Scientific Computing

Maastricht Science Program

Week 2

Frans Oliehoek
<frans.oliehoek@maastrichtuniversity.nl>

Recap

- Overview scientific computing
  - exact science: mathematical models (precise understanding!)
  - (high-level) programming: Mathematica, Matlab
  - topics
- Introduction Mathematica
  - population models via difference equations (also 'recurrence equations' hence '**R**Solve') $p_{n+1} = f(p_n)$

This Lecture

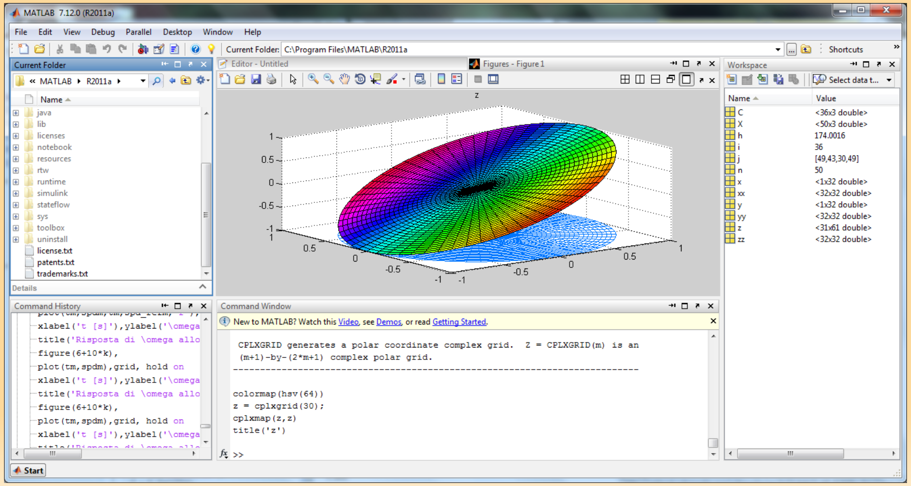
- A introduction to Matlab
- Principle Component Analysis
- Floating Point Numbers

Matlab

- 'Matrix' laboratory
- primarily: numerical computing

- learn by doing!
- different numbers:
  - no π or 'how many digits you like'
  - but: floating point numbers

- Octave



Principal Component Analysis

Dimension Reduction

- High dimensional data $(x_{1,} x_{2,} \dots, x_D)$
  - apple: weight, length, circumference, color, taste, etc.
- Hard to understand / visualize!
- Dimension reduction:
  - reduce the number of variables $(x_{1,} x_{2,} \dots, x_D) \rightarrow (z_{1,} z_{2,} \dots, z_d)$
  - i.e., reduce the number of dimensions from D to d

Dimension Reduction

- In the lab: measurements about Brachiopods

	A	B	C	D	E	F	G	H	I	J
1	0	1	2	60	1	16.55	13.6	8.05	16.2	0
2	0	1	2	60	1	12.95	11.7	5	11.6	0
3	1	0	0	0	0	30.3	30.8	23.2	26.1	1
4	1	0	0	0	0	31.5	35.35	24.9	21.35	0
5	0	2	0	0	0	31.6	29.4	21.25	31.5	0
6	0	2	0	0	0	18.35	18.95	12.75	16.25	0
7	1	1	1	8	0	31	35.45	20.2	26.4	0
8	1	1	1	8	0	29.85	31.95	21.8	25.7	0
9	1	1	0	0	0	26.9	23.8	15	24.65	0
10	1	1	0	0	0	30.2	26.75	17.8	26.4	0
11	1	0	2	14	0	17.15	16.6	10.55	15.3	0
12	1	0	2	16	1	15.9	12.9	10.2	13.45	0
13										
14										

PCA – Goals

$$N = n + 1$$

- Given a data set X of N data point of D variables
  $\rightarrow$ convert to data set Z of N data points of d variables

$$\left(x_1^{(0)}, x_2^{(0)}, \ldots, x_D^{(0)}\right) \rightarrow \left(z_1^{(0)}, z_2^{(0)}, \ldots, z_d^{(0)}\right)$$
$$\left(x_1^{(1)}, x_2^{(1)}, \ldots, x_D^{(1)}\right) \rightarrow \left(z_1^{(1)}, z_2^{(1)}, \ldots, z_d^{(1)}\right)$$
$$\ldots$$
$$\left(x_1^{(n)}, x_2^{(n)}, \ldots, x_D^{(n)}\right) \rightarrow \left(z_1^{(n)}, z_2^{(n)}, \ldots, z_d^{(n)}\right)$$

PCA – Goals

- Given a data set X of N data point of D variables
  $\rightarrow$ convert to data set Z of N data points of d variables

$$\left(x_1^{(0)}, x_2^{(0)}, \dots, x_D^{(0)}\right) \rightarrow \left(z_1^{(0)}, z_2^{(0)}, \dots, z_d^{(0)}\right)$$
$$\left(x_1^{(1)}, x_2^{(1)}, \dots, x_D^{(1)}\right) \rightarrow \left(z_1^{(1)}, z_2^{(1)}, \dots, z_d^{(1)}\right)$$
$$\dots$$
$$\left(x_1^{(n)}, x_2^{(n)}, \dots, x_D^{(n)}\right) \rightarrow \left(z_1^{(n)}, z_2^{(n)}, \dots, z_d^{(n)}\right)$$

The vector $\left(z_i^{(0)}, z_i^{(1)}, \dots, z_i^{(n)}\right)$

is called the *i*-th **principal component** (of the data set)

PCA – Goals

- Given a data set X of N data point of D variables
  $\rightarrow$ convert to data set Z of N data points of d variables

$$\left(x_1^{(0)}, x_2^{(0)}, \ldots, x_D^{(0)}\right) \rightarrow \left(z_1^{(0)}, z_2^{(0)}, \ldots, z_d^{(0)}\right)$$
$$\left(x_1^{(1)}, x_2^{(1)}, \ldots, x_D^{(1)}\right) \rightarrow \left(z_1^{(1)}, z_2^{(1)}, \ldots, z_d^{(1)}\right)$$
$$\ldots$$
$$\left(x_1^{(n)}, x_2^{(n)}, \ldots, x_D^{(n)}\right) \rightarrow \left(z_1^{(n)}, z_2^{(n)}, \ldots, z_d^{(n)}\right)$$

The vector $\left(z_i^{(0)}, z_i^{(1)}, \ldots, z_i^{(n)}\right)$

is called the *i*-th **principal component** (of the data set)

- PCA performs a **linear** transformation:
  $\rightarrow$ variables z_i are linear combinations of $x_1,\ldots,x_D$

PCA Goals – 2

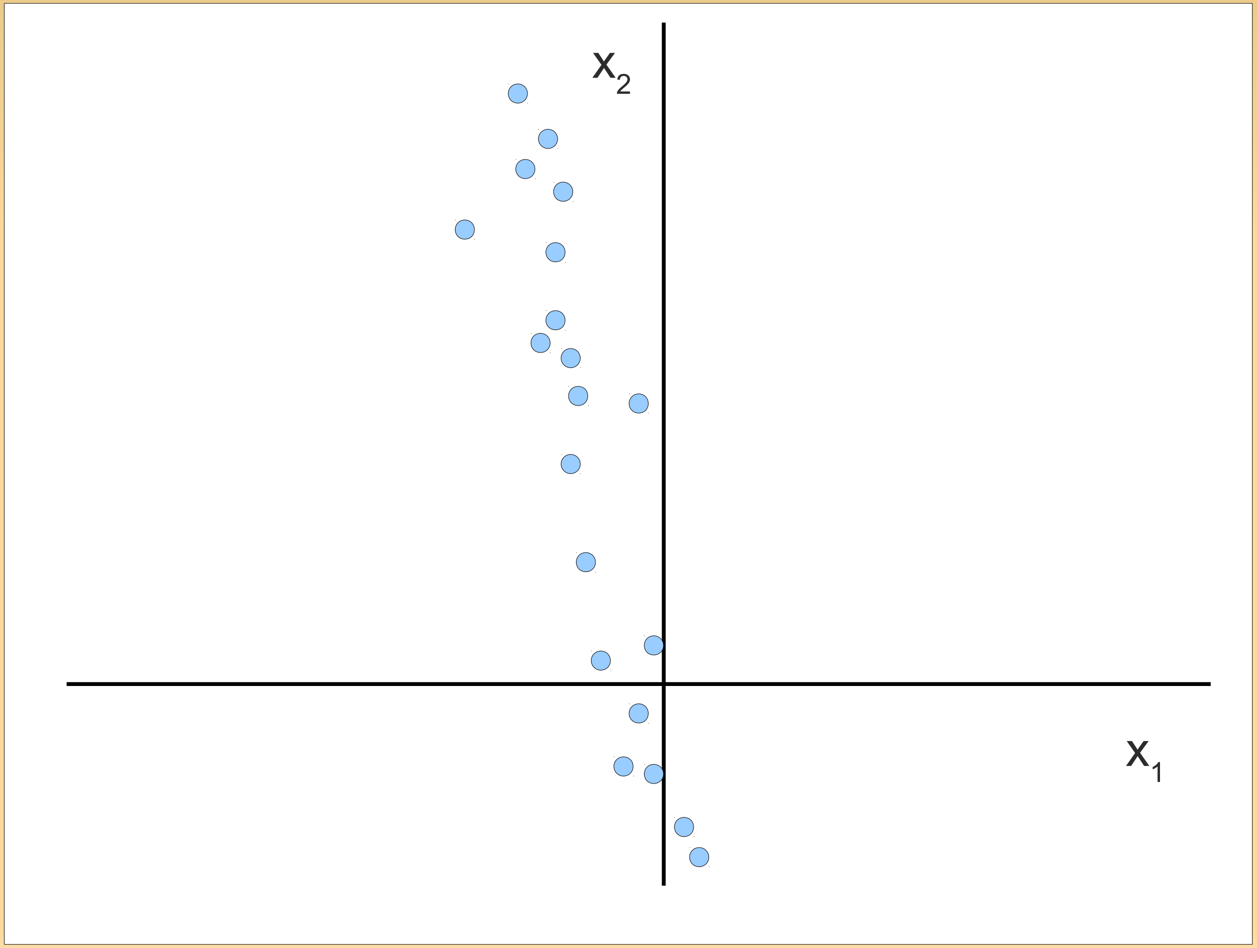
- Of course many possible transformations possible...
  - Reducing the number of variables: loss of information
  - PCA makes this loss minimal

- PCA is very useful
  - Exploratory analysis of the data
  - Visualization of high-D data
  - Data preprocessing
  - Data compression

PCA – Intuition

- How would you summarize this data using 1 dimension?

