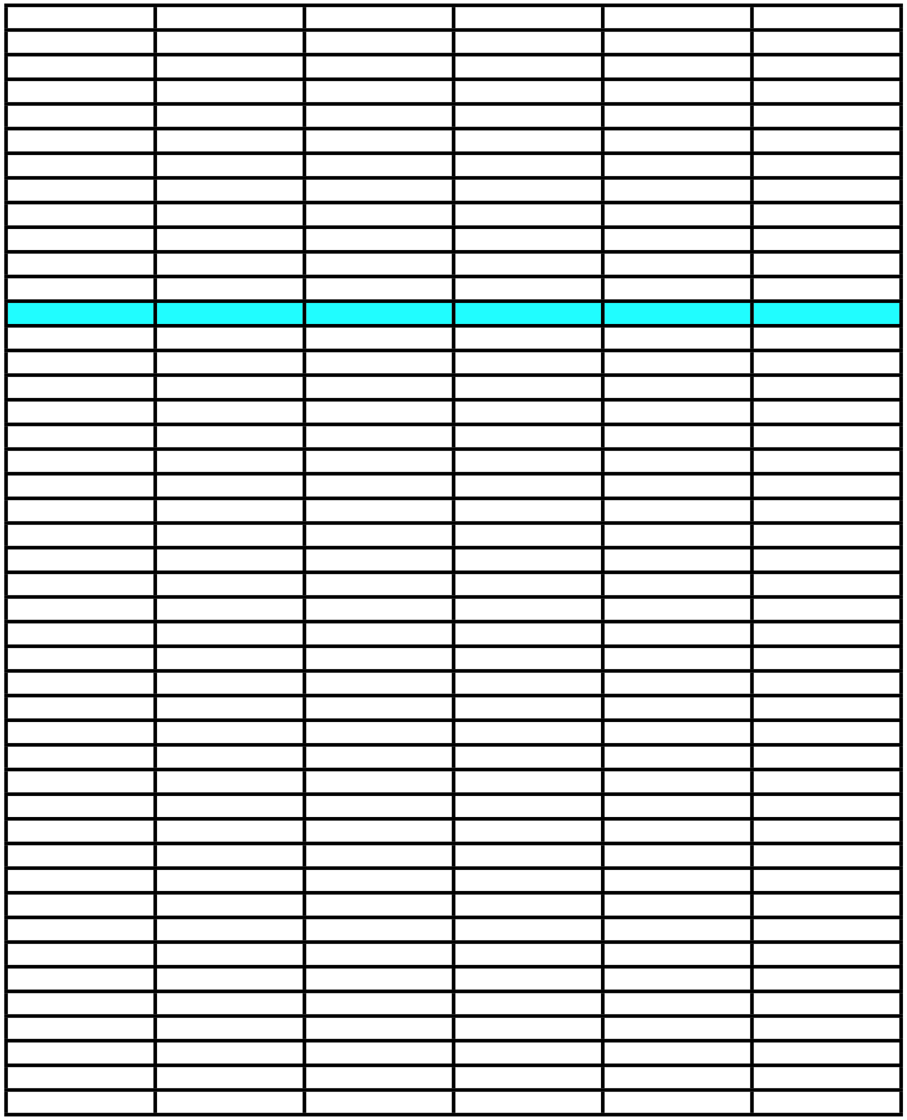


The table is one important constraint, after which we make an important conceptual leap (one that's often invisible)



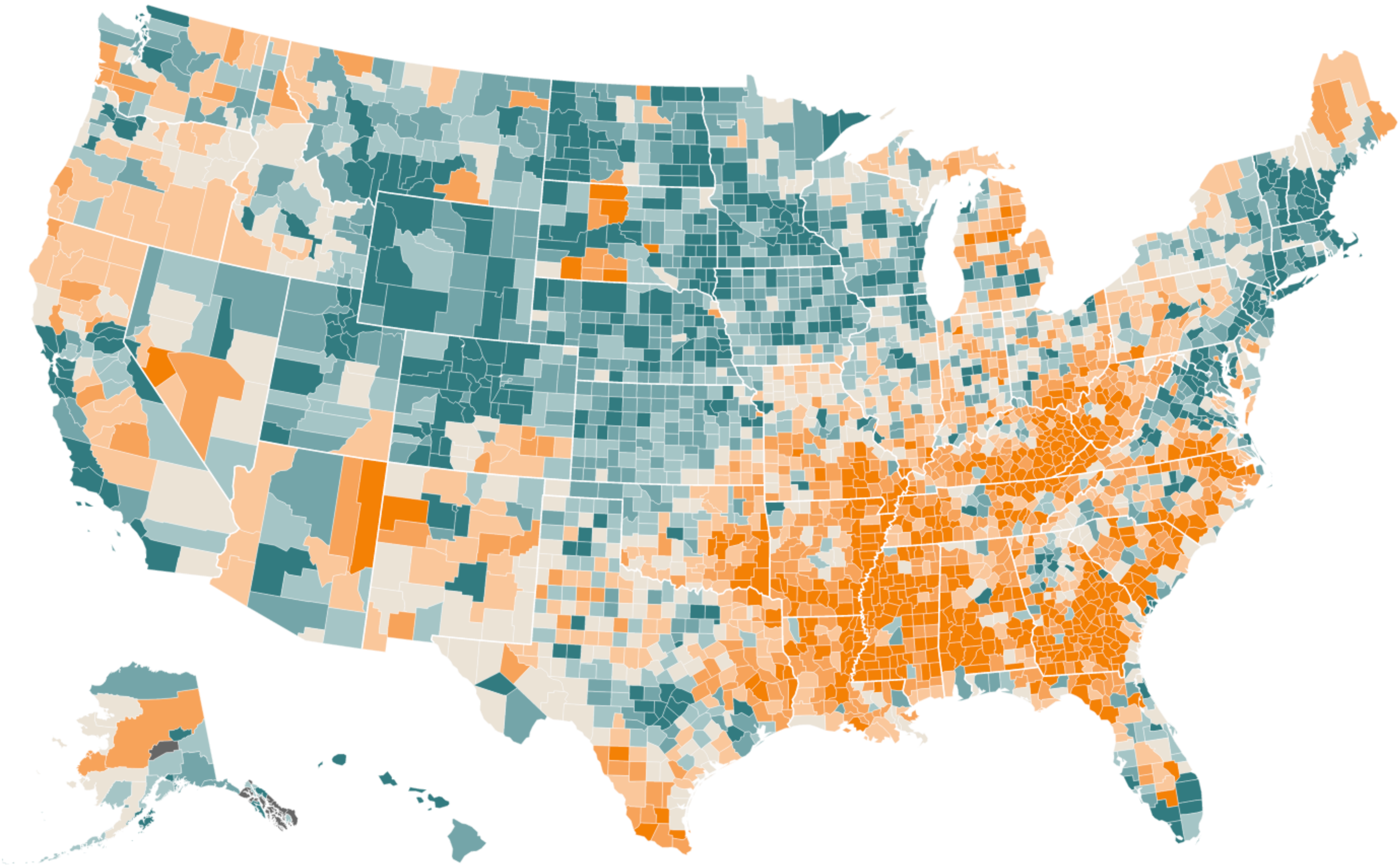
Each row represents a point in d-dimensional Euclidean space

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix}$$

Where Are the Hardest Places to Live in the U.S.?

Alan Flippen @alflip JUNE 26, 2014

56



A composite ranking of where Americans are healthy and wealthy, or struggling.

County Ranking

DOING BETTER DOING WORSE

At the right, we have the data associated with this graphic — The ranking is compiled by ranking the average ranks of counties using different indicators

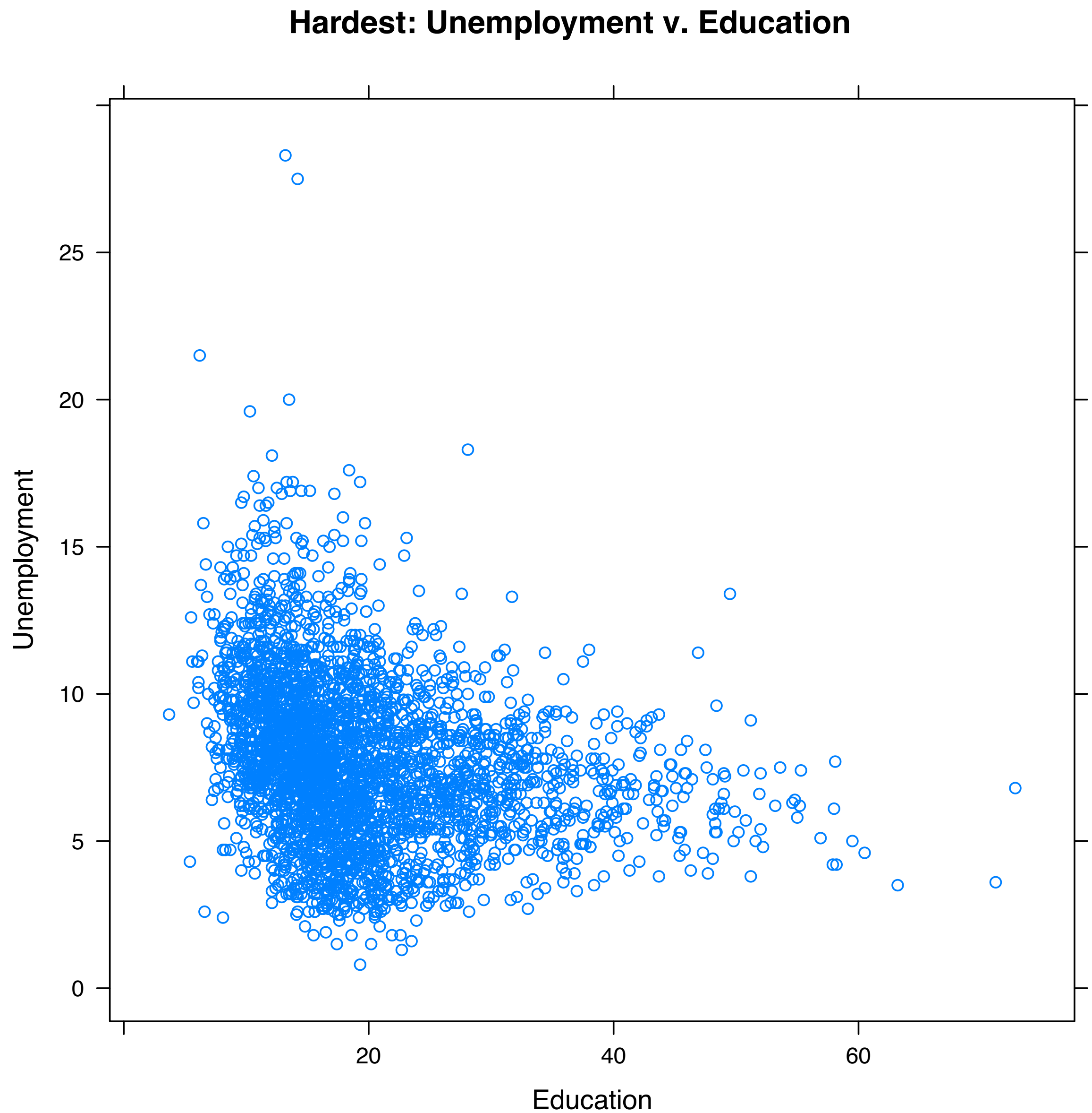
What kinds of variables do we have here? Qualitative or Quantitative?

