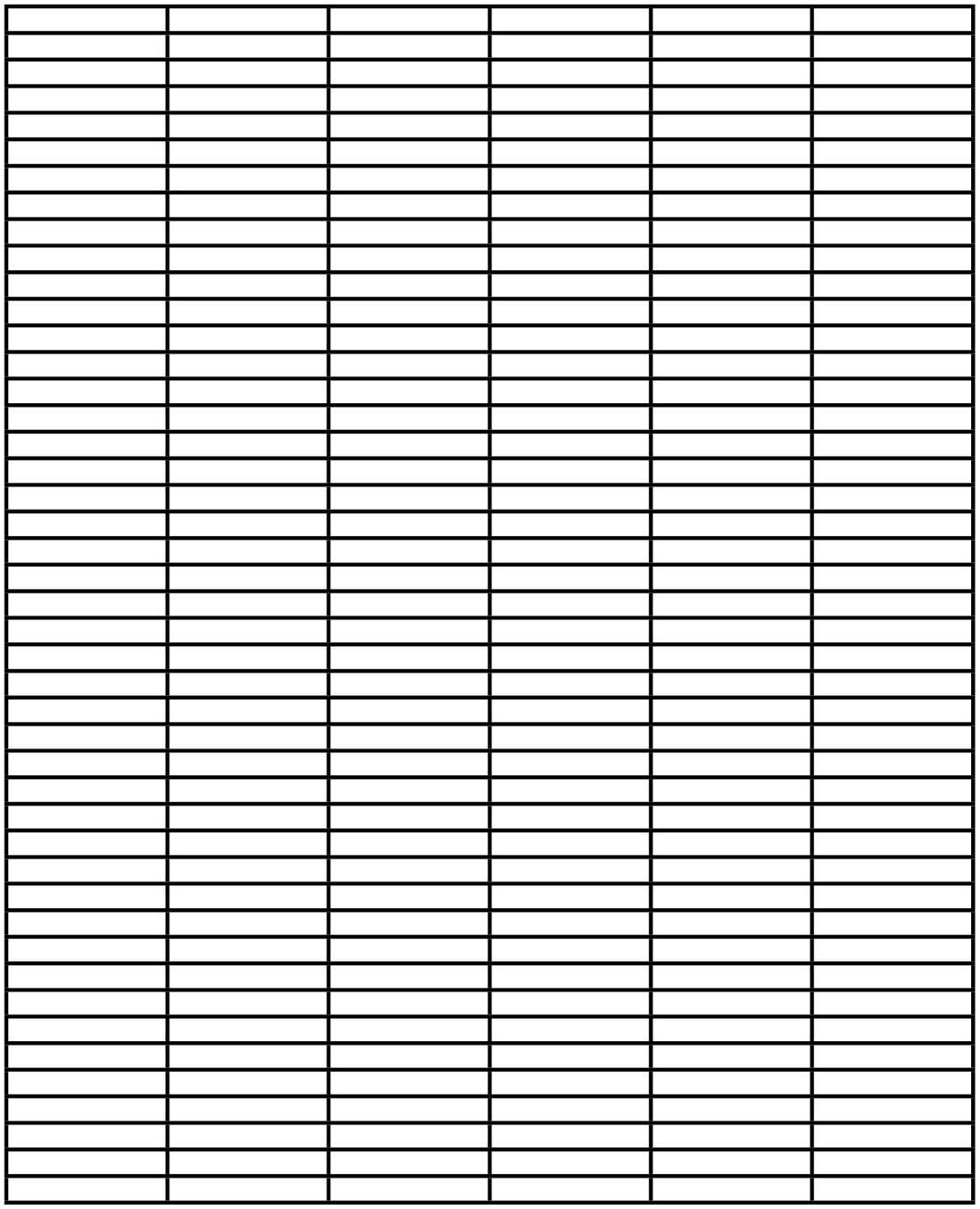
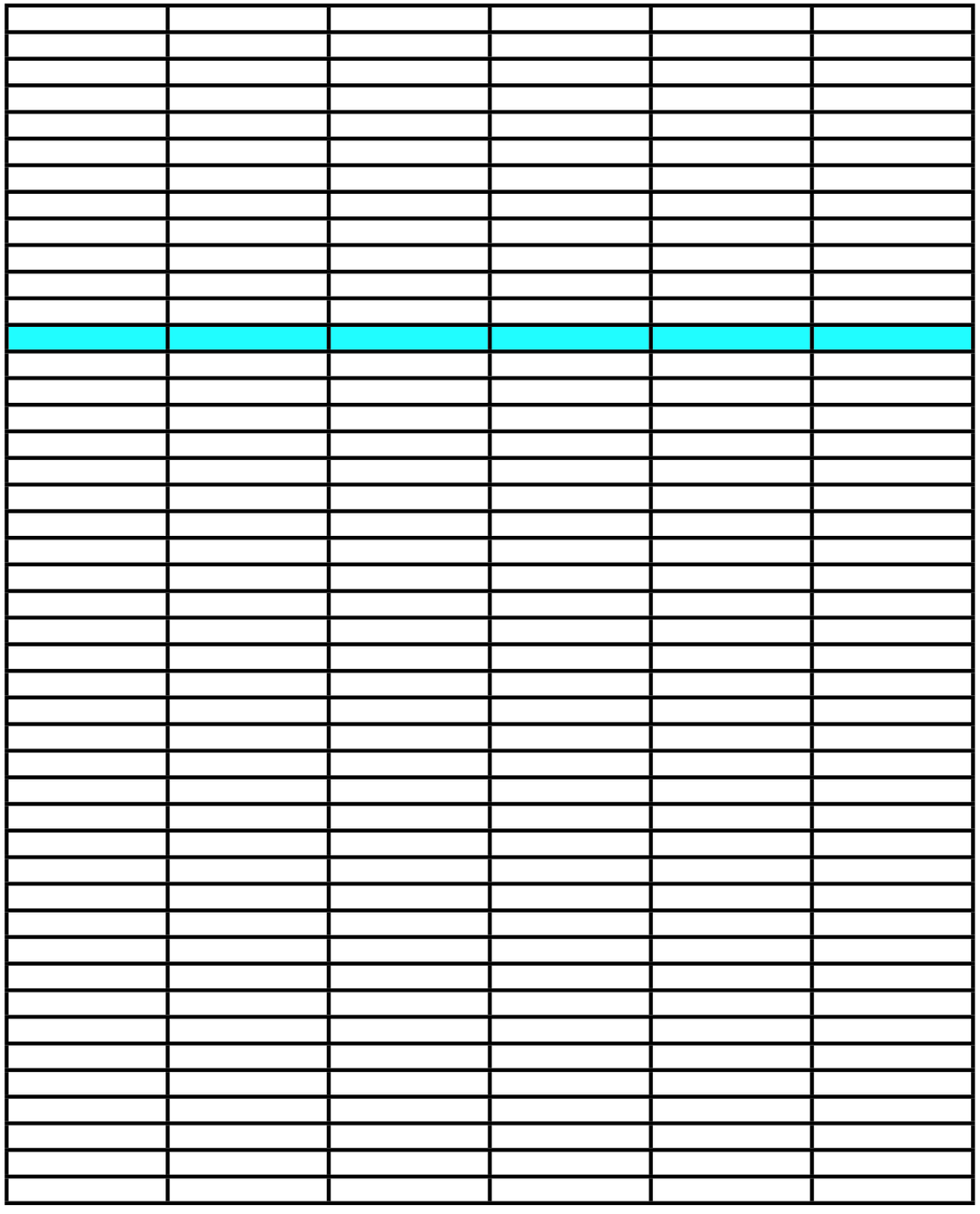
Working with Scatterplots — Ode to the humble `geom_point( )`



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

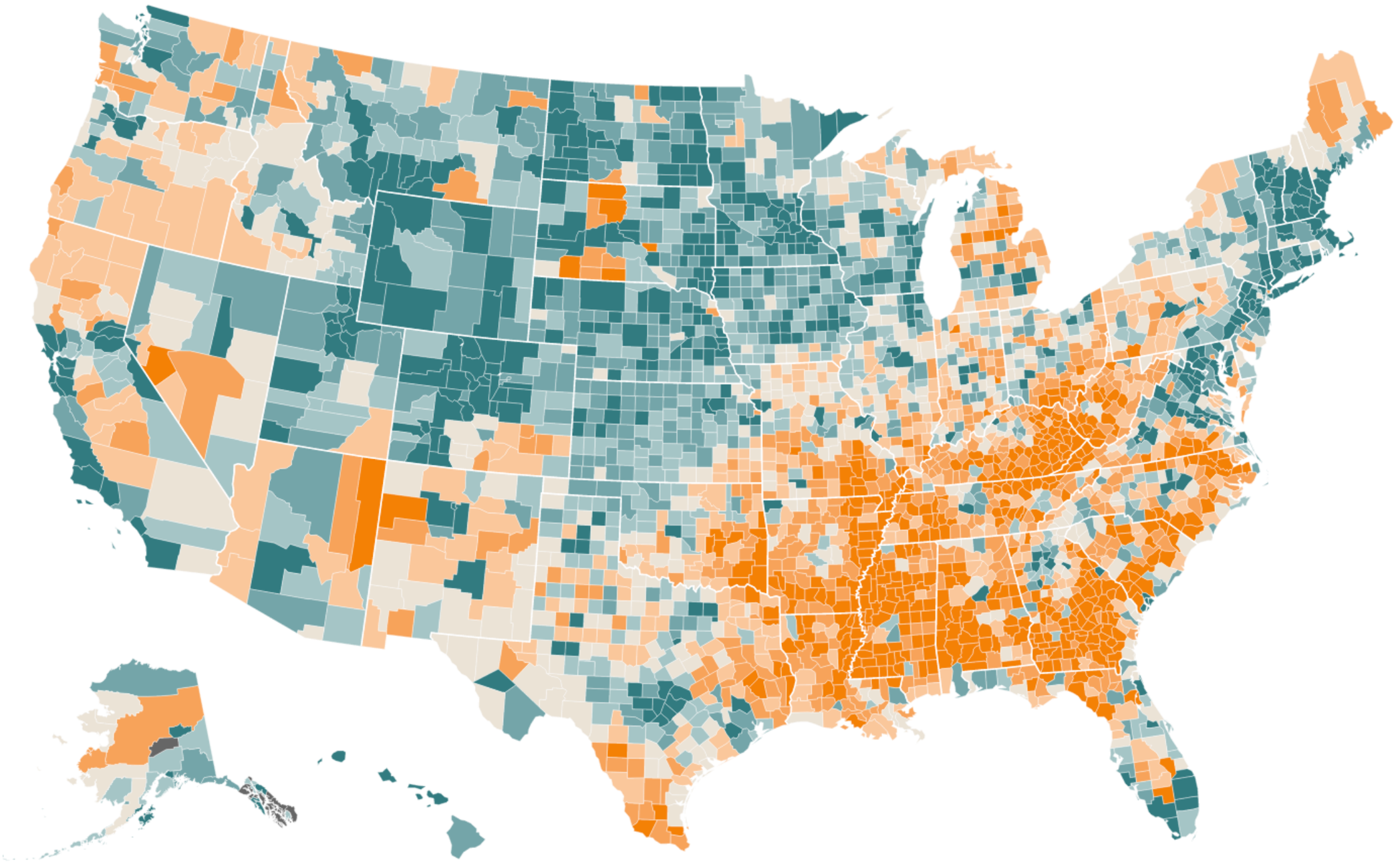


Each row represents a point in d-dimensional Euclidean space

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

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

Alan Flippen @alflip JUNE 26, 2014



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

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

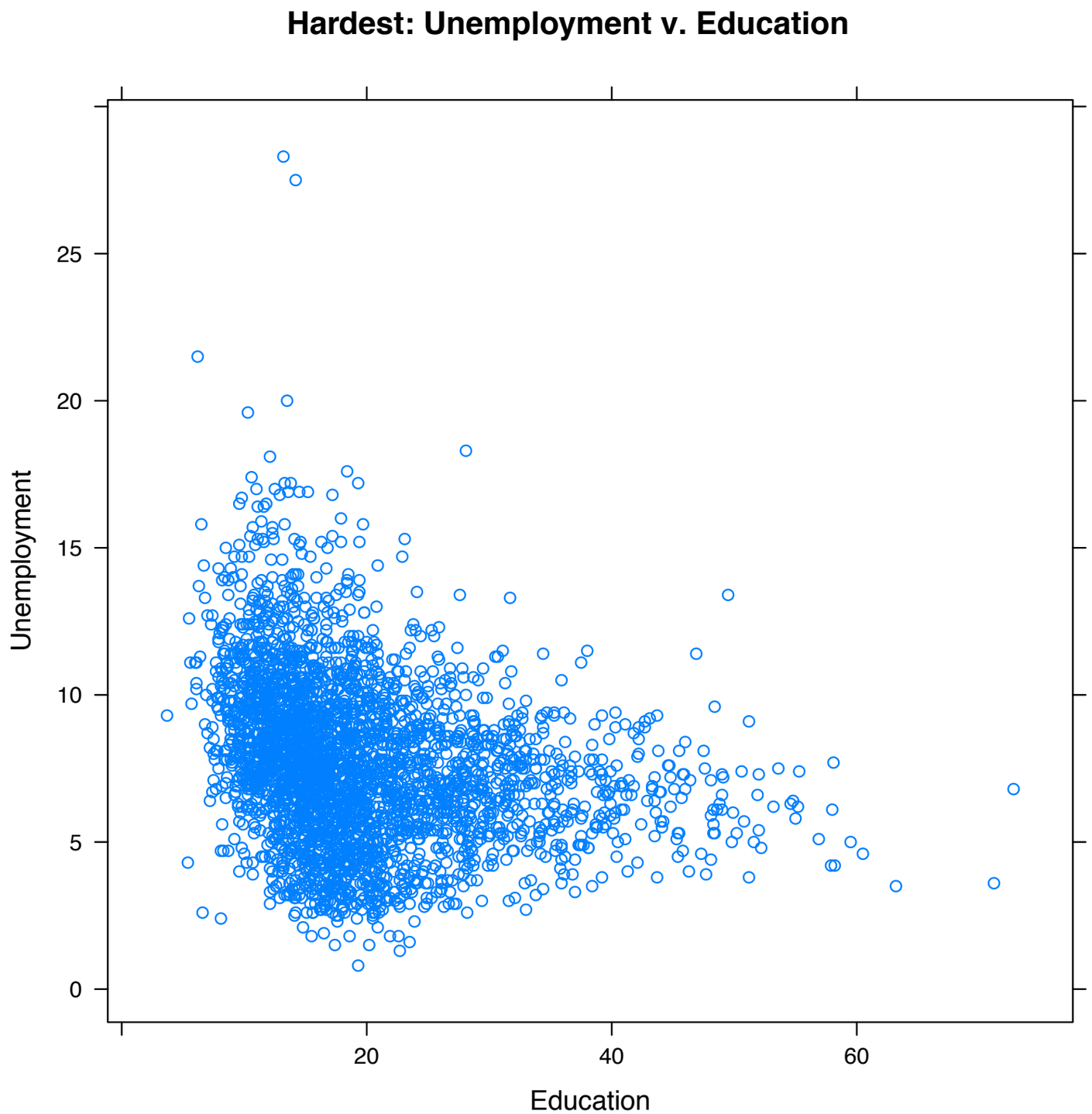
What kinds of variables do we have here? Qualitative or Quantitative?