  (what variable contains the most information?)



PCA – Intuition

- How would you summarize this data using 1 dimension?

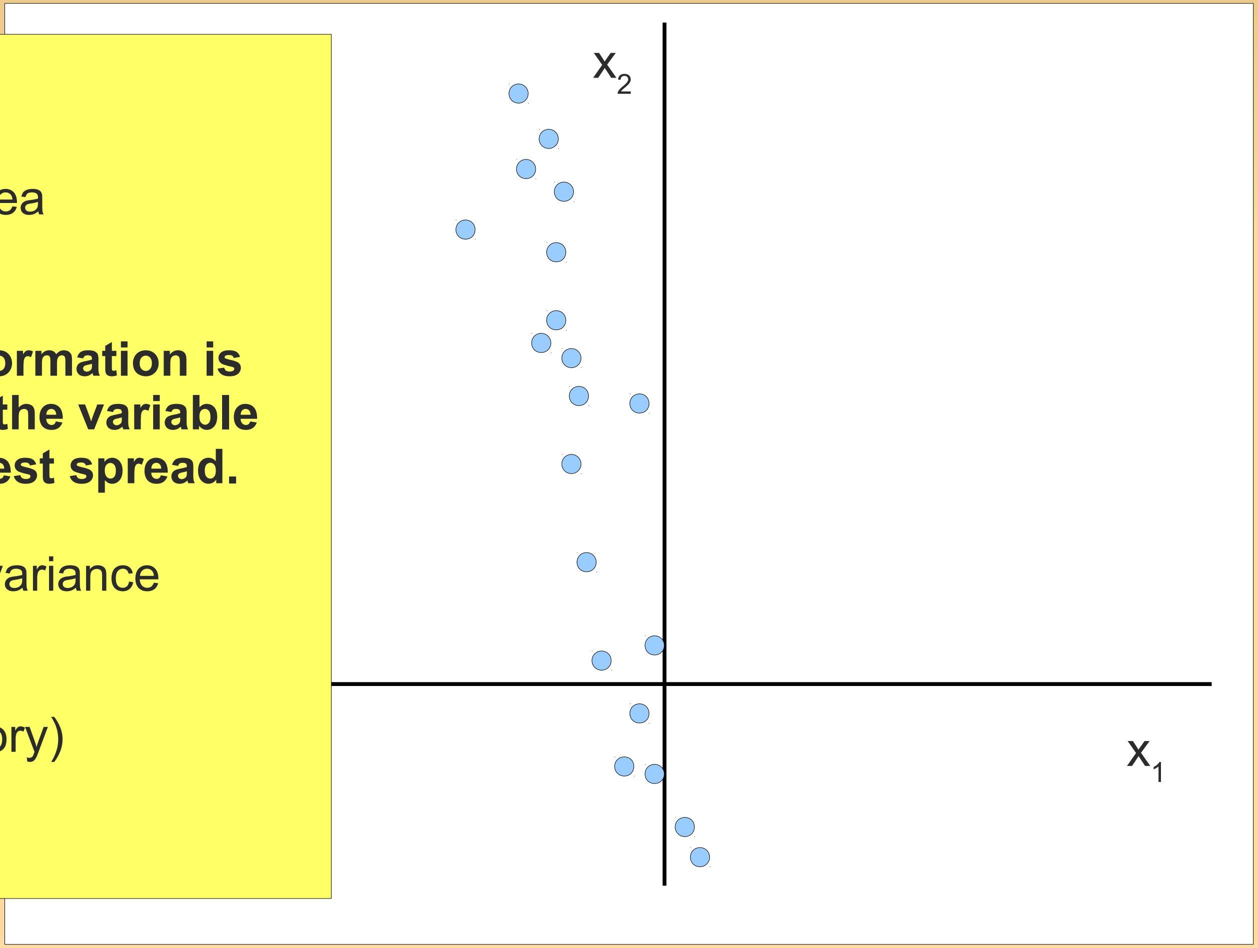
  (what variable contains the most information?)

Very important idea

The most information is contained by the variable with the largest spread.

- i.e., highest variance

(Information Theory)



PCA – Intuition

- How would you summarize this data using 1 dimension?

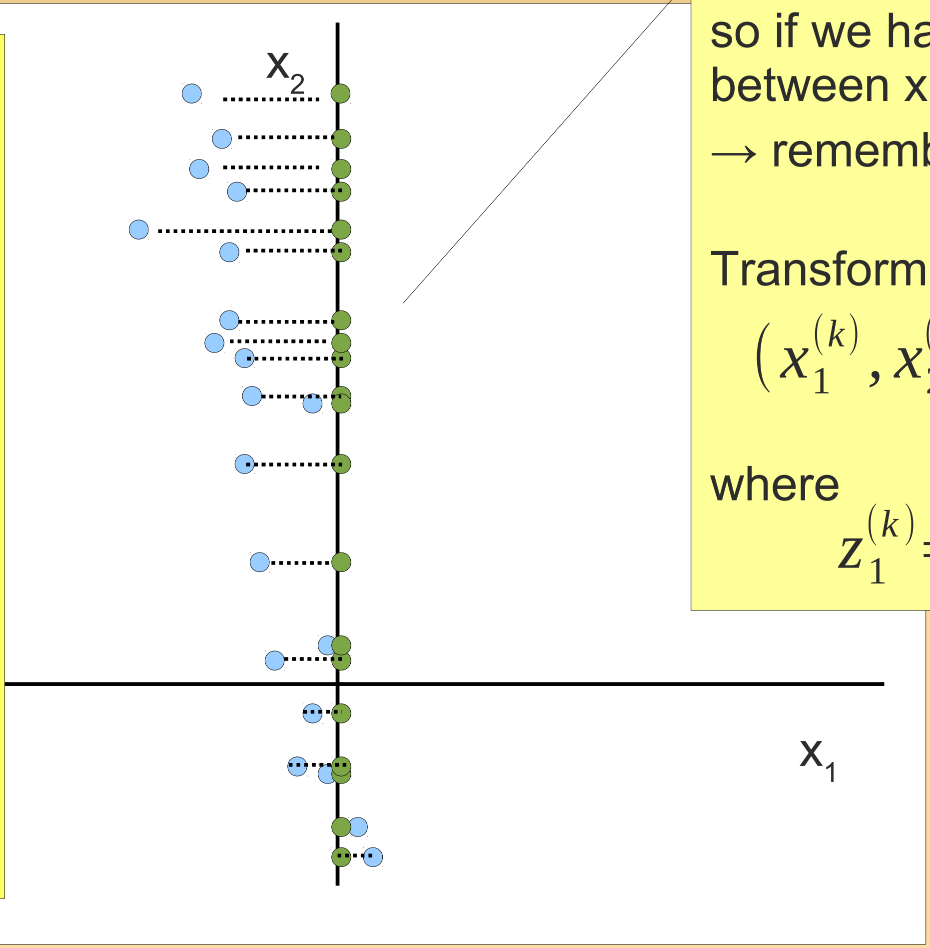
  (what variable contains the most information?)

Very important idea

The most information is contained by the variable with the largest spread.

