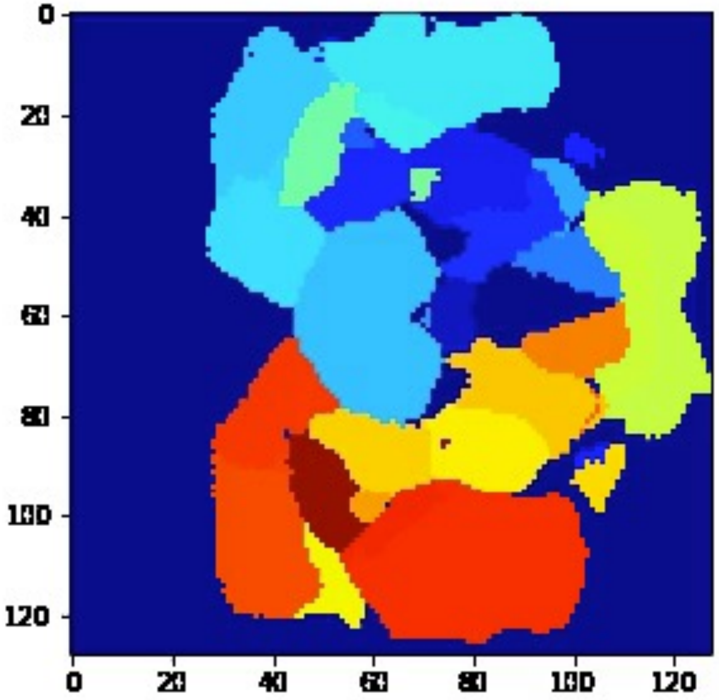
real time x 10



A case study: whole brain calcium imaging in the fruit fly

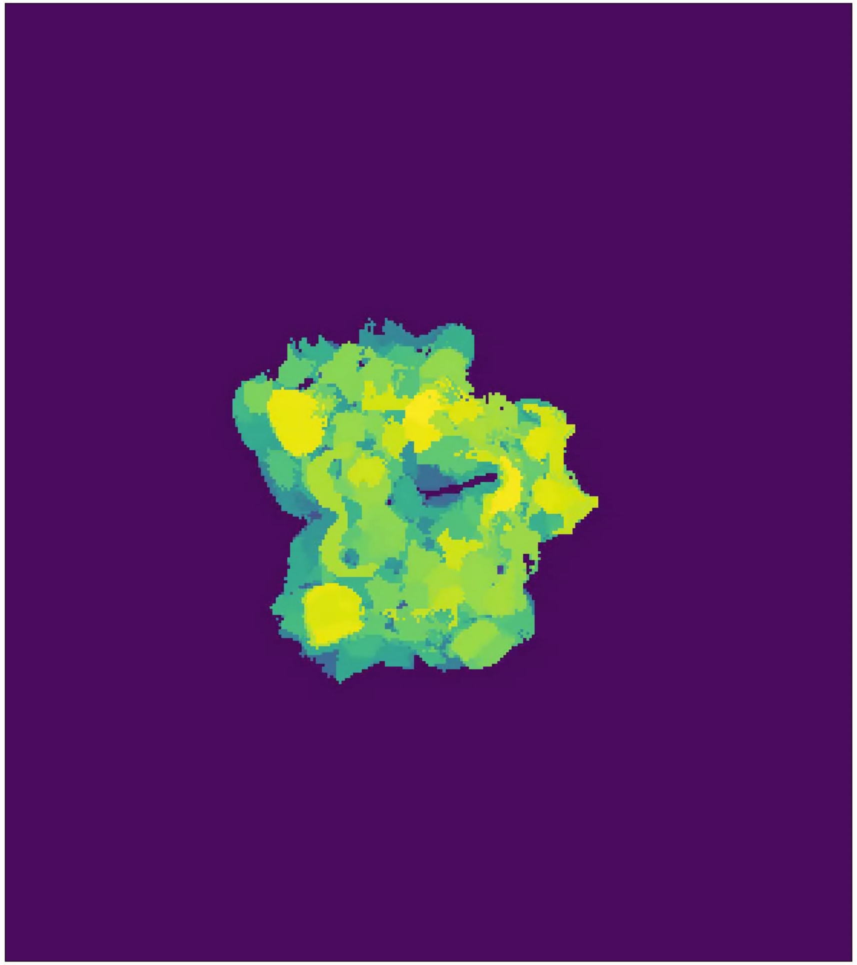
subtracting the mean activity of every voxel





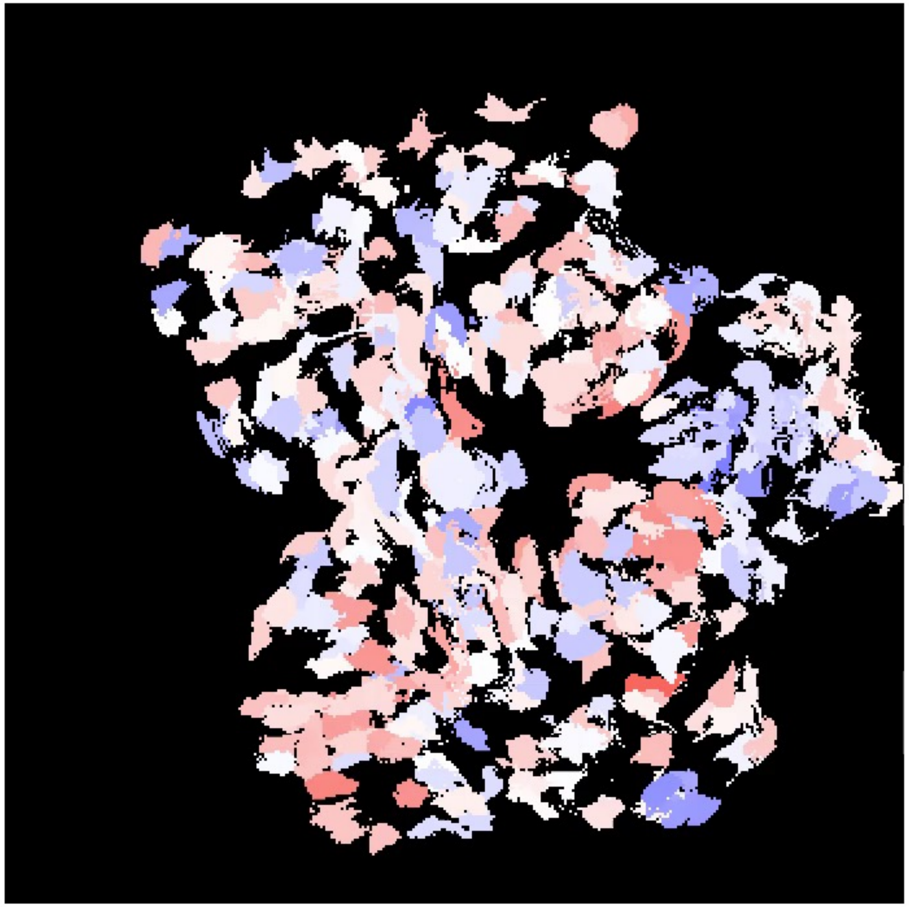
A case study: whole brain calcium imaging in the fruit fly

averaging the activity over regions:



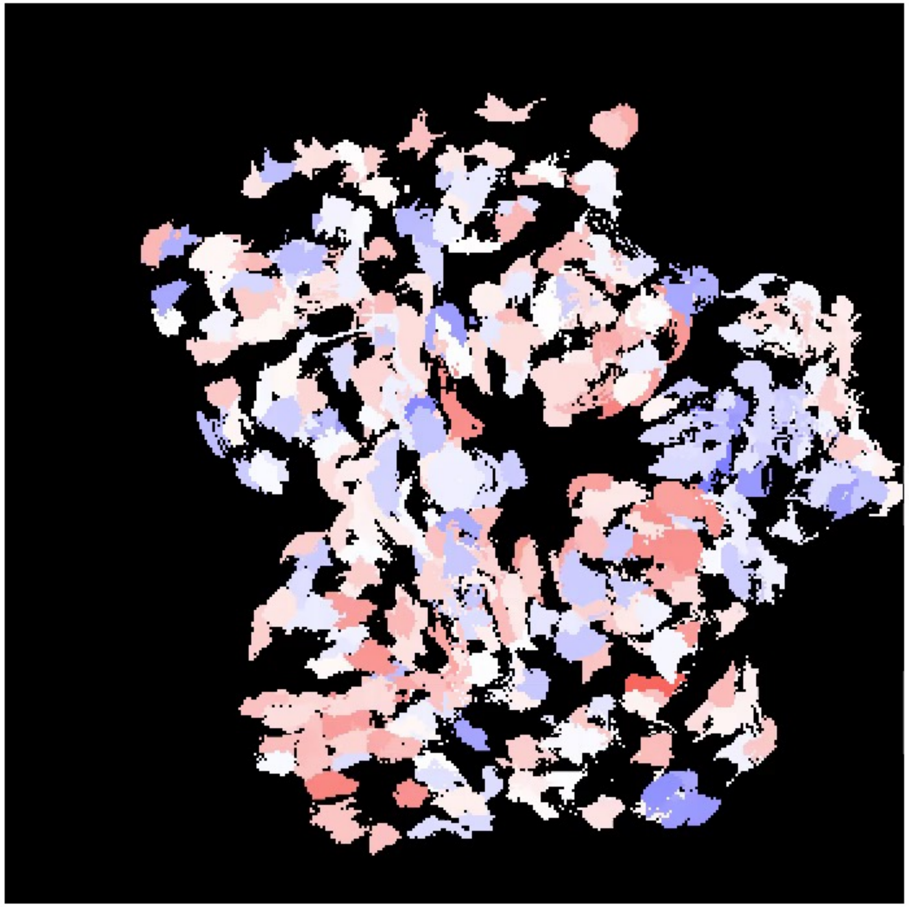
A case study: whole brain calcium imaging in the fruit fly

averaging the activity over regions:



A case study: whole brain calcium imaging in the fruit fly

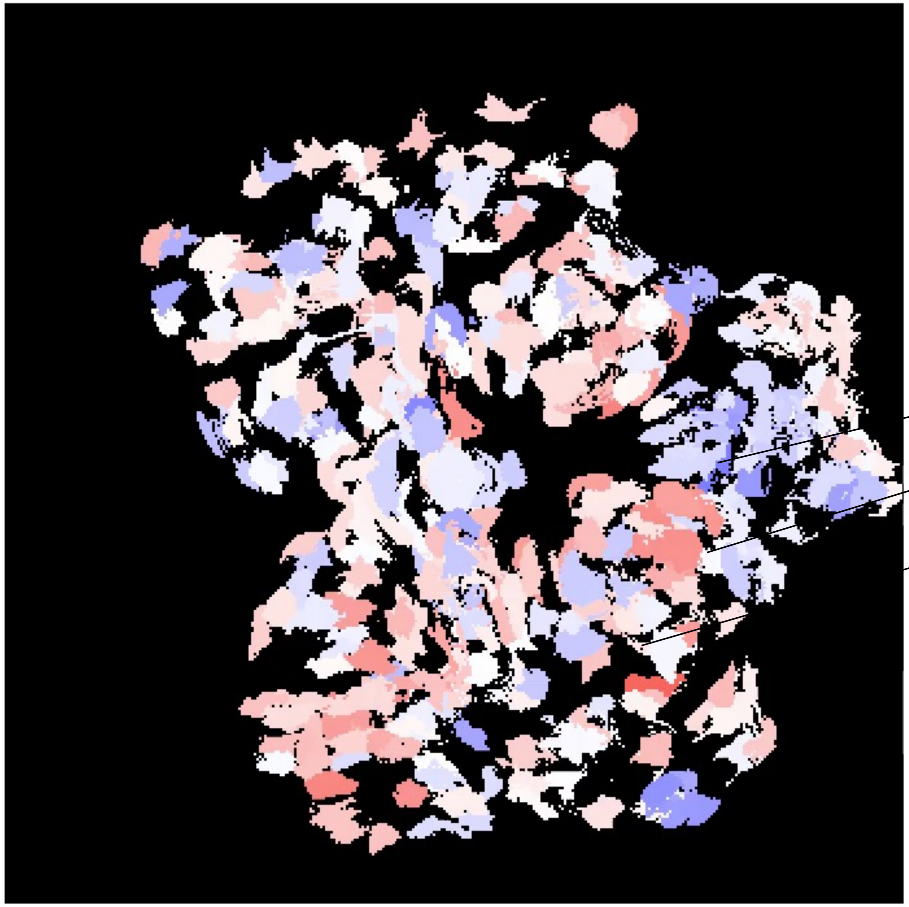
PCA consists in "lumping together" regions that **cofluctuate** in order to maximize the signal (informal definition).



Use-cases of PCA: a good first step to analyze high-dimensional datasets

- Exploratory data analysis of a high-dimensional dataset
- Revealing hidden structure in a dataset
- Reducing the dimensionality of a dataset for modelling purposes
- Removing noise

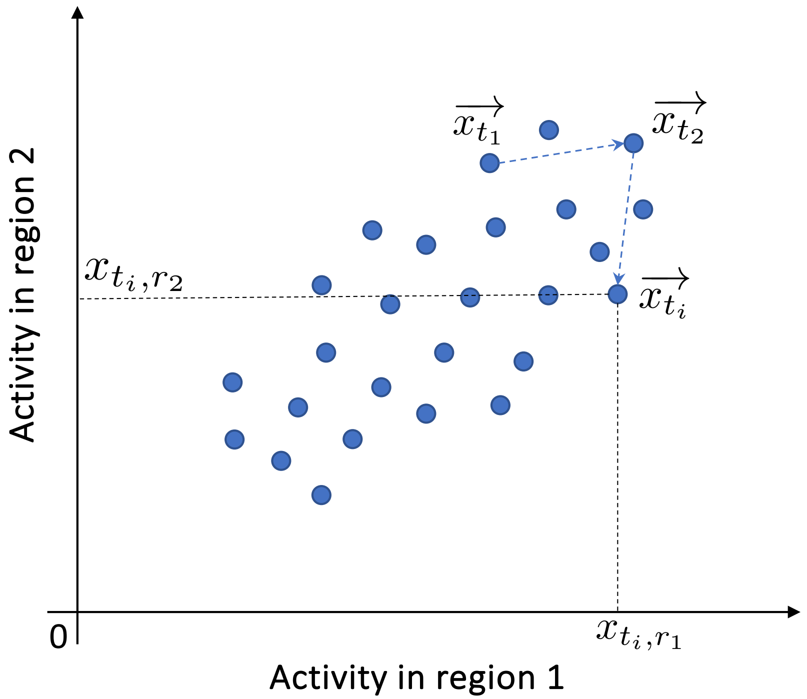
A case study: whole brain calcium imaging in the fruit fly



Vectorial notation of activity:

$$\overrightarrow{x_{t_i}} = \left(x_{t_i,r_1} \quad x_{t_i,r_2} \quad \cdots \quad x_{t_i,r_i} \quad \cdots \right)$$