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

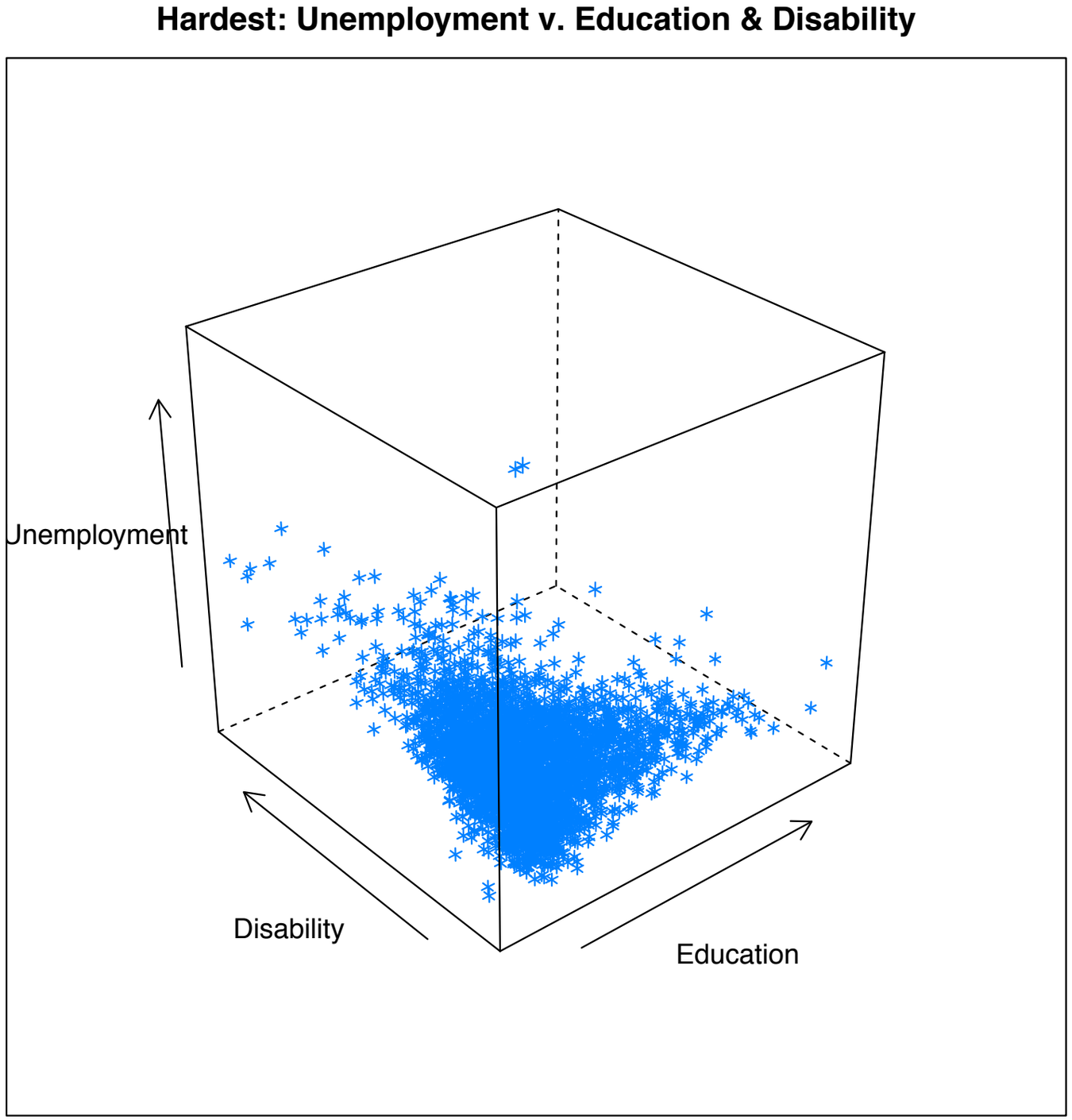
If we had only two quantitative variables in our data set, we would do the (now) obvious thing of simply plotting one variable against another

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



Geometry

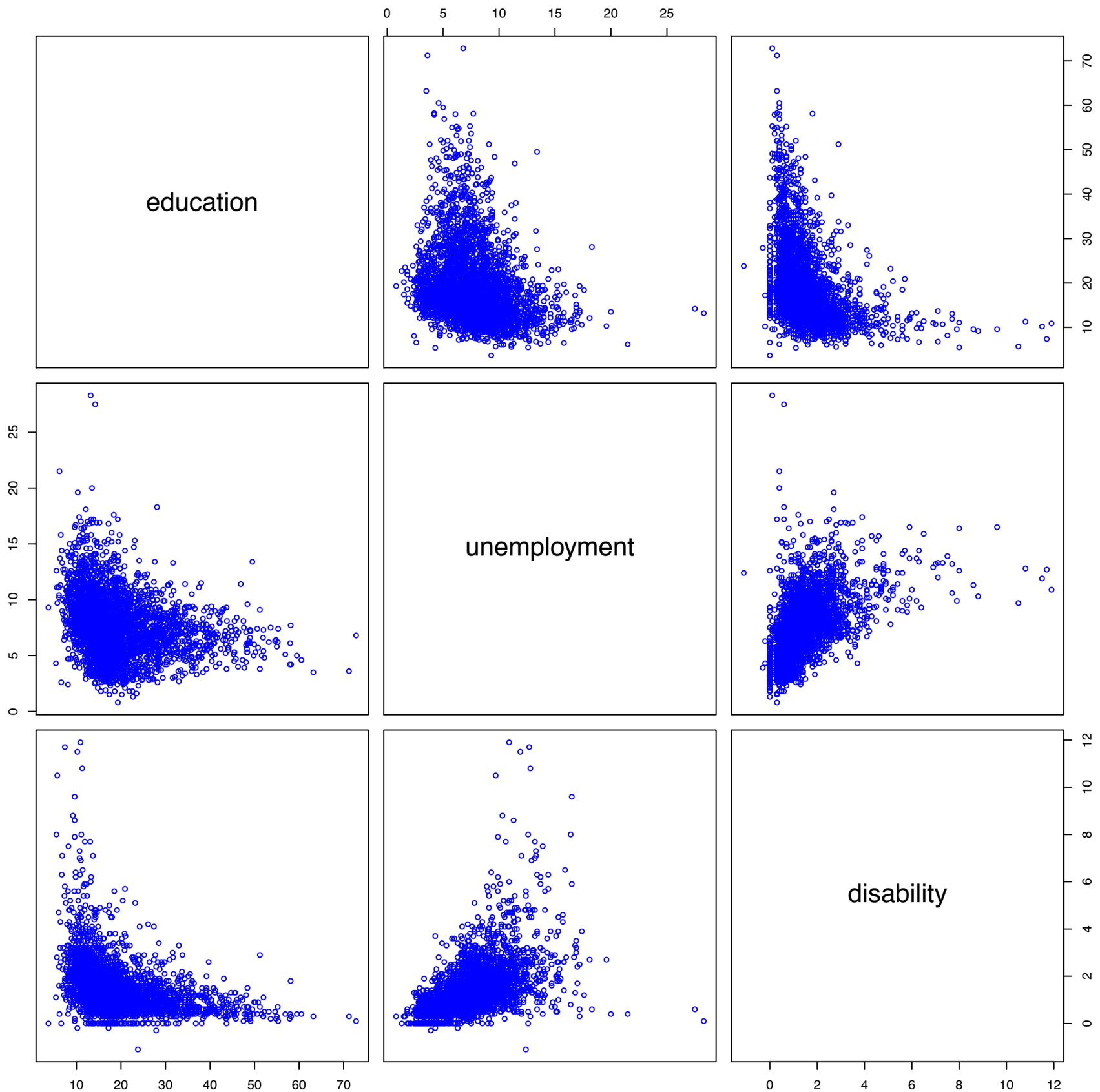
With 2- and 3-dimensional data, we can **invoke a spatial metaphor** and create artificial axes to plot two variables against each other

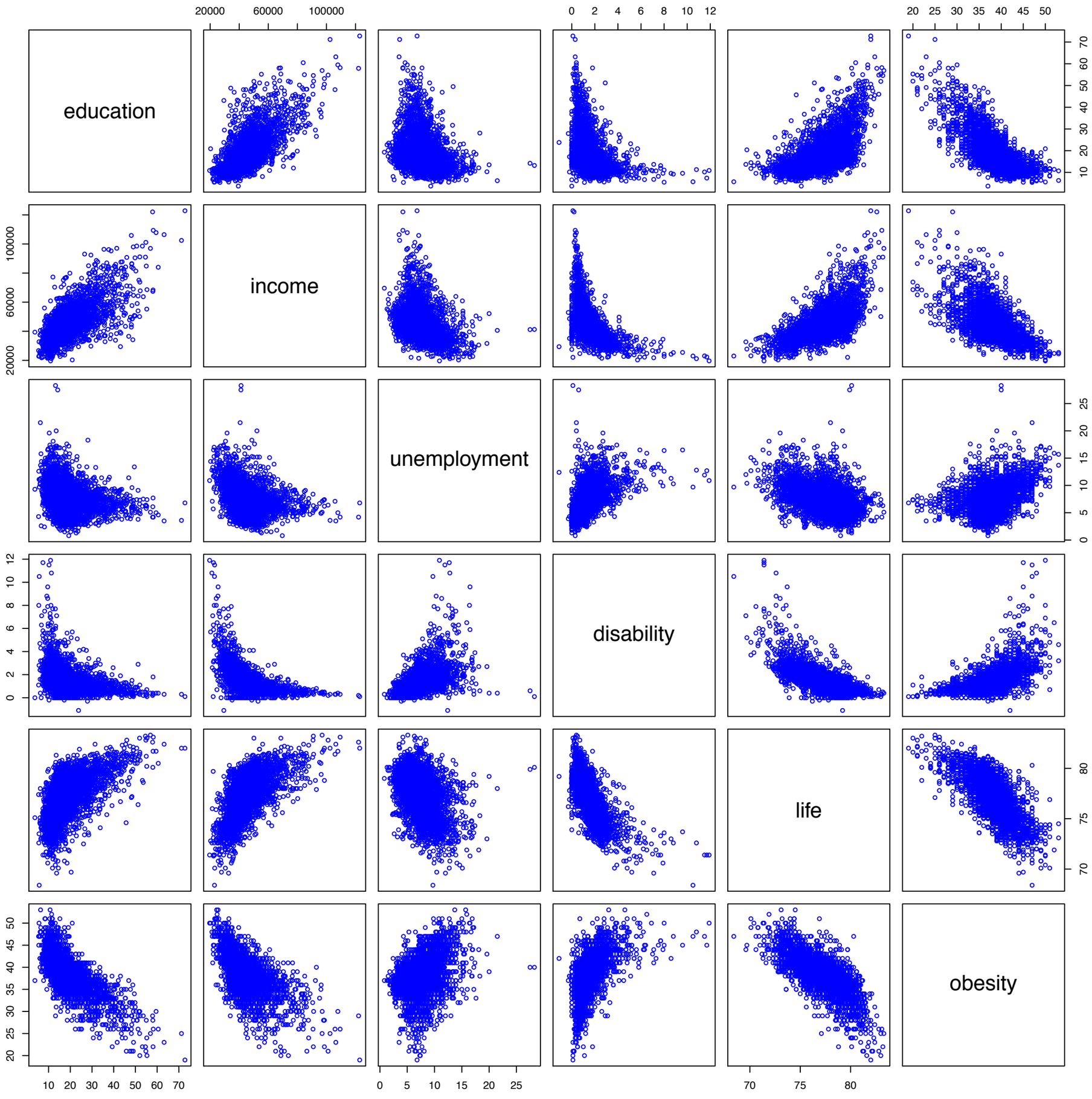
In each case, we are associating **the “x”, “y” and “z” axes with a different variable** or column in the dataset and then plotting locating each row in our table using this coordinate system — the first data point is at (0.75, 2.3, -0.71), for example, or 0.75 out along the HDI axis, 2.3 units along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

The obvious question is what do you do when you have more than three variables on each observational unit (row)? 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

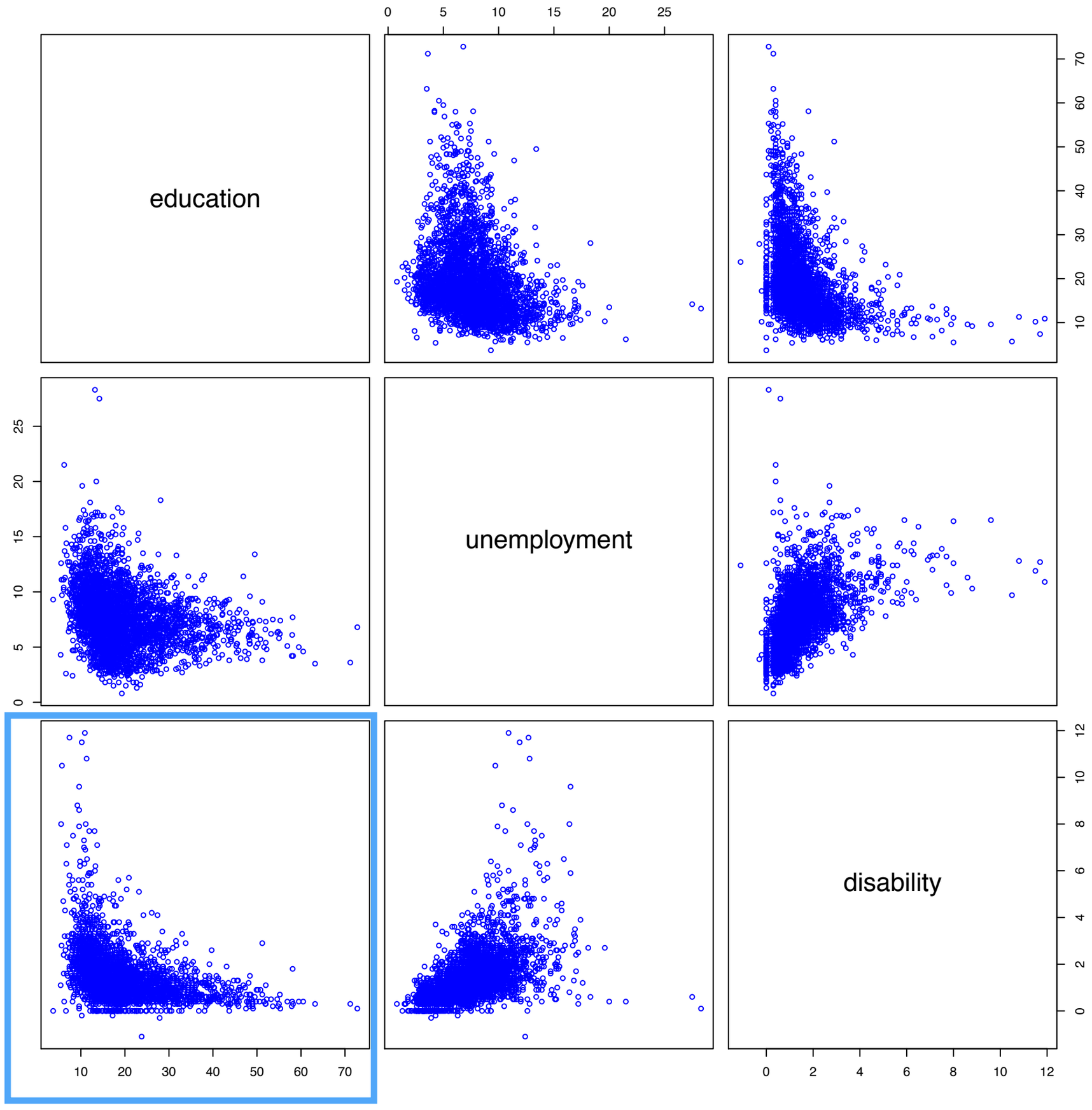
The resulting **scatterplot matrix** represents all the pairwise relationships between columns in our data set at one time




education
income
unemployment
disability
life
obesity

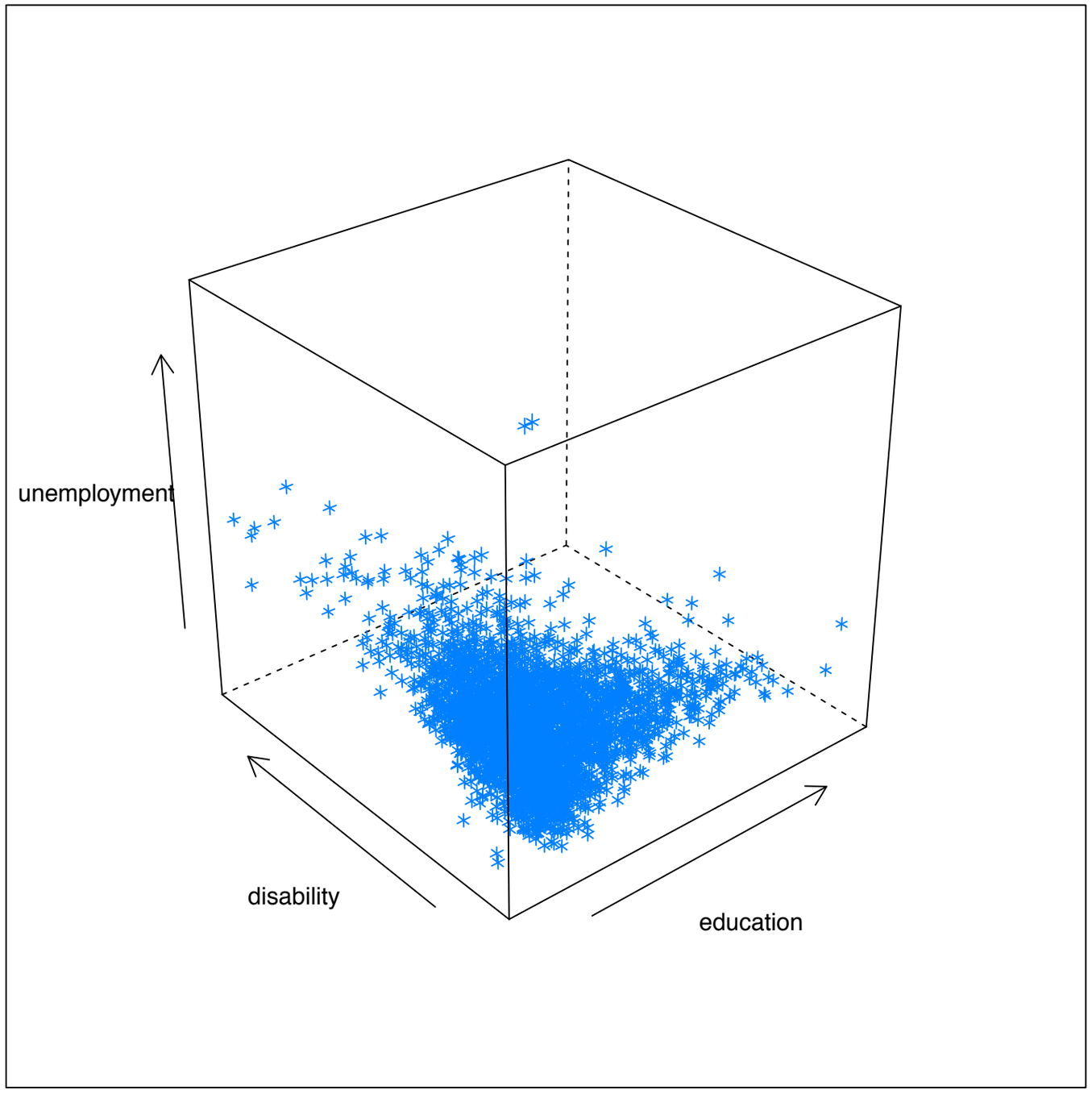
Just to be super clear here, if we had 3 variables in our data set, each entry in the scatterplot matrix represents an extreme view on the data -- That is, we take our 3-dimensional box of data and look at it along different axes