- i.e., highest variance

(Information Theory)



so if we have to chose between x_1 and x_2
→ remember x_2

Transform of *k*-th point:

$$\left(x_1^{(k)}, x_2^{(k)}\right) \rightarrow \left(z_1^{(k)}\right)$$

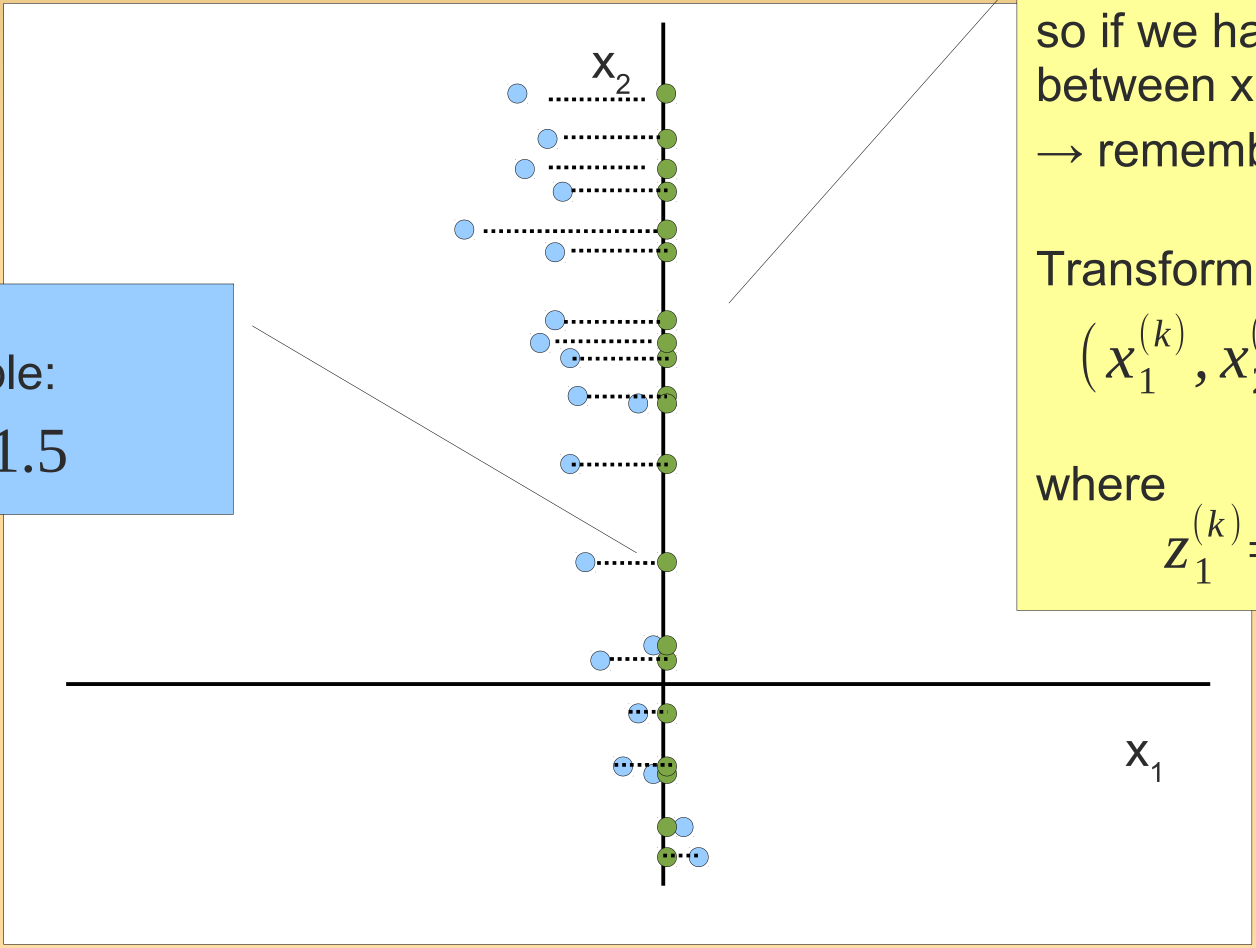
where

$$z_1^{(k)} = x_2^{(k)}$$

PCA – Intuition

- How would you summarize this data using 1 dimension?

  (what variable contains the most information?)



Example:

$z_1^{(k)} = 1.5$

so if we have to chose between x_1 and x_2
→ remember x_2

Transform of *k*-th point:

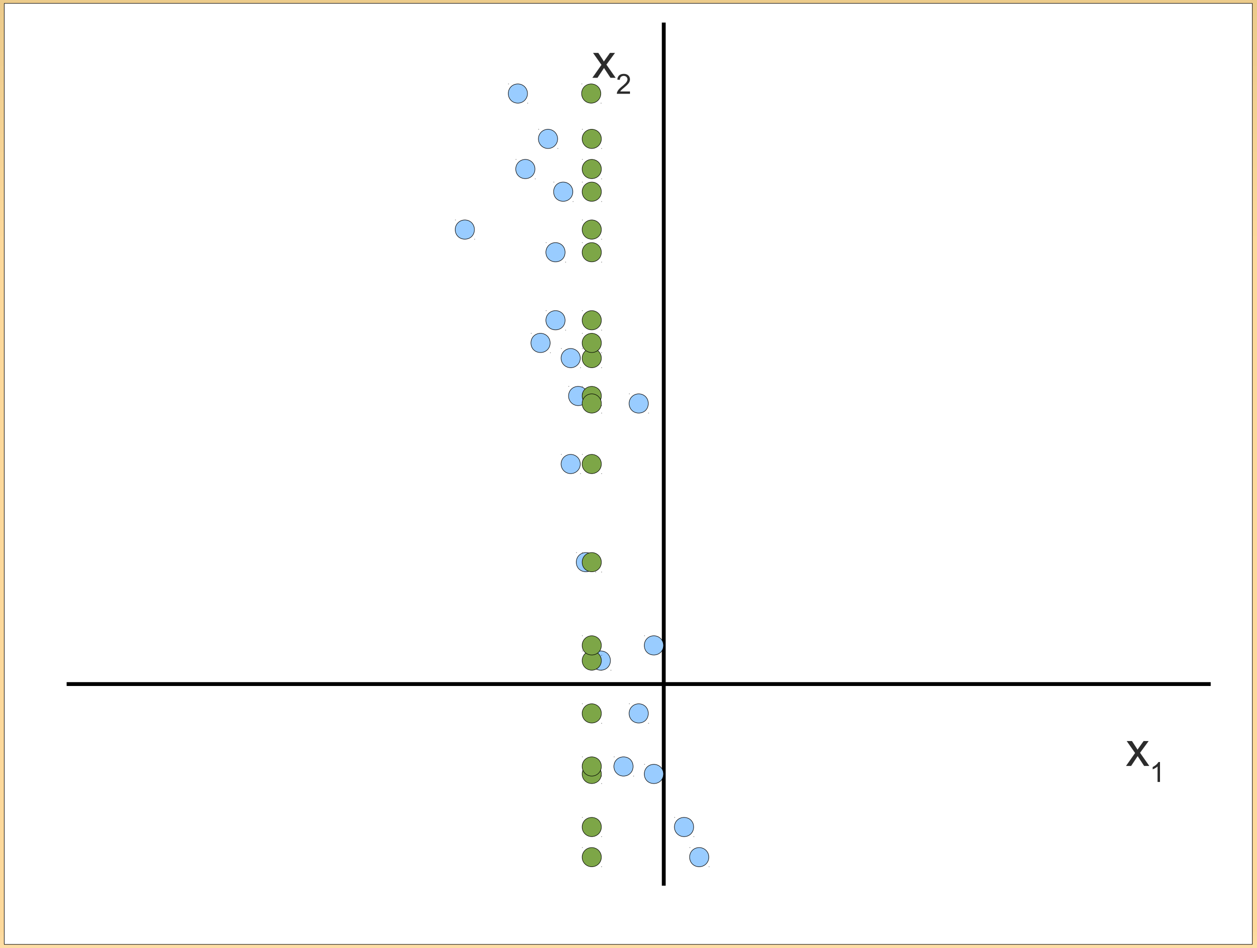
$$\left(x_1^{(k)}, x_2^{(k)}\right) \rightarrow \left(z_1^{(k)}\right)$$