Geometric representation of the data

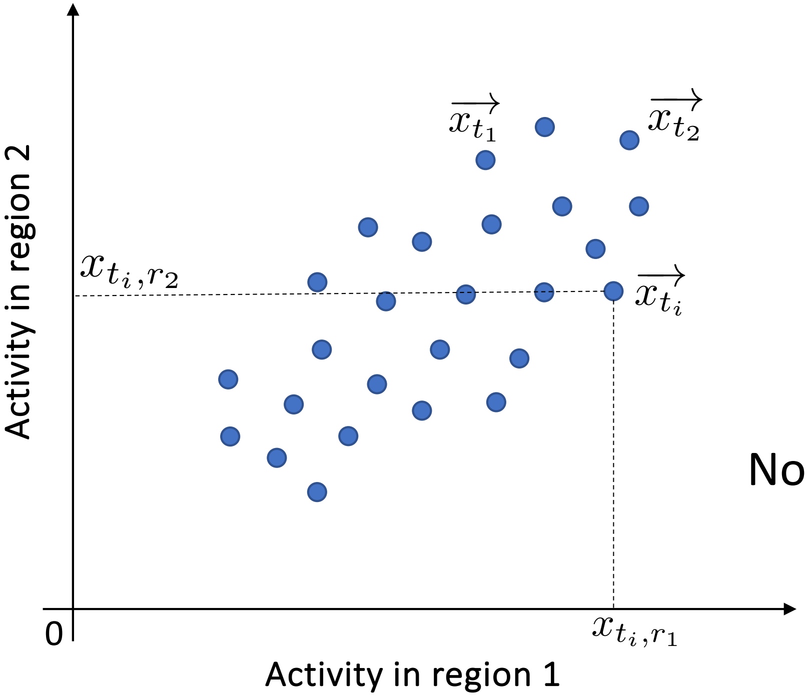


$\overrightarrow{x_{t_1}}$
$\overrightarrow{x_{t_2}}$
$\overrightarrow{x_{t_i}}$
x_{t_i,r_2}
x_{t_i,r_1}
0
Activity in region 2
Activity in region 1

Vectorial notation of activity:

$$\overrightarrow{x_{t_i}} = \begin{pmatrix} x_{t_i,r_1} & x_{t_i,r_2} & \cdots & x_{t_i,r_i} & \cdots \end{pmatrix}$$

Geometric representation of the data



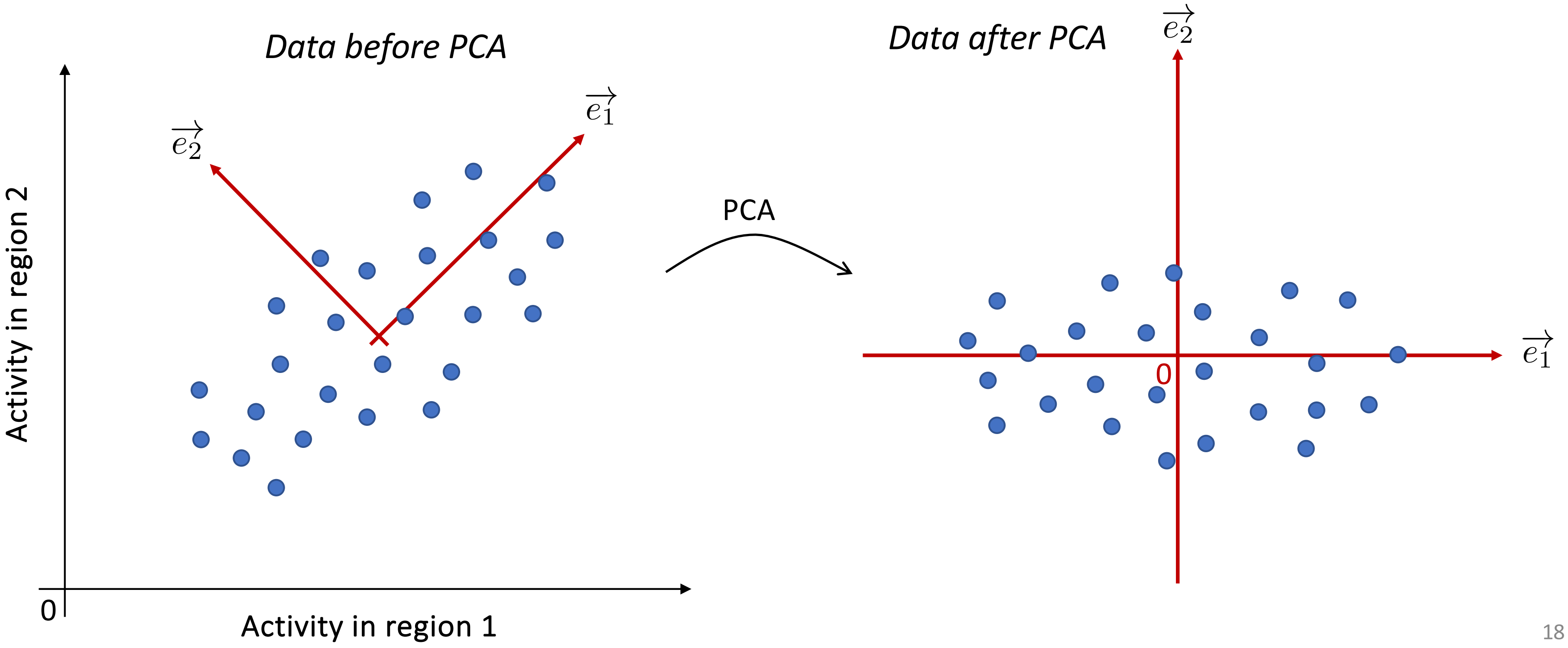
Vectorial notation of activity:

$$\overrightarrow{x_{t_i}} = \begin{pmatrix} x_{t_i,r_1} & x_{t_i,r_2} & \cdots & x_{t_i,r_i} & \cdots \end{pmatrix}$$

Note: the activity in these two regions cofluctuates.

A geometric view of PCA

PCA consists in “lumping together” regions that cofluctuate in order to maximize the signal (informal definition).