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



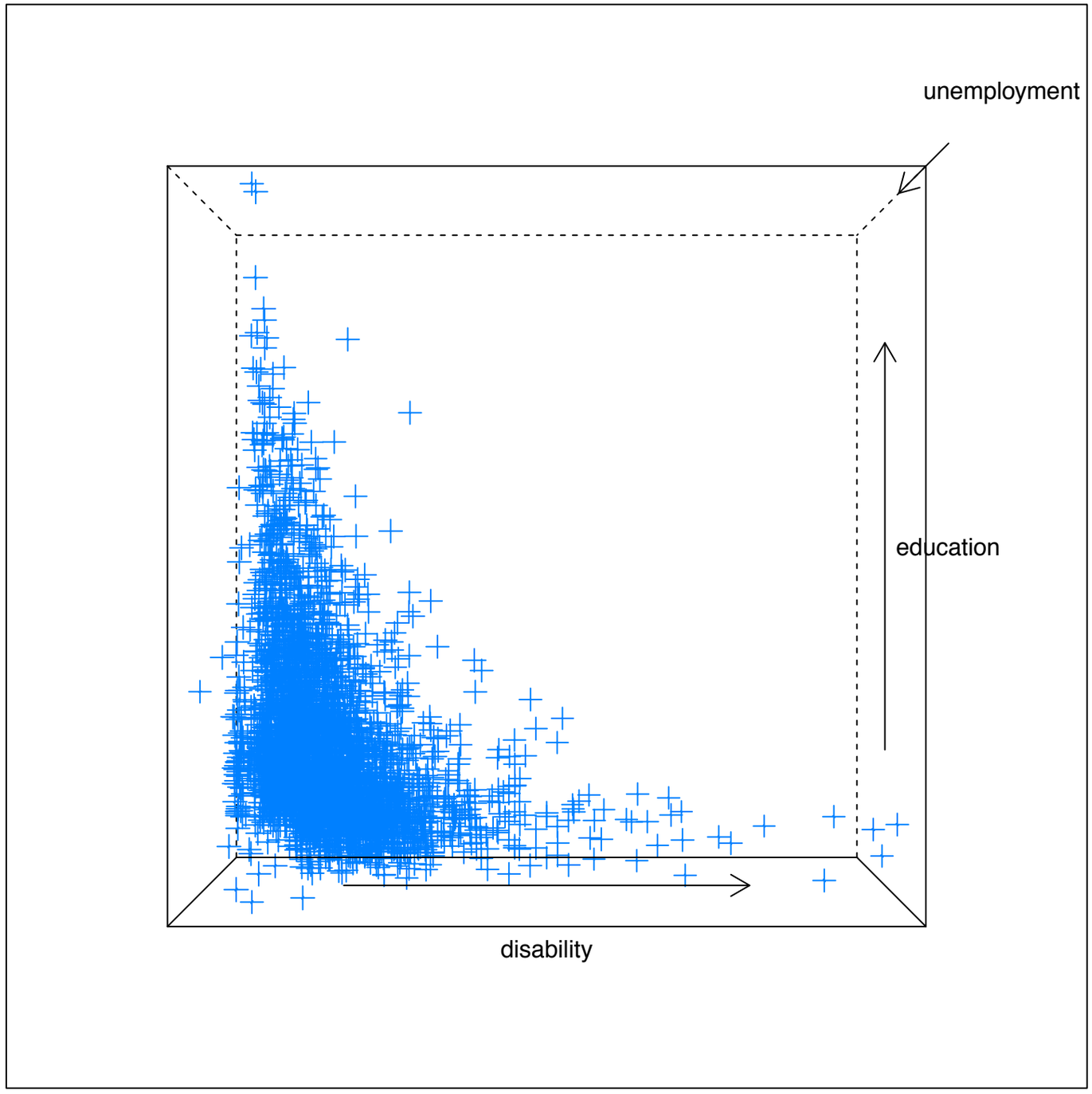
Just to be super clear here, if we had 3 variables in our data set, each entry in the scatterplot matrix represents an extreme view on the data -- That is, we take our 3-dimensional box of data and look at it along different axes

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

Thinking about data as a table is a reduction — there are other structures people use that offer greater flexibility

A Mongo data base, for example, stores JSON strings (which you can think of as the basic Python data types of dictionaries, lists, numeric and character data, etc.)

But reducing things to a table means we have rows and columns — each row is a unit of observation and each column represents some measurement or characteristic related to that unit — and that reduction will let us invoke concepts from (essentially) high school geometry

So far we've looked at 2- and 3-dimensional space, or d-space via a series of 2-dimensional "marginal" plots (although you could imagine a 3-d version of a scatterplot matrix)

The notion of "nearby" will hep 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

The most popular database for

mongodb.com

Nominate a project for the MongoDB Innovation Awards today!

mongoDB.

The database for modern applications

MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era.

No database makes you more productive.

Try MongoDB free in the cloud!

Start free

The most popular database for

mongodb.com

mongoDB.