where

$$z_1^{(k)} = x_2^{(k)}$$

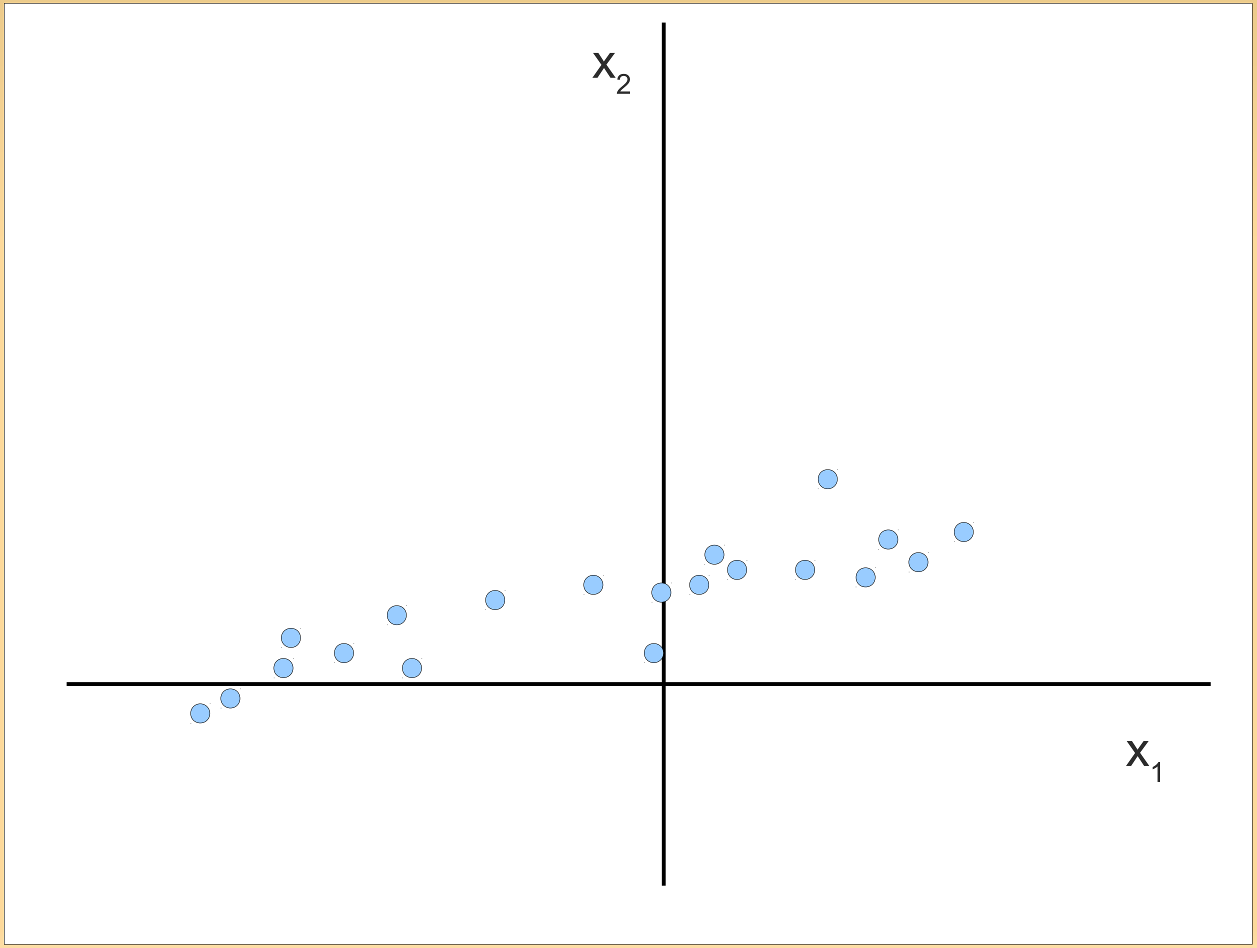
PCA – Intuition

- Reconstruction based on x_2
  $\rightarrow$ only need to remember mean of x_1



PCA – Intuition

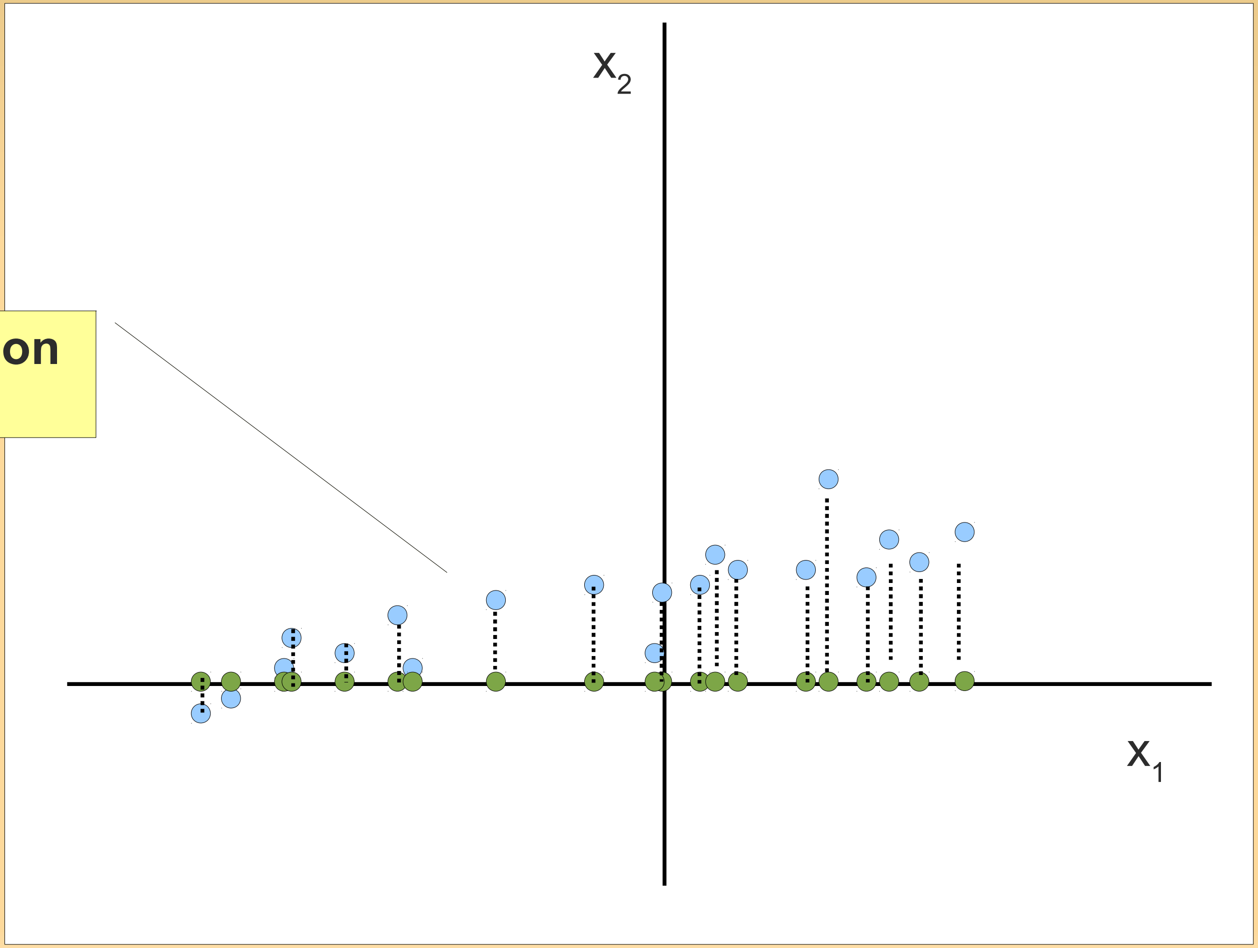
- How would you summarize this data using 1 dimension?



PCA – Intuition

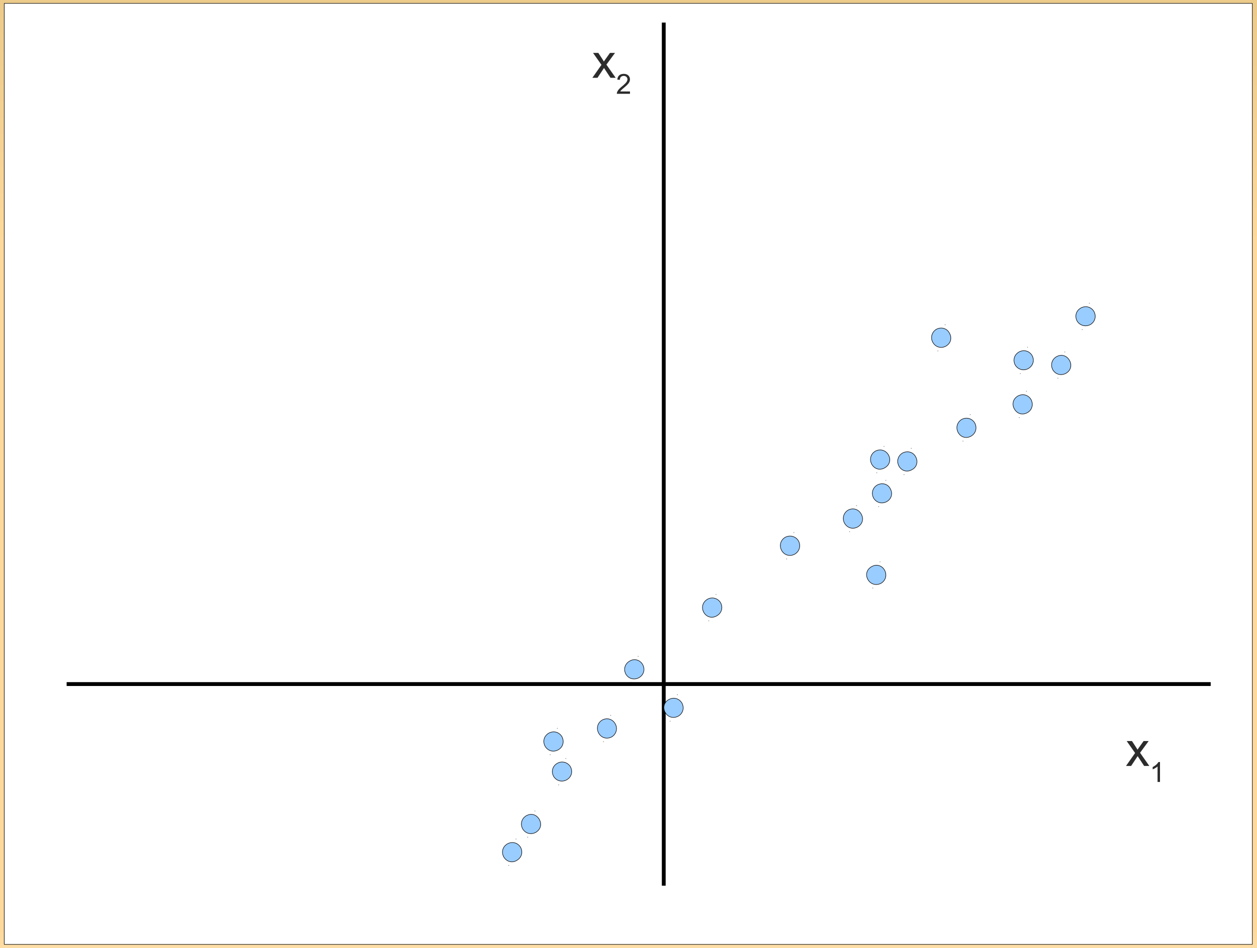
- How would you summarize this data using 1 dimension?