Data before PCA

Data after PCA

Definition: covariance

- Given two paired variables $X = \{x_1, x_2, ..., x_N\}$

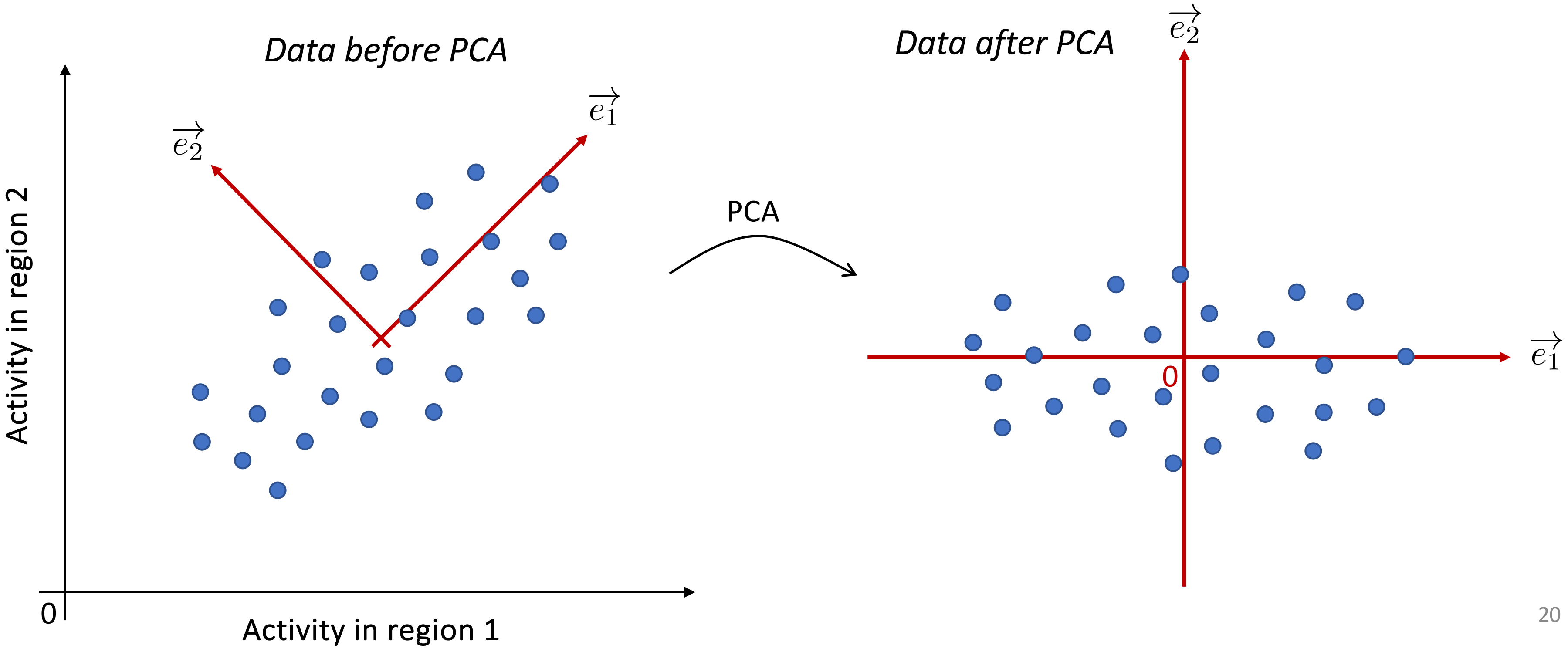
$$Y = \{y_1, y_2, ..., y_N\}$$

- Their ***covariance*** is given by

$$Cov(X,Y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \underset{\substack{\uparrow \\ \text{mean of signal X}}}{\mu_X})(y_i - \underset{\substack{\uparrow \\ \text{mean of signal Y}}}{\mu_Y})$$

A geometric view of PCA

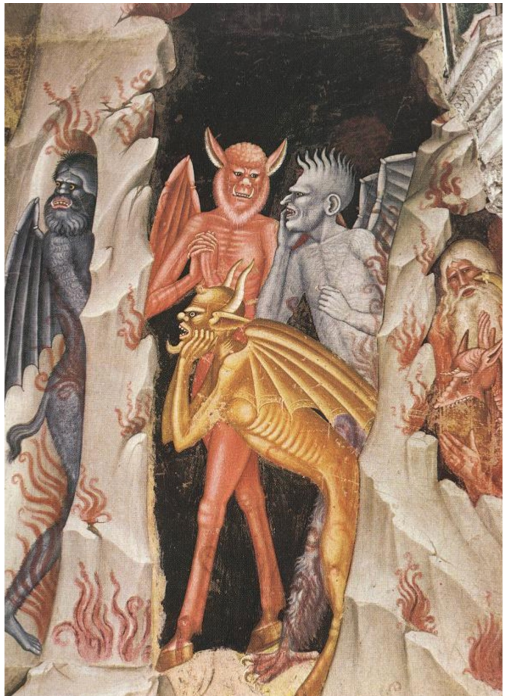
PCA consists in finding a new ccordinate system aligned with the *covariance* of the data, such that the covariance between variables is null in that new basis (more formal definition).



Algebraic view of PCA

“Algebra is the offer made by the devil to the mathematician. The devil says: I will give you this powerful machine, it will answer any question you like. All you need to do is give me your soul: give up geometry and you will have this marvelous machine.”

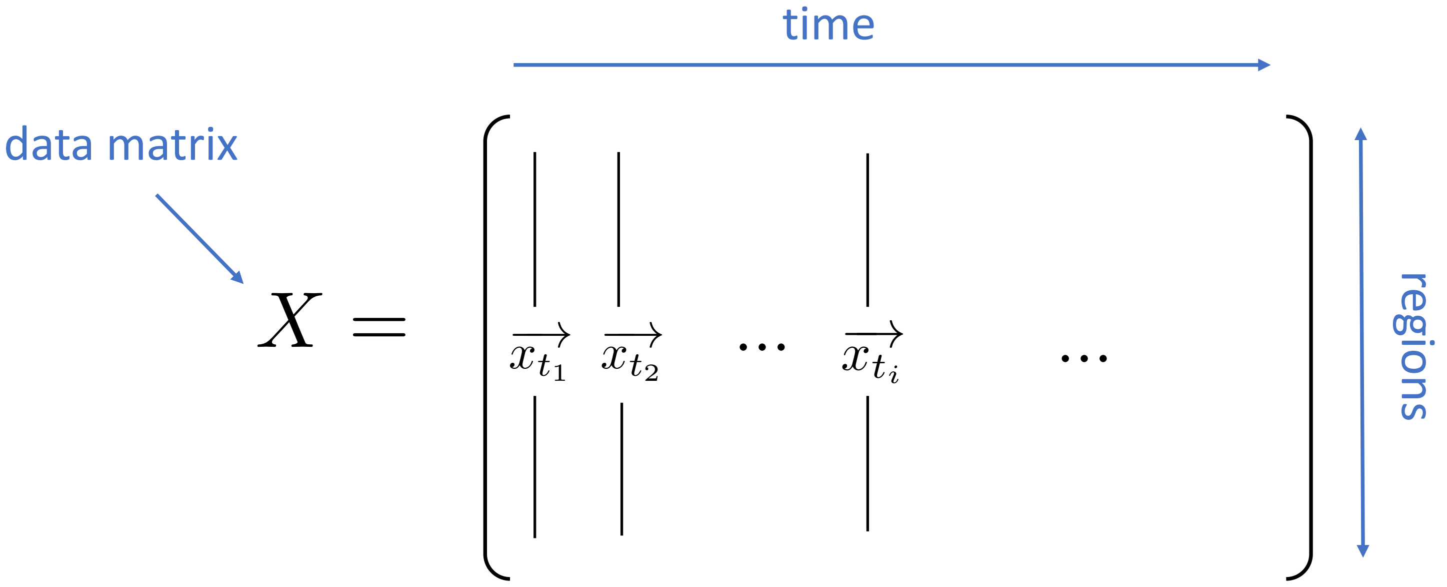
— Michael Francis Atiyah



source: https://www.goodreads.com/quotes/9026708-algebra-is-the-offer-made-by-the-devil-to-the
https://uscatholic.org/articles/201409/what-do-catholics-believe-about-the-devil/

Algebraic view of PCA (1)