county	state	id	rank	education	income	unemployment	disability	life	obesity
Autauga	Alabama	1001	1371	21.7	53773	6.5	1.6	76.1	38
Baldwin	Alabama	1003	657	27.7	50706	6.8	1	77.7	34
Barbour	Alabama	1005	2941	14.5	31889	11.2	2.9	74.7	47
Bibb	Alabama	1007	2803	9	36824	7.6	2.6	74.2	43
Blount	Alabama	1009	2000	12.4	45192	6.2	1.4	75.9	40
Bullock	Alabama	1011	3083	11.9	34500	13.4	3.8	71.8	49
Butler	Alabama	1013	2981	12.9	30752	10.9	3.2	73.8	45
Calhoun	Alabama	1015	2451	16	40093	7.6	2.4	73.3	40
Chambers	Alabama	1017	2967	11	32181	9.3	2.6	73.3	44
Cherokee	Alabama	1019	2584	13.1	36241	7.1	2.2	74.7	41
Chilton	Alabama	1021	2546	12.5	40834	6.5	2.1	73.9	43
Choctaw	Alabama	1023	2873	11.9	35123	9	3.3	75.1	46
Clarke	Alabama	1025	3011	12.7	30954	12.1	3.1	74.9	44
Clay	Alabama	1027	2914	8.8	34556	9.3	2.6	74.2	42
Cleburne	Alabama	1029	2564	9.5	37244	6.9	2.1	74.2	39
Coffee	Alabama	1031	1602	22.6	44626	6.2	1.5	76.3	39
Colbert	Alabama	1033	2398	18	40158	7.6	2.3	74.1	41
Conecuh	Alabama	1035	3088	9.7	27064	11.6	3.6	73.8	45
Coosa	Alabama	1037	2872	9.7	37425	8.2	2.7	73.9	45
Covington	Alabama	1039	2591	13.8	35321	7.5	2.1	74.9	41
Crenshaw	Alabama	1041	2739	11	37309	7.2	2.4	73.3	43
Cullman	Alabama	1043	2194	14.2	39244	6.4	1.7	75	39
Dale	Alabama	1045	2127	17.5	45247	7.3	2.1	75.7	42
Dallas	Alabama	1047	3094	13.3	26178	13.7	6.2	72	48

A scatterplot

If we had only two quantitative variables in our data set, we would do the (now) obvious thing of simply plotting one variable against another

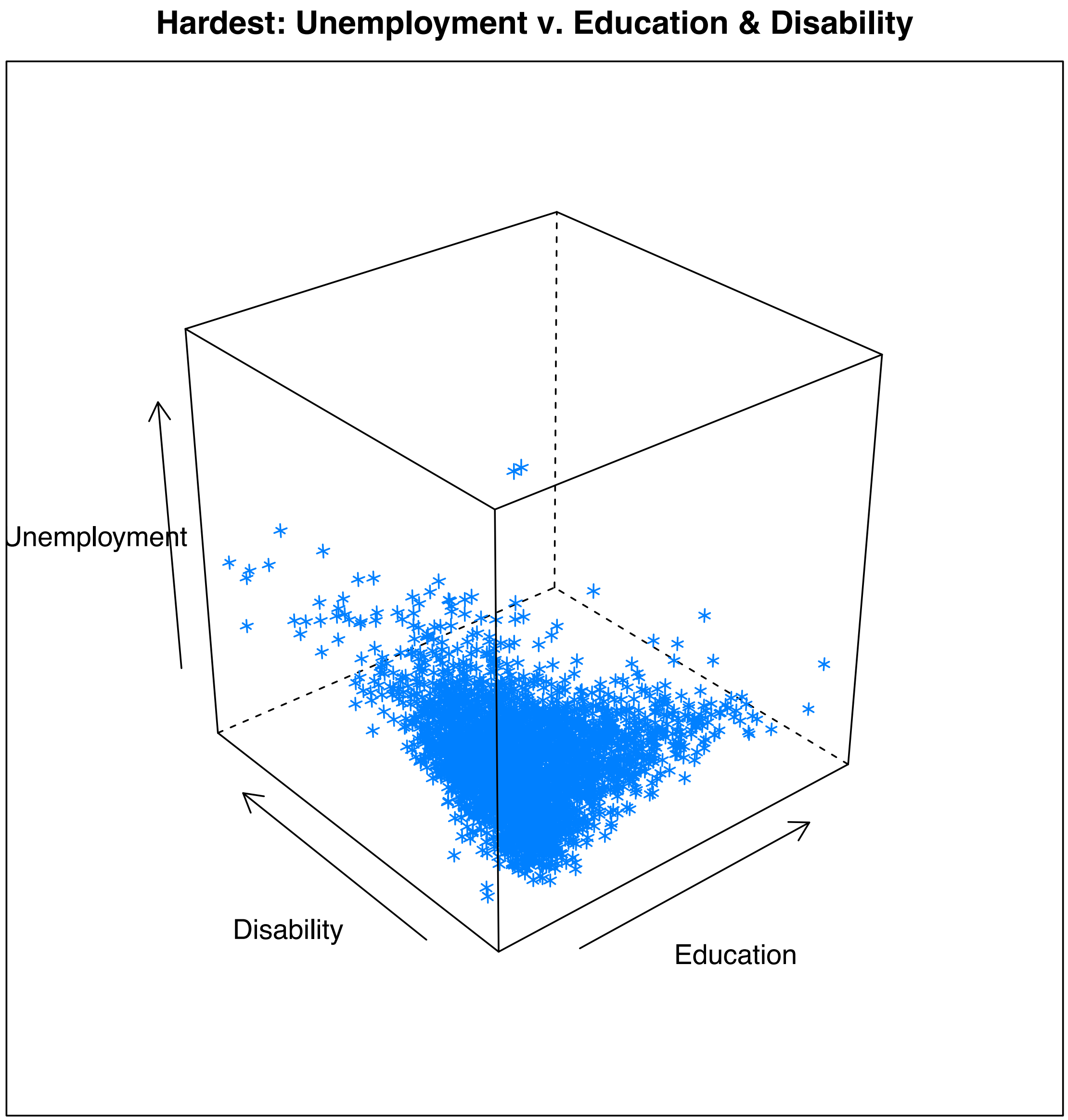
The result is a scatterplot...



A 3-d scatterplot

It’s 3-d cousin aligns data on 3 variables along the x, y and z axes placing each point in 3-space

Here’s an example from R



Geometry

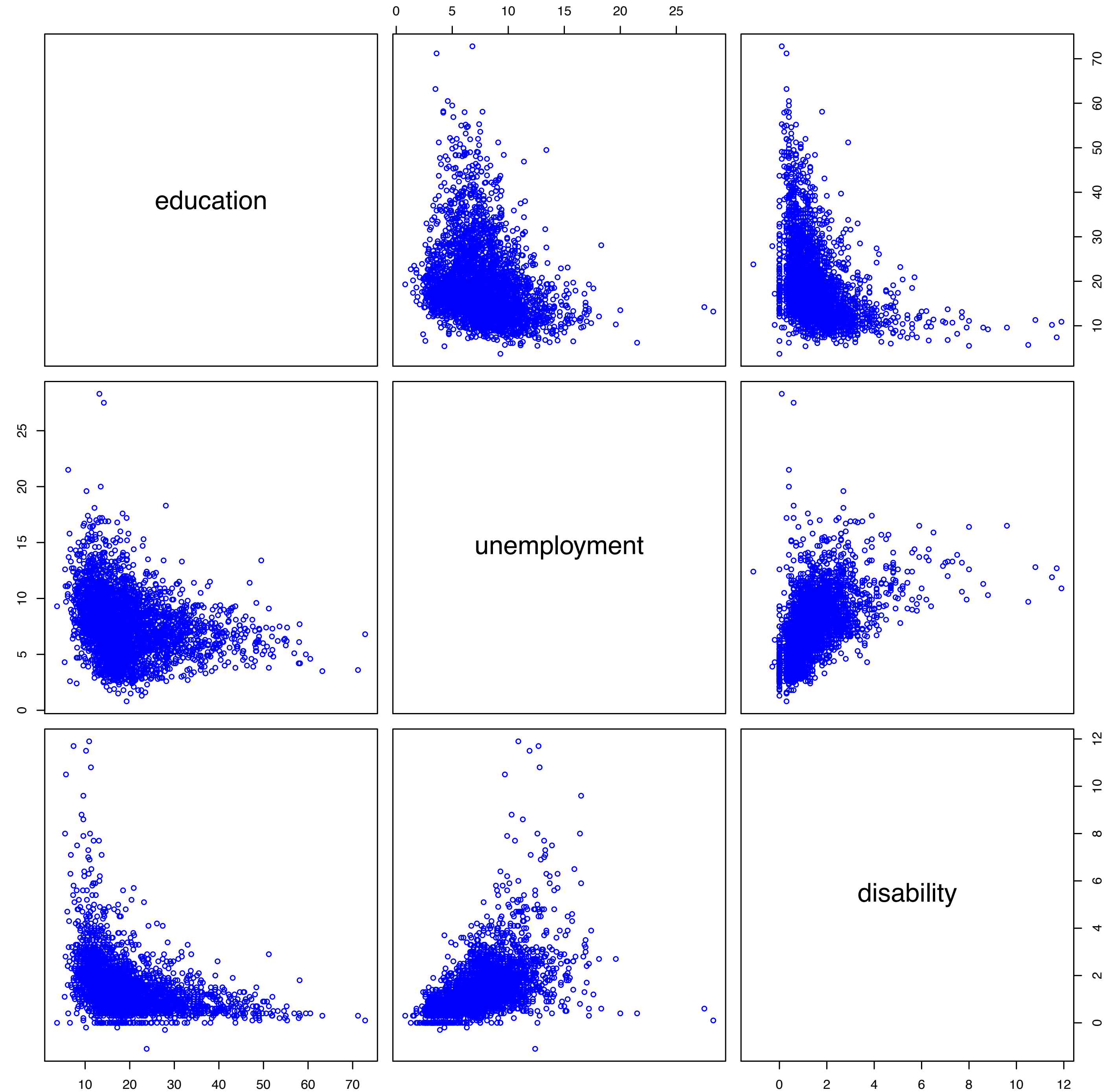
With 2- and 3-dimensional data, we can **invoke a spatial metaphor** and create artificial axes to plot two variables against each other -- You should all be intimately familiar with a 3-d coordinate system given your practice with Processing

