Carnegie Mellon University
Heinz College

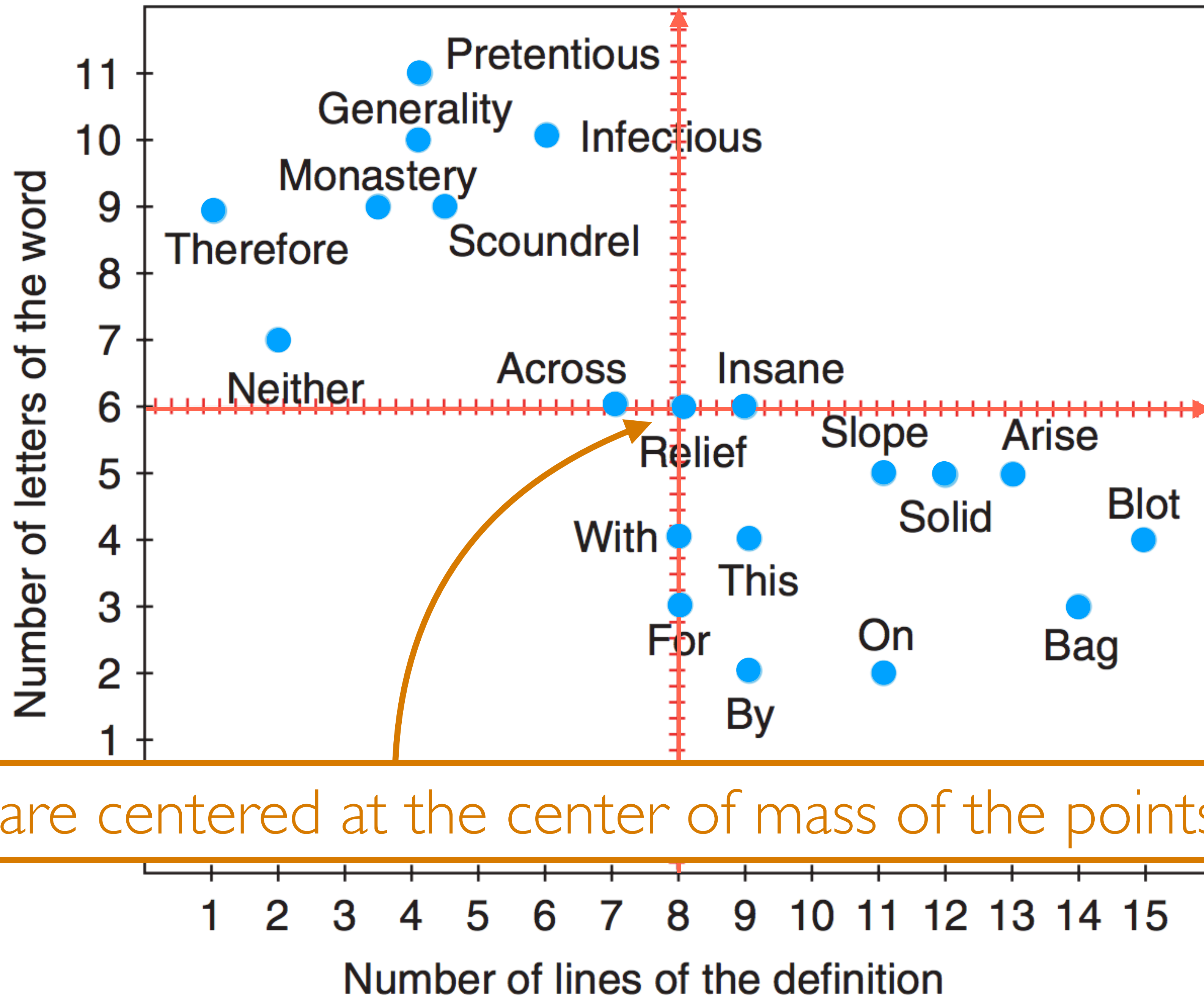
Unstructured Data Analytics for Policy

Lecture 4: PCA, manifold learning

George Chen

Principal Component Analysis (PCA)

How to project 2D data down to 1D?

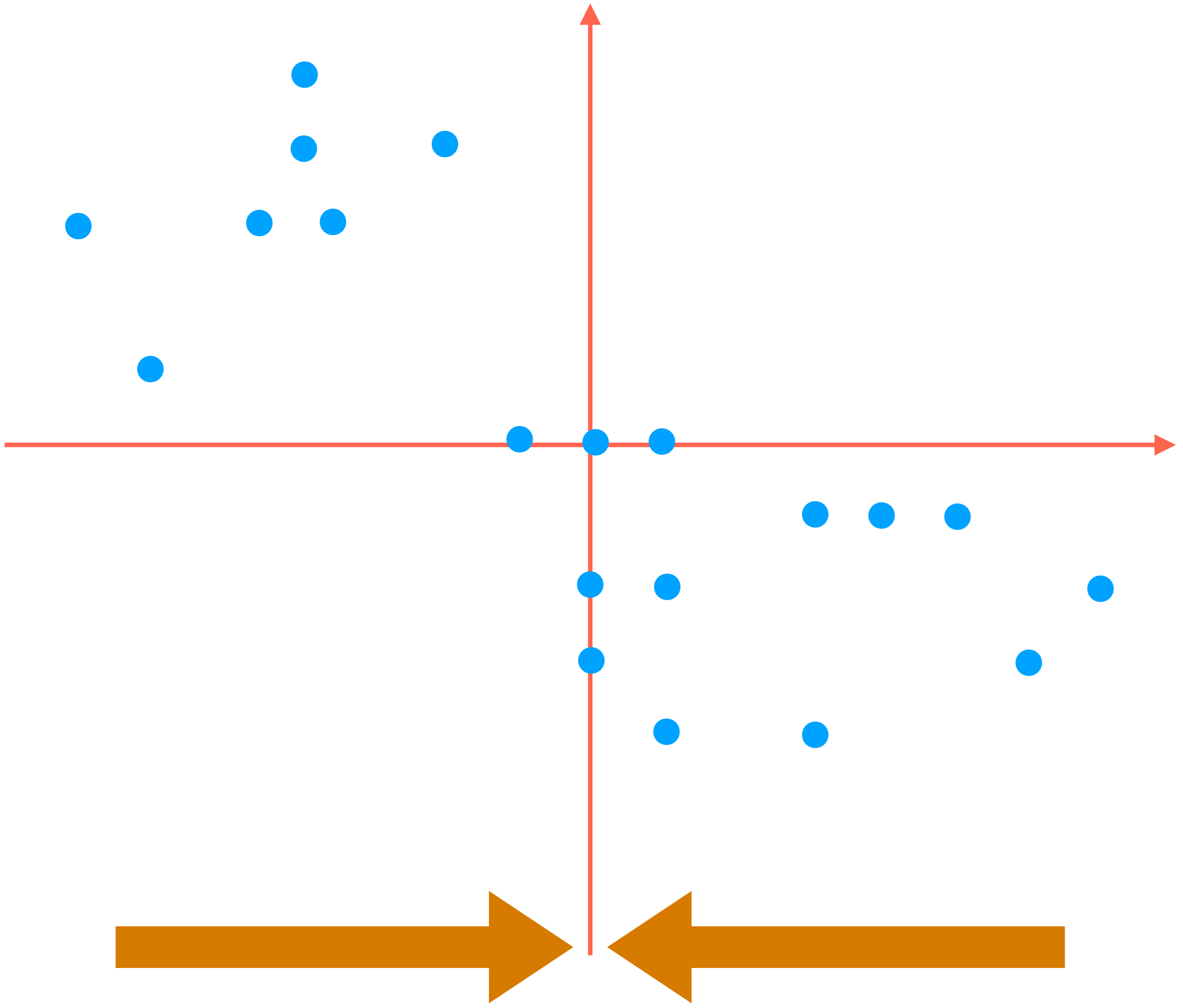


Note: axes are centered at the center of mass of the points and not at (0, 0)

Hervé Abdi and Lynne J. Williams. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics. 2010.