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  },
  "hobbies": ["surfing", "coding"]
}
```

Rich JSON D

- The most natural an
- Supports arrays and
- Allows for flexib ar

MBot from MongoDB

Hi there!

We have product specialists ready to help with MongoDB. Do you want to talk to them?

Talk to a specialist now

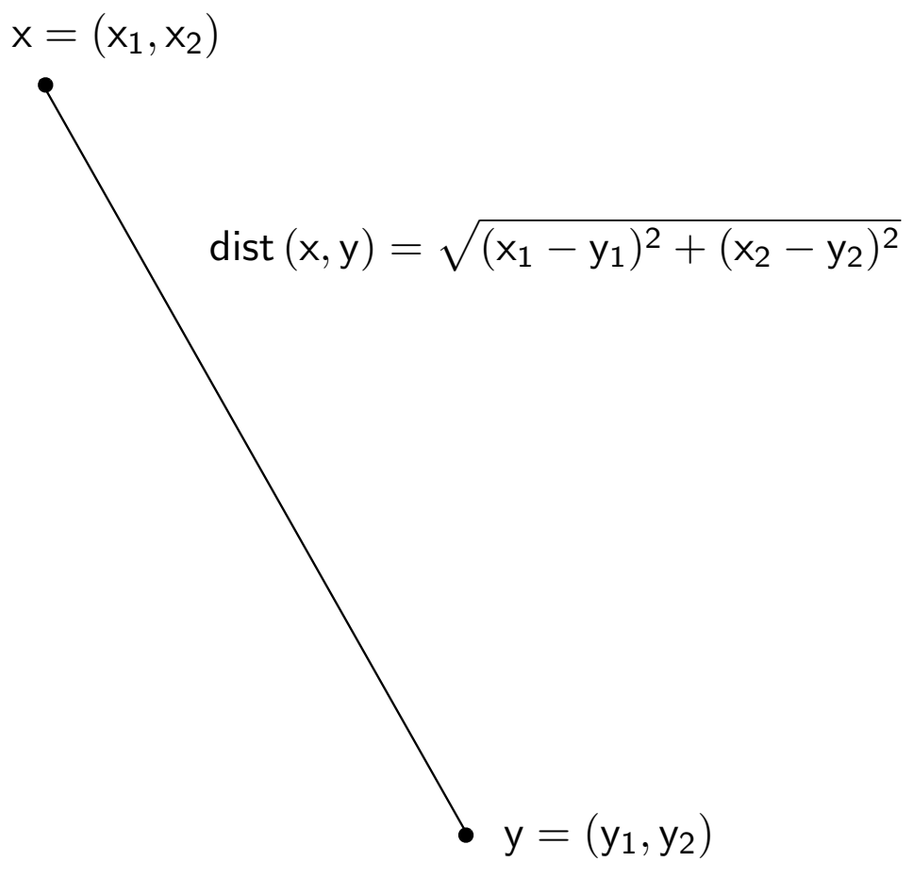
No, I'm just browsing

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

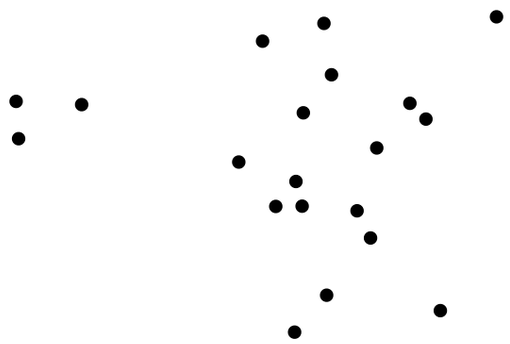
x

y

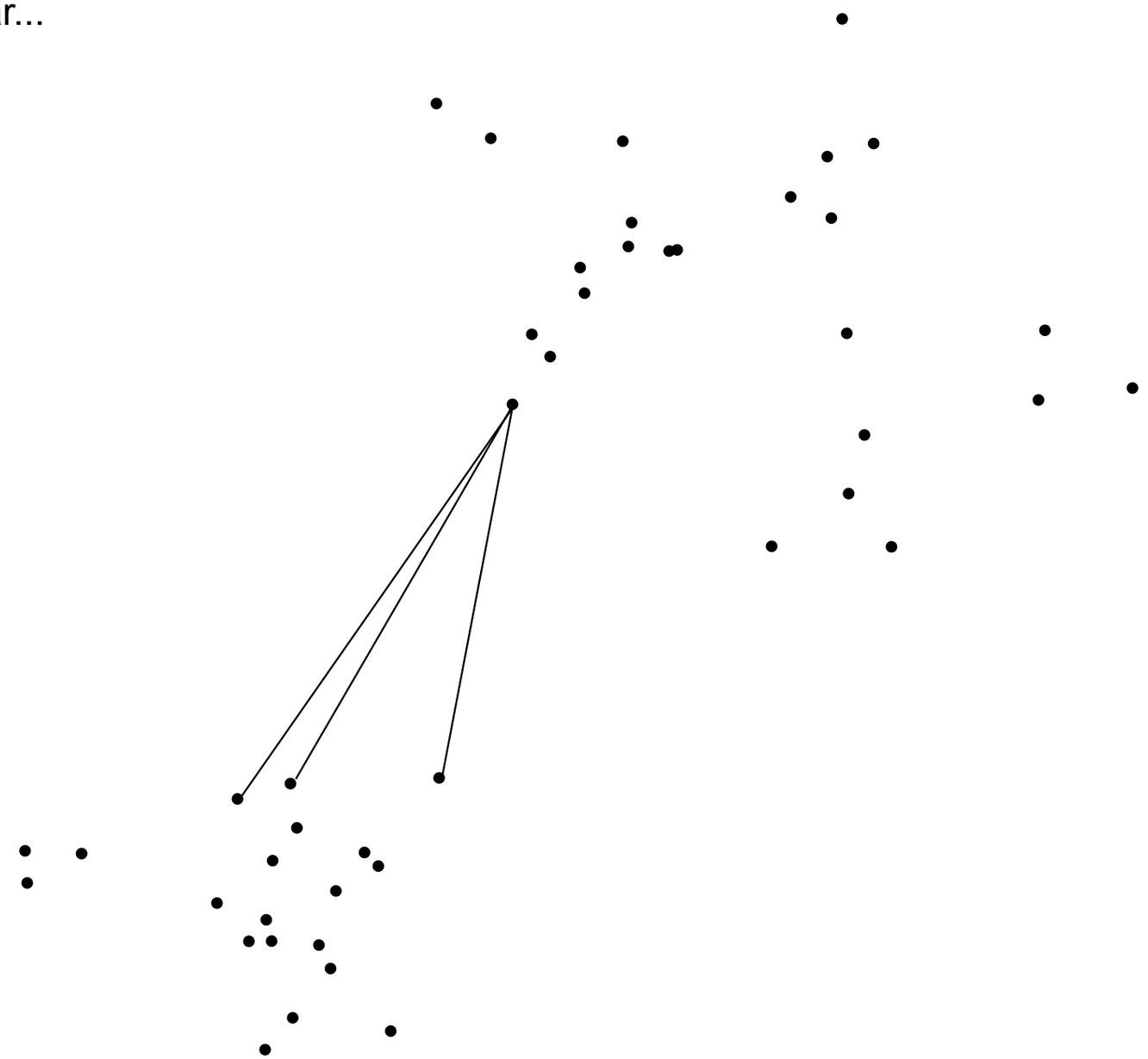
We can also talk how far apart they are using standard Euclidean distance



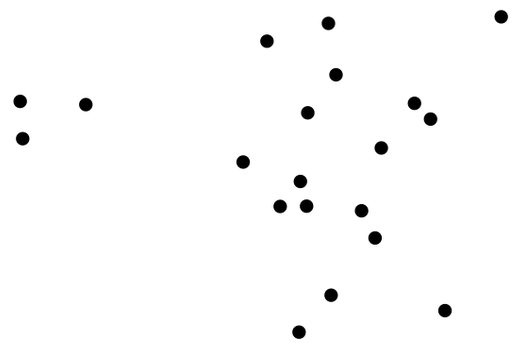
With distance, we can compare points based on whether they are far...



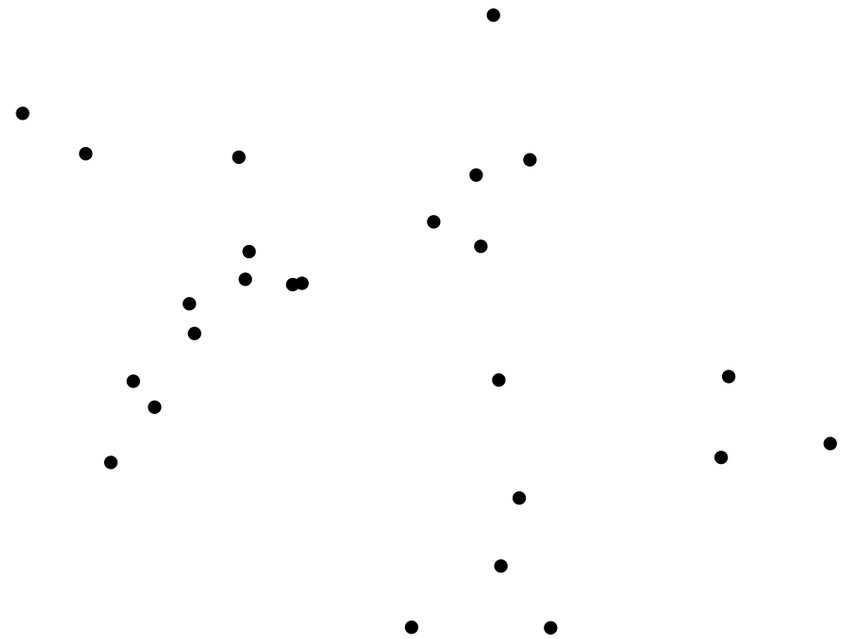
With distance, we can compare points based on whether they are far...

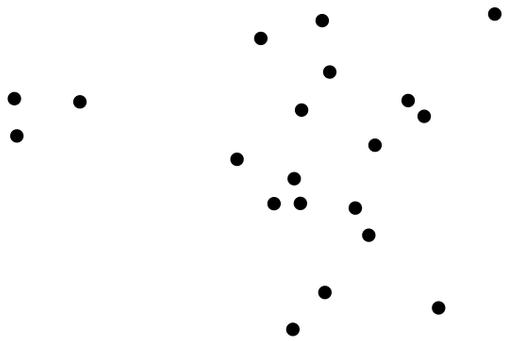


With distance, we can compare points based on whether they are far or near

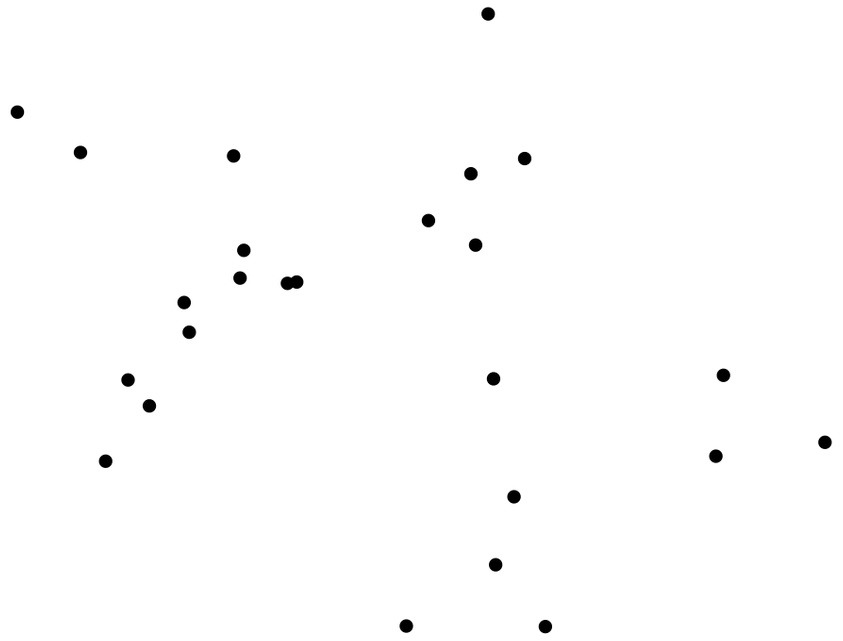


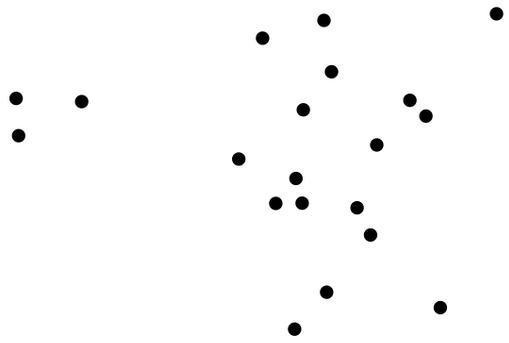
Which leads us to the idea of clusters, points that fall naturally into groups based on proximity



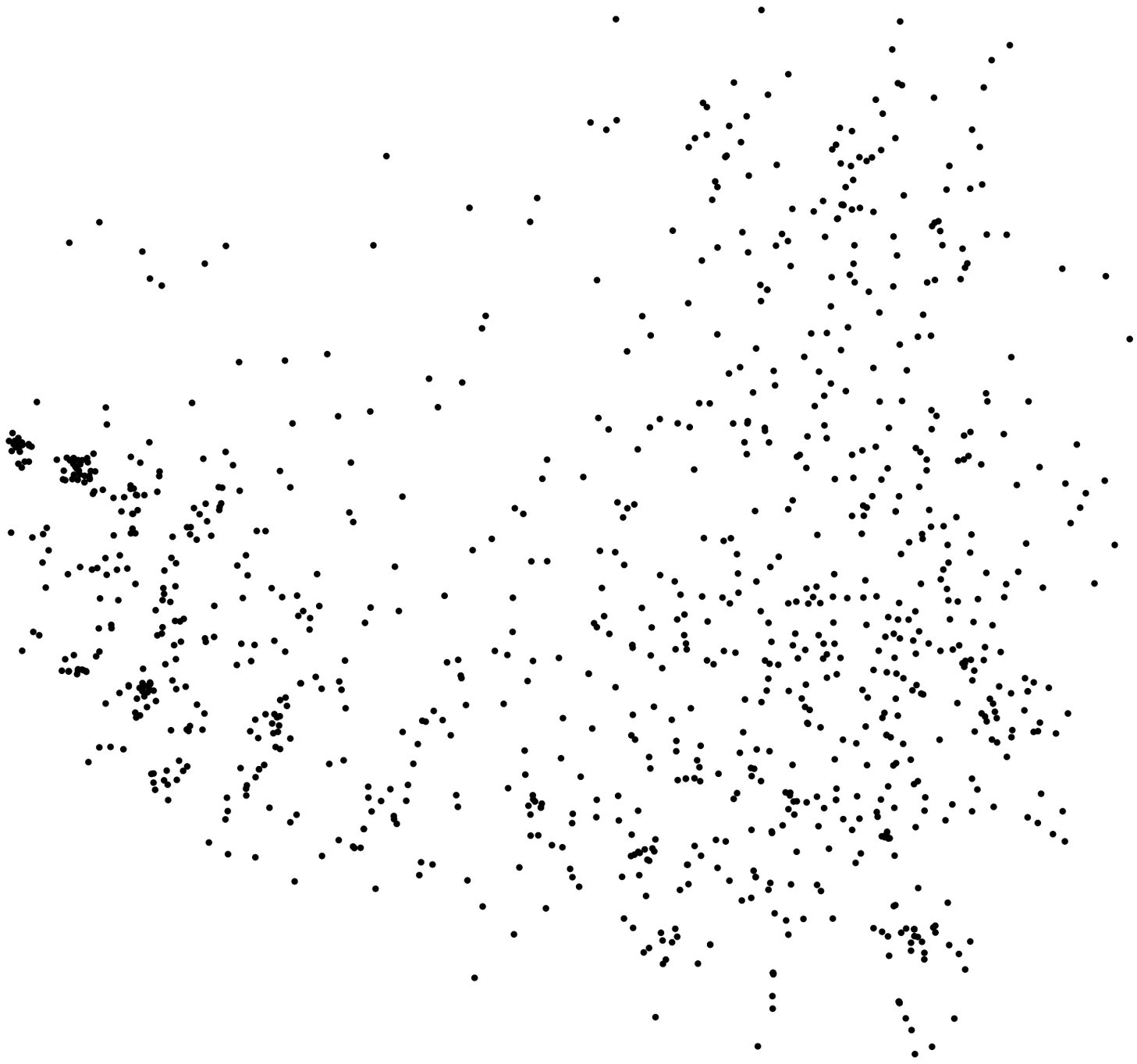


How many clusters do you see here? How do you identify them?

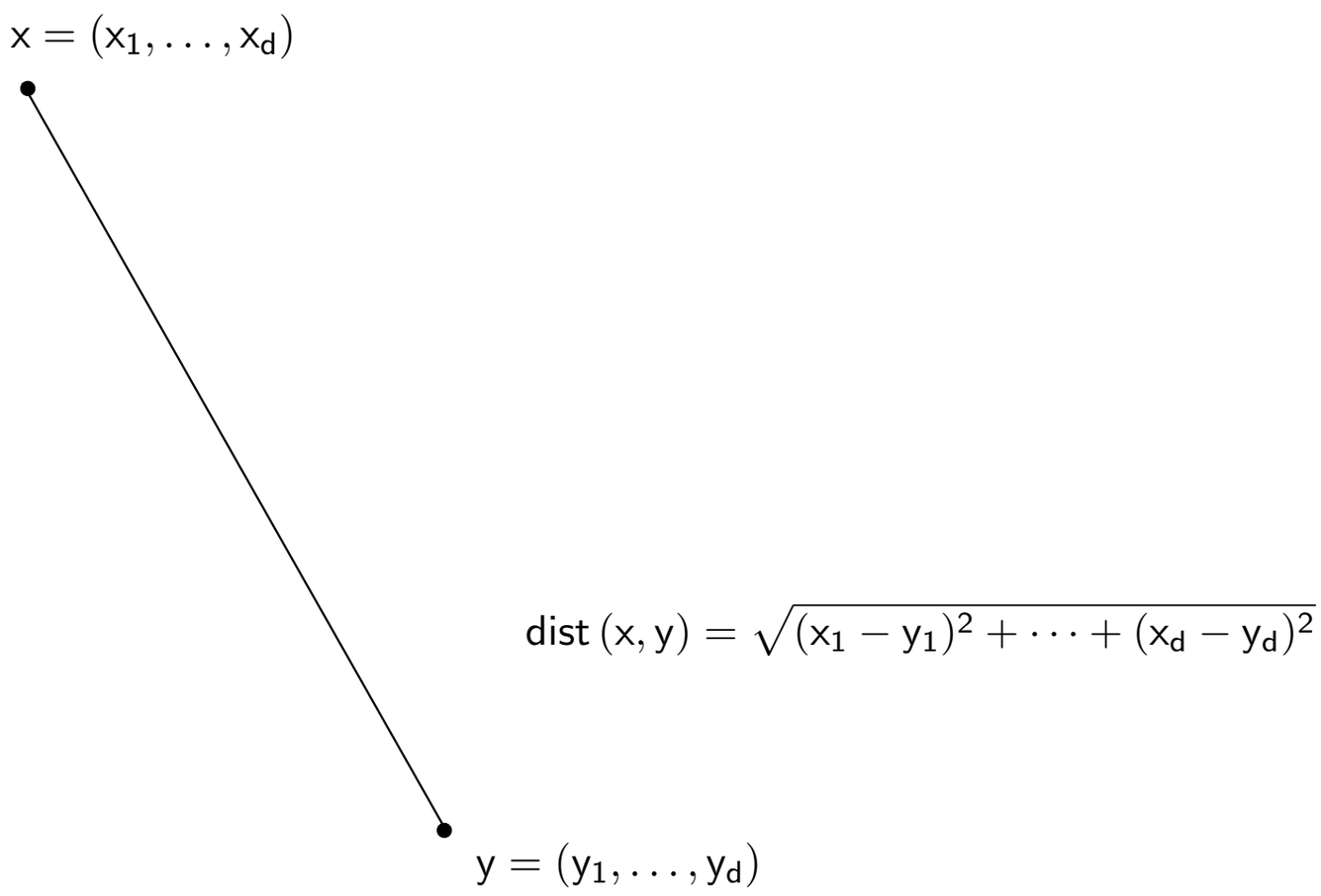




What about here?



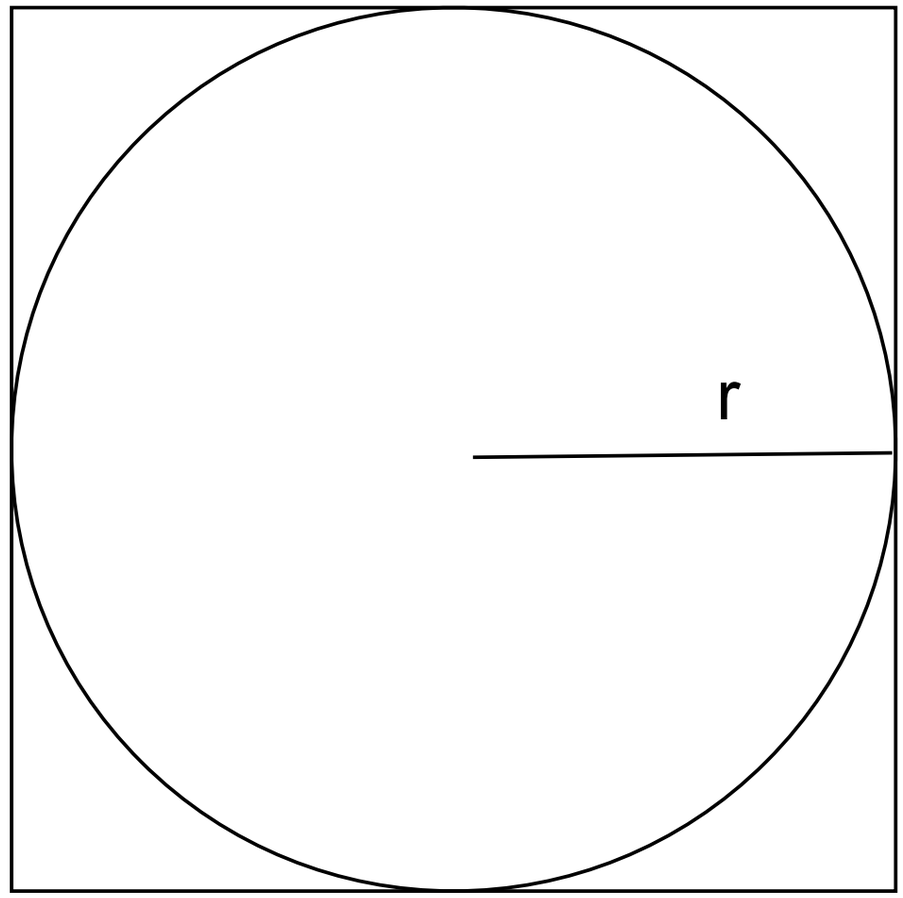
The idea of distance is completely general and we can compute the distance between points in d-dimensional space



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

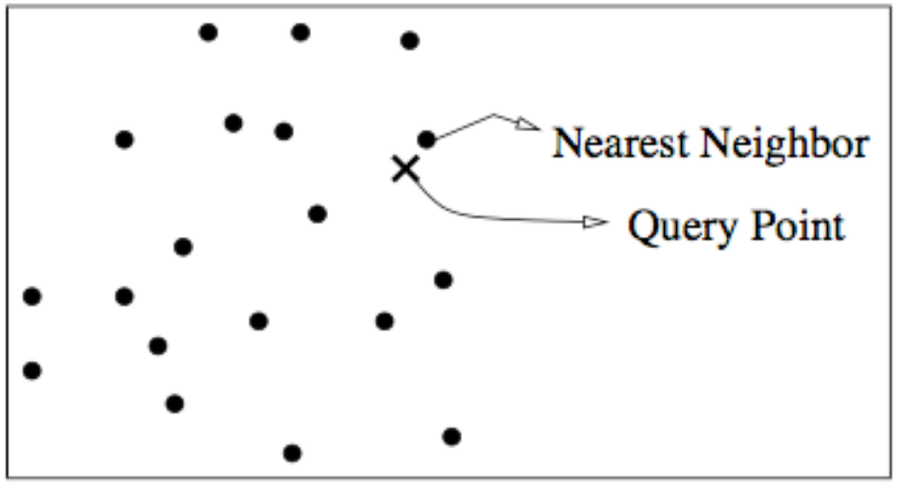
$$\text{volume of the enclosing box with side } 2r = 2r^d$$

and after a little work

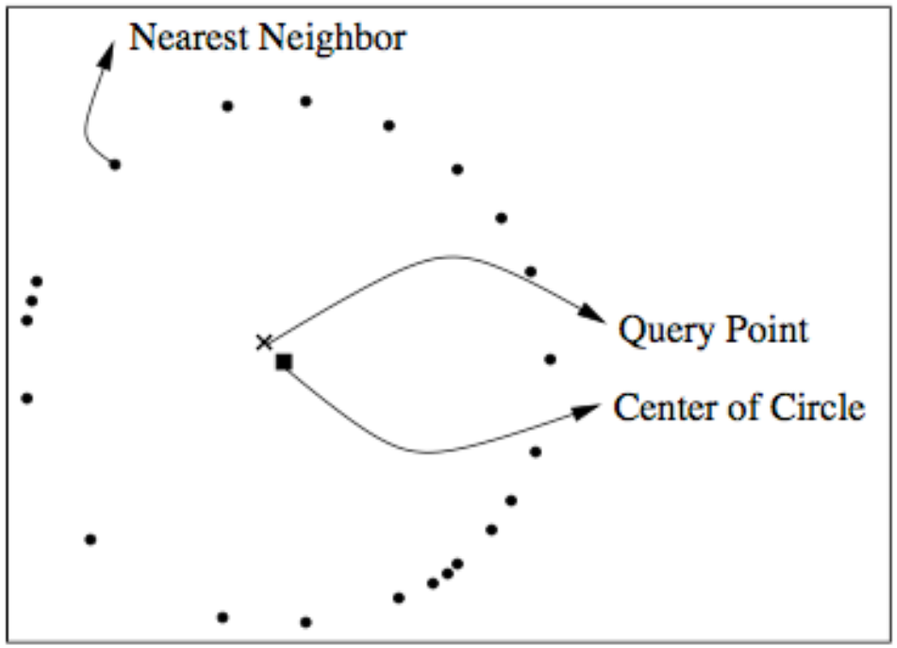
$$\text{ratio of their volumes} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} \to 0 \text{ as } d \to \text{big}$$

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)



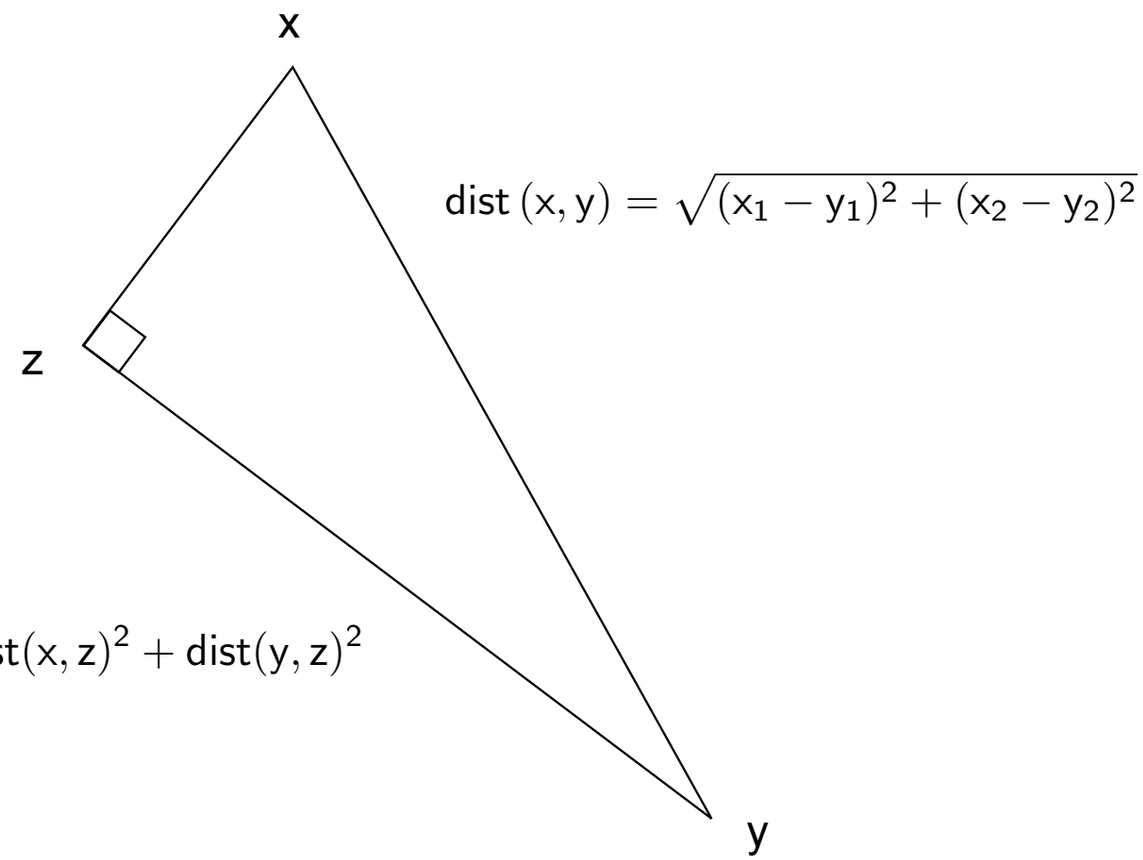
Returning to data, under certain mathematical assumptions, you can also show that in high-dimensional spaces, the distance to **the point nearest you in a data set isn't that much closer than the point farthest from you**