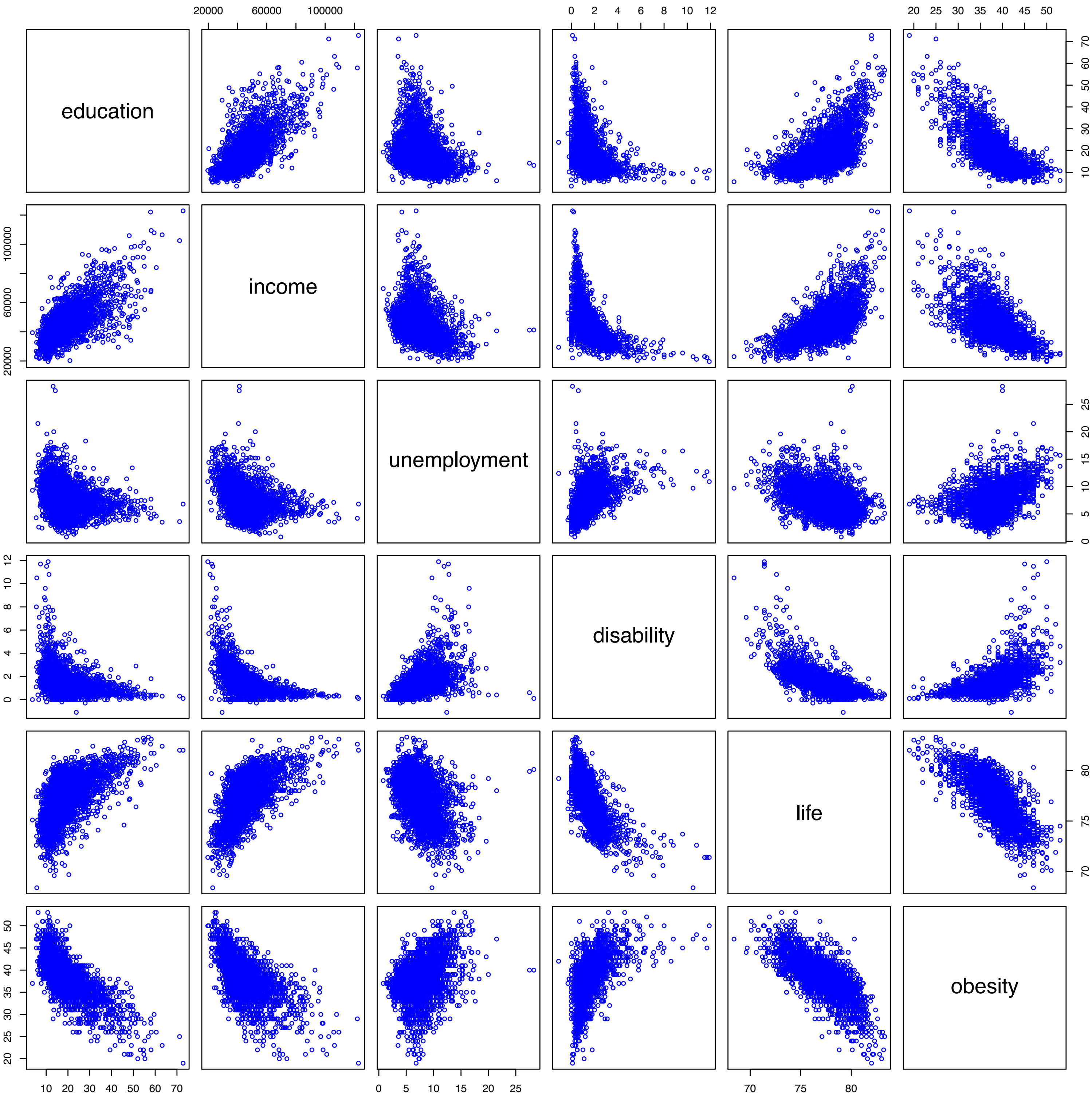
In each case we are associating **the “x”, “y” and “z” axes with a different variable** or column in the data set and then plotting locating each row in our table using this coordinate system -- The first data point is at (0.75, 2.3, -0.71), for example, or 0.75 out along the HDI axis, 2.3 units along the ln_events axis, etc.

The question arises, however, what do you do when you have more than 3 variables measured on each observational unit? How do we “see” tables with this form?

One technique to attempt to see the relationship between multiple variables involves, well, **multiple views** -- With three variables we have three different pairings that can each be represented as a scatterplot

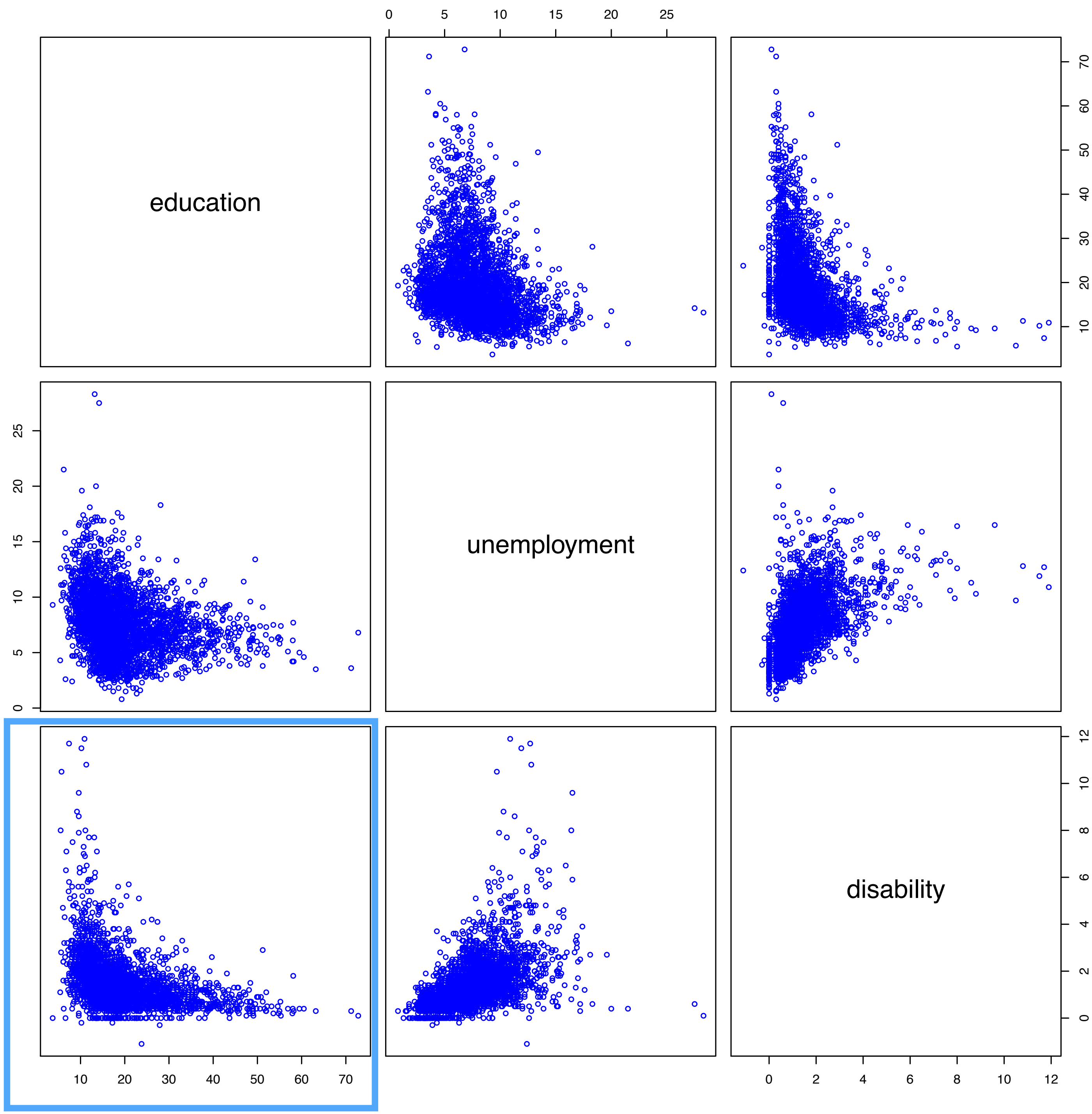
The resulting **scatterplot matrix** represents all the pairwise relationships between columns in our data set at one time -- It's not a huge advance, but in R there's some nice layout added to the




education
income
unemployment
disability
life
obesity
20000
60000
100000
0
2
4
6
8
10
12
20
25
30
35
40
45
50
10
20
30
40
50
60
70
0
5
10
15
20
25
75
80

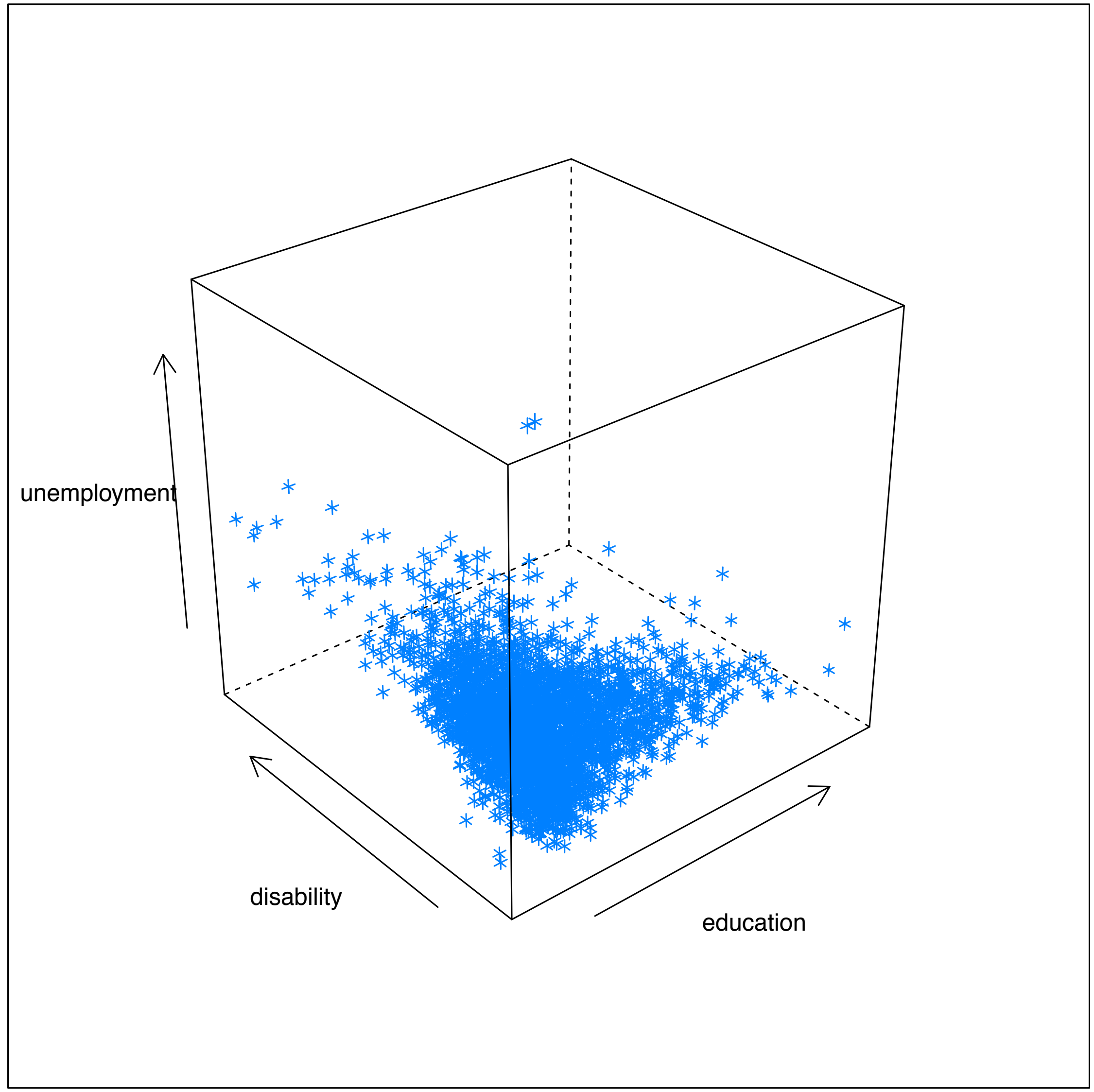
Just to be super clear here, if we had 3 variables in our data set, each entry in the scatterplot matrix represents an extreme view on the data -- That is, we take our 3-dimensional box of data and look at it along different axes

Changing the view in this way, looking along a single axis, produces (in technical parlance) a projection of our data into a 2-d "plane", the space of the remaining pair of variables