Principal Component Analysis (PCA)

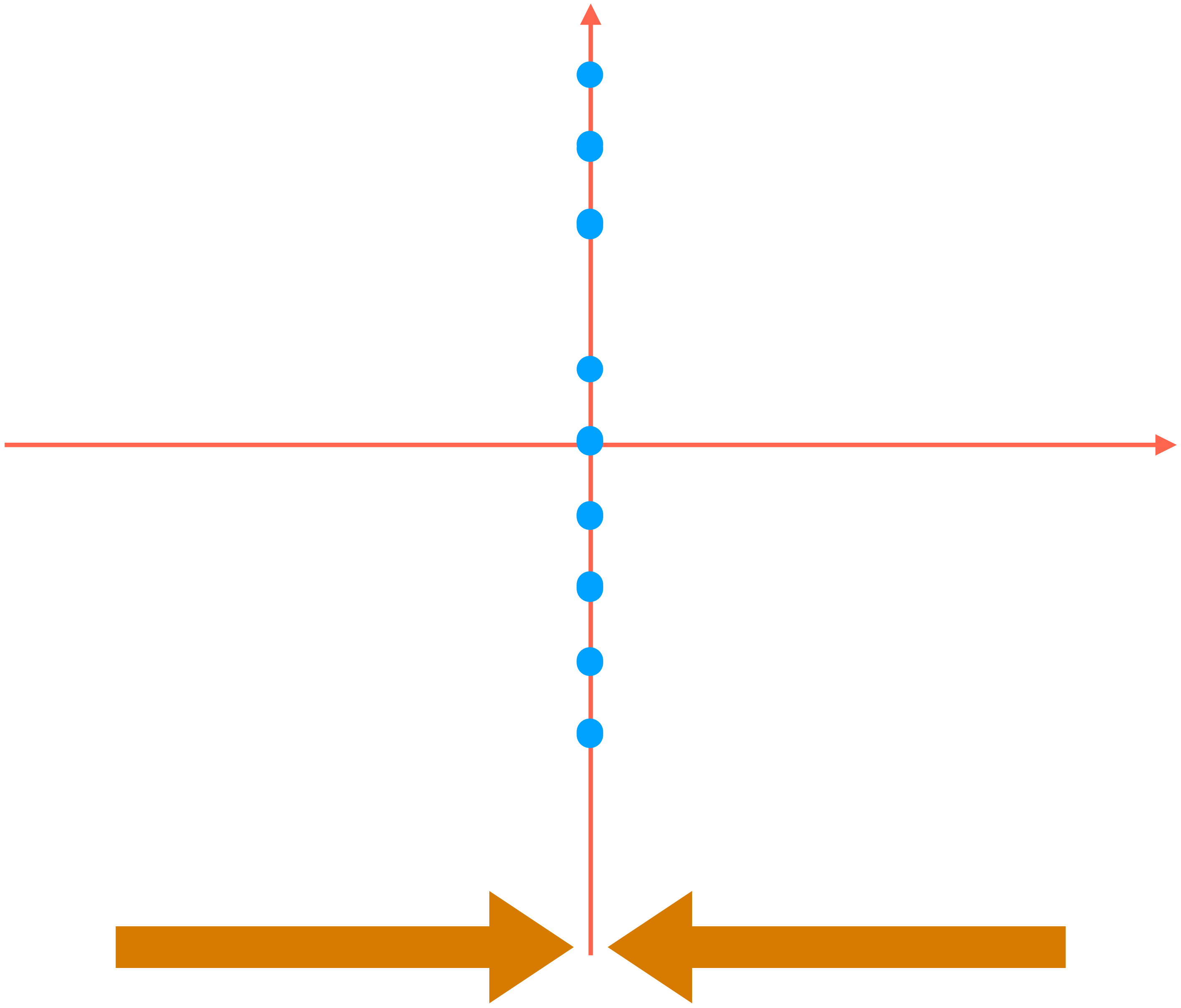
How to project 2D data down to 1D?



Simplest thing to try: flatten to one of the red axes

Principal Component Analysis (PCA)

How to project 2D data down to 1D?

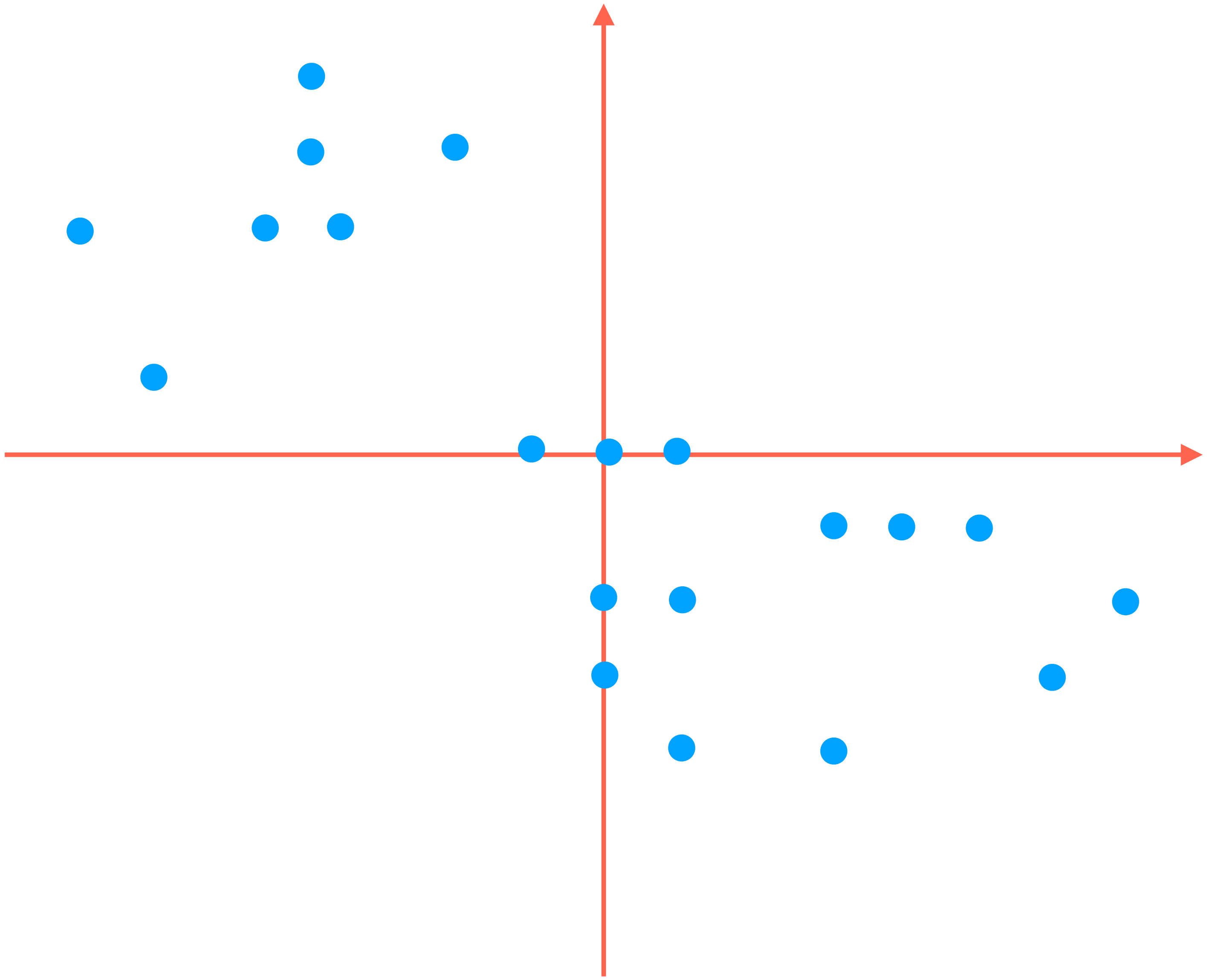


Simplest thing to try: flatten to one of the red axes

(We could of course flatten to the other red axis)