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?

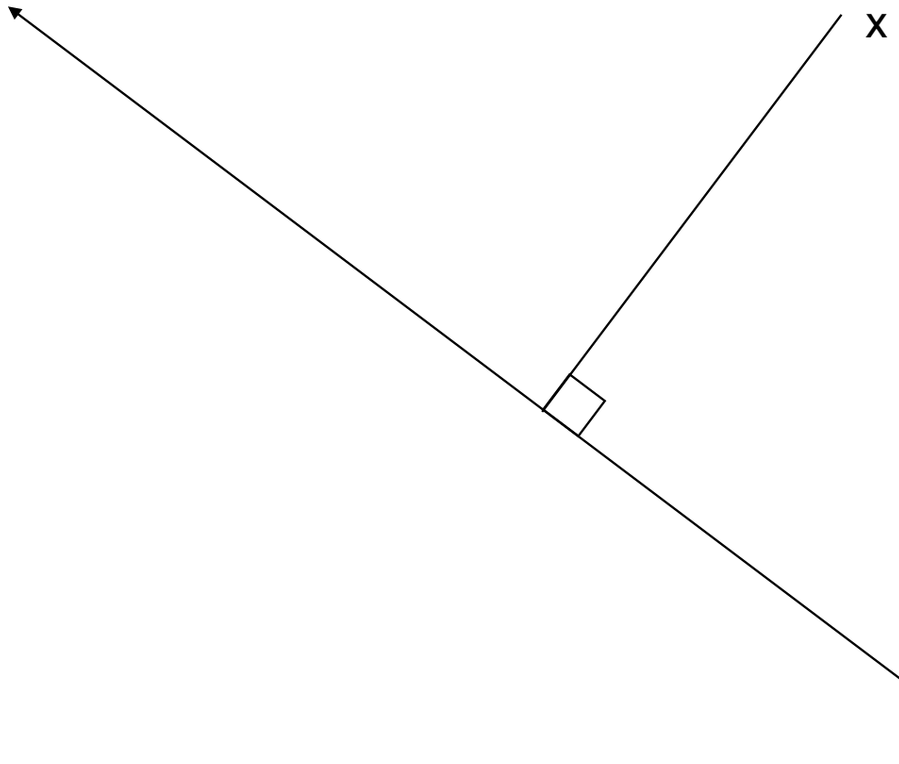
With distance, we also get a right-angle relationship -- Remember the Pythagorean theorem?



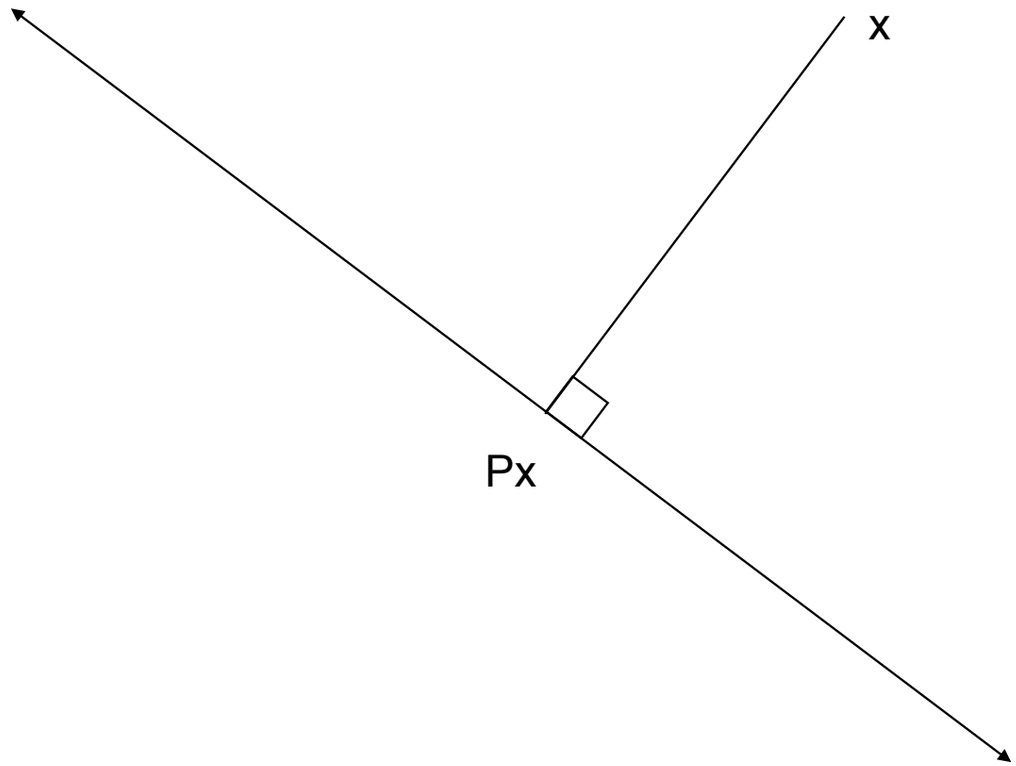
$$\text{dist}\,(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

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

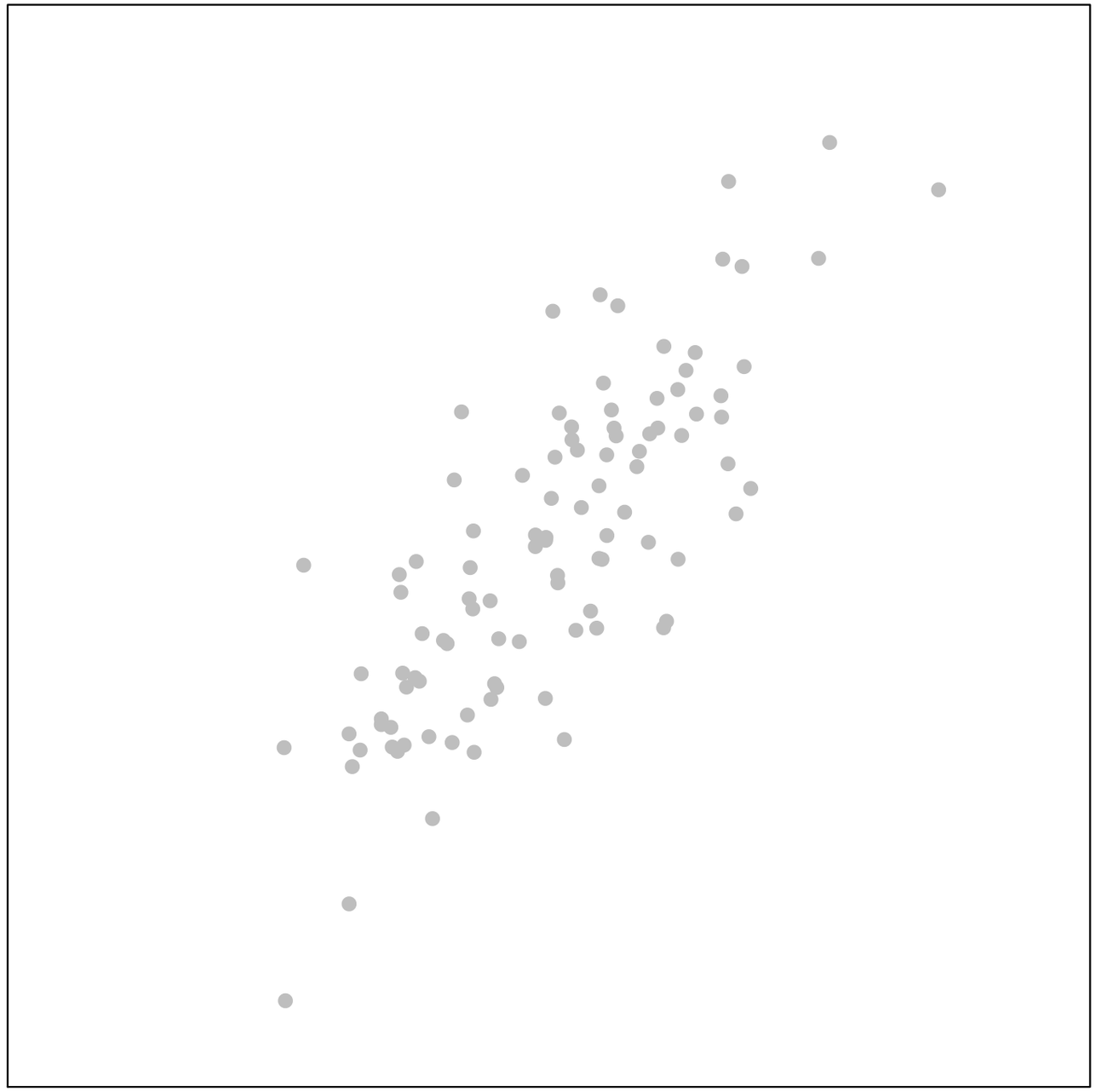
We also might remember right angles appearing when you talk about the nearest point to a line...



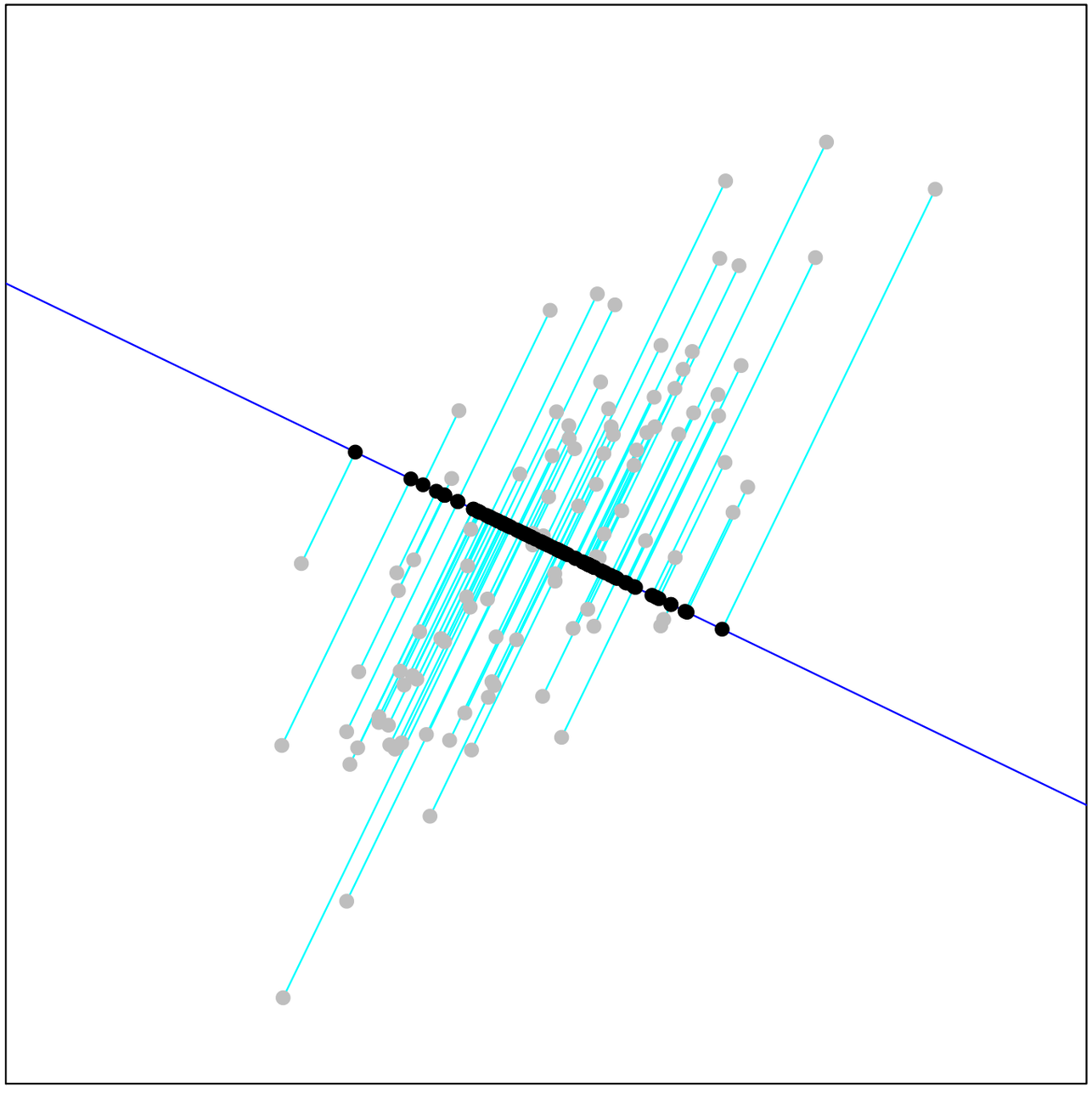
We refer to this point as the orthogonal projection of x onto the line...

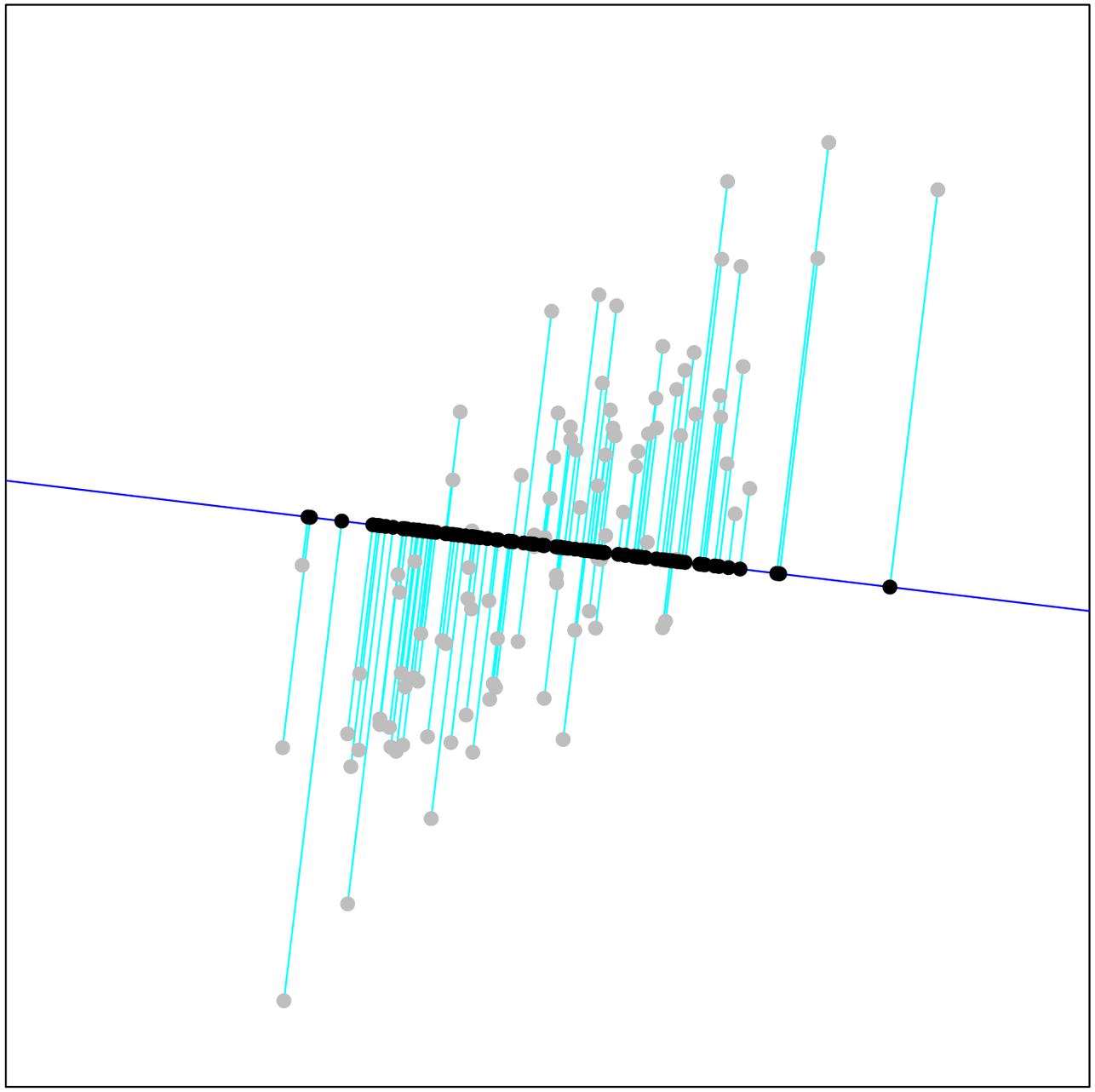


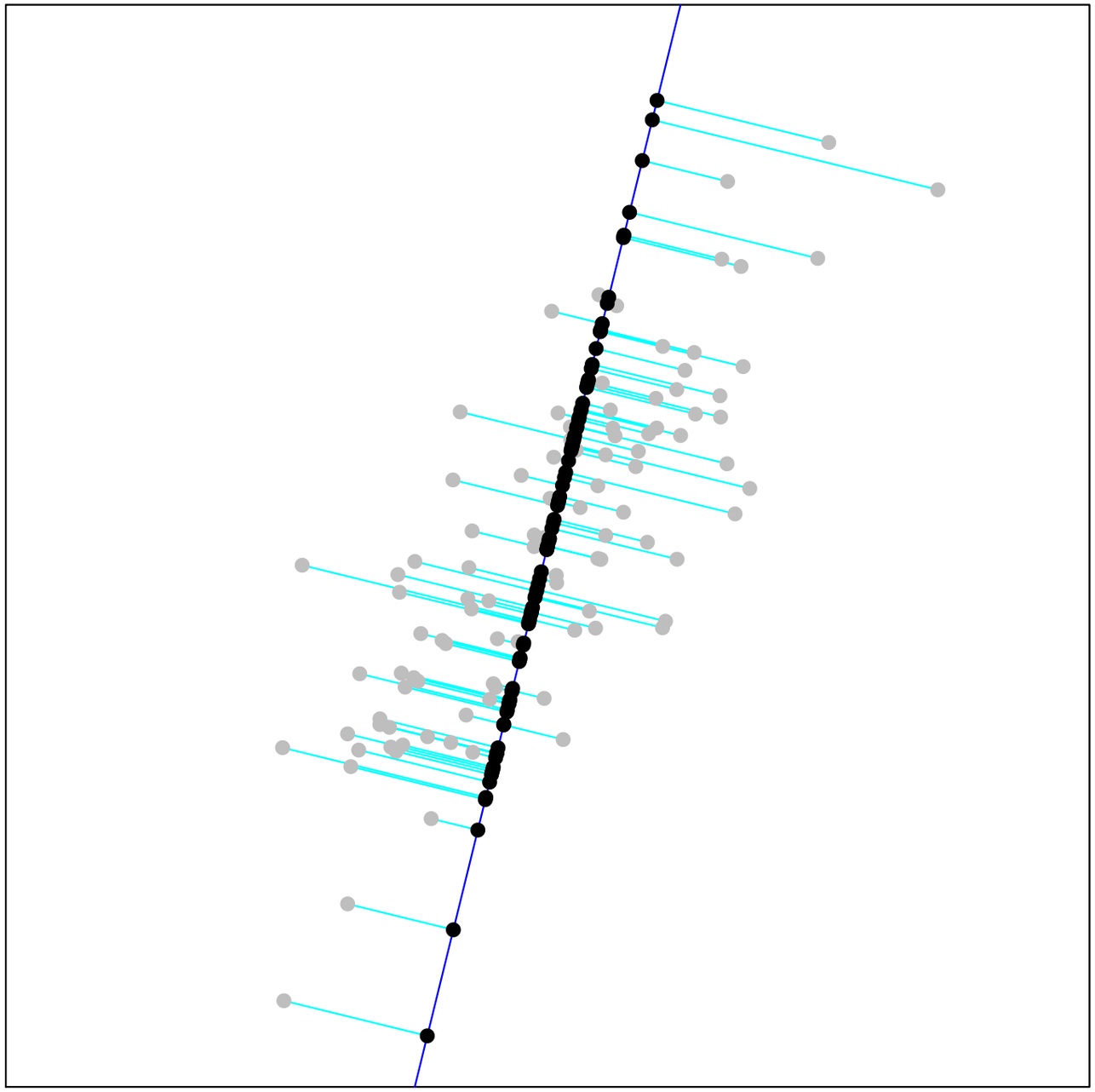
Let’s consider a 2-d data set and projections onto various lines



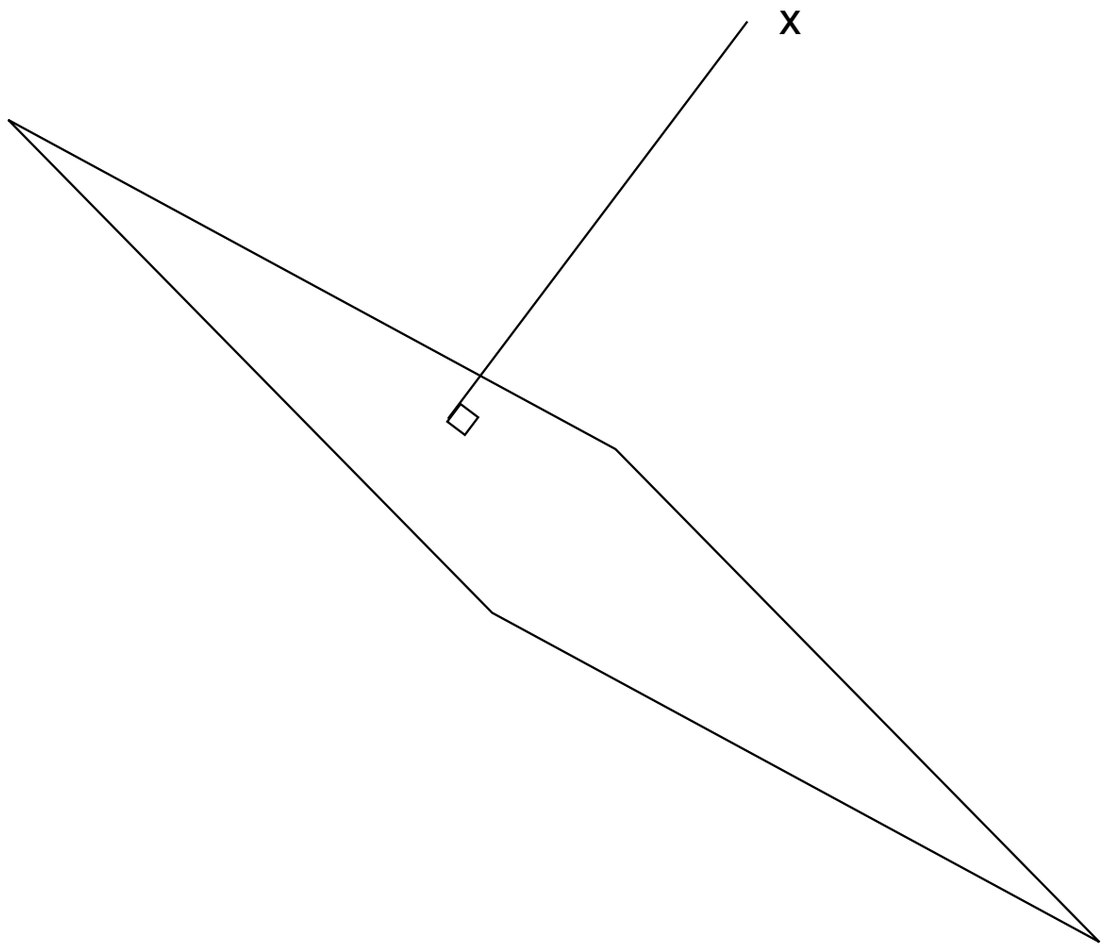
Here is one, indicated by the blue line with the black points indicating the projections of our data



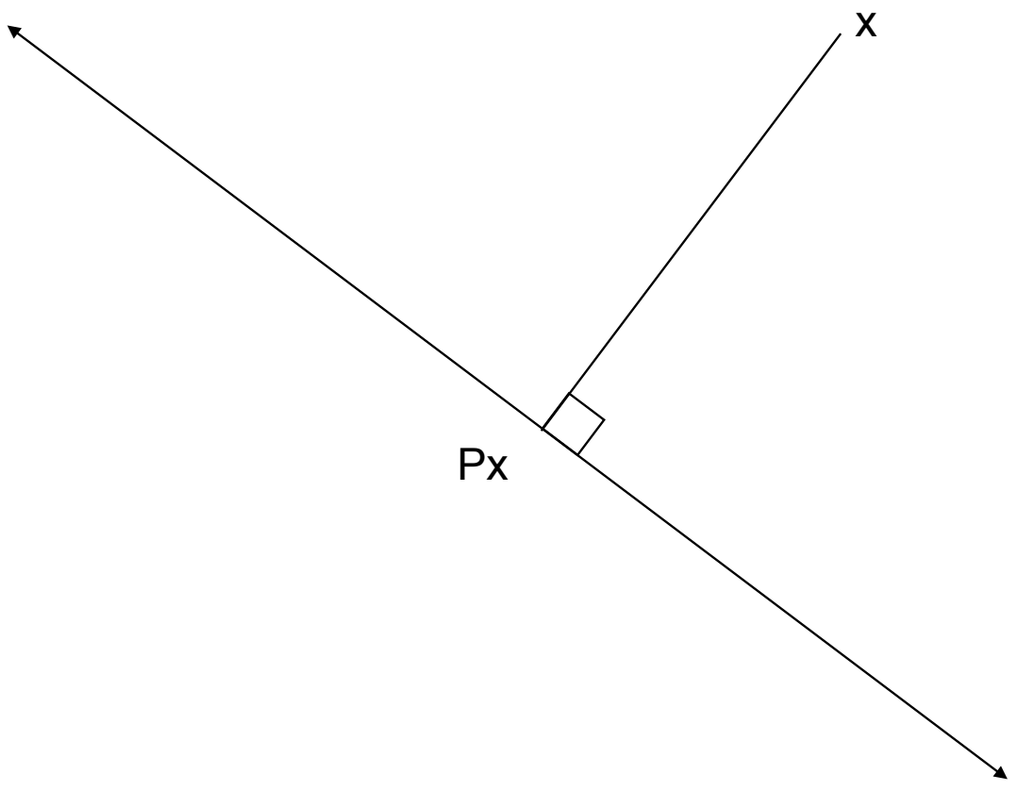


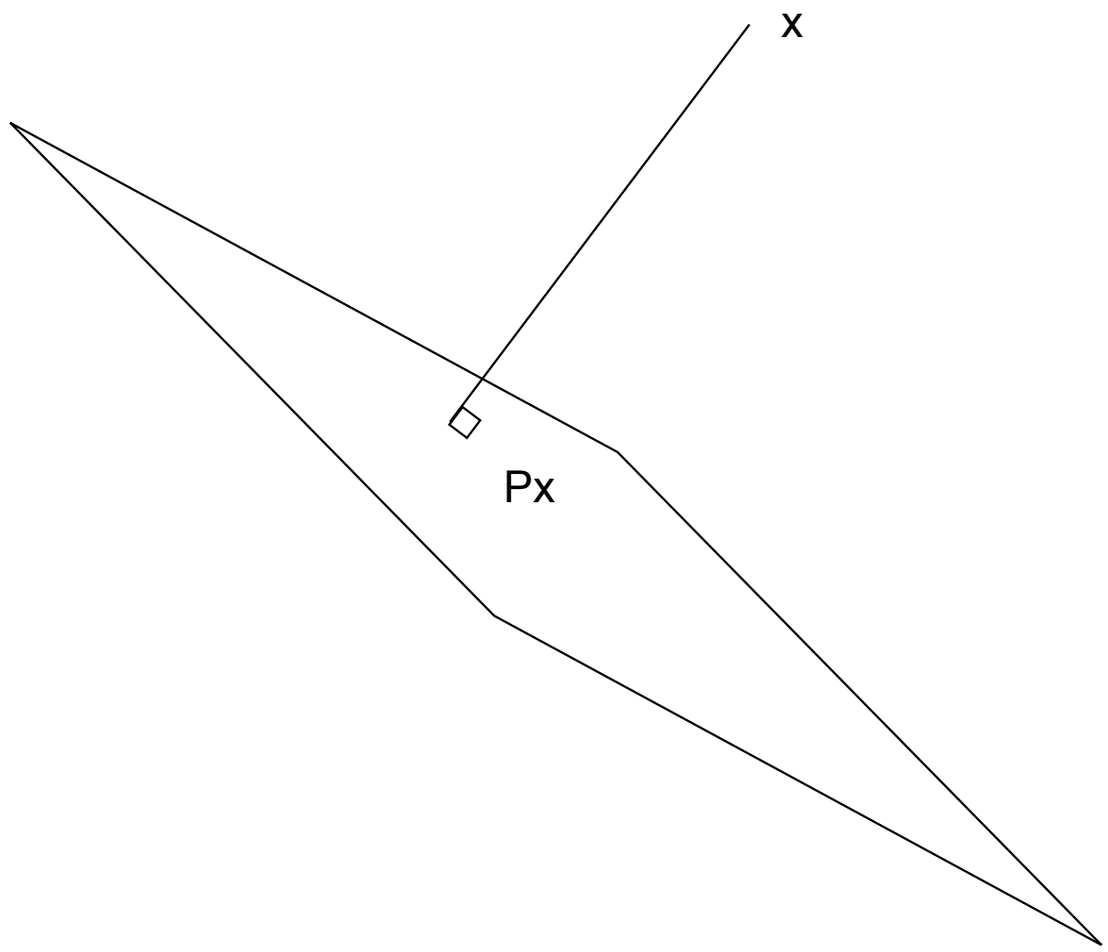


The same essential approach works when you're projecting a point onto a plane instead

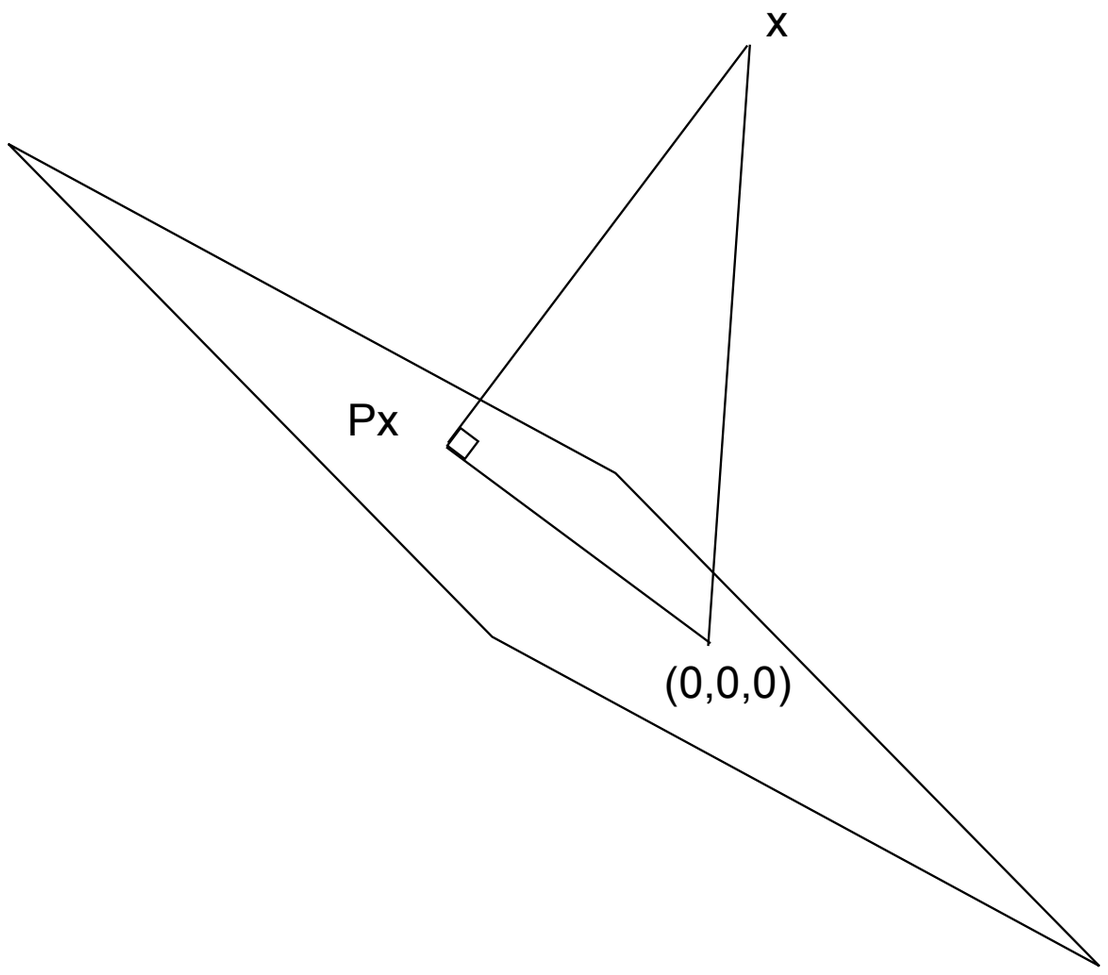


With these pictures, we have another view of the concept of projection -- The projection of a point onto a line (left) or a plane (right) is taken to be the nearest point marked Px





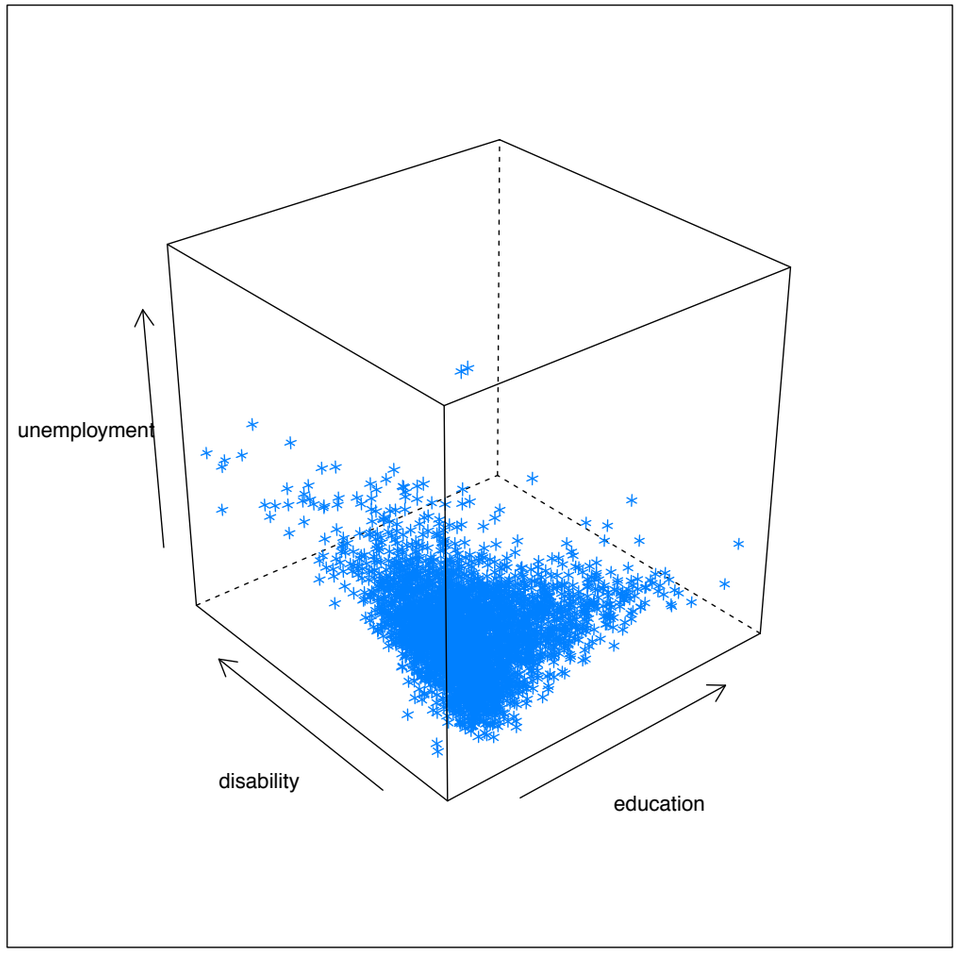
The Pythagorean theorem helps us see things a little more clearly (or not if this doesn't speak to you)

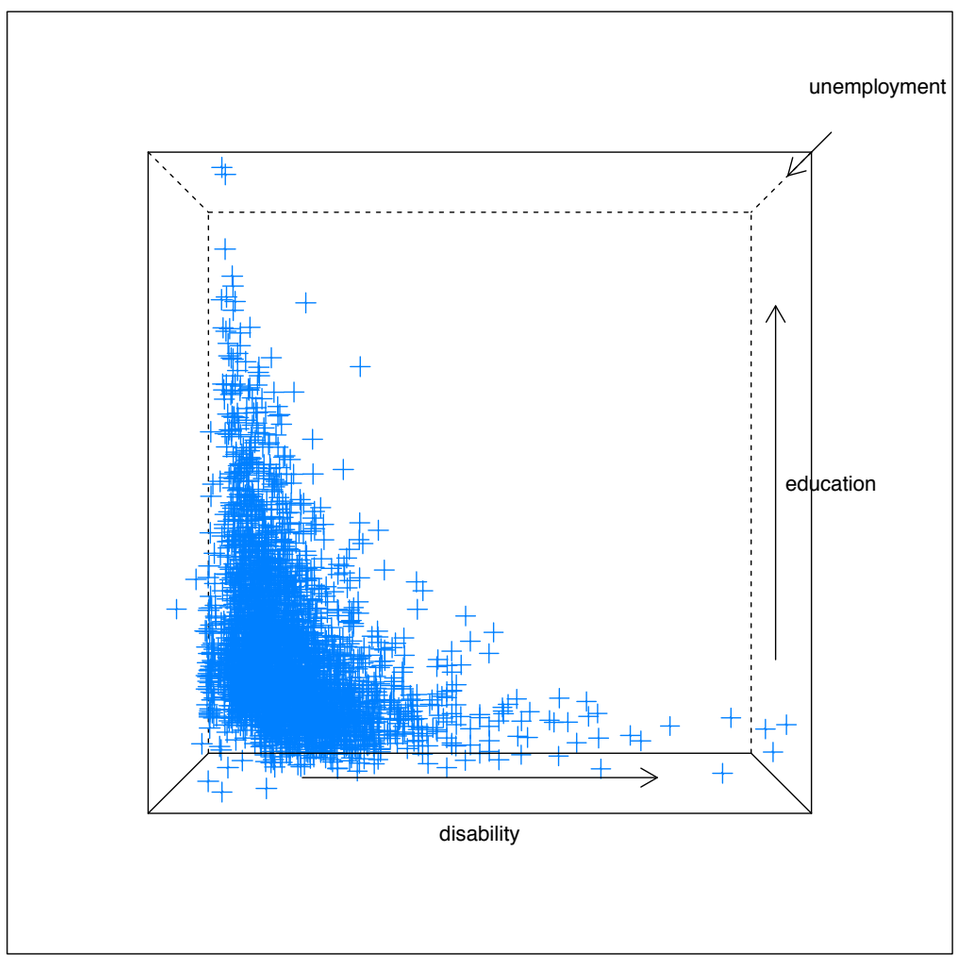


Projections

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





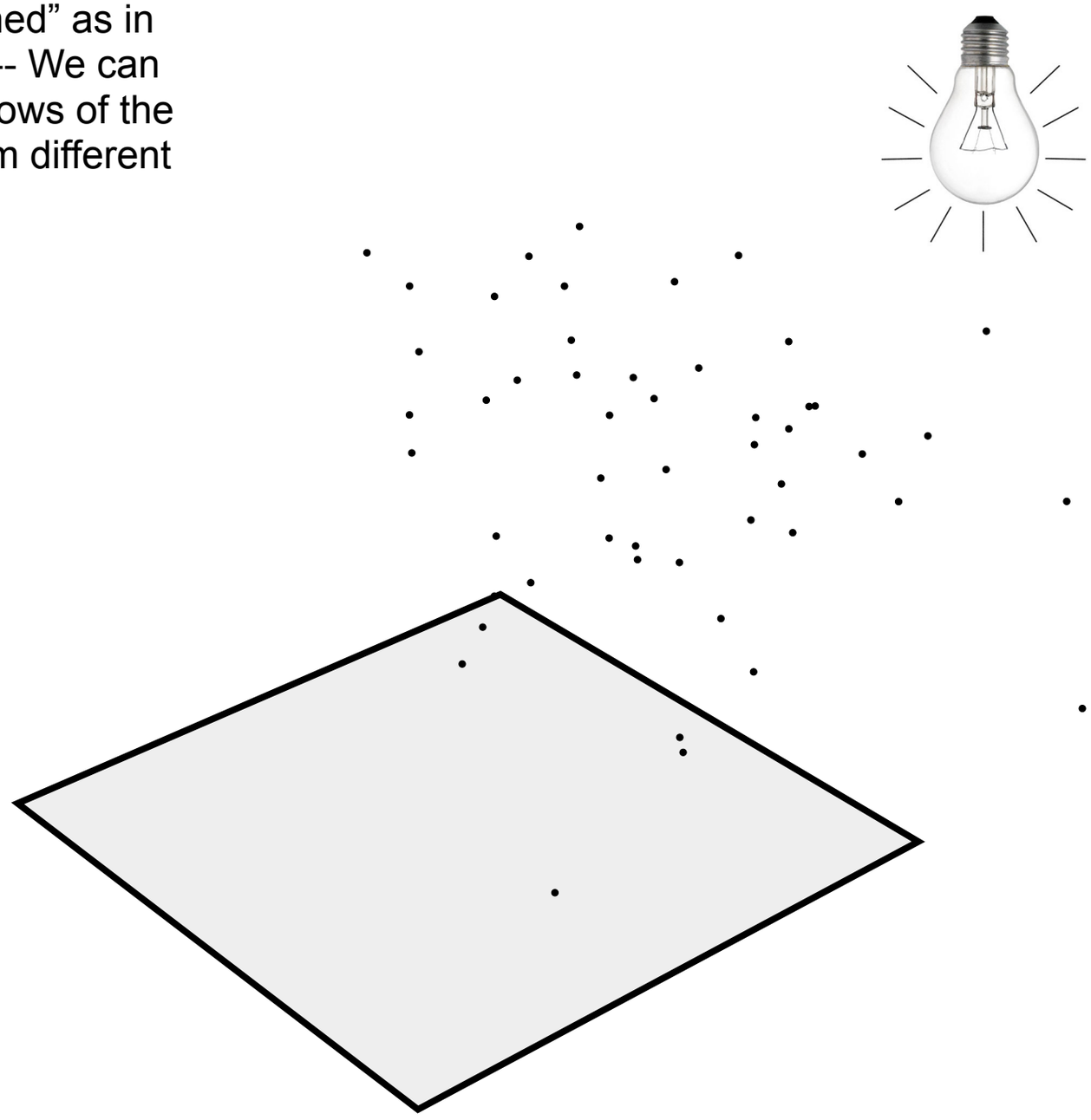
Multiple views

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

News: **Hack-at-it 2010**

Download GGobi for Windows, Mac and Linux

Introduction

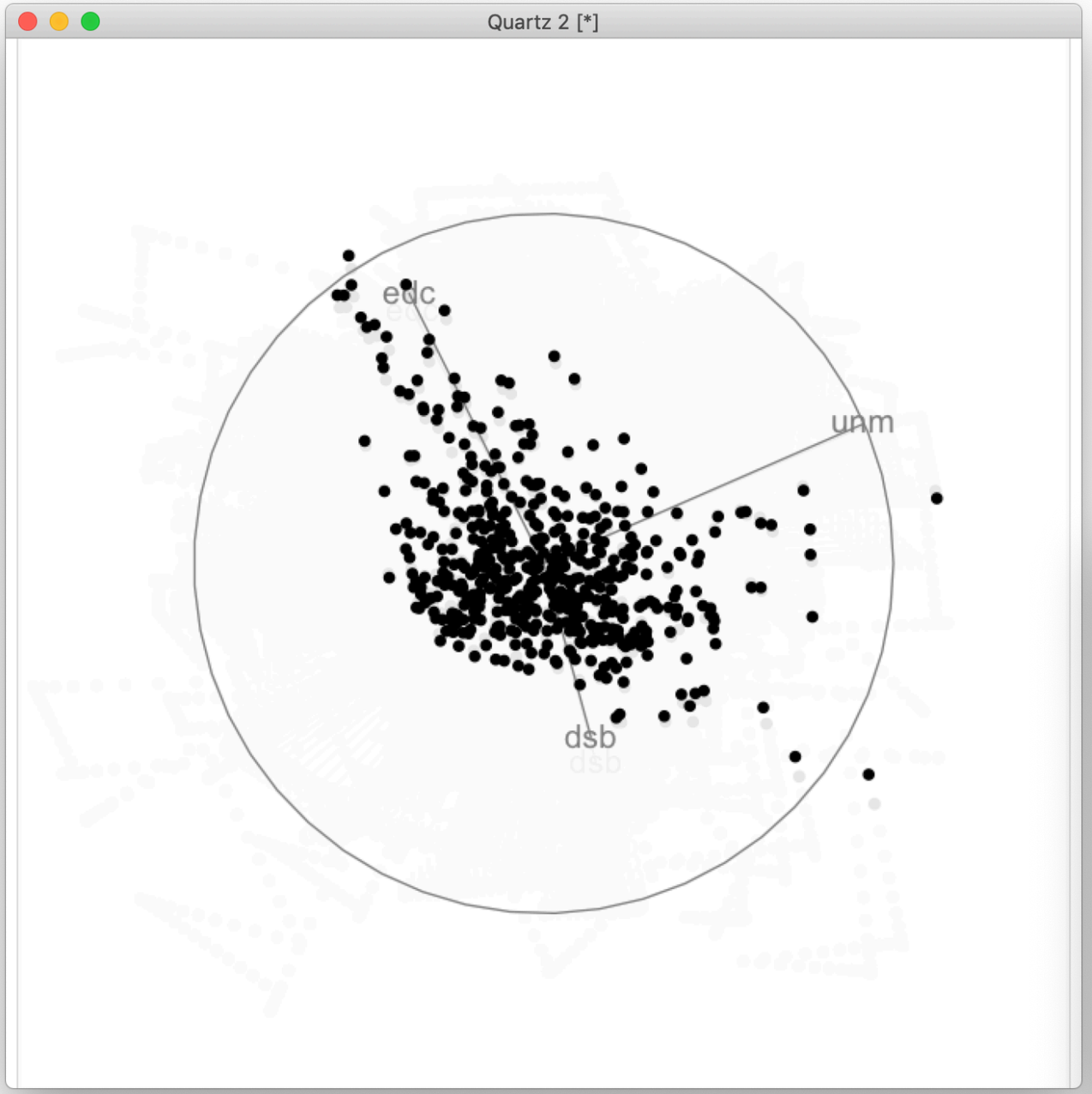
GGobi is an open source visualization program for exploring high-dimensional data. It provides highly dynamic and interactive graphics such as tours, as well as familiar graphics such as the scatterplot, barchart and parallel coordinates plots. Plots are interactive and linked with brushing and identification.