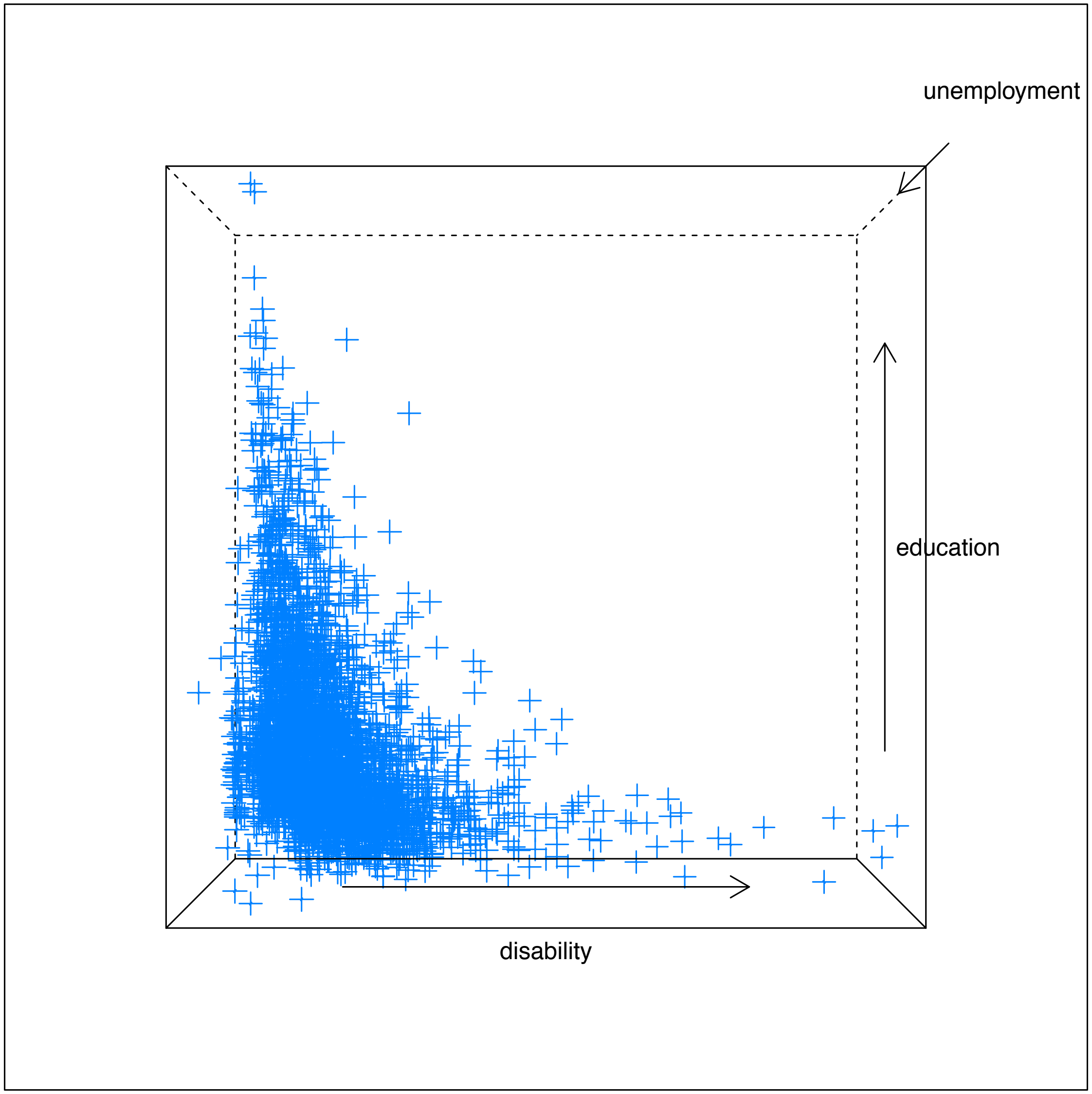
Just to be super clear here, if we had 3 variables in our data set, each entry in the scatterplot matrix represents an extreme view on the data -- That is, we take our 3-dimensional box of data and look at it along different axes

Changing the view in this way, looking along a single axis, produces (in technical parlance) a projection of our data into a 2-d “plane”, the space of the remaining pair of variables



Just to be super clear here, if we had 3 variables in our data set, each entry in the scatterplot matrix represents an extreme view on the data -- That is, we take our 3-dimensional box of data and look at it along different axes

Changing the view in this way, looking along a single axis, produces (in technical parlance) a projection of our data into a 2-d “plane”, the space of the remaining pair of variables



Geometry

So, after an enormous reduction of our data from a collection of attributes assembled somewhat arbitrarily about each object we've studied, we've reduced things to a regular table of numerical data

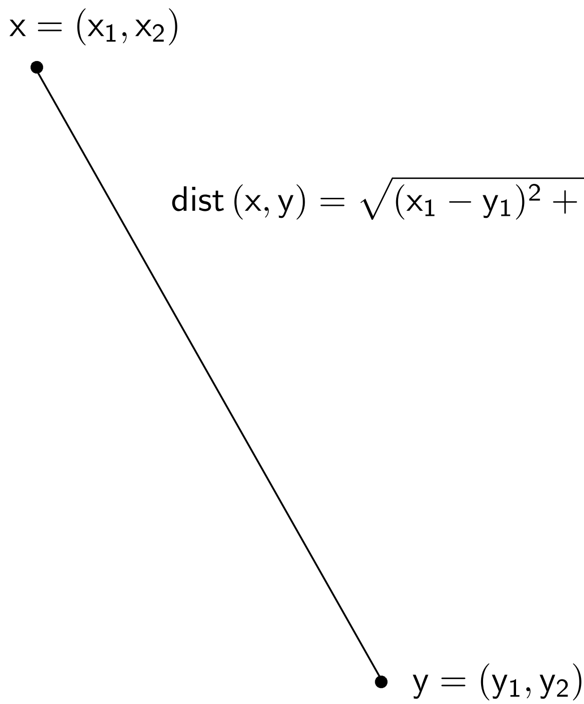
This step lets us invoke a number of concepts from (essentially) high school geometry -- We are able to identify axes with the different variables and place each observation in a d-dimensional Euclidean space

So far, we've looked at 2- and 3-space or d-space via a series of 2-dimensional "marginal" plots (although one could imagine a 3-d version of the scatterplot matrix) -- Let's recall a bit more geometry and see where it might take us

The notion of "nearby" will help us find natural groupings in data, make predictions, almost all of statistics comes from an understanding of geometry and the clever mobilization of a geometric understanding

If x and y are in 2-space,
then we can plot them

We can also talk how far apart they are using standard Euclidean distance



$$\mathrm{dist}\,(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

With distance, we can
compare points based on
whether they are far...

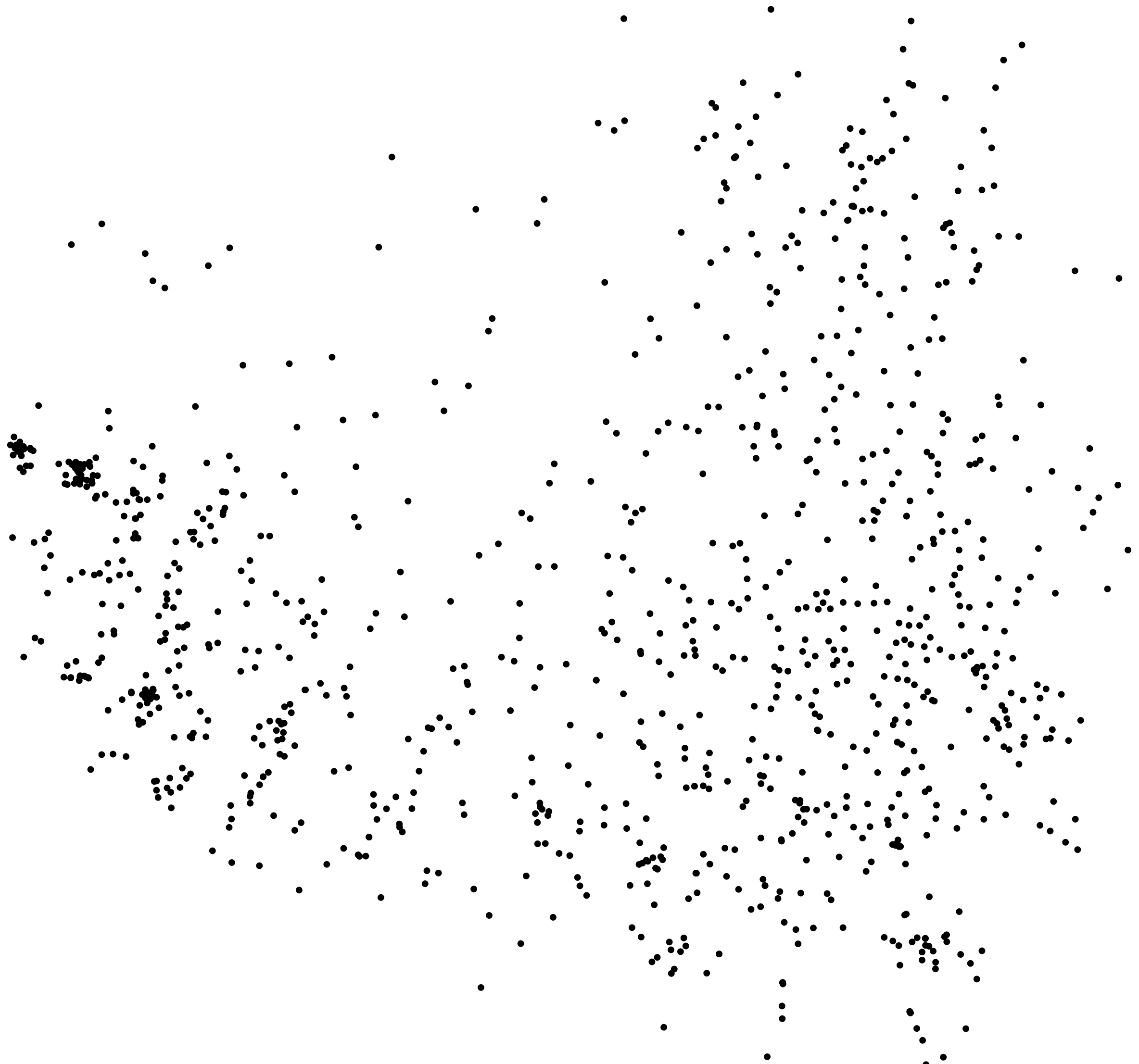
With distance, we can compare points based on whether they are far...

With distance, we can compare points based on whether they are far or near

Which leads us to the idea of clusters, points that fall naturally into groups based on proximity

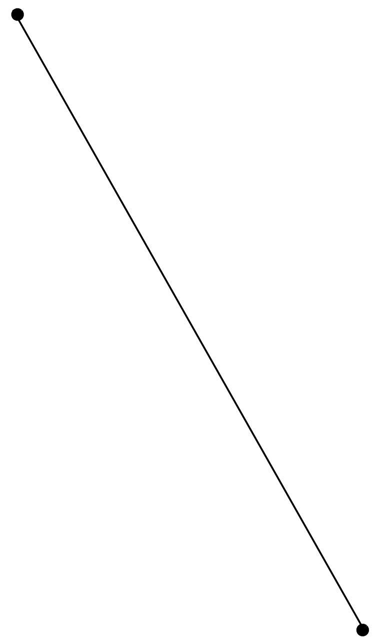
How many clusters do you see here? How do you identify them?

What about here?