This is a **projection** on the x1 axis.



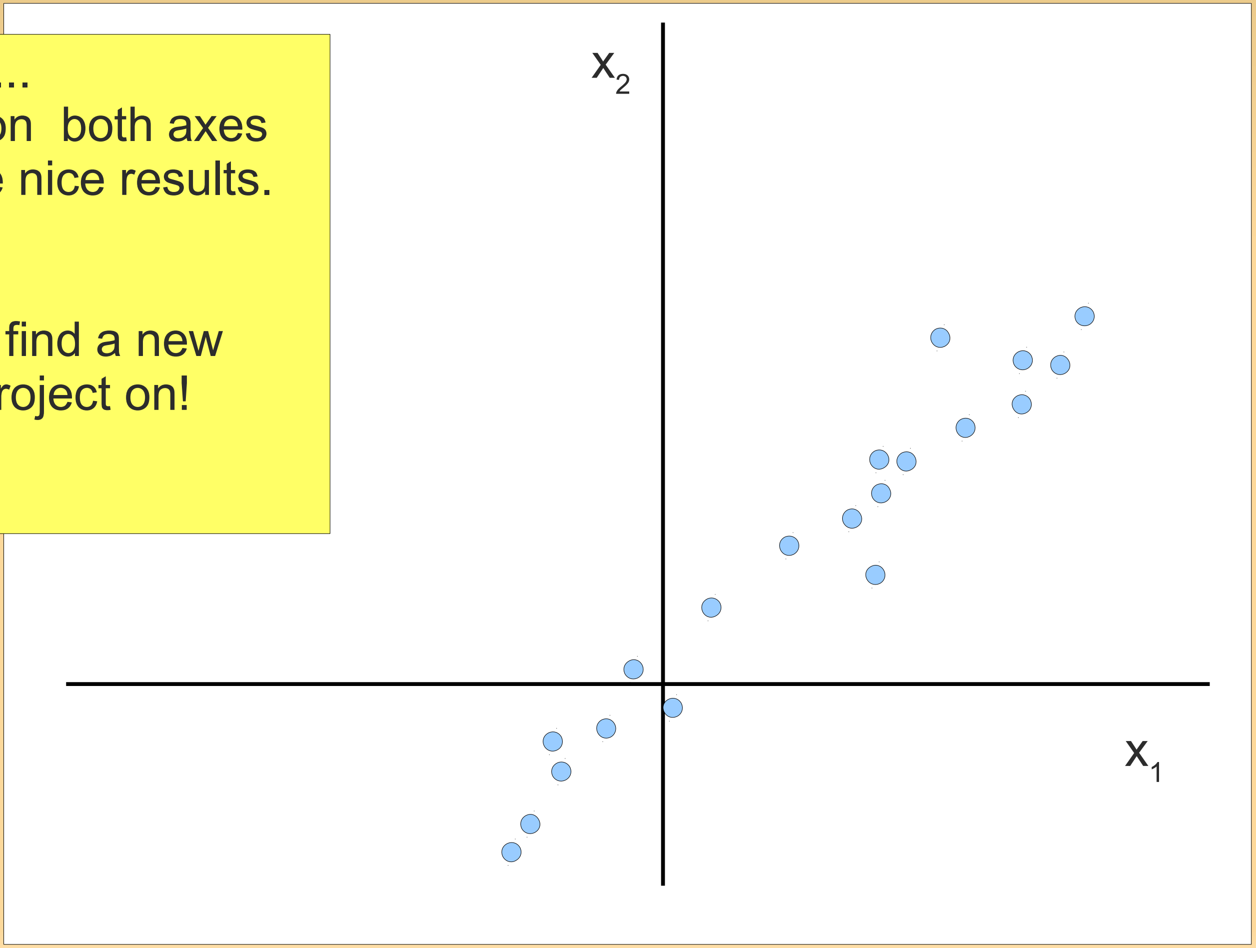
PCA – Intuition

- How would you summarize this data using 1 dimension?



PCA – Intuition

- How would you summarize this data using 1 dimension?



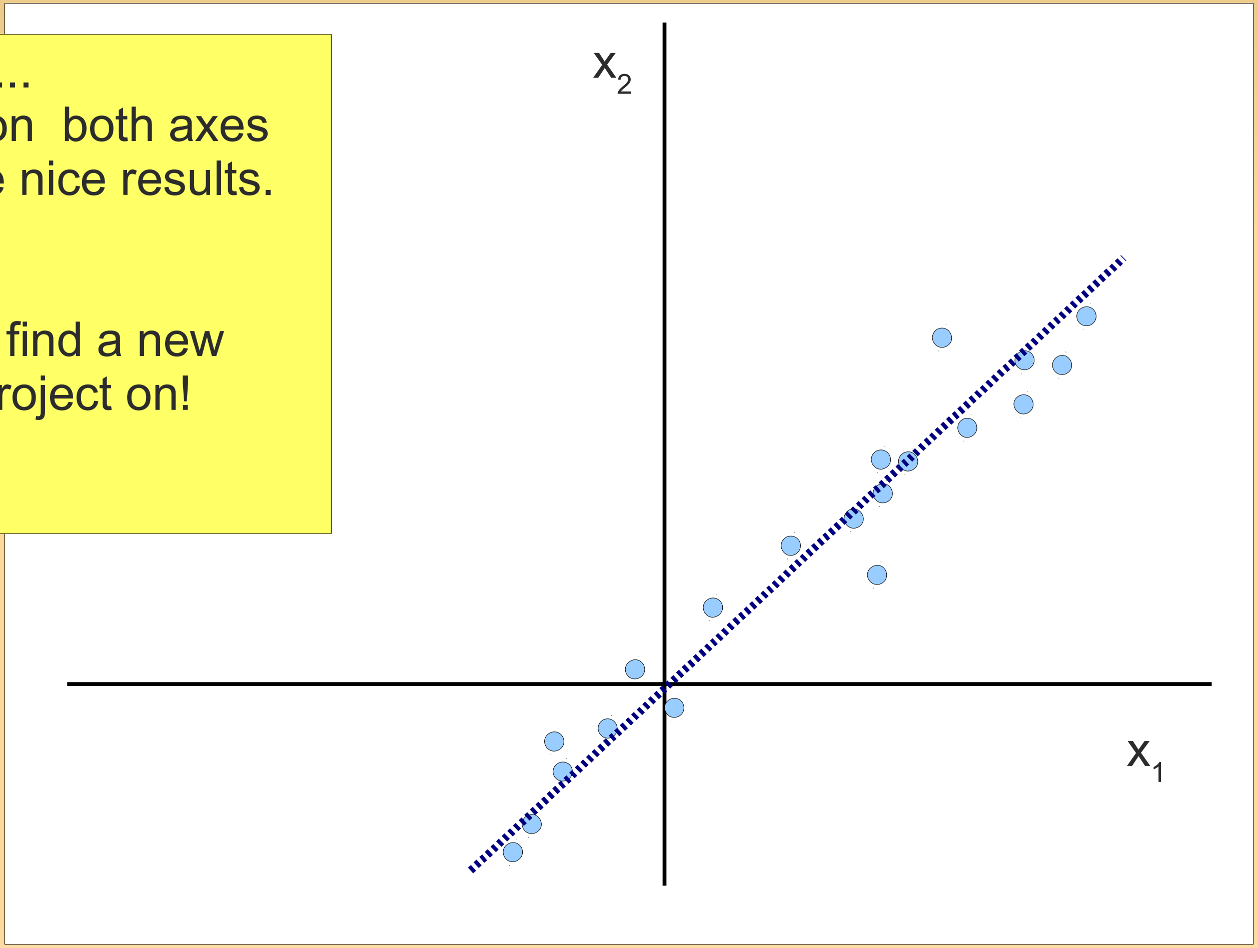
- More difficult...
  ...projection on both axes does not give nice results.
- Idea of PCA: find a new direction to project on!

PCA – Intuition

- How would you summarize this data using 1 dimension?