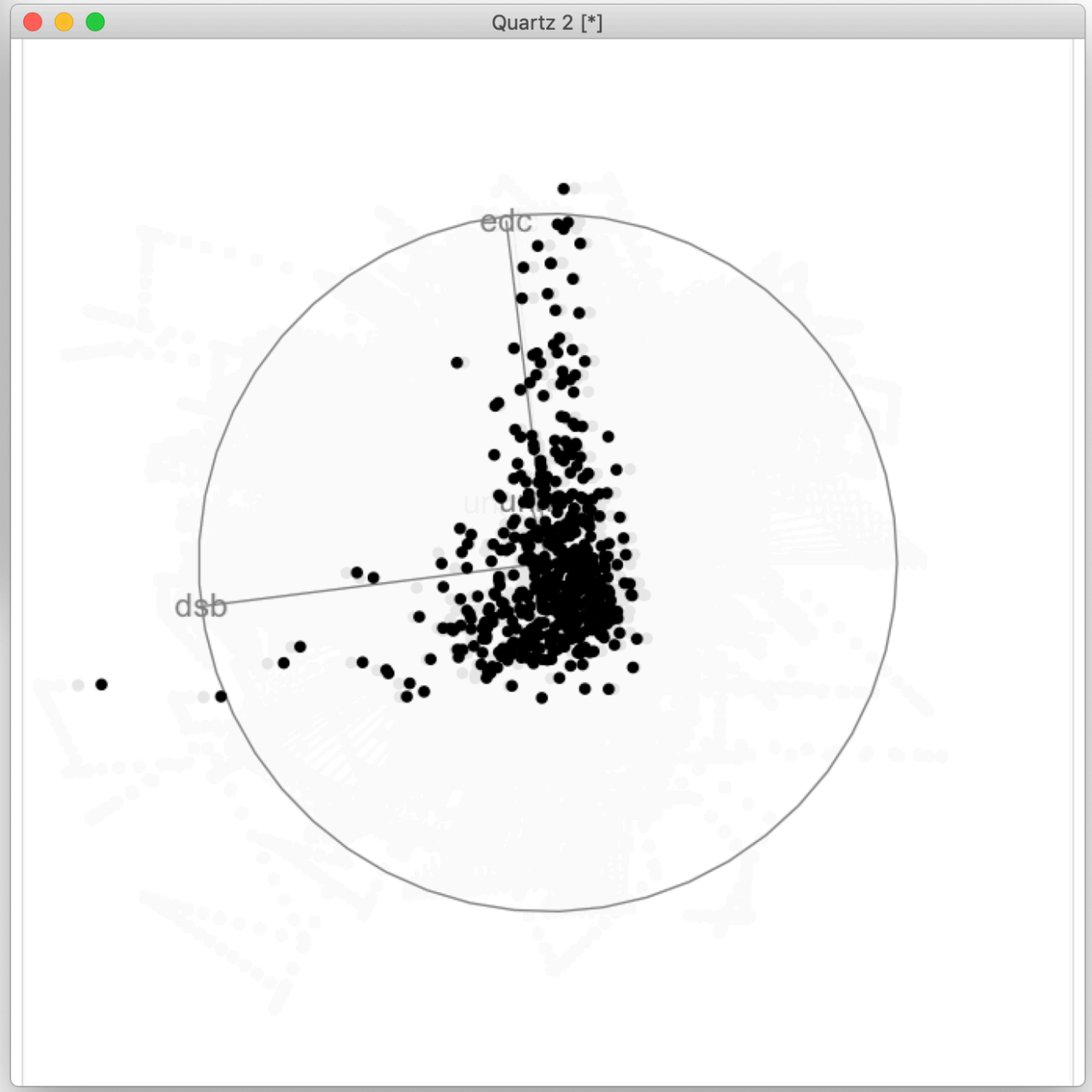
GGobi is fully documented in the GGobi book: "Interactive and Dynamic Graphics for Data Analysis".

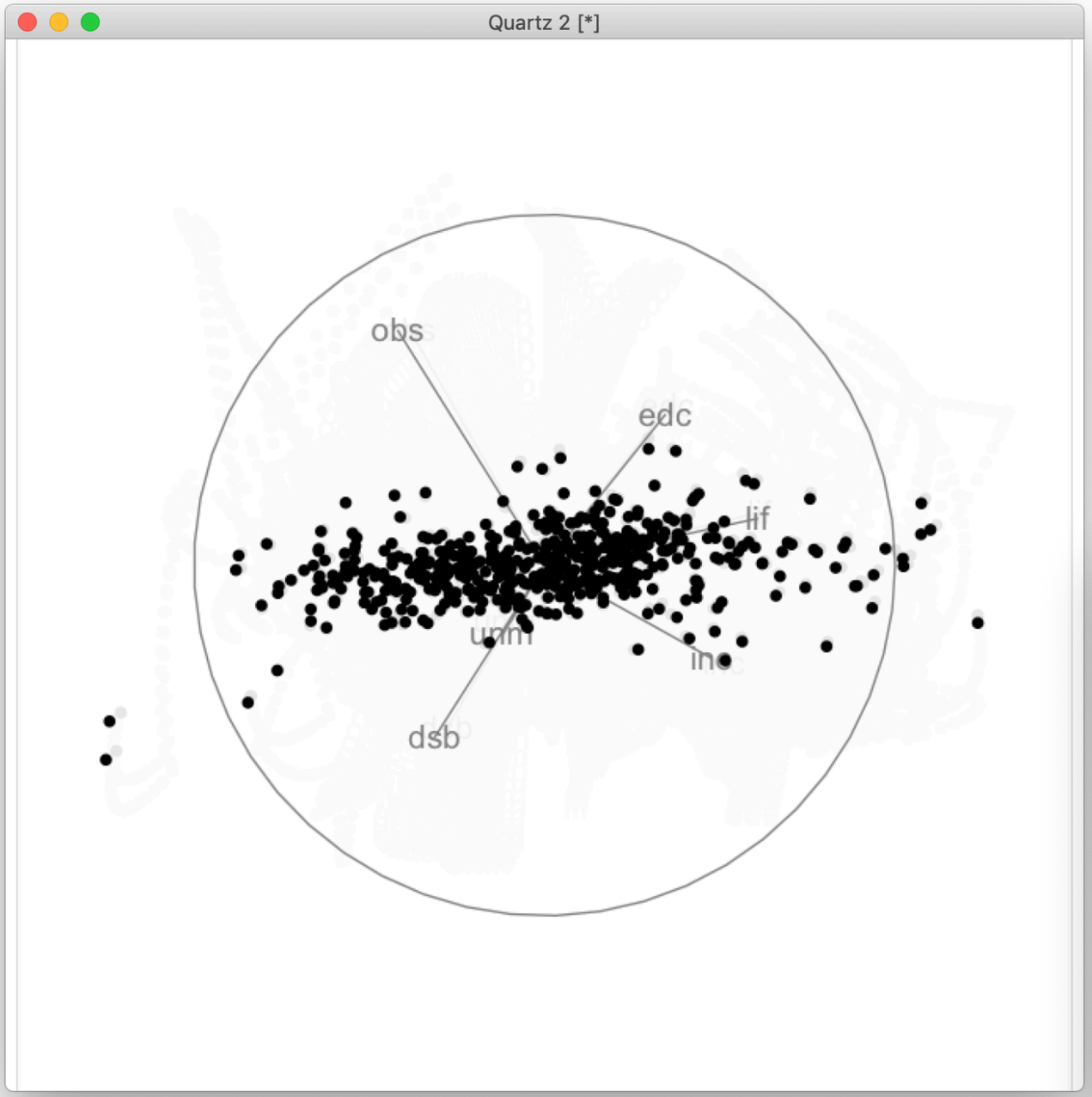
If you are interested in how GGobi came to be, you can read more about it on our history page.

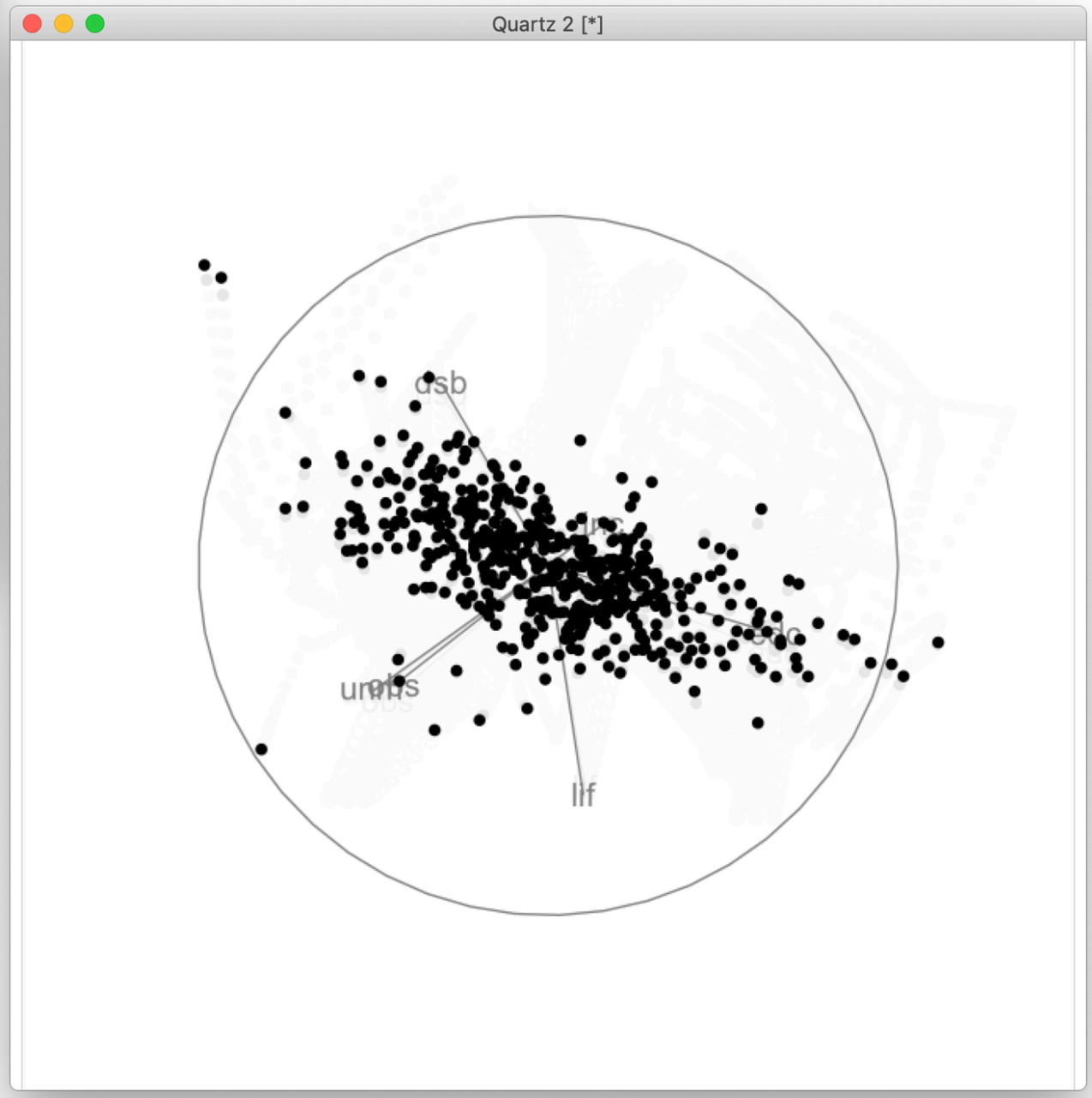
Features

- 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.


Quartz 2 [*]
edc
unm
dsb


Quartz 2 [*]
edc
unm
dsb


Quartz 2 [*]
obs
edc
lif
umn
ine
dsb


Quartz 2 [*]
dsb
lif

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…

Documents as data

We are going to start with something simple and build up some of the statistical issues associated with handling and modeling text -- **We'll take as an initial case study a collection of recipes**

Recipes are interesting because they are not as unstructured as a tweet or an email message or a news article or a novel, but instead have a form -- You can usually point to **an ingredient list and then a list of instructions**, for example

Our recipes come from a site that distributes its data in an open way...

White wine Recipes with White wine, Butter - Recipe Puppy

http://www.recipepuppy.com/?i=white+wine%2C+butter&q=white+wine

Most Visited Getting Started Latest Headlines

White wine Recipes with White wi...

Recipe Puppy beta

Search by Ingredients (comma separated): Advanced Search

white wine, butter, Search

Results **1-10** of **6,091** for recipes with **white wine, butter** and with keywords **'white wine'** (0.052 seconds)

Butter Sauce Recipes LandOLakesFoodservice.com/Recipes
Quality ingredients to make butter sauces. Great flavor in every dish.

Seafood Pasta Recipes deliciousrecipeideas.com
Over 100 Seafood Pasta Recipes Quick & Easy Seafood Pasta Recipes

Making Shrimp & Pasta? healthyha.newworldpasta.com
Get Recipe for Shrimp & Pasta. Plus $1OFF Ronzoni Healthy Harvest®.

Free Granola Bars www.Facebook.com/NaturesPath
Visit Nature's Path and receive a Free Granola Bar Sample!

Ads by Google

Stuffed Mussels With White Wine Recipe