The idea of distance is completely general and we can compute the distance between points in d-dimensional space

$x = (x_1, \ldots, x_d)$



$$\mathrm{dist}\,(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_d - y_d)^2}$$

$y = (y_1, \ldots, y_d)$

High-dimensional spaces

The notion of near and far starts to break down a little as we increase d from 2 to 3 to 4 to 100 -- In short, as we increase the dimension, all points start to look far apart

There are several arguments usually put forward to support this — Suppose, for example, we consider a sphere in d-dimensional space

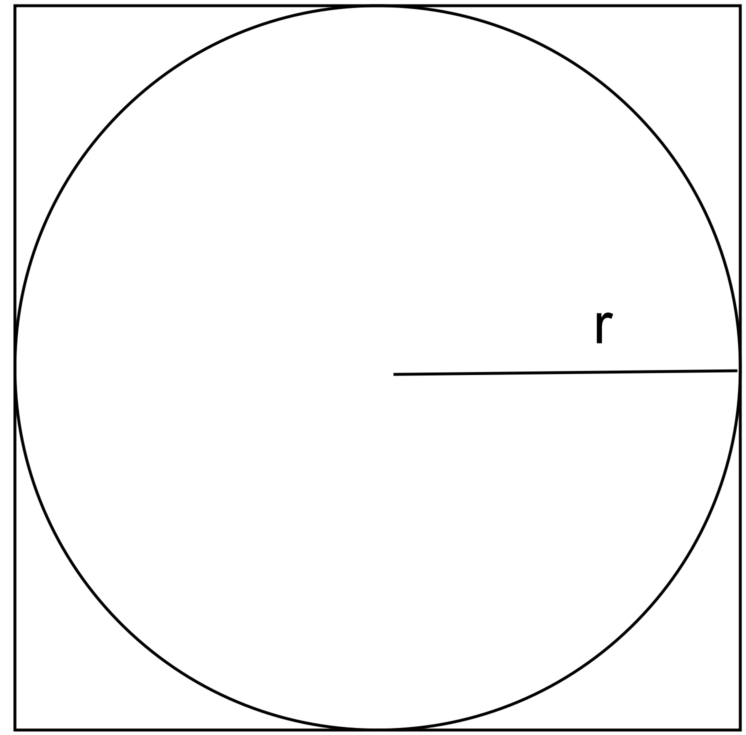
$$\text{volume of a sphere of radius } r = \frac{2r^d\pi^{d/2}}{d\Gamma(d/2)}$$

which we can put in a box

$$\text{volume of the enclosing box with side } 2r = 2r^d$$

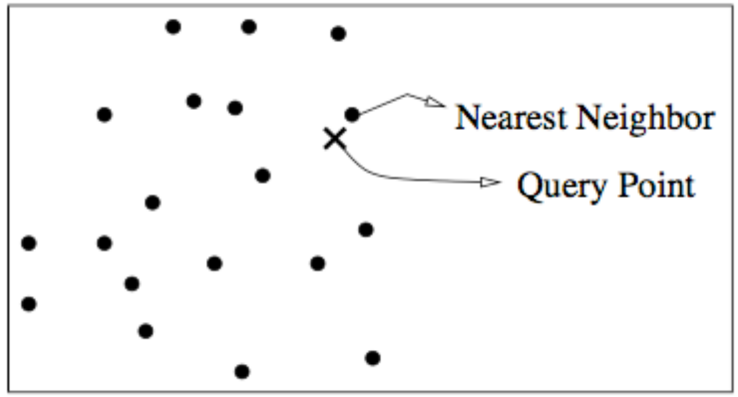
and after a little work

$$\text{ratio of their volumes} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} \rightarrow 0 \text{ as } d \rightarrow \text{big}$$

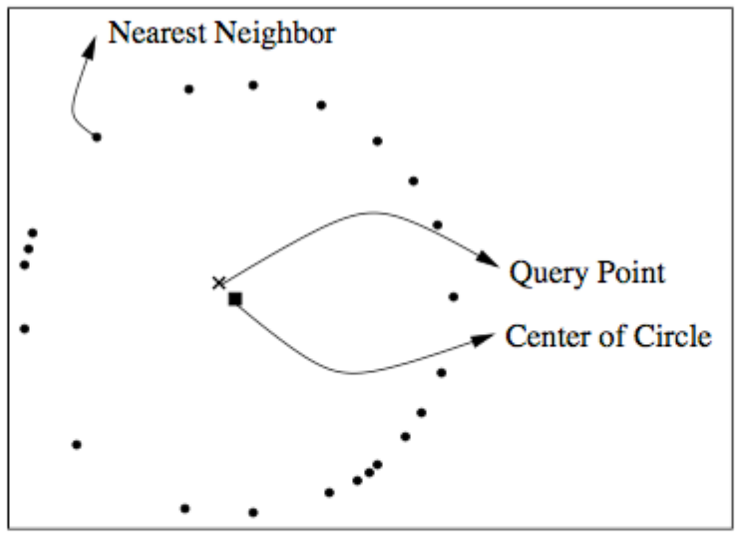


High-dimensional spaces

This means that somehow **all the "mass" in the box is in at the edges** as we increase the dimension of our data (increase the number of variables we measure)



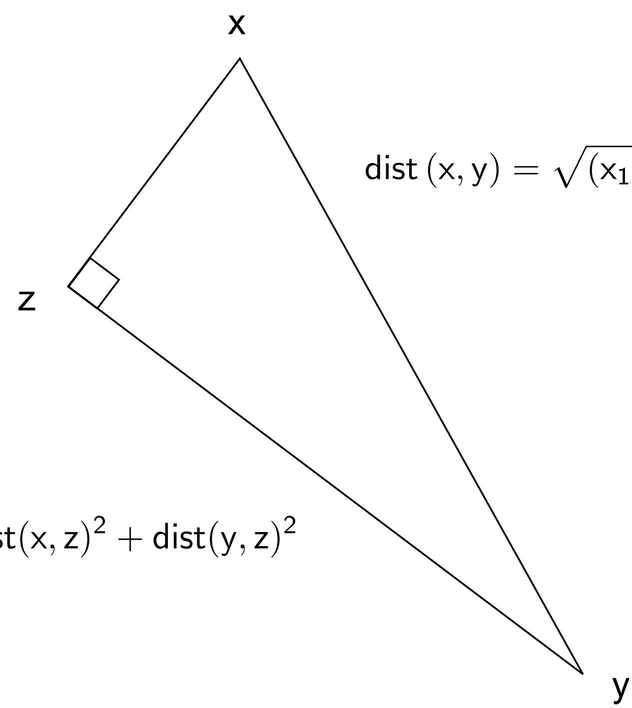
Returning to data, under certain mathematical assumptions, you can also show that in high-dimensional spaces, the distance to **the point nearest you in a data set isn't that much closer than the point farthest from you**



The fact that things spread out in high- dimensional spaces is one manifestation of the **"curse of dimensionality"** (every good pirate story needs a curse!)

What are the practical implications of this?

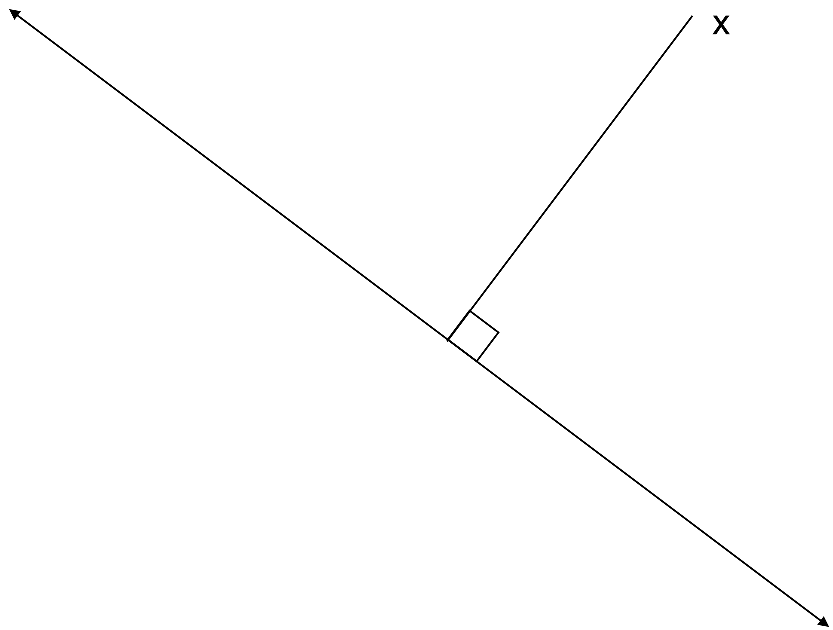
With distance, we also get a right-angle relationship -- Remember the Pythagorean theorem?



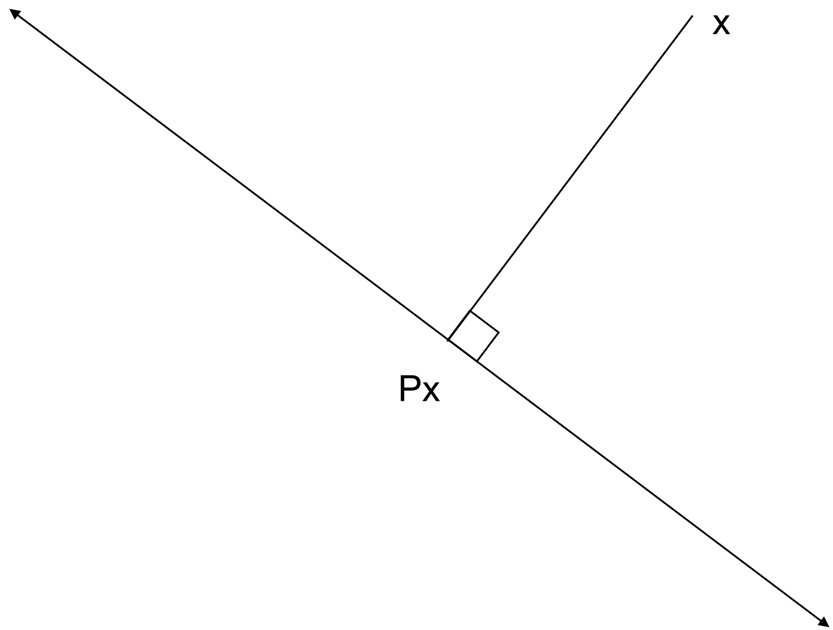
$$\mathrm{dist}\,(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