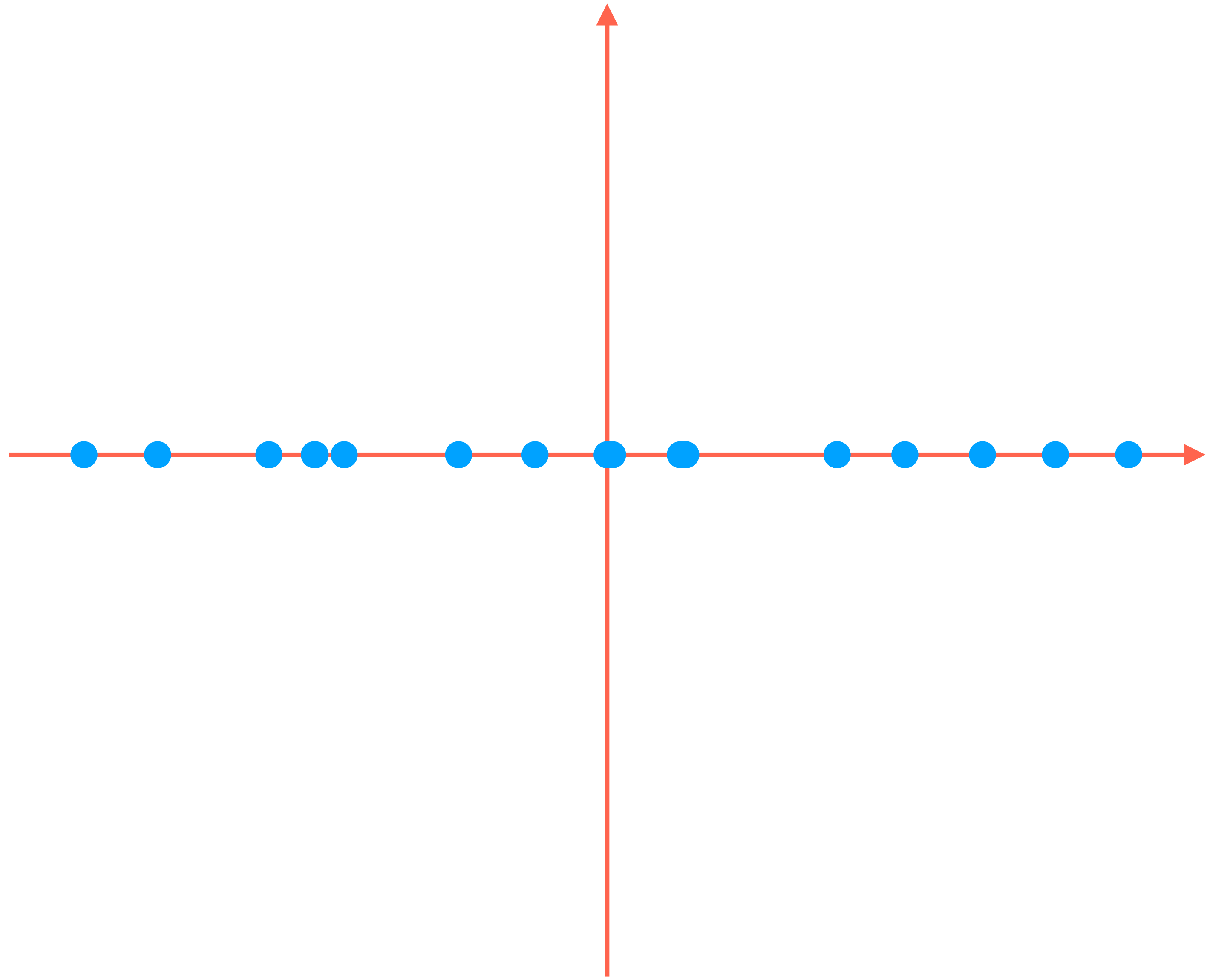
Principal Component Analysis (PCA)

How to project 2D data down to 1D?



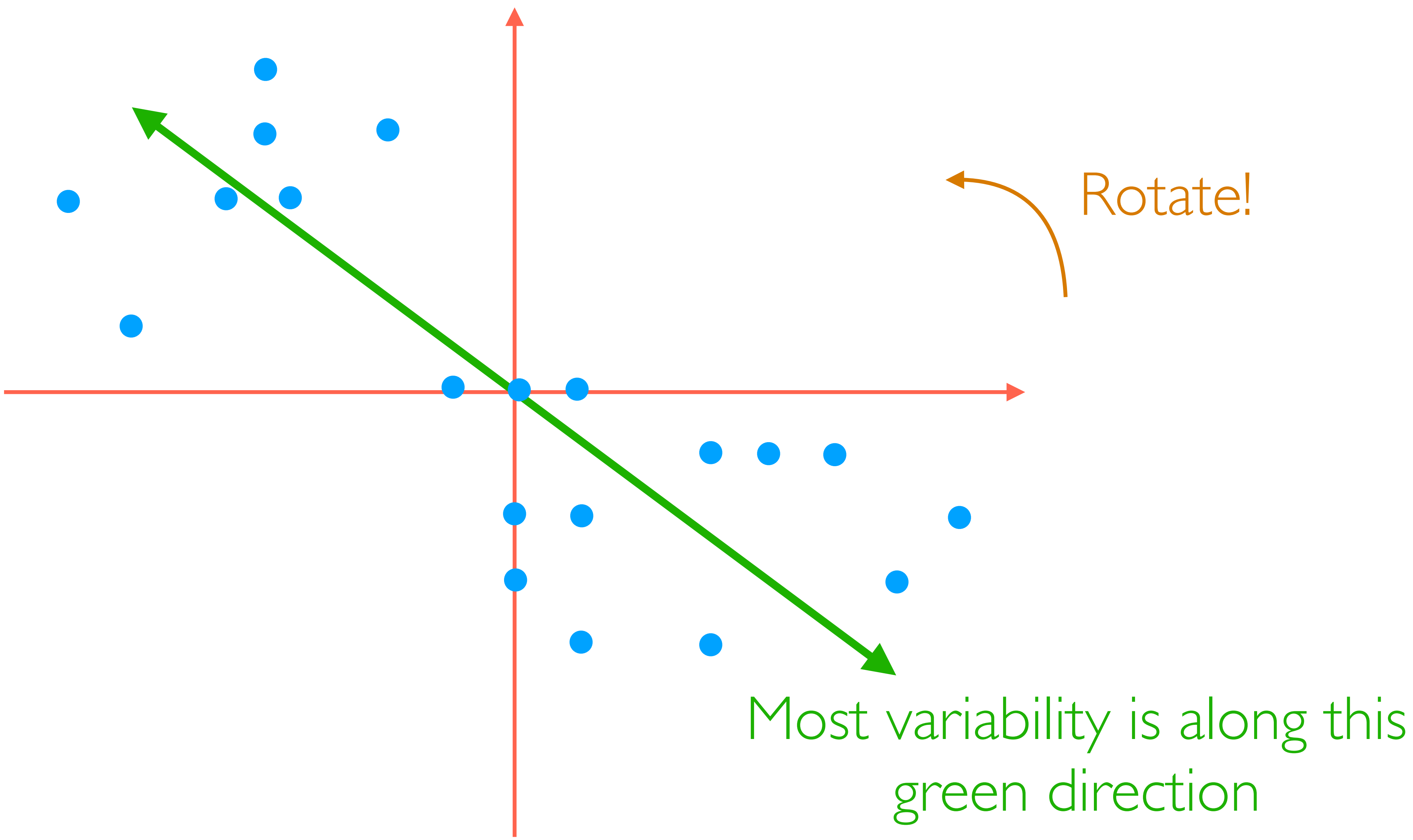
Principal Component Analysis (PCA)

How to project 2D data down to 1D?



Principal Component Analysis (PCA)