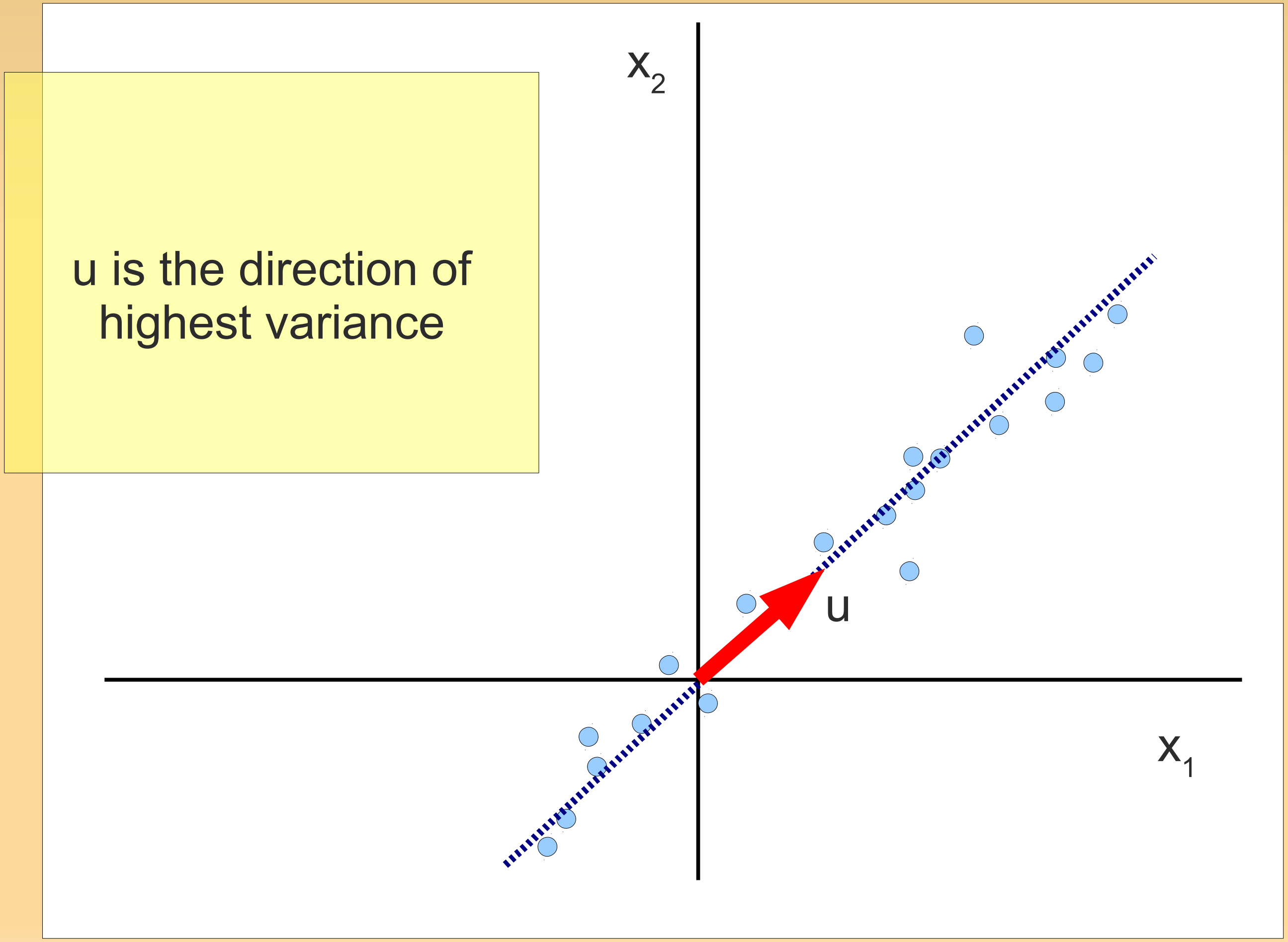
- More difficult...
  ...projection on both axes does not give nice results.

- Idea of PCA: find a new direction to project on!



PCA – Intuition

- How would you summarize this data using 1 dimension?



PCA – Intuition

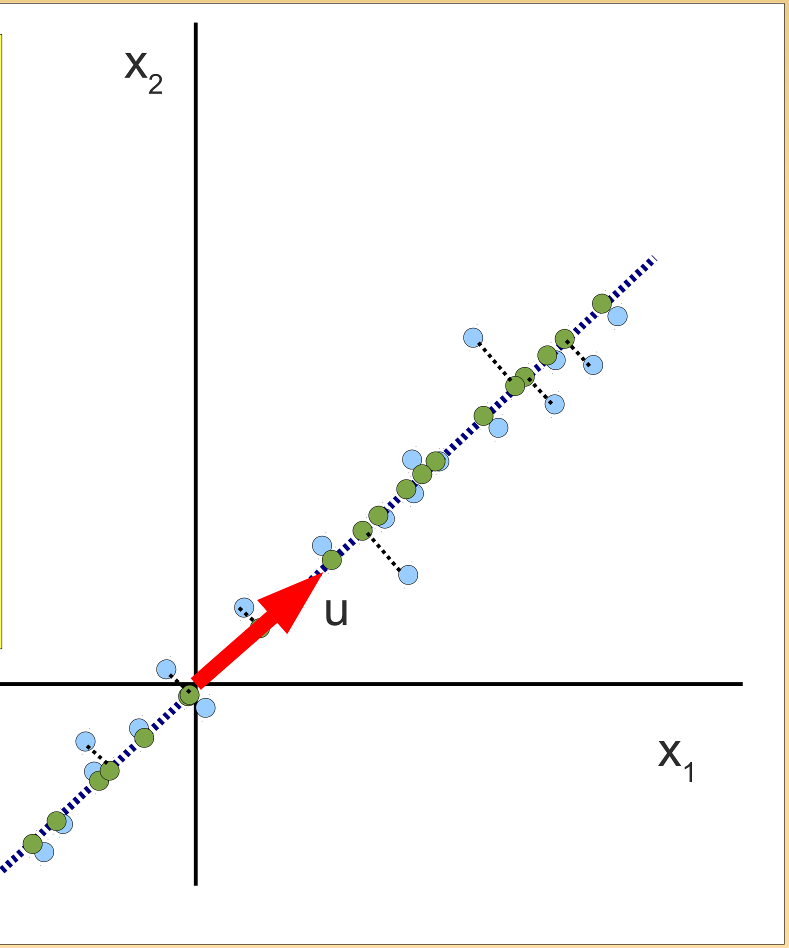
- How would you summarize this data using 1 dimension?

Transform of *k*-th point:

$$\left(x_1^{(k)}, x_2^{(k)}\right) \rightarrow \left(z_1^{(k)}\right)$$

where z_1 is the **orthogonal scalar projection** on u:

$$z_1^{(k)} = u_1 x_1^{(k)} + u_2 x_2^{(k)} = \left(u, x^{(k)}\right)$$



PCA – Intuition

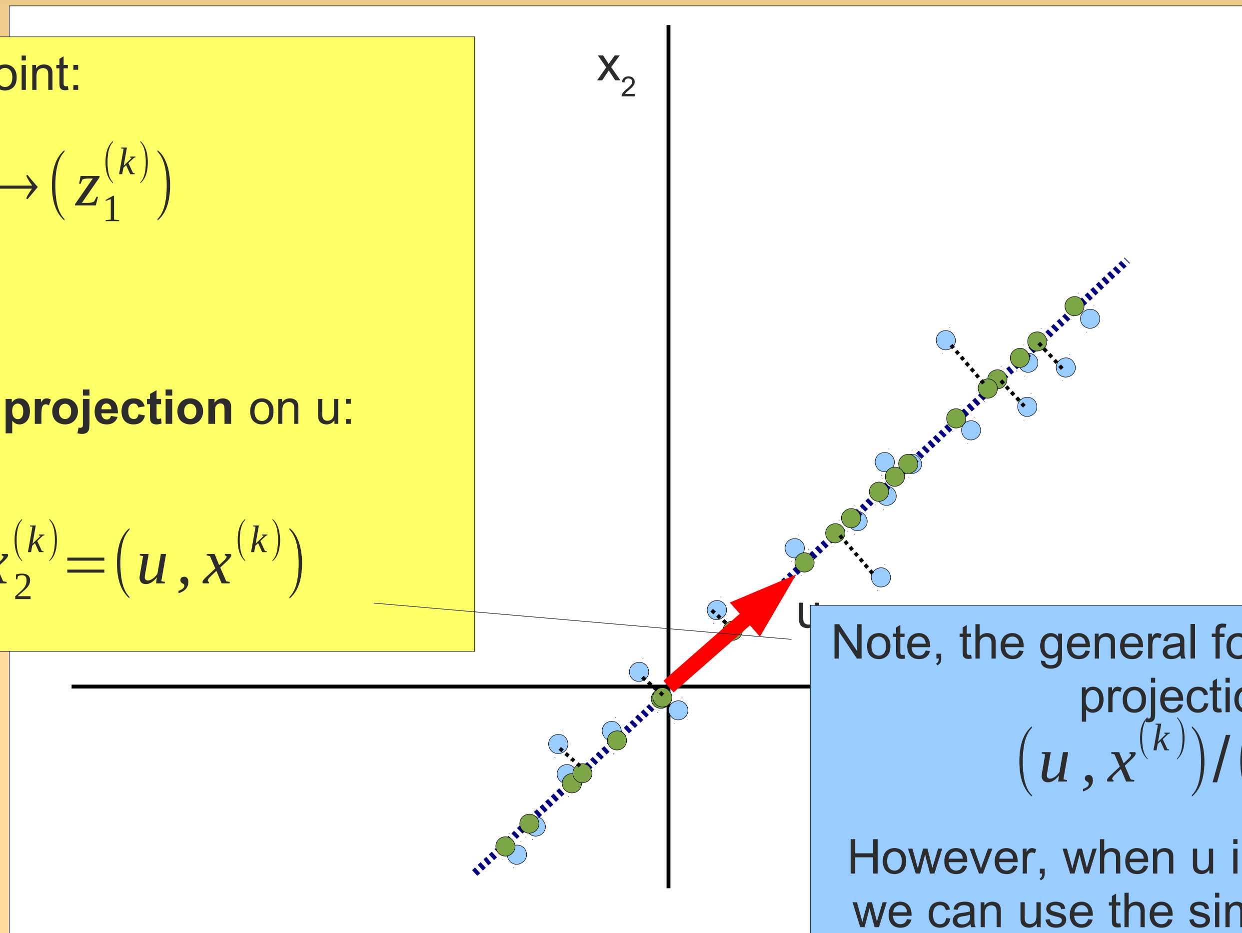
- How would you summarize this data using 1 dimension?