$$\mathrm{dist}\,(x, y)^2 = \mathrm{dist}(x, z)^2 + \mathrm{dist}(y, z)^2$$

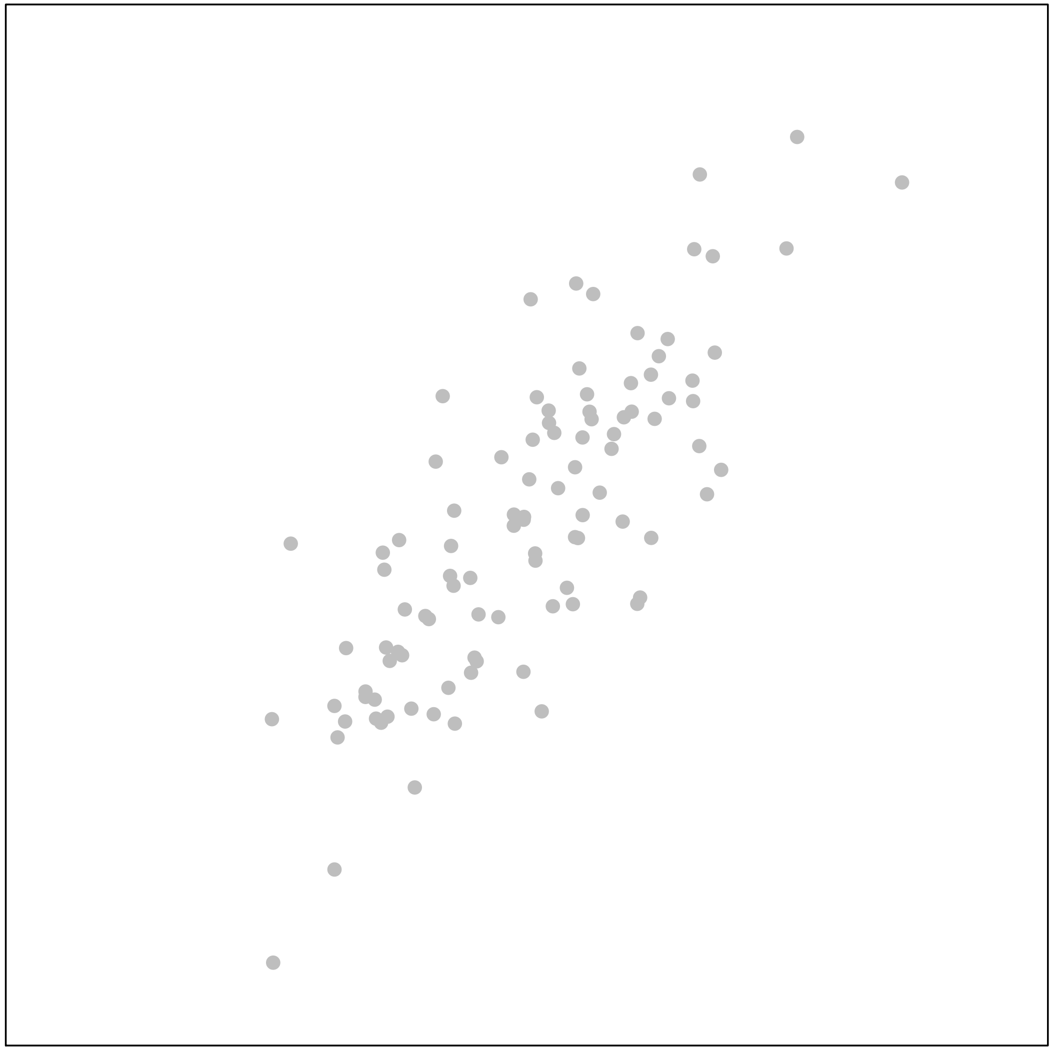
We also might remember right angles appearing when you talk about the nearest point to a line...



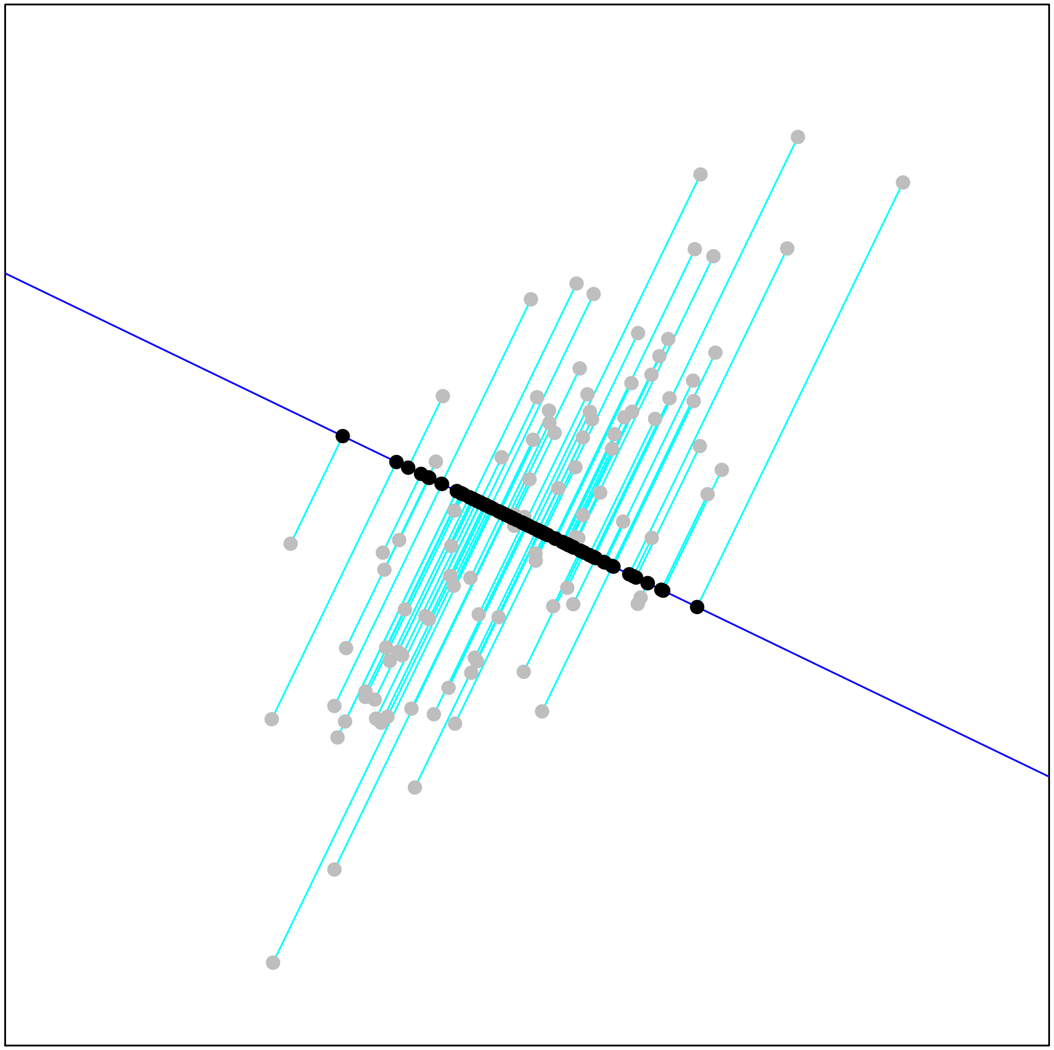
We refer to this point as
the orthogonal projection
of x onto the line...

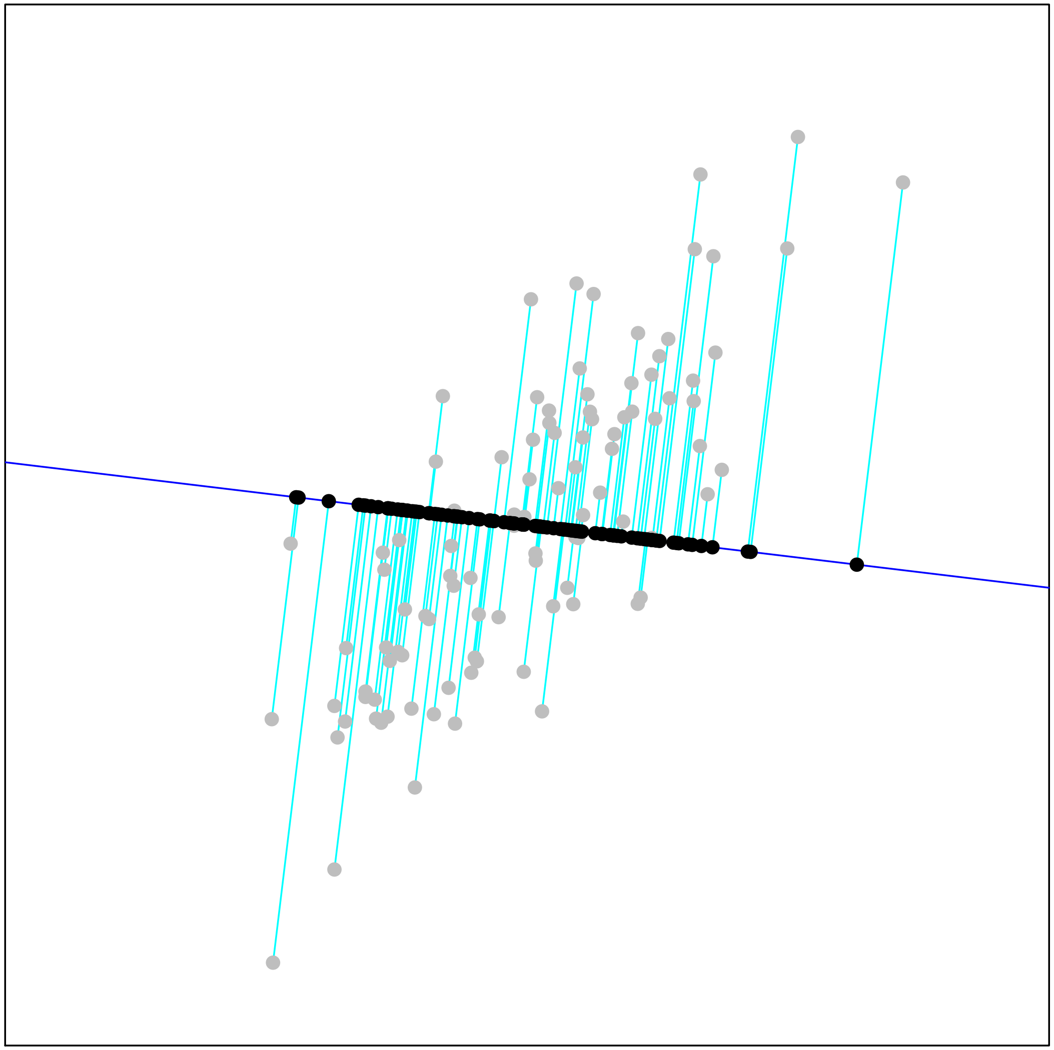


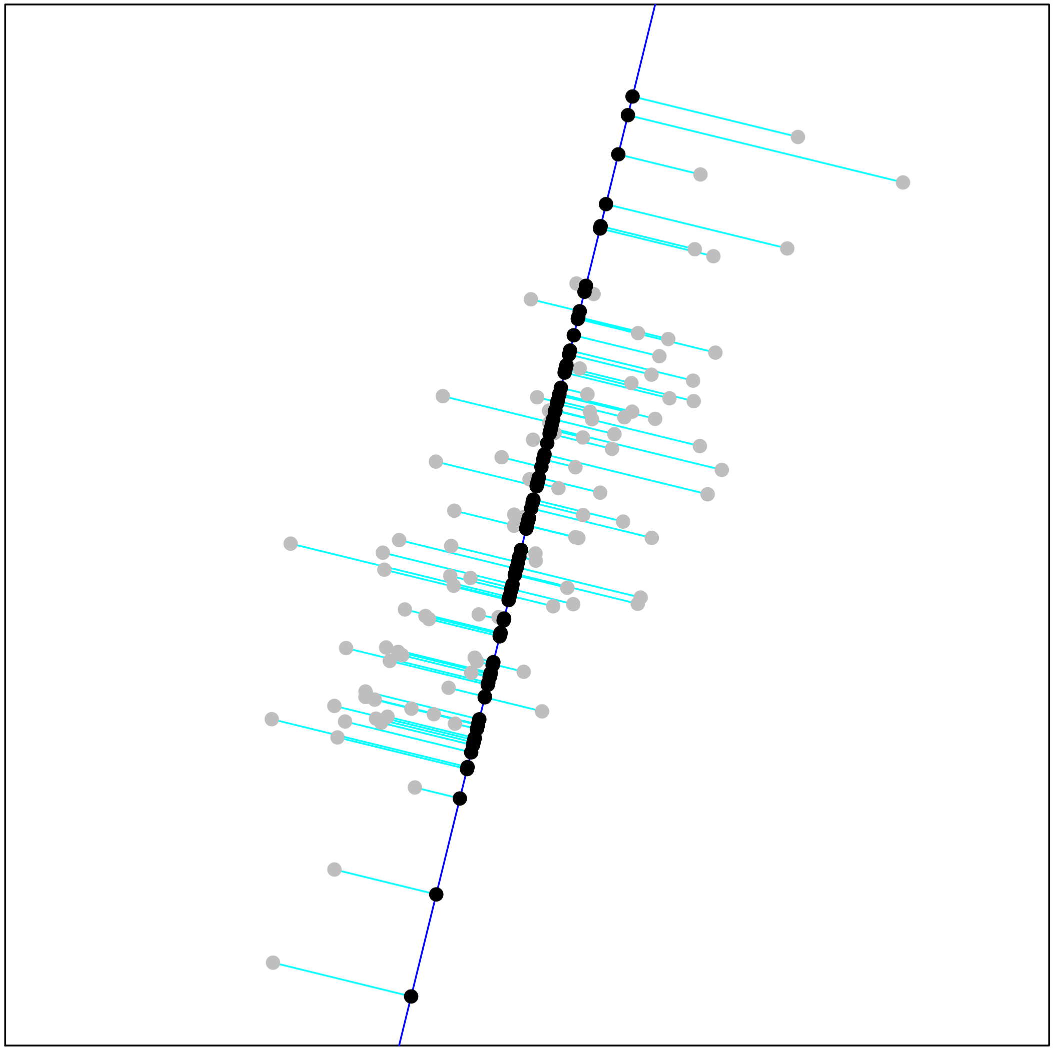
Let’s consider a 2-d data set and projections onto various lines