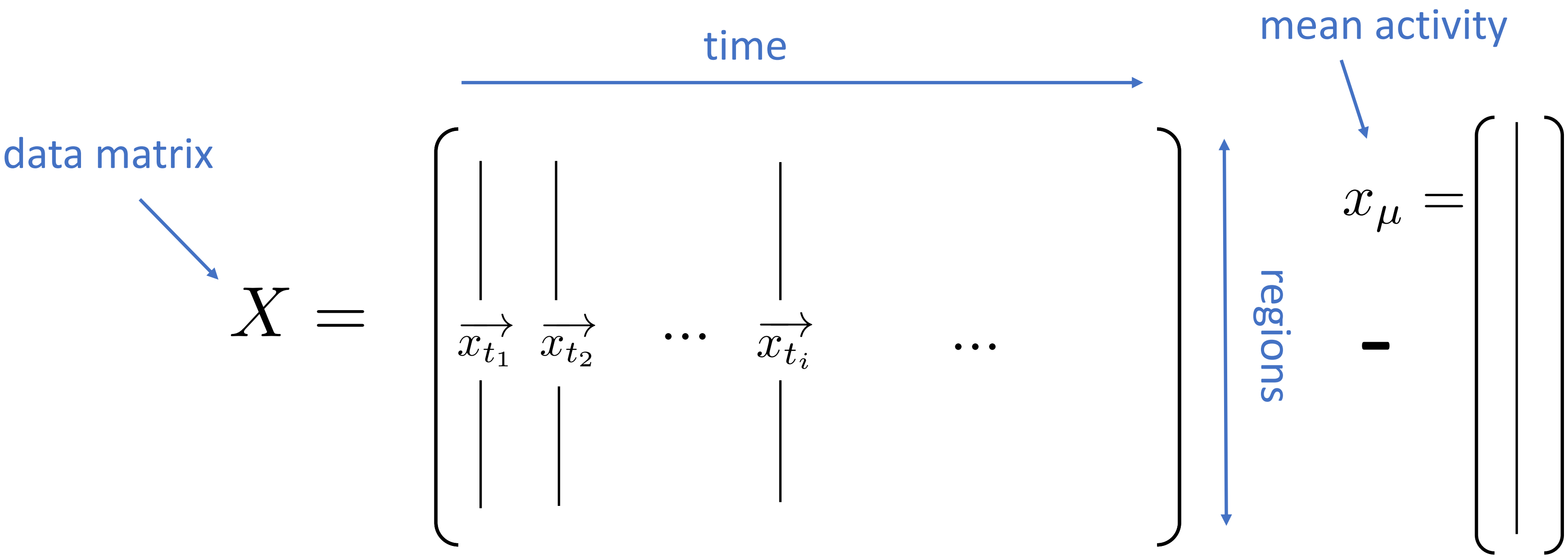
To arrive to a formal (algebraic) definition of PCA, we first need to store the data into a matrix:



$$X = \begin{bmatrix} | & | & & | & & \\ \overrightarrow{x_{t_1}} & \overrightarrow{x_{t_2}} & \cdots & \overrightarrow{x_{t_i}} & \cdots & \\ | & | & & | & & \end{bmatrix}$$

Algebraic view of PCA (2)

PCA consists in subtracting the mean of the data...



data matrix

time

regions

mean activity

$$X = \begin{pmatrix} \overrightarrow{x_{t_1}} & \overrightarrow{x_{t_2}} & \cdots & \overrightarrow{x_{t_i}} & \cdots \end{pmatrix}$$