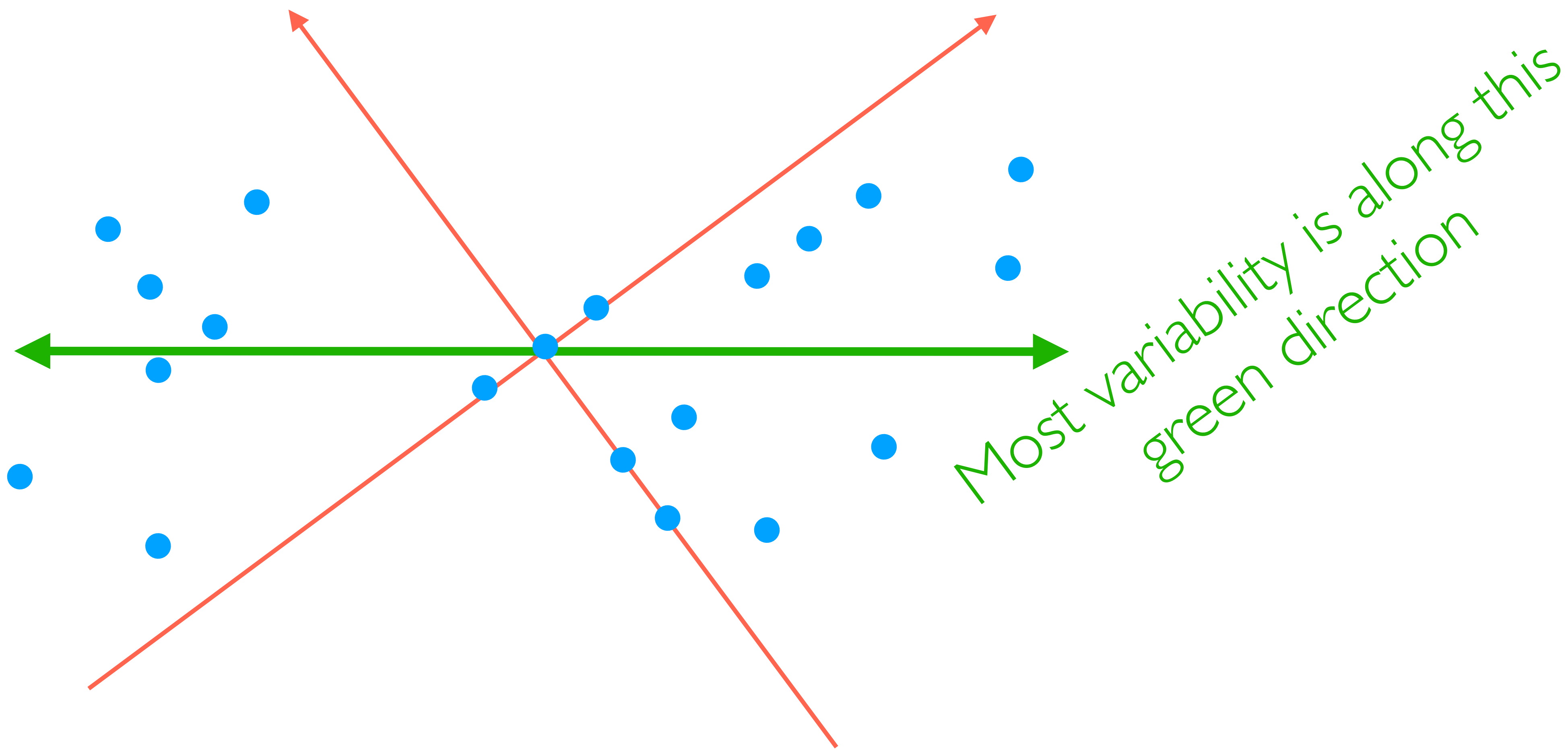
How to project 2D data down to 1D?



But notice that most of the variability in the data is *not* aligned with the red axes!

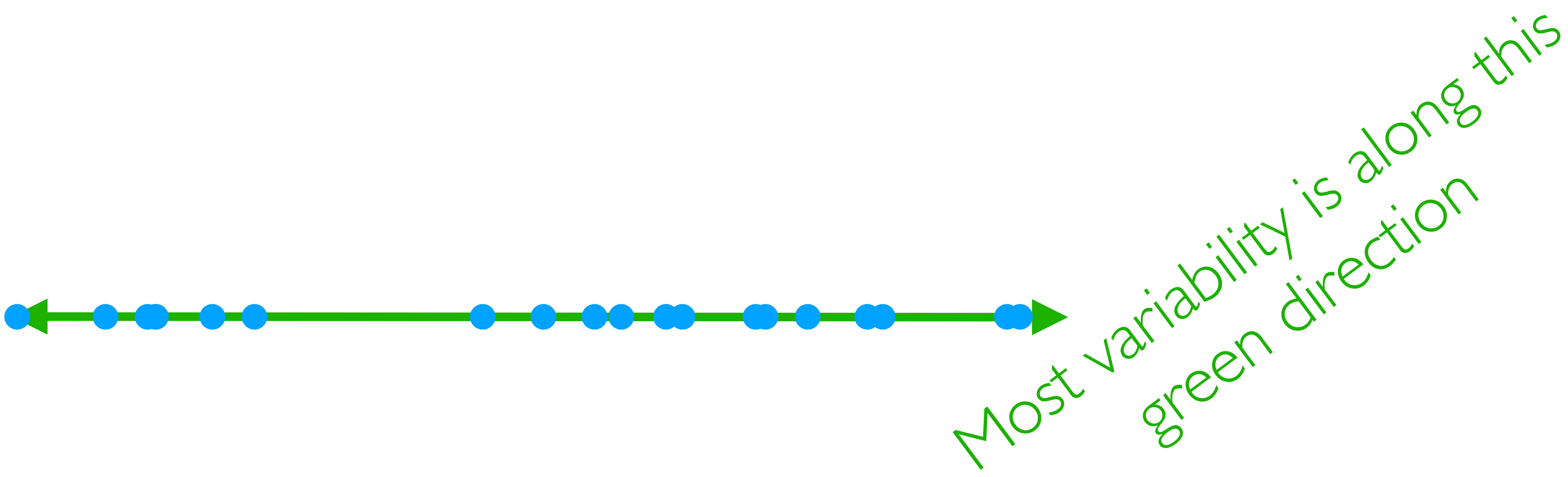
Principal Component Analysis (PCA)

How to project 2D data down to 1D?



Principal Component Analysis (PCA)

How to project 2D data down to 1D?

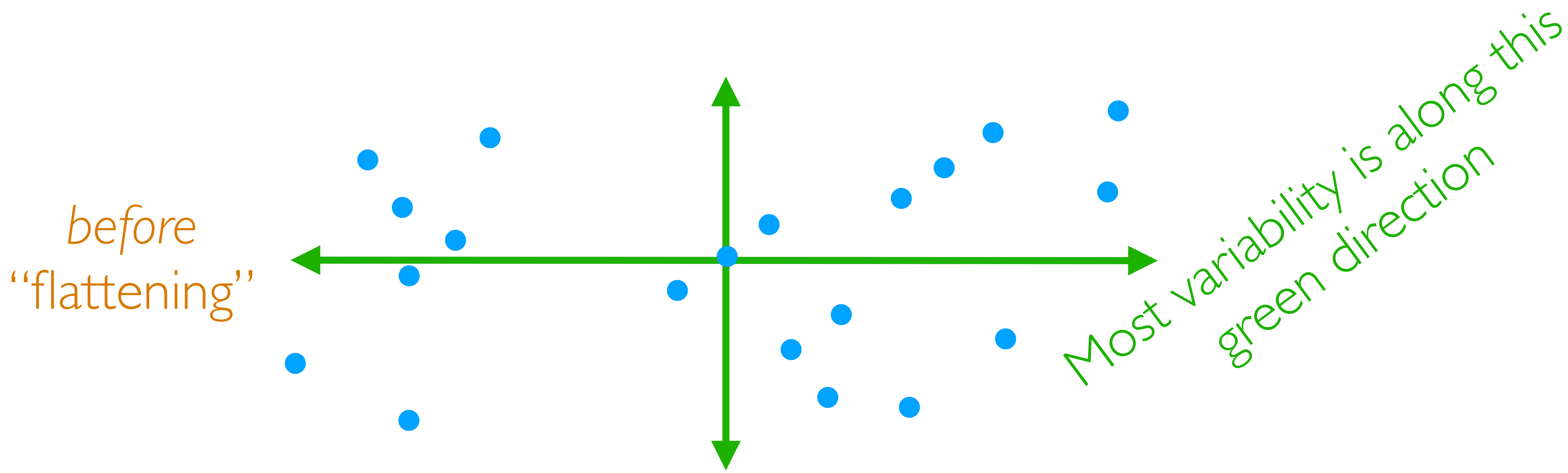


The idea of PCA actually works for 2D ➔ 2D as well
(and just involves rotating, and not "flattening" the data)

Principal Component Analysis (PCA)