Transform of *k*-th point:

$$\left(x_1^{(k)}, x_2^{(k)}\right) \rightarrow \left(z_1^{(k)}\right)$$

where z_1 is the
orthogonal scalar projection on u:

$$z_1^{(k)} = u_1 x_1^{(k)} + u_2 x_2^{(k)} = \left(u, x^{(k)}\right)$$



Note, the general formula for scalar projection is

$$\left(u, x^{(k)}\right) / \left(u, u\right)$$

However, when u is a unit vector, we can use the simplified formula

PCA – Intuition

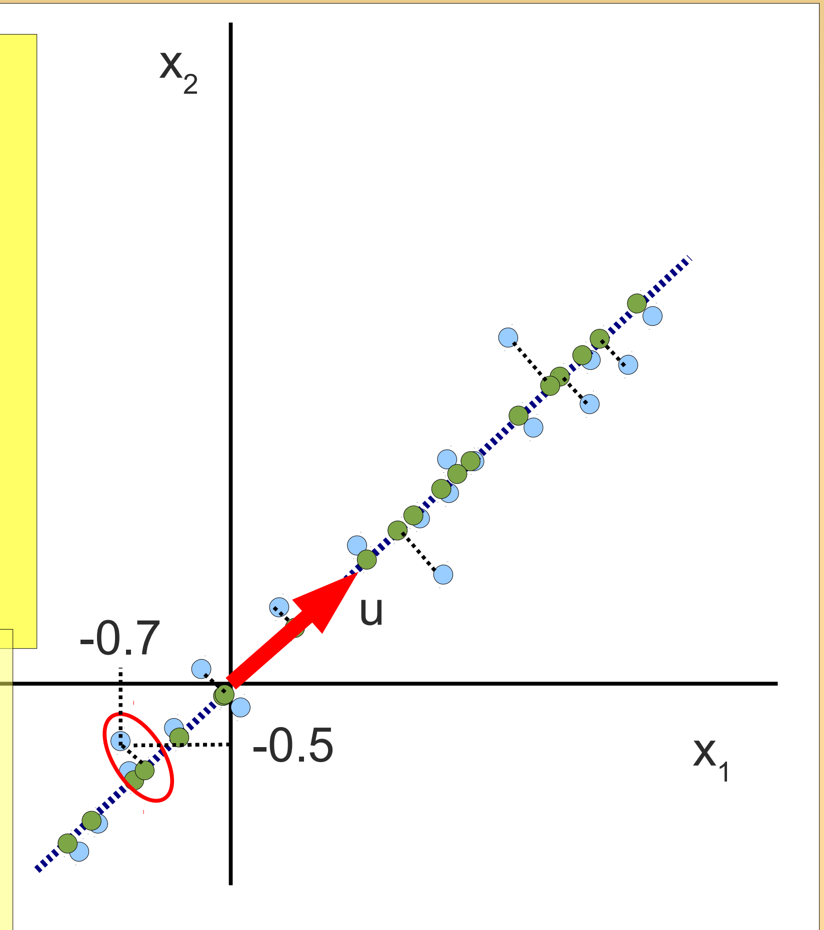
- How would you summarize this data using 1 dimension?

Transform of *k*-th point:

$$\left(x_1^{(k)}, x_2^{(k)}\right) \rightarrow \left(z_1^{(k)}\right)$$

where z_1 is the **orthogonal scalar projection** on u:

$$z_1^{(k)} = u_1 x_1^{(k)} + u_2 x_2^{(k)} = \left(u, x^{(k)}\right)$$

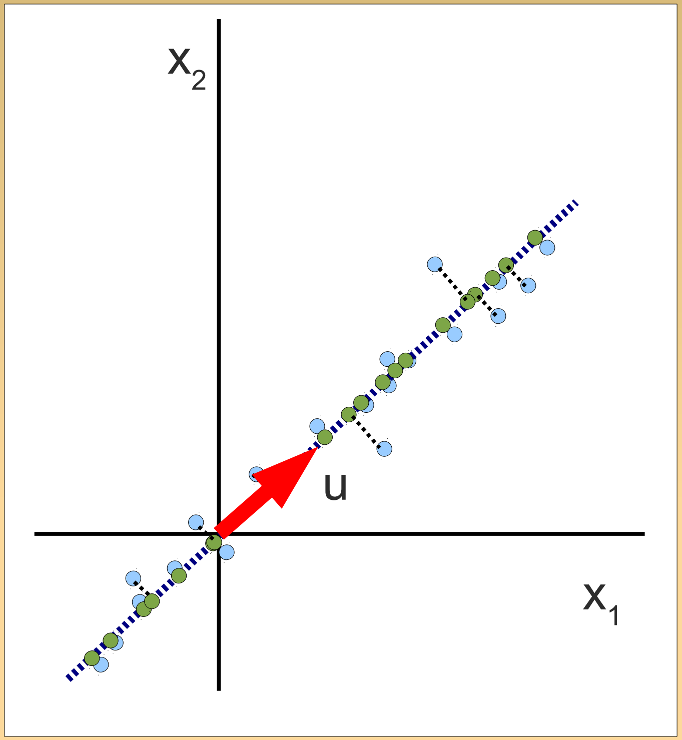


E.g.:

$$z_1 = 0.7(-0.7) + 0.7(-.5) = -0.84$$

is the first principal component of this data point

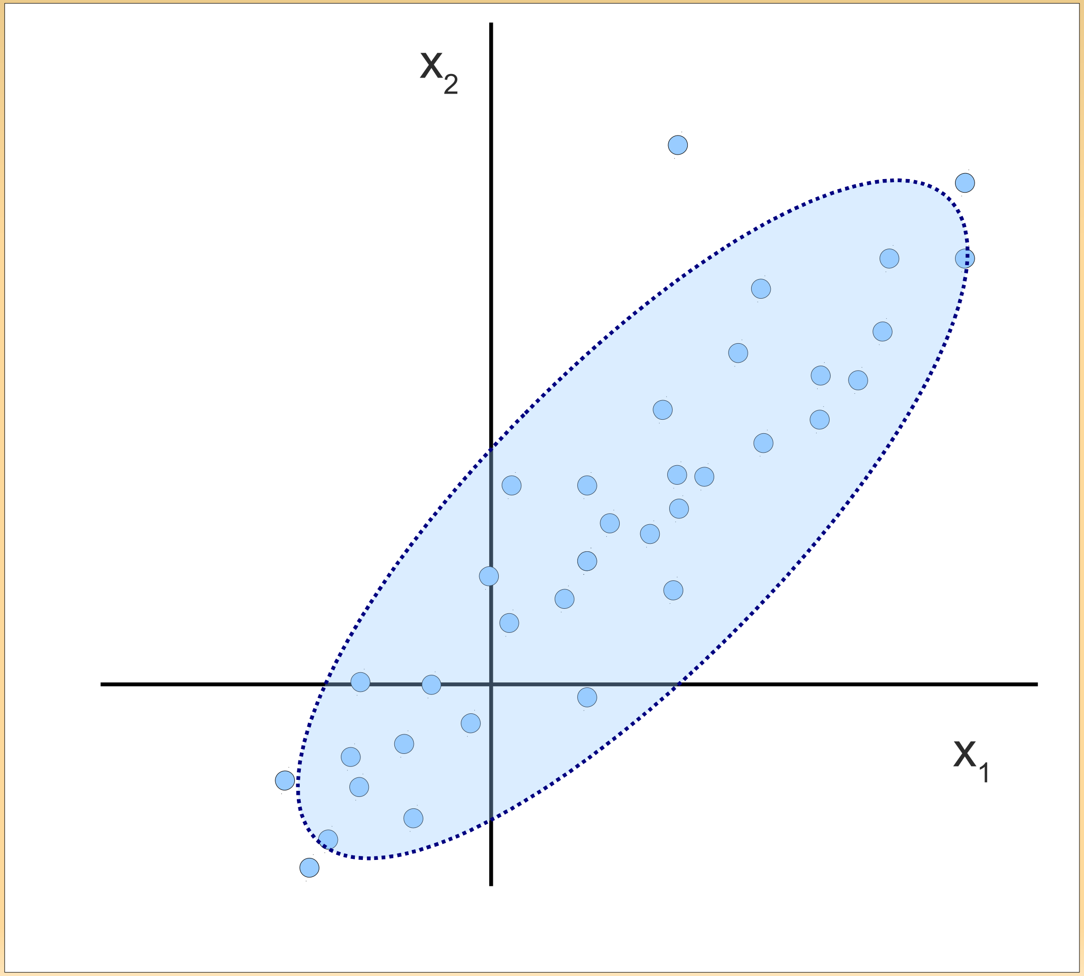
PCA – Intuition



- PCA so far...
  - find the direction *u* of highest variance
  - project data on $u \rightarrow z_1$ the **first** principle component (PC)
- Next...
  - find **more directions** of high variance
    $\rightarrow$ u is $u^{(1)}$, the direction of the first PC
    $\rightarrow$ find $u^{(2)}$, $u^{(3)}$,..., $u^{(D)}$
    (the directions of the other PCs)