Tasty recipe to go with a nice dry **white wine**. I love mussels and was tired or just making them in a **wine** sauce. Something different. Prep:15m
white wine, **butter**, +garlic, +eggs, +mussels, +bread crumbs, +salt, +rosemary, +parsley, +saffron, +paprika, +tomato
www.grouprecipes.com - Similiar recipes

Scallops with White Wine Sauce II
"**White wine**, butter, and shallots make a great sauce for scallops. This is easy and non-creamy for those that don't like cream sauces."
butter, **white wine**, +chicken broth, +garlic, +lemon, +olive oil, +salt, +sea scallops,

Search Options:
Only Recipes with Images All Recipes

Keyword Search: white wine
Search

Your Ingredients:
butter X white wine X
Start Over

Done

Recipe Puppy API

http://www.recipepuppy.com/about/api/

Most Visited Getting Started Latest Headlines

Recipe Puppy API

RECIPE PUPPY beta

Recipe Puppy API

Recipe Puppy has a very simple API. This api lets you search through recipe puppy database of over a million recipes by keyword and/or by search query. We only ask that you link back to Recipe Puppy and let me know if you are going to perform more than 1,000 requests a day.

The api is accessible at http://www.recipepuppy.com/api/.

For example:
http://www.recipepuppy.com/api/?i=onions,garlic&q=omelet&p=3

Optional Parameters:
i : comma delimited ingredients
q : normal search query
p : page
format=xml : if you want xml instead of json

No parameters are required. Let me know if you have any questions or if you want to share the project built on top of our api.

Done

APIs

APIs (application programming interfaces) make data available as a kind of web service -- The Recipe Puppy API is just one of many you will find, but it was somehow the simplest and hence a reasonable choice to play with on pedagogical grounds

Aside from complications with authentication (often organizations want to know who is accessing their data) the broad principle is largely the same -- A specialized URL represents data, the results of a search, say, and not necessarily an HTML page

For example, enter these two into a Chrome browser -- What do you get?