~~How to project 2D data down to 1D?~~

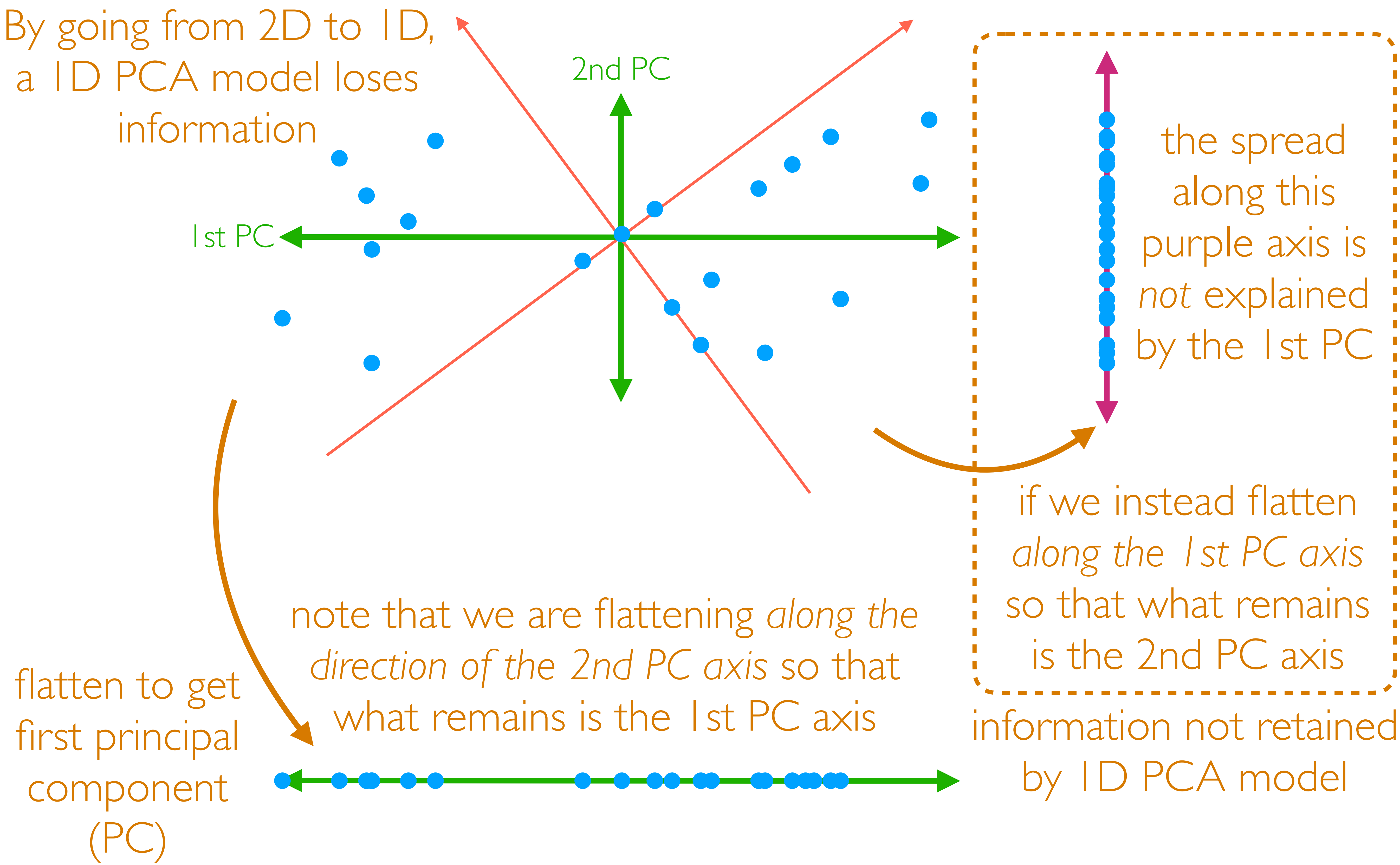
How to rotate 2D data so 1st axis has most variance



The idea of PCA actually works for 2D ➔ 2D as well
(and just involves rotating, and not "flattening" the data)

2nd green axis chosen to be 90° ("orthogonal") from first green axis

PCA: The Unexplained Variability



3D Dataset Example

http://setosa.io/ev/principal-component-analysis/

PCA in Higher Dimensions

- Finds top *k* orthogonal directions that explain the most variance in the data
  - 1st component: explains most variance along 1 dimension
  - 2nd component: explains most of **remaining** variance along next dimension that is orthogonal to 1st dimension
  - …
- "Flatten" data by retaining only the top *k* dimensions (if $k <$ original dimension, then we are doing dimensionality reduction)

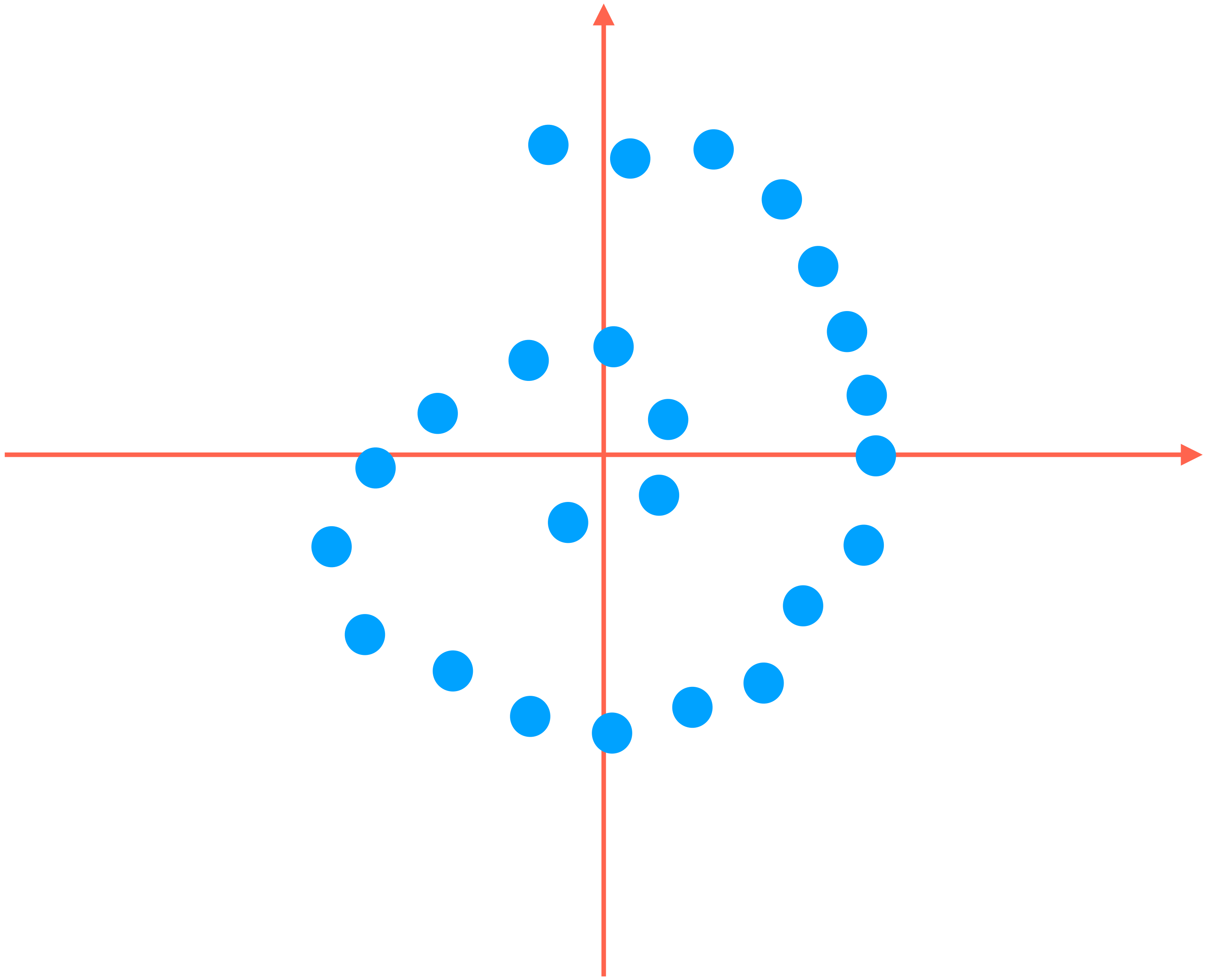
Principal Component Analysis (PCA)

Demo

PCA reorients data so axes explain variance in "decreasing order"
→ can "flatten" (*project*) data onto a few axes that captures most variance

Image source: http://4.bp.blogspot.com/-USQEgoh1jCU/VfncdNOETcI/AAAAAAAAGp8/Hea8UtE_1c0/s1600/Blog%2B1%2BIMG_1821.jpg

2D Swiss Roll



PCA would just flatten this thing and
lose the information that the data actually lives on a 1D line that has been curved!