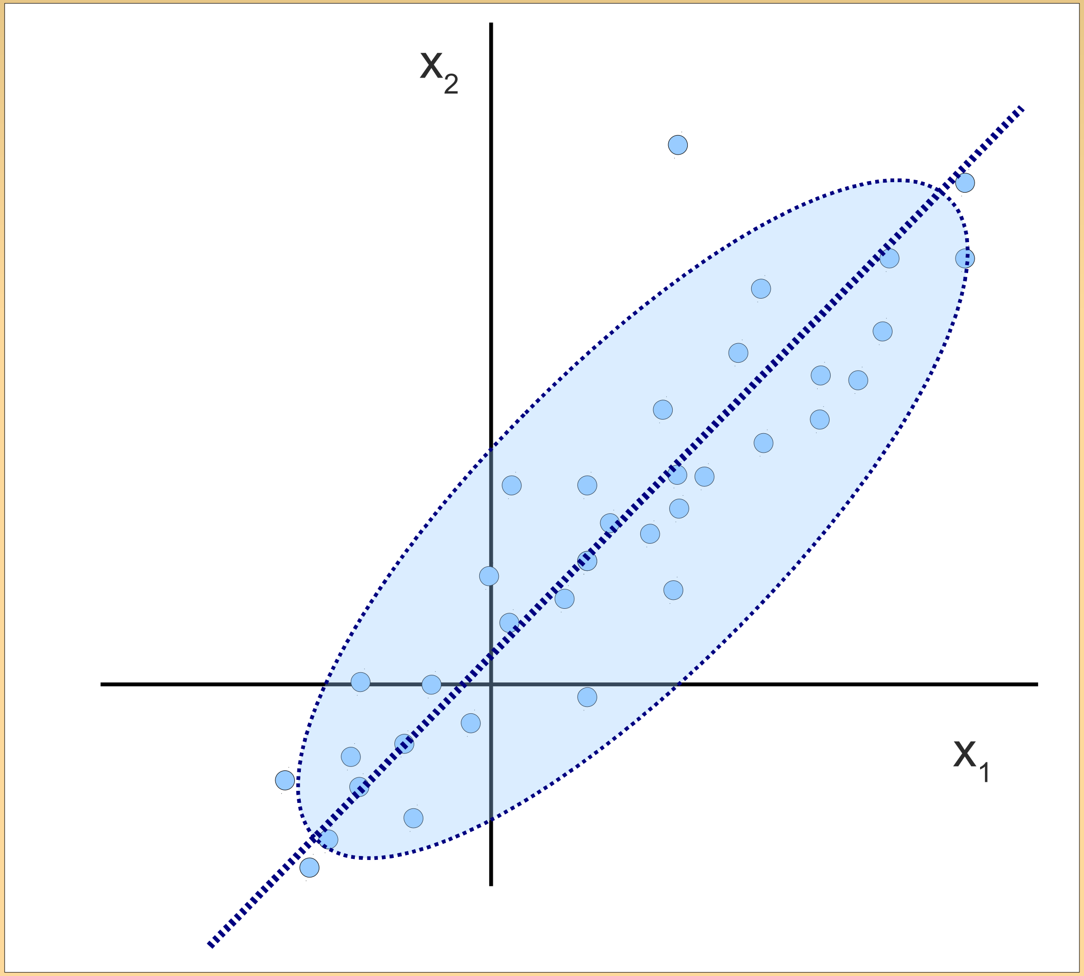
More Principle Components

- Given this data, what is $u^{(1)}$?
  (i.e., the direction of the first PC)



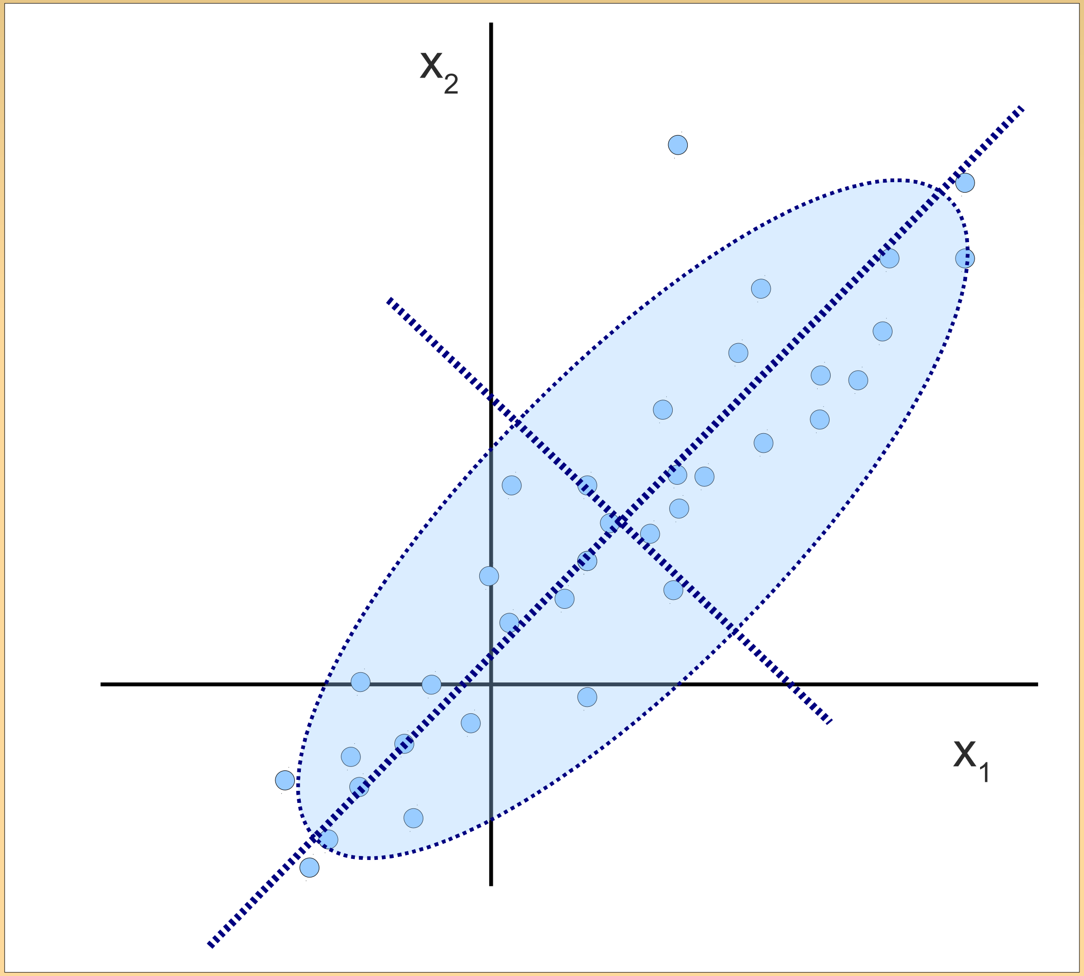
More Principle Components

- $u^{(1)}$ explains the most variance
- What is $u^{(2)}$? (the direction of the 2nd PC) ?



More Principle Components

- $u^{(2)}$ is the direction with most 'remaining' variance
  - orthogonal to $u^{(1)}$!



- Data is 2D, so can find only two directions
- Each point $x^{(k)}$ can be converted to $z^{(k)}$

$$\left(x_1^{(k)}, x_2^{(k)}\right) \Leftrightarrow \left(z_1^{(k)}, z_2^{(k)}\right)$$

$$z_i^{(k)} = \left(u^{(i)}, x^{(k)}\right)$$

Floating Point Numbers

How are number represented?

- Matlab represents numbers using a **floating point representation**

$$(-1)\cdot(0.b_1 b_2 \ldots b_{53})\cdot 2^e$$

sign

mantissa
- 53 bits - 2^{53}=9.0072e+15
- normalized: b_1 ~= 0

  (unique representation)

Exponent
-1021< e < 1024

How are number represented?

- Matlab represents numbers using a **floating point representation**

$$(-1)\cdot(0.b_1 b_2 \dots b_{53})\cdot 2^e$$

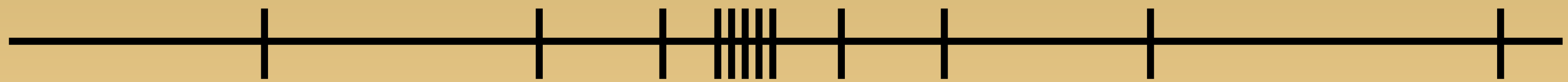
sign

mantissa
•53 bits - 2^{53}=9.0072e+15
•normalized: b_1 ~= 0
(unique representation)

Exponent
-1021< e < 1024

- Smallest
  - normalized $(0.100\dots00)\cdot 2^{-1021} = 2.2251\text{e-}308$
  - non-norm. $(0.000\dots01)\cdot 2^{-1021} = 4.9407\text{e-}324$
- Largest $(0.111\dots11)\cdot 2^{1024} = 1.7977\text{e+}308$

Spacing between numbers



- Spacing for the largest numbers

$$\left(0.000\ldots001\right)\cdot 2^{1024}$$

$$\left(0.000\ldots010\right)\cdot 2^{1024}$$

$$diff = \left(0.000\ldots001\right)\cdot 2^{1024} = 1\cdot 2^{\left(1024-53\right)} = 1.9958\text{e+}292$$

- Spacing for smallest numbers 4.9407e-324

- “eps(n)” gives spacing around n
  - eps(realmax), eps(0)

Round Off Errors

- set of floating point numbers *F*
- when real number *x* is replaced by number *fl(x)* in *F*

  $\rightarrow$ round off error

- Absolute error can be large: 0.5 *eps(realmax)

- However: *relative error* is bounded $$\frac{|x - fl(x)|}{|x|} \leqslant \frac{1}{2}\epsilon$$
  - where $\epsilon = eps(1) = 2.2204\text{e-}16$