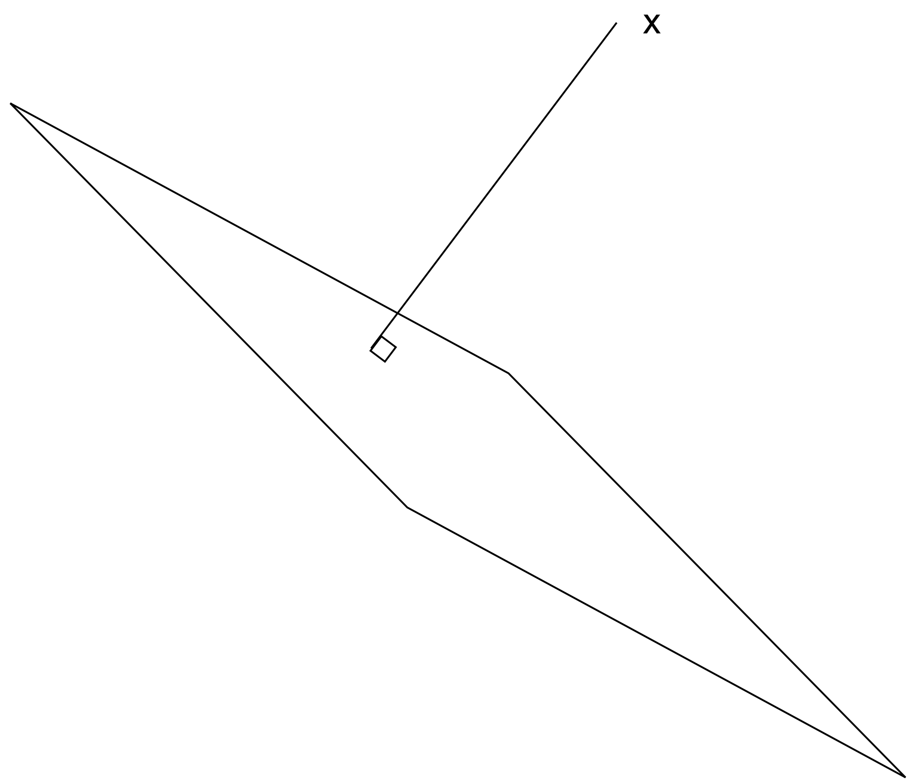
Here is one, indicated by the blue line with the black points indicating the projections of our data



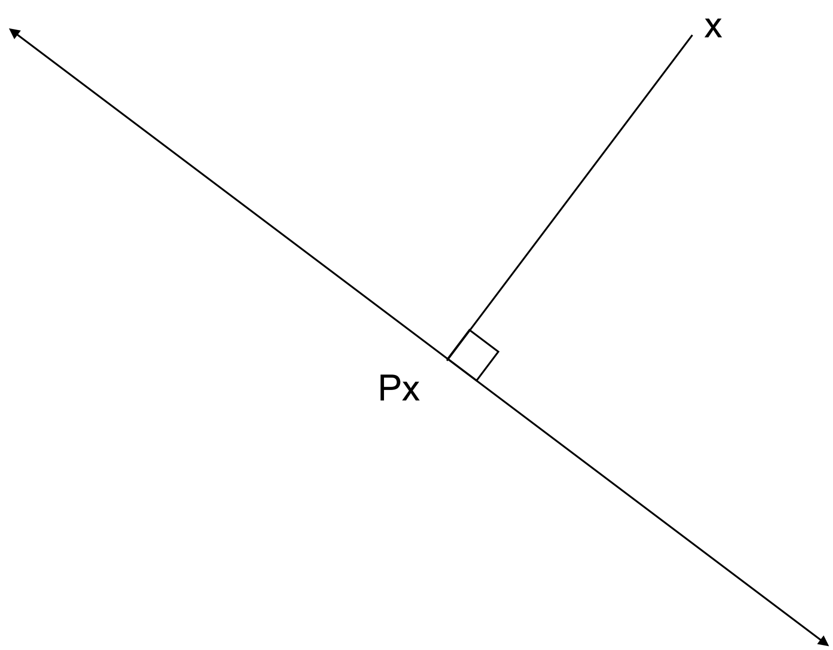


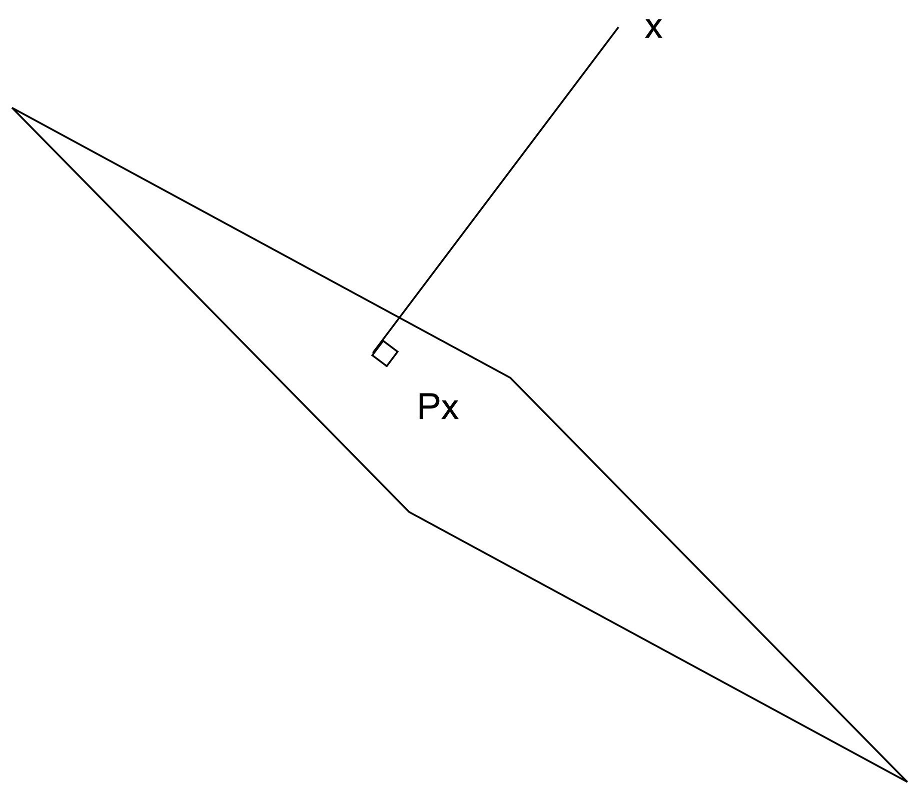


The same essential approach works when you're projecting a point onto a plane instead

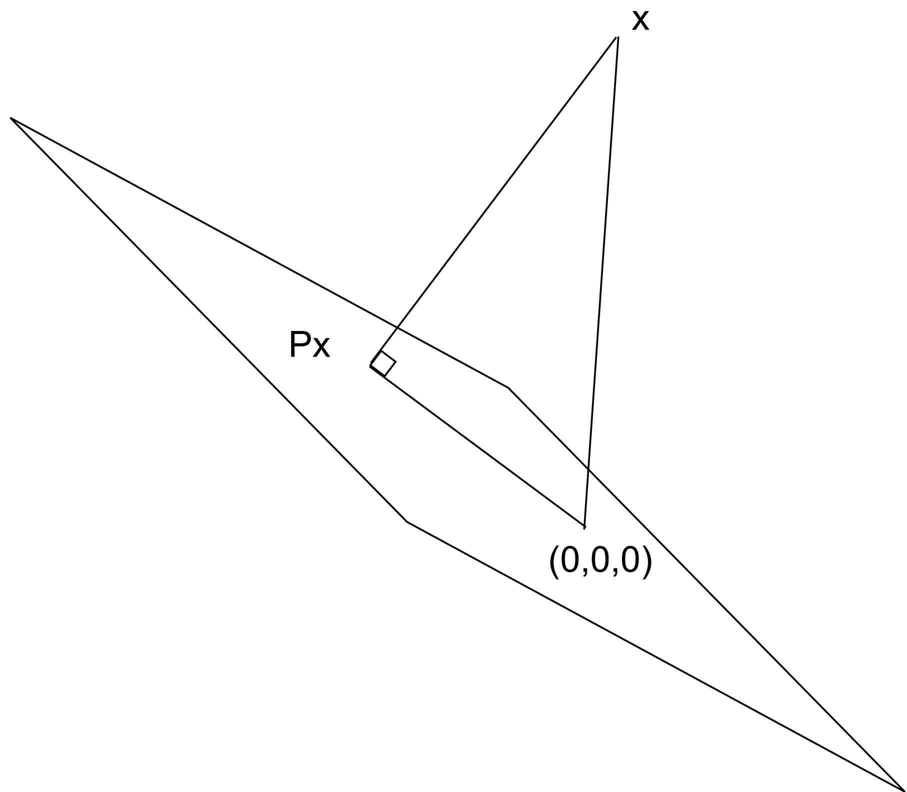


With these pictures, we have another view of the concept of projection -- The projection of a point onto a line (left) or a plane (right) is taken to be the nearest point marked Px





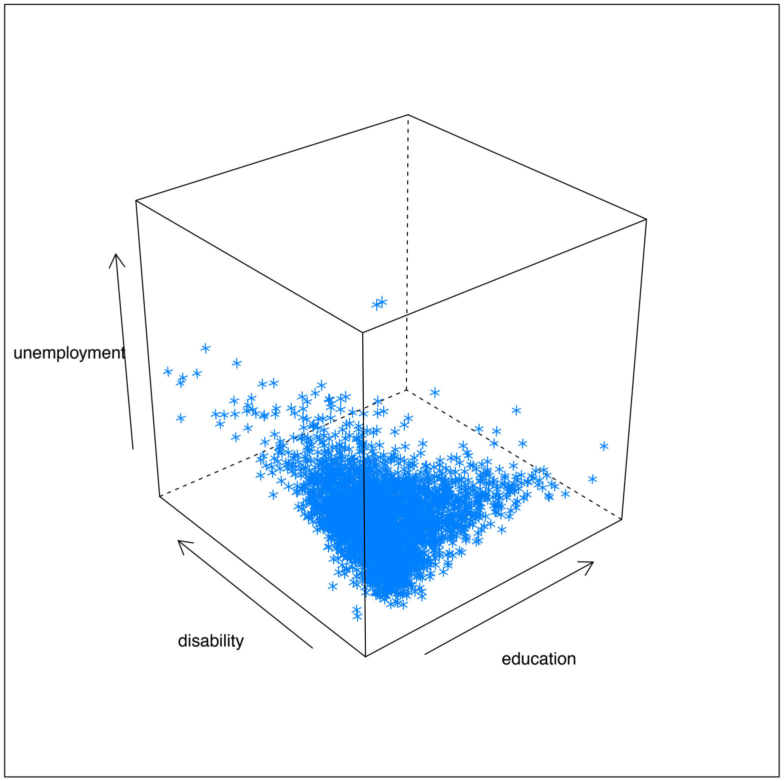
The Pythagorean theorem helps us see things a little more clearly (or not if this doesn't speak to you)