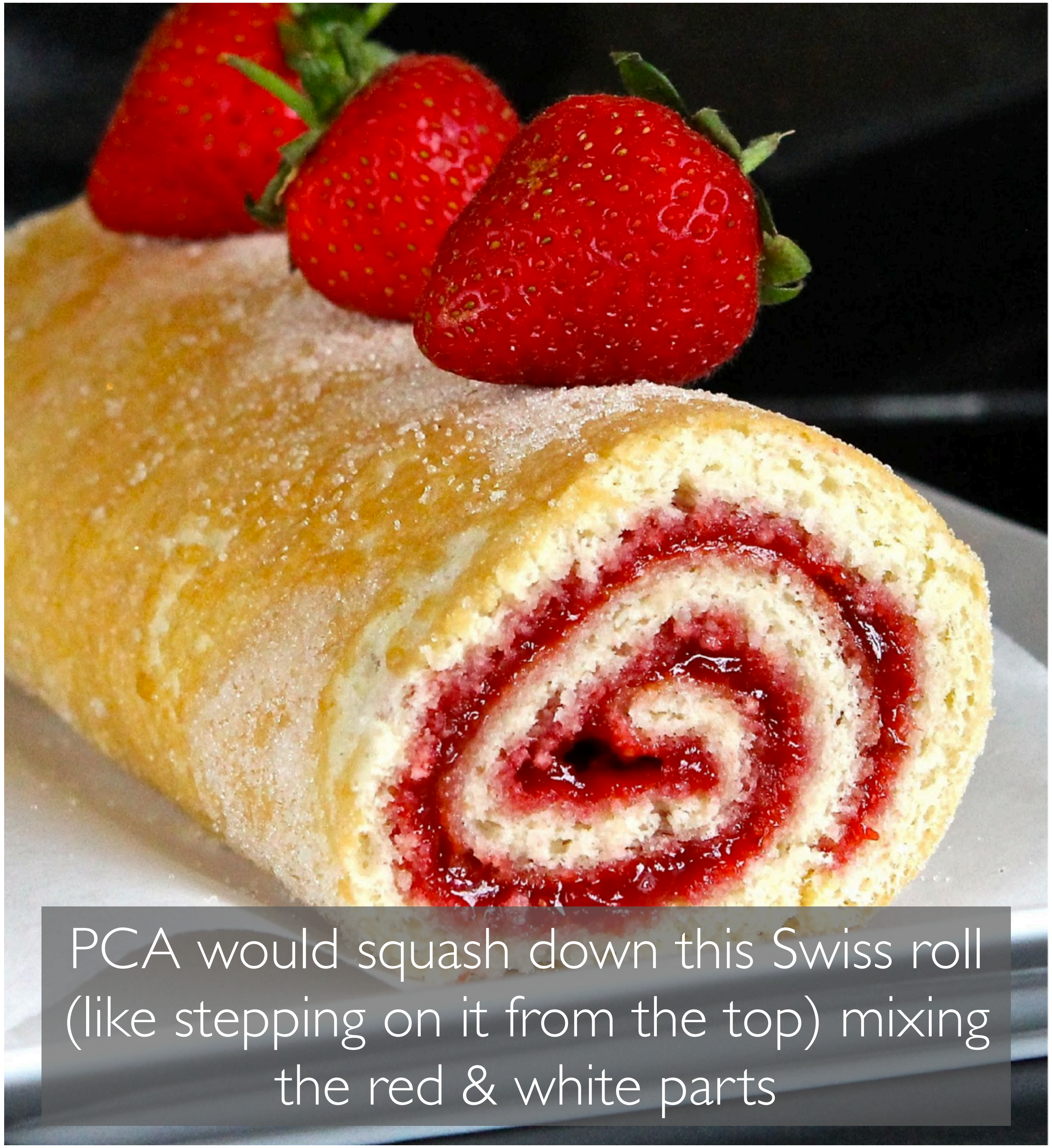
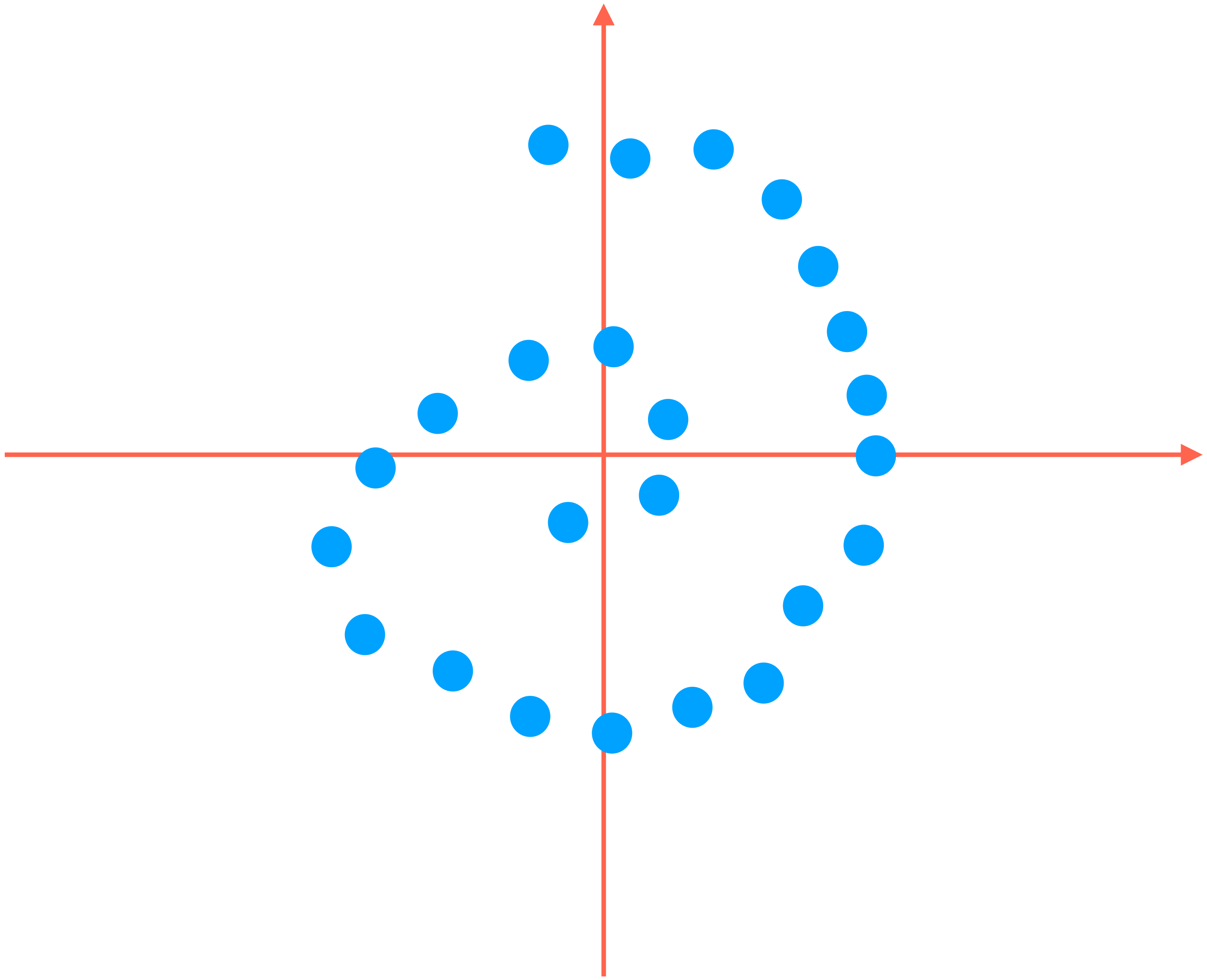
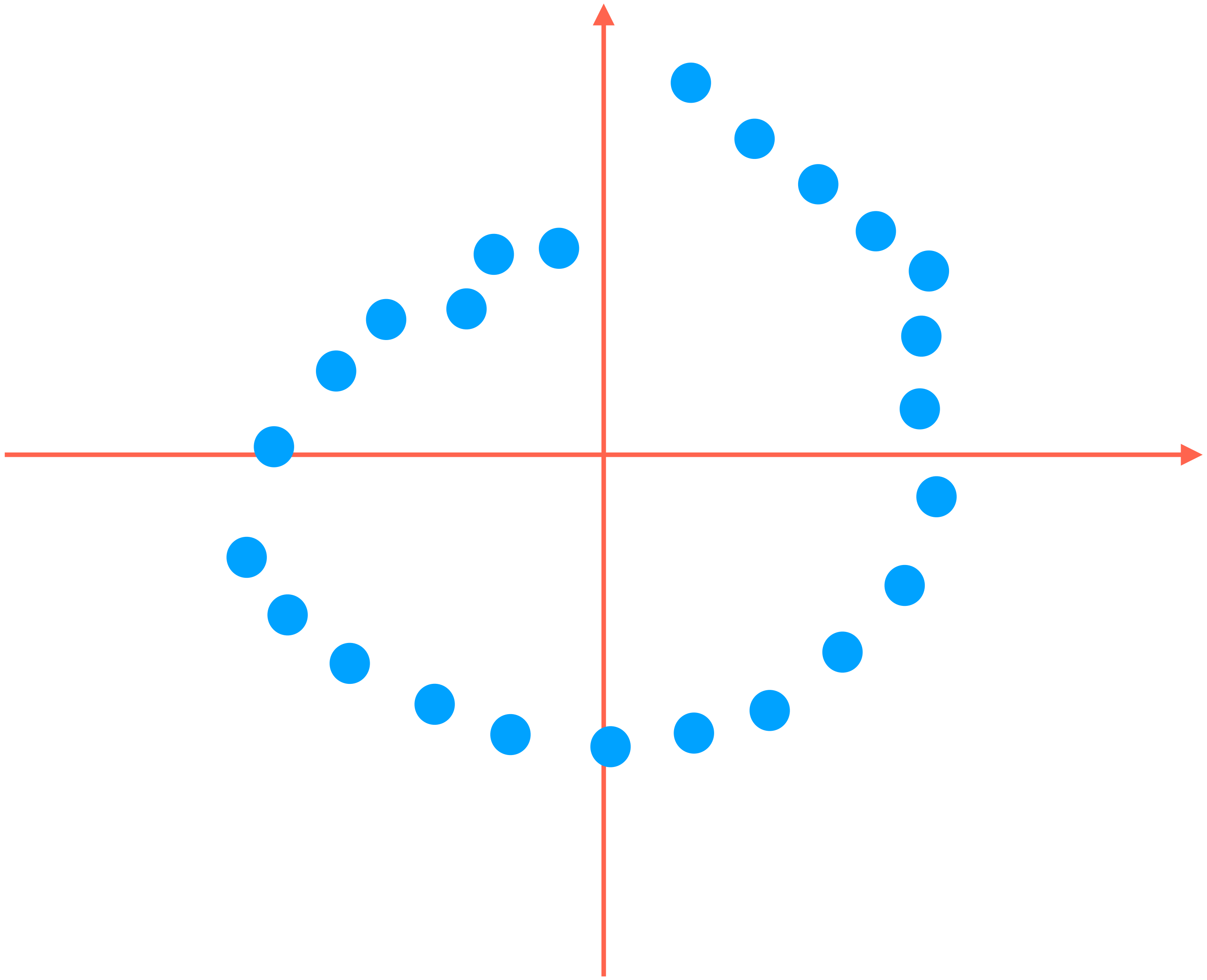