$$x_\mu = \begin{pmatrix} \; \end{pmatrix}$$

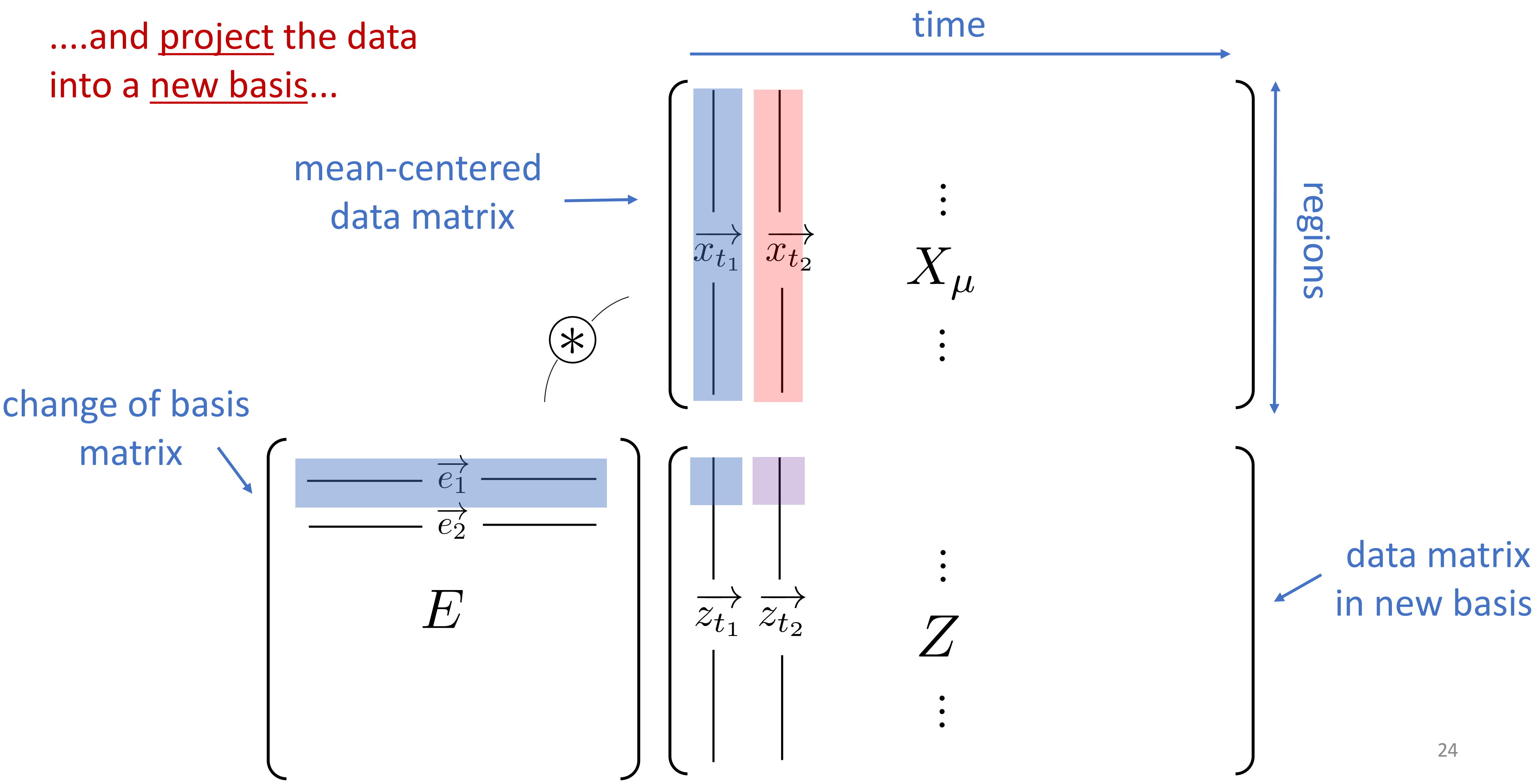
-

mean-centered data matrix

$$X_\mu = X - \overrightarrow{x_\mu}$$

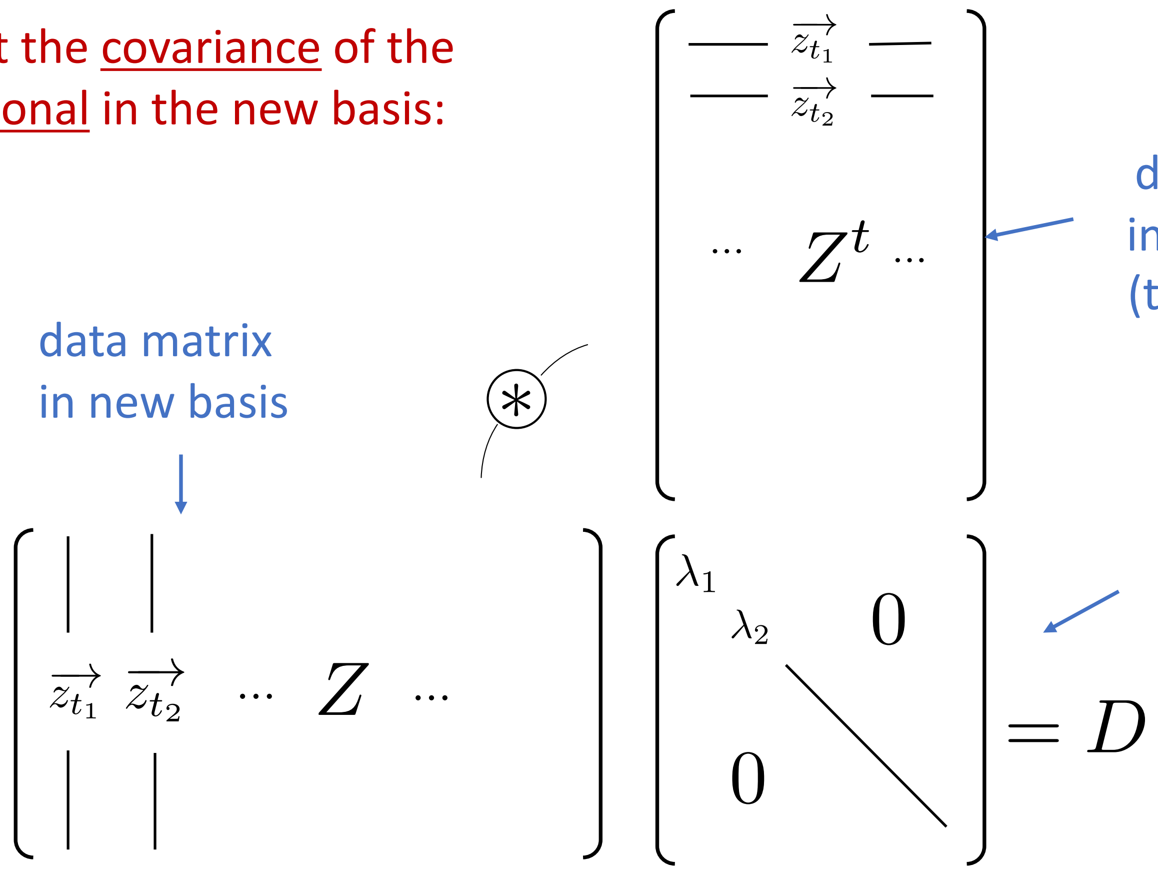
Algebraic view of PCA (3)

....and project the data into a new basis...


time
mean-centered data matrix
$\overrightarrow{x_{t_1}}$
$\overrightarrow{x_{t_2}}$
X_μ
regions
$\circledast$
change of basis matrix
$\overrightarrow{e_1}$
$\overrightarrow{e_2}$
E
$\overrightarrow{z_{t_1}}$
$\overrightarrow{z_{t_2}}$
Z
data matrix in new basis

Algebraic view of PCA (4)

... such that the covariance of the data is diagonal in the new basis:


$\overrightarrow{z_{t_1}}$
$\overrightarrow{z_{t_2}}$
$\cdots$ Z^t $\cdots$
$\circledast$
$\overrightarrow{z_{t_1}}$ $\overrightarrow{z_{t_2}}$ $\cdots$ Z $\cdots$
λ_1
λ_2
0
0
$= D$

data matrix in new basis (transposed)

data matrix in new basis

covariance matrix of Z

Formal definition of PCA

PCA consists in finding an (orthonormal) basis such that the covariance of the data is diagonal in that basis.

In equations: we want to find a change of basis E such that, for

$Z = EX_{\mu}$, the dataset expressed in this new basis,

we have $Cov(Z) = D$ where D is a diagonal matrix

Properties of E such the E qualifies as an orthonormal change of basis:

$||\overrightarrow{e_i}||_2 = 1$ for all i

$\overrightarrow{e_i}.\overrightarrow{e_j} = 0$ for all i≠j

After calculations (see supplementary slide), we can show that, if E satisfies the conditions of PCA, then the covariance matrix of the data X can be decomposed in the following product:

$Cov(X) = E^t DE$ where D is a diagonal matrix and E orthonormal

source: https://en.wikipedia.org/wiki/Principal_component_analysis

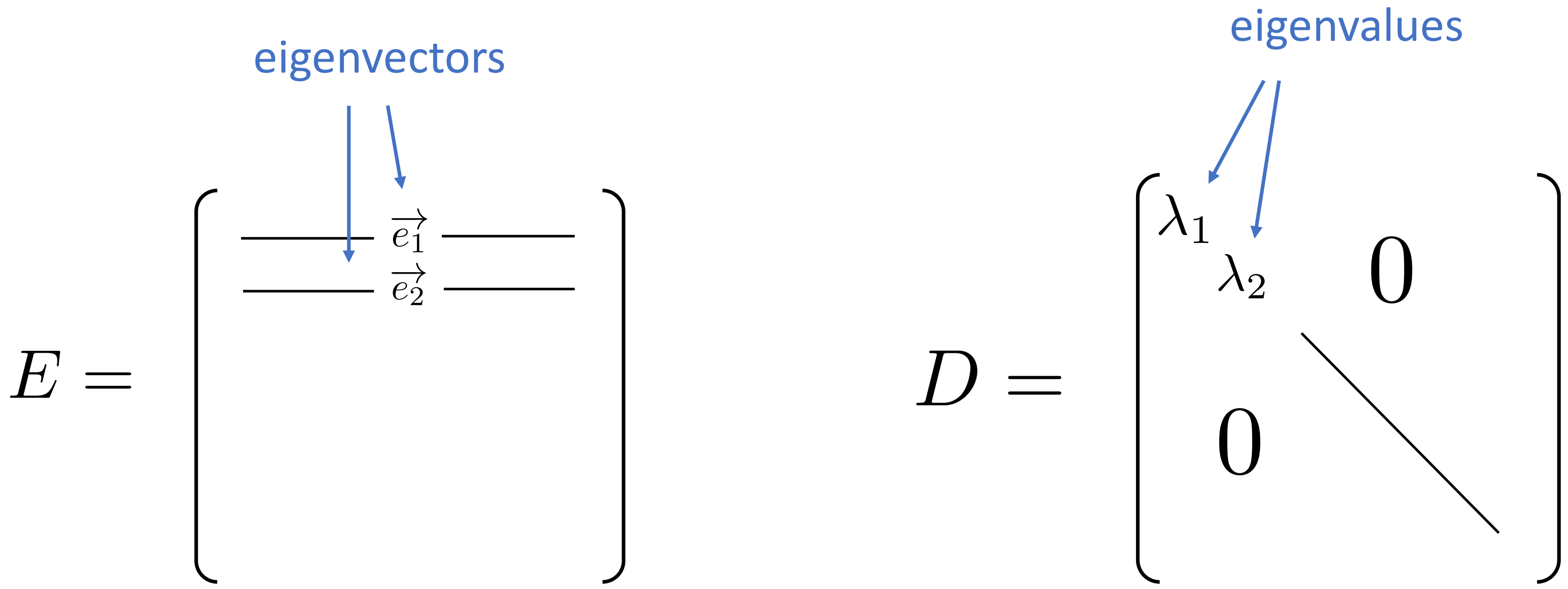
Definitions: *eigenvectors*, *eigenvalues* and *scores*