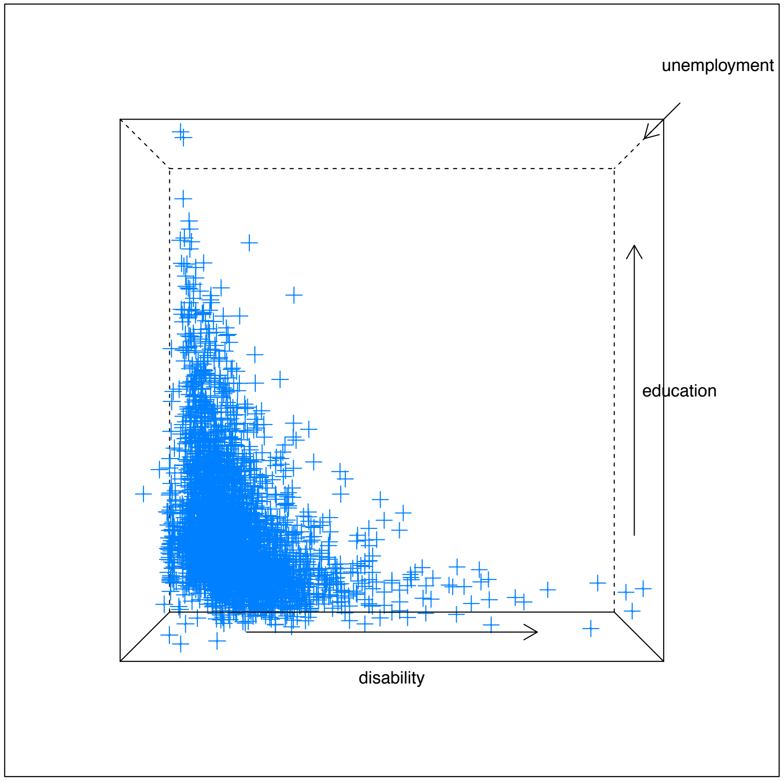


Projections

When we were rotating our 3-d cube and looking at it along different axes, we were projecting the data down to plane spanned by the remaining two variables

Again, the nearest points are simply those directly below, removing the third dimension





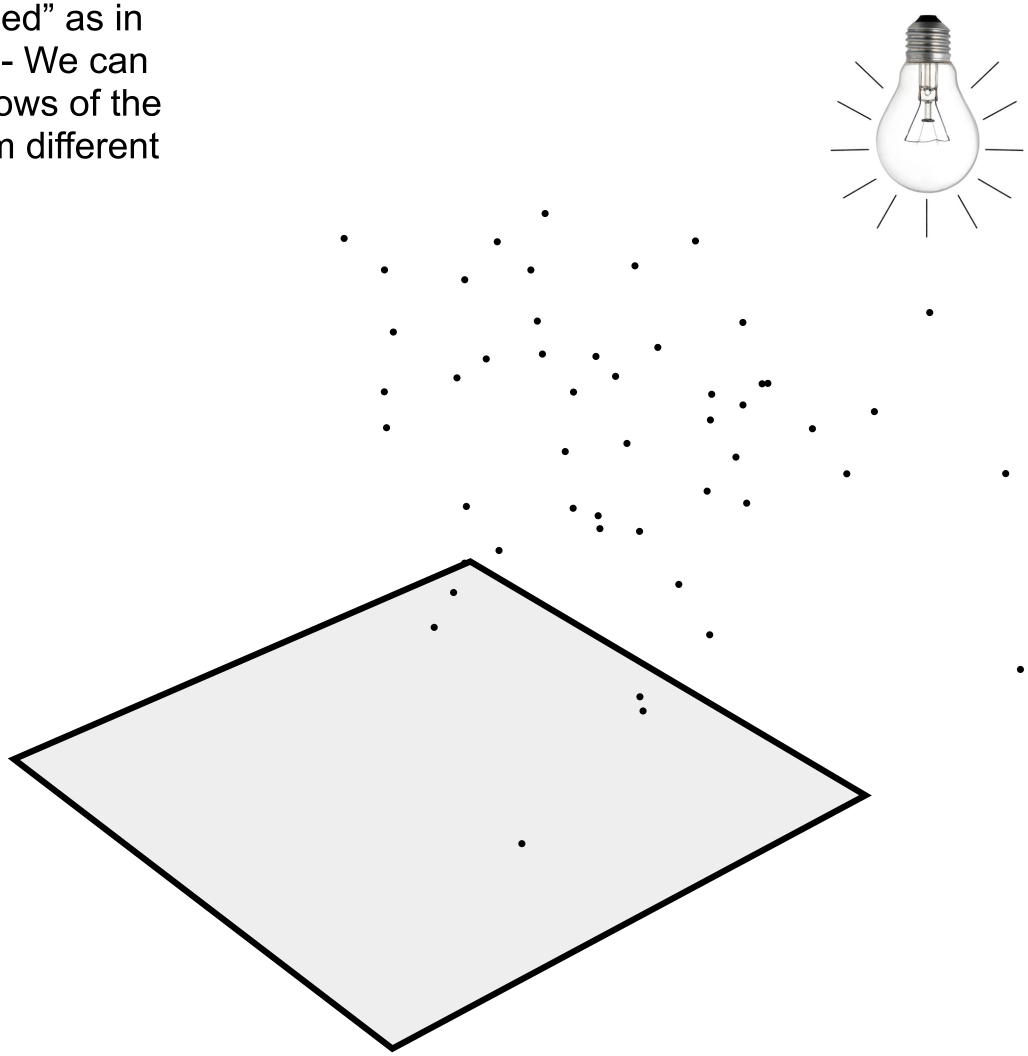
Multiple views

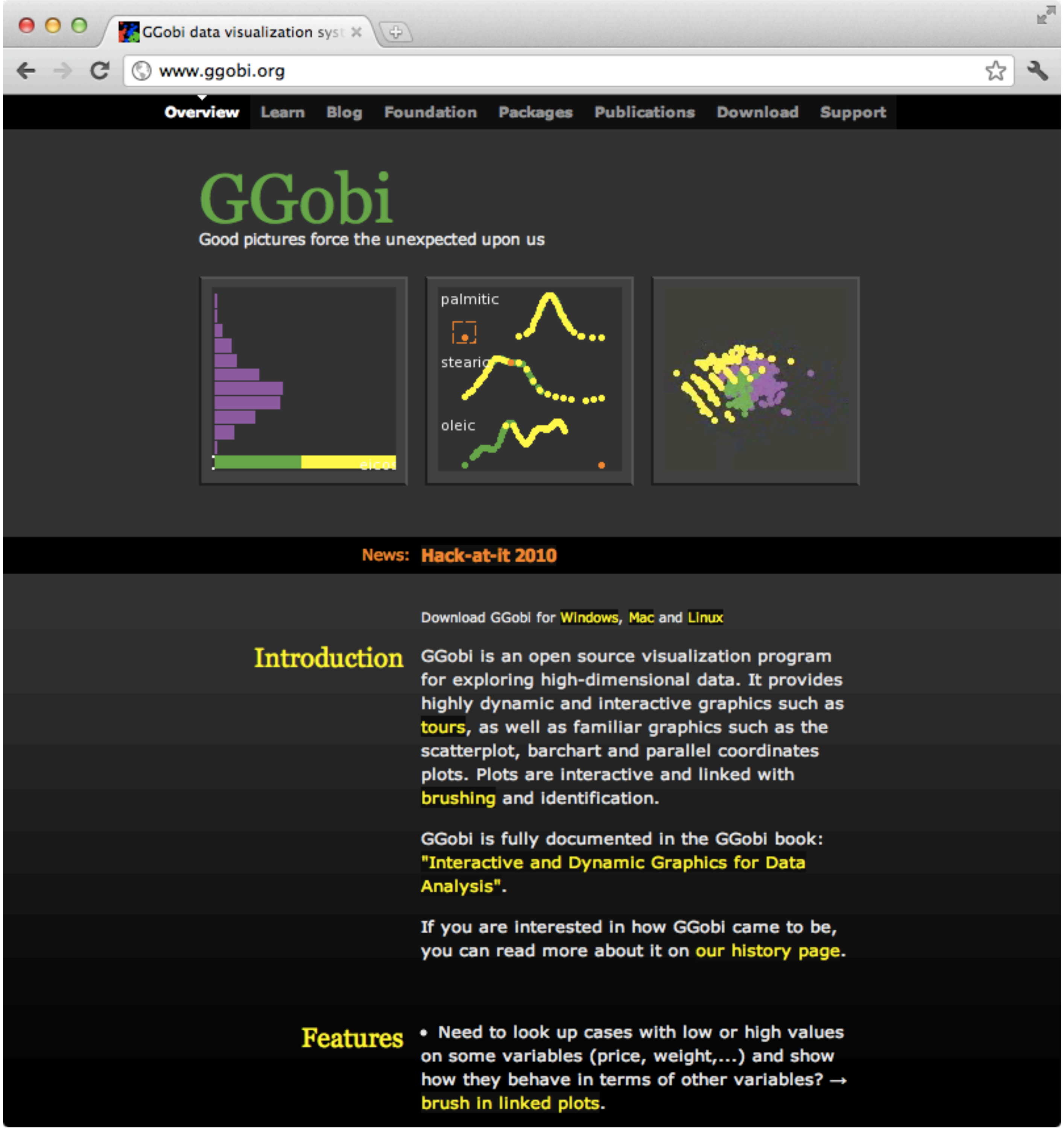
The axis-aligned views are simple to think about but are by no means the end of the story -- We can choose any vantage point from which to look at the data

Why might we investigate these different views? What might they show us? What strategy could we employ to come up with different views?

An alternate interpretation

We can carry this idea farther and examine two-dimensional marginal views of our data set that are not “axis-aligned” as in the scatterplot matrix -- We can consider casting shadows of the data when viewed from different angles




GGobi data visualization syst
www.ggobi.org
Overview
Learn
Blog
Foundation
Packages
Publications
Download
Support
GGobi
Good pictures force the unexpected upon us
palmitic
stearic
oleic
News: Hack-at-it 2010
Download GGobi for Windows, Mac and Linux
Introduction
GGobi is an open source visualization program for exploring high-dimensional data. It provides highly dynamic and interactive graphics such as tours, as well as familiar graphics such as the scatterplot, barchart and parallel coordinates plots. Plots are interactive and linked with brushing and identification.
GGobi is fully documented in the GGobi book: "Interactive and Dynamic Graphics for Data Analysis".
If you are interested in how GGobi came to be, you can read more about it on our history page.
Features
• Need to look up cases with low or high values on some variables (price, weight,...) and show how they behave in terms of other variables? → brush in linked plots.

Multiple views

As you watch the data dance across the screen, we are scanning for directions that are "interesting", providing us with a view into the clustering or grouping of data that might not be immediately evident otherwise

It turns out (a consequence of the Central Limit Theorem) that these projected views of the data will be "uninteresting" in that they will look like a bivariate normal distribution

This, then, becomes one possible definition of "uninteresting" and we can score views by how dissimilar they are from this distribution -- In the late 1970s and early 1980s, this led to a statistical technique known as projection pursuit

Viewing indices were designed to respond to various features in a scatter (say, the presence of holes) -- The Grand Tour then becomes a kind of stochastic search for these "interesting" aspects of the data

Let's talk a little about what we mean by clustering…