Image source: http://4.bp.blogspot.com/-USQEgoh1jCU/VfncdNOETcI/AAAAAAAAGp8/Hea8UtE_1c0/s1600/Blog%2B1%2BIMG_1821.jpg

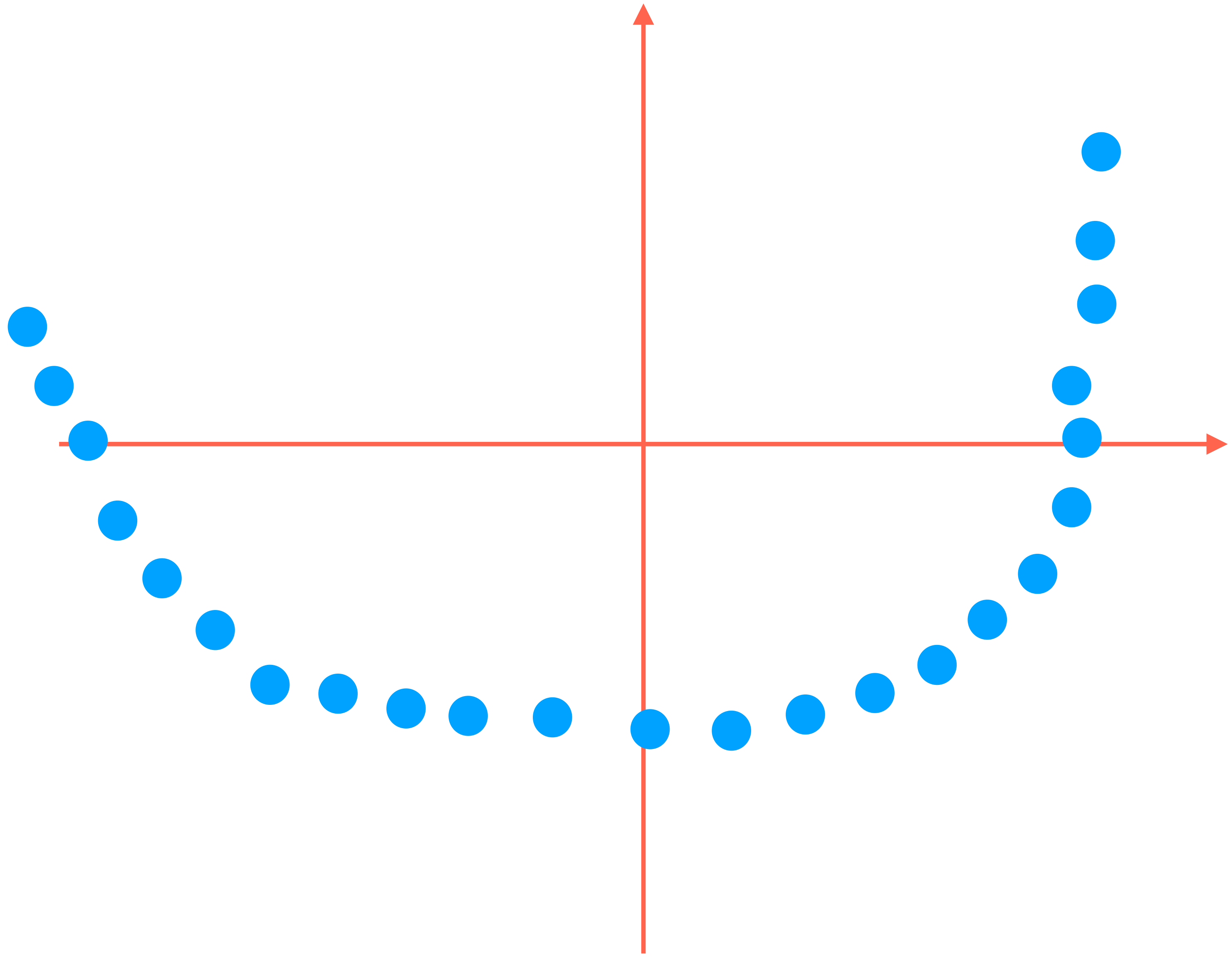
2D Swiss Roll



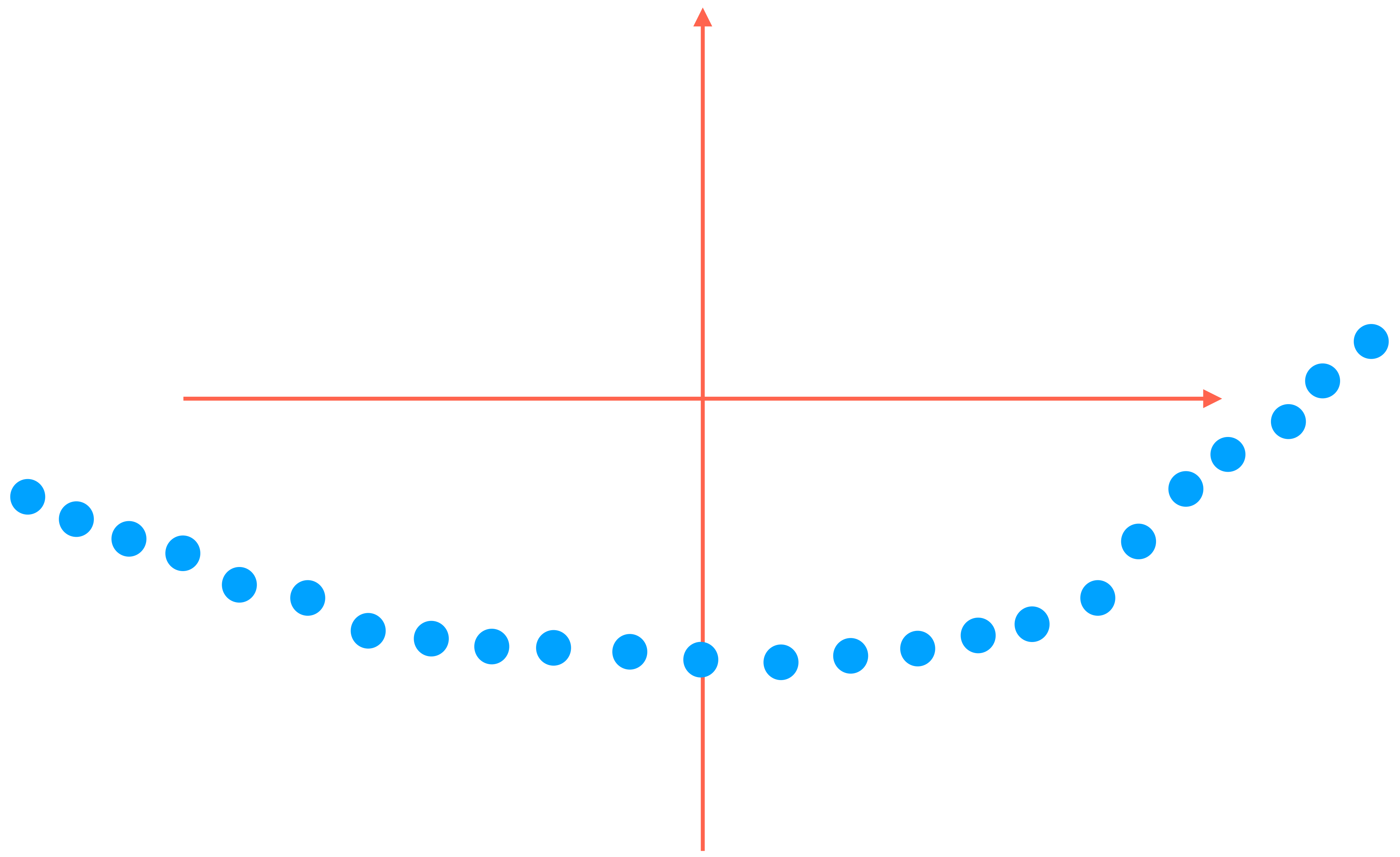
2D Swiss Roll



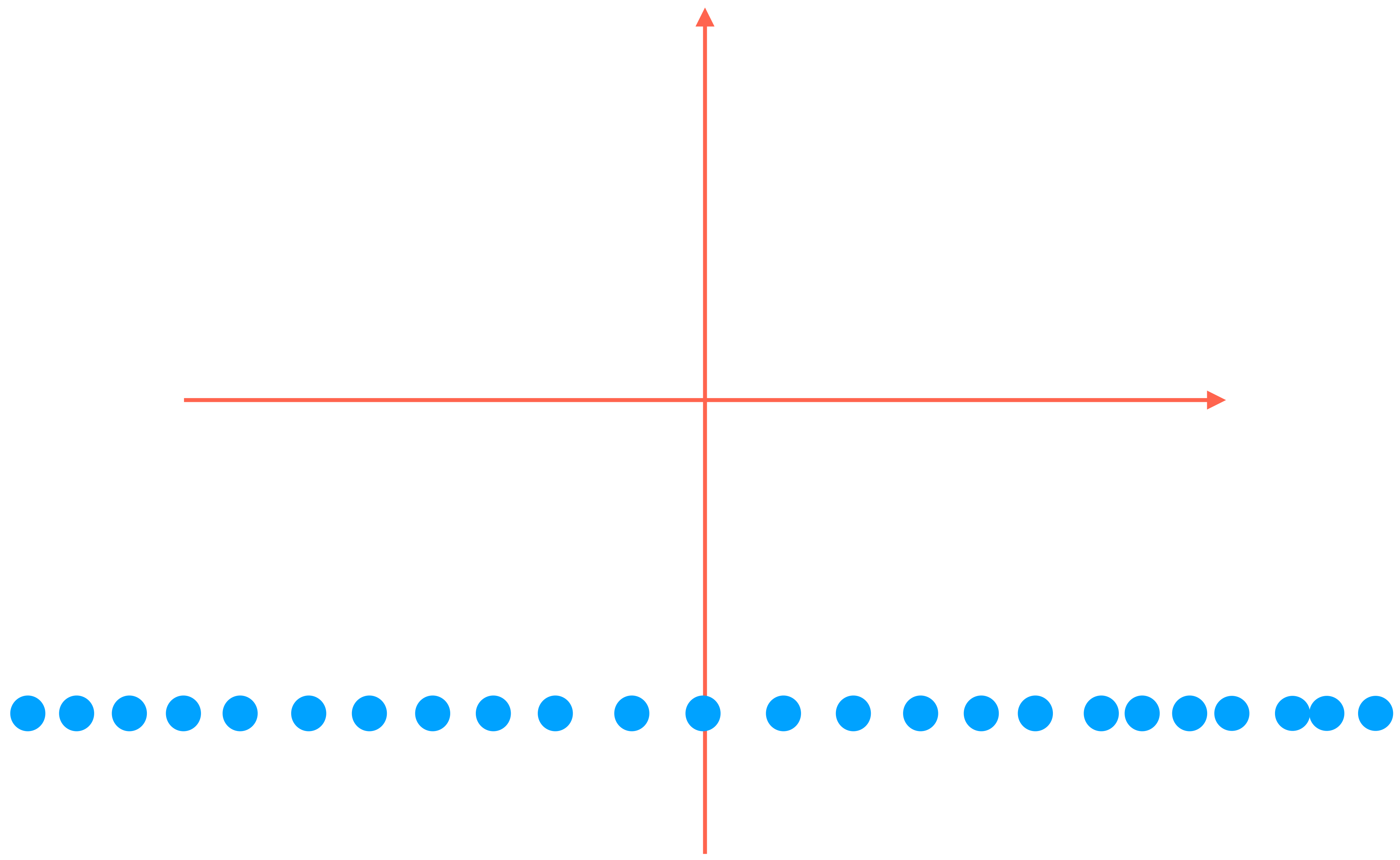
2D Swiss Roll



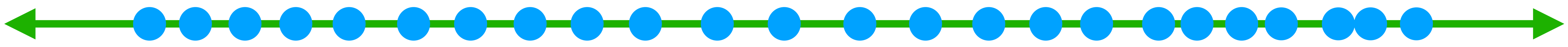
2D Swiss Roll



2D Swiss Roll

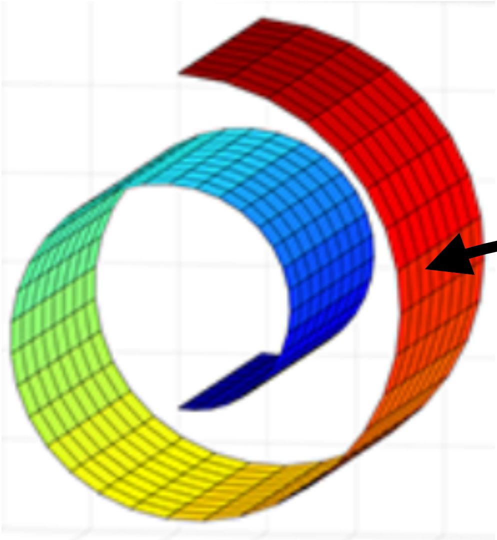


2D Swiss Roll



This is the desired result

Manifold Learning



The dataset here is clearly 3D

But when we zoom in a lot on any point, around the point it looks like a flat 2D sheet!

Another example: Earth is approximately a 3D sphere, but zooming a lot on any point, around the point it's approximately a 2D sheet

In general: if we have ***d***-dimensional data where when you zoom in a lot, the data dimensionality is smaller than ***d***, then the lower-dimensional object is called a **manifold**

- We have the data's high-dim. coordinates, but we want to find the low-dim. coordinates (on the manifold) ➜ this is **manifold learning**
- Manifold learning is *nonlinear* whereas PCA is linear (this will make more sense after we see code demos)

Image source: "Head Pose Estimation via Manifold Learning" (Wang et al 2017)