$$Cov(X) = E^t DE$$

eigendecomposition of Cov(X)

$$Z = EX_\mu$$

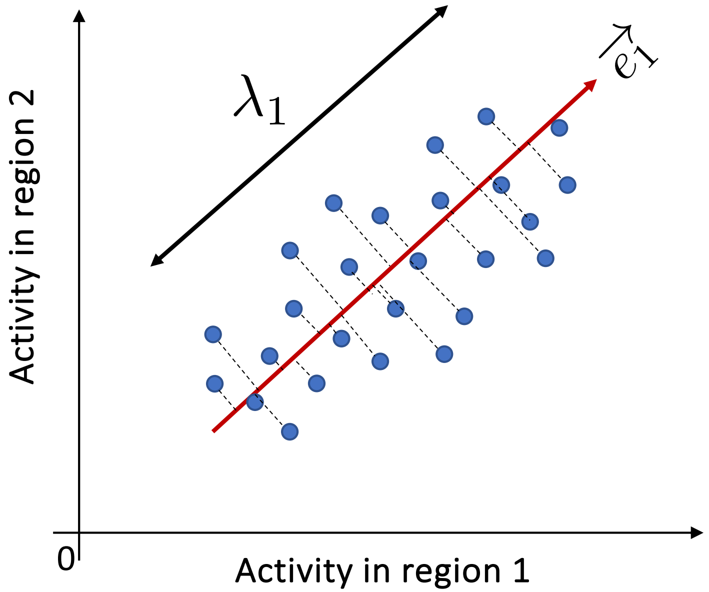
PC scores



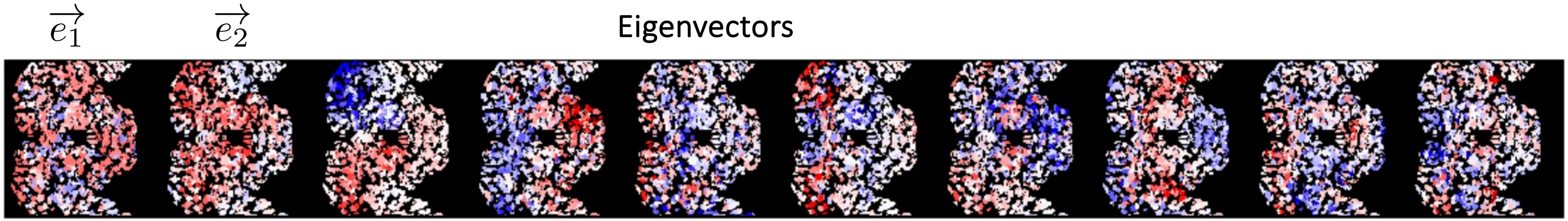
Eigenvalues and Explained Variance

The eigenvalues are equal to the variance of the data projected along their corresponding eigenvector:

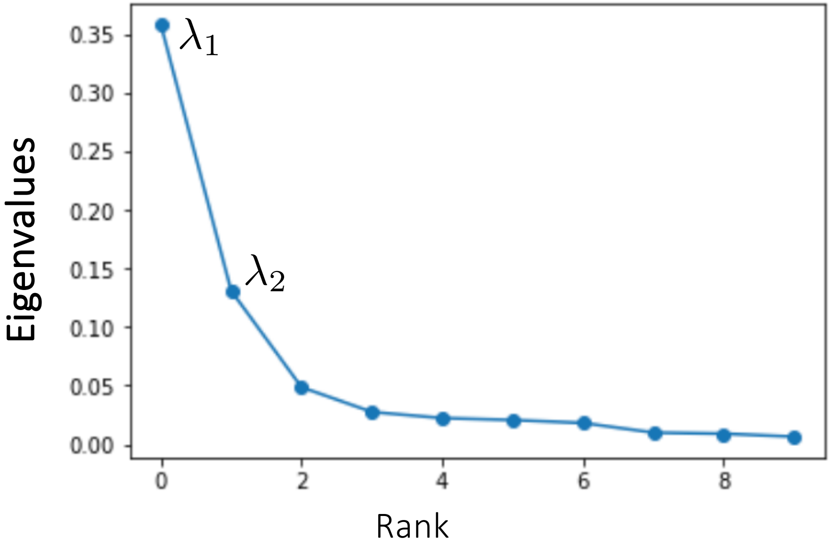
$$Var(X.\vec{e_i}) = \lambda_i \quad \text{for all i}$$



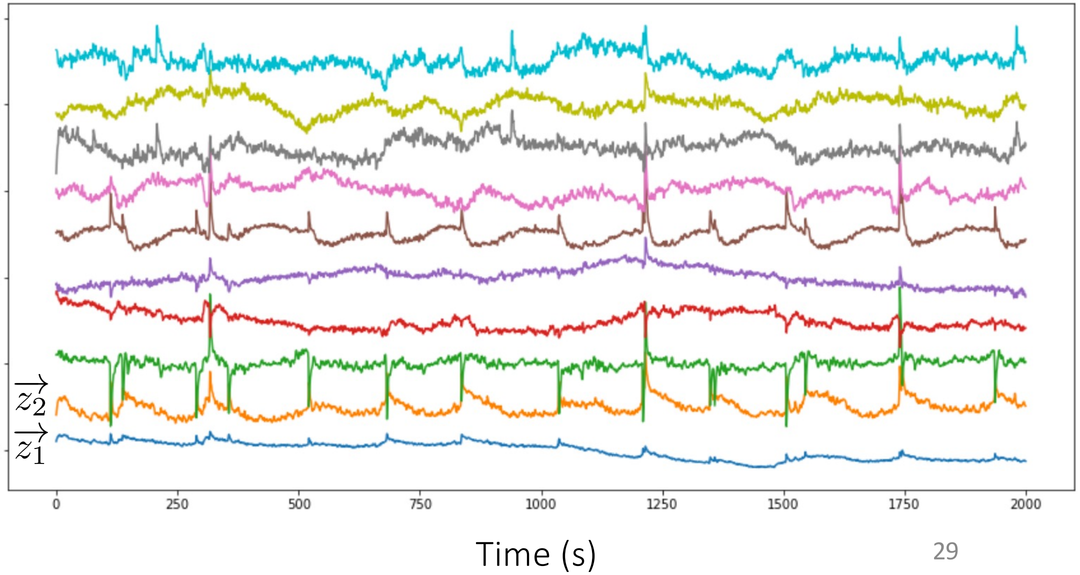
Case study: applying PCA to the fly brain



Eigenvectors



Scores



Insights from applying PCA to the fly brain

- A few dimensions can explain quite a lot of the variability in the data.
- The whole brain activity is the dominant source of variability in the data.
- There is some interesting structure in the activity: for example, the two hemispheres exhibit sharp events which look pretty independant.

Steps to perform PCA on a computer

1. Subtract the mean of the data (computed over samples): $X_\mu = X - x_\mu$

2. Compute the covariance matrix of the data: $Cov(X) = X_\mu X_\mu^t$

3. Compute eigendecomposition of Cov(X) using a precoded algorithm: $Cov(X) = E^t DE$

4. Plot (ranked) eigenvalues and interpret the dimensionality of the dataset.

5. Plot eigenvectors and interpret them.

6. (Sometimes) Compute and plot PC scores $Z = EX_\mu$ and interpret them.

Risks associated with interpreting PCA

- Reducing dimensionality can result in losing some small but important signals.

- PCA can be blind to some complex structure present in the data (see board for examples).

- PCA always gives a decomposition, even on a random dataset. This decomposition is not always meaningful.

Next lecture

- Clustering methods

Supplementary material (can be ignored)

Calculation showing how to arrive to the equation of PCA

We are looking for a change of basis E for X such that $Cov(Z) = D$ where D is diagonal

and where $Z = EX_\mu$

We can rewrite Cov(Z) as: $Cov(Z) = ZZ^t$ by the definition of covariance and because Z is mean-centered

$Cov(Z) = (EX_\mu)(EX_\mu)^t$ by the definition of Z

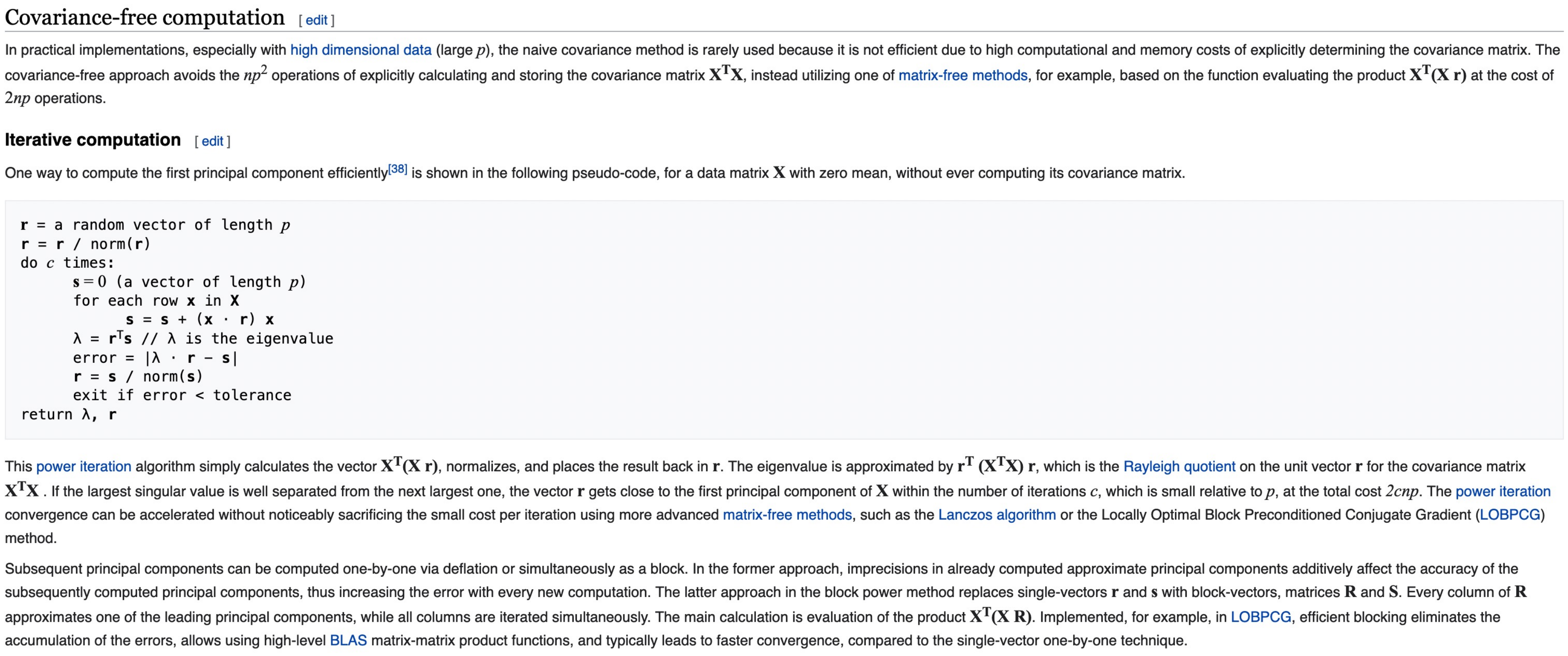
$Cov(Z) = EX_\mu X_\mu^t E^t$ this is how the transpose operator applies to a matrix product

$Cov(Z) = ECov(X)E^t$ by definition of the covariance of X

And so we are looking for E such that $ECov(X)E^t = D$ where D is diagonal

Finally we can rewrite this equation as $\boxed{Cov(X) = E^t DE}$ as $E^{-1} = E^t$ for any orthonormal matrix

Iterative algo to compute PCA



Covariance-free computation [edit]

In practical implementations, especially with high dimensional data (large p), the naive covariance method is rarely used because it is not efficient due to high computational and memory costs of explicitly determining the covariance matrix. The covariance-free approach avoids the np^2 operations of explicitly calculating and storing the covariance matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$, instead utilizing one of matrix-free methods, for example, based on the function evaluating the product $\mathbf{X}^\mathsf{T}(\mathbf{X}\ \mathbf{r})$ at the cost of $2np$ operations.

Iterative computation [edit]

One way to compute the first principal component efficiently[38] is shown in the following pseudo-code, for a data matrix $\mathbf{X}$ with zero mean, without ever computing its covariance matrix.

```
$\mathbf{r}$ = a random vector of length $p$
$\mathbf{r}$ = $\mathbf{r}$ / norm($\mathbf{r}$)
do $c$ times:
      $\mathbf{s} = 0$ (a vector of length $p$)
      for each row $\mathbf{x}$ in $\mathbf{X}$
            $\mathbf{s}$ = $\mathbf{s}$ + ($\mathbf{x}$ · $\mathbf{r}$) $\mathbf{x}$
      $\lambda$ = $\mathbf{r}^\mathsf{T}\mathbf{s}$ // $\lambda$ is the eigenvalue
      error = |$\lambda$ · $\mathbf{r}$ − $\mathbf{s}$|
      $\mathbf{r}$ = $\mathbf{s}$ / norm($\mathbf{s}$)
      exit if error < tolerance
return $\lambda$, $\mathbf{r}$
```

This power iteration algorithm simply calculates the vector $\mathbf{X}^\mathsf{T}(\mathbf{X}\ \mathbf{r})$, normalizes, and places the result back in $\mathbf{r}$. The eigenvalue is approximated by $\mathbf{r}^\mathsf{T}\ (\mathbf{X}^\mathsf{T}\mathbf{X})\ \mathbf{r}$, which is the Rayleigh quotient on the unit vector $\mathbf{r}$ for the covariance matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$. If the largest singular value is well separated from the next largest one, the vector $\mathbf{r}$ gets close to the first principal component of $\mathbf{X}$ within the number of iterations c, which is small relative to p, at the total cost $2cnp$. The power iteration convergence can be accelerated without noticeably sacrificing the small cost per iteration using more advanced matrix-free methods, such as the Lanczos algorithm or the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method.

Subsequent principal components can be computed one-by-one via deflation or simultaneously as a block. In the former approach, imprecisions in already computed approximate principal components additively affect the accuracy of the subsequently computed principal components, thus increasing the error with every new computation. The latter approach in the block power method replaces single-vectors $\mathbf{r}$ and $\mathbf{s}$ with block-vectors, matrices $\mathbf{R}$ and $\mathbf{S}$. Every column of $\mathbf{R}$ approximates one of the leading principal components, while all columns are iterated simultaneously. The main calculation is evaluation of the product $\mathbf{X}^\mathsf{T}(\mathbf{X}\ \mathbf{R})$. Implemented, for example, in LOBPCG, efficient blocking eliminates the accumulation of the errors, allows using high-level BLAS matrix-matrix product functions, and typically leads to faster convergence, compared to the single-vector one-by-one technique.

source: https://en.wikipedia.org/wiki/Principal_component_analysis