```
http://www.recipepuppy.com/api/?q=cake
```

```
http://www.recipepuppy.com/api/?q=cake&format=xml
```

www.recipepuppy.com/api/?

www.recipepuppy.com/api/?q=cake&p=100


```
{"title":"Recipe Puppy","version":0.1,"href":"http:\/\/www.recipepuppy.com\/","results":[{"title":"Best Lemon
Blueberry Bundt Cake","href":"http:\/\/www.recipezaar.com\/Best-Lemon-Blueberry-Bundt-Cake-
176927","ingredients":"baking powder, baking soda, blueberries, butter, buttermilk, eggs, flour, flour, lemon juice,
lemon zest, salt, sugar, vanilla extract","thumbnail":"http:\/\/img.recipepuppy.com\/166414.jpg"},{"title":"Best
Southern Pound Cake","href":"http:\/\/www.recipezaar.com\/Best-Southern-Pound-Cake-79972","ingredients":"baking
powder, butter, shortening, eggs, milk, flour, salt, sugar, vanilla extract","thumbnail":""},{"title":"Best Wacky
Cake","href":"http:\/\/www.recipezaar.com\/Best-Wacky-Cake-125923","ingredients":"flour, baking soda, butter, water,
salt, semisweet chocolate, sugar, cocoa powder, vanilla extract, white
vinegar","thumbnail":"http:\/\/img.recipepuppy.com\/166944.jpg"},{"title":"Better Than Grandma's Pound
Cake","href":"http:\/\/www.recipezaar.com\/Better-Than-Grandmas-Pound-Cake-104343","ingredients":"crisco, eggs,
margarine, milk, flour, flour, sugar, vanilla extract","thumbnail":""},{"title":"Better Than Sex Cake (With Bananas,
Coconut, and Pineapple)","href":"http:\/\/www.recipezaar.com\/Better-Than-Sex-Cake-With-Bananas-Coconut-and-
Pineapple-163122","ingredients":"banana, coconut, cool whip, pineapple, pudding, eggs, cream cheese, vegetable oil,
sugar, water","thumbnail":""},{"title":"Better-For-You Pound Cake","href":"http:\/\/www.recipezaar.com\/Better-For-
You-Pound-Cake-126395","ingredients":"buttermilk, egg whites, condensed milk, flour, almonds, applesauce, cake
mix","thumbnail":""},{"title":"Betty Crocker Coffee Cake Circa 1973","href":"http:\/\/www.recipezaar.com\/Betty-
Crocker-Coffee-Cake-Circa-1973-224163","ingredients":"baking powder, eggs, flour, milk, salt, sugar, vegetable
oil","thumbnail":""},{"title":"Better Than Sex Cake (Toffee Bar Cake)","href":"http:\/\/www.recipezaar.com\/Better-
Than-Sex-Cake-Toffee-Bar-Cake-38550","ingredients":"chocolate cake, chocolate syrup, cool whip, candy
bars","thumbnail":""},{"title":"Bijan's Pina Colada Birthday Cake","href":"http:\/\/www.recipezaar.com\/Bijans-Pina-
Colada-Birthday-Cake-125679","ingredients":"cool whip, cream cheese, pineapple, cream of coconut, vanilla pudding,
milk, cake mix","thumbnail":"http:\/\/img.recipepuppy.com\/169120.jpg"},{"title":"Birdseed
Cake","href":"http:\/\/www.recipezaar.com\/Birdseed-Cake-57924","ingredients":"flour, baking soda, banana, cinnamon,
pineapple, eggs, vegetable oil, pecan, salt, sugar, vanilla extract","thumbnail":""}]}
```

www.recipepuppy.com/api/?

www.recipepuppy.com/api/?q=cake&p=100&format=xml


```
▼<recipes>
  ▼<recipe>
     <title>Best Lemon Blueberry Bundt Cake</title>
    ▼<href>
       http://www.recipezaar.com/Best-Lemon-Blueberry-Bundt-Cake-176927
     </href>
    ▼<ingredients>
       baking powder, baking soda, blueberries, butter, buttermilk, eggs, flour, flour, lemon juice, lemon zest, salt,
       sugar, vanilla extract
     </ingredients>
   </recipe>
  ▼<recipe>
     <title>Best Southern Pound Cake</title>
    ▼<href>
       http://www.recipezaar.com/Best-Southern-Pound-Cake-79972
     </href>
    ▼<ingredients>
       baking powder, butter, shortening, eggs, milk, flour, salt, sugar, vanilla extract
     </ingredients>
   </recipe>
  ▼<recipe>
     <title>Best Wacky Cake</title>
     <href>http://www.recipezaar.com/Best-Wacky-Cake-125923</href>
    ▼<ingredients>
       flour, baking soda, butter, water, salt, semisweet chocolate, sugar, cocoa powder, vanilla extract, white
       vinegar
     </ingredients>
   </recipe>
  ▼<recipe>
     <title>Better Than Grandma's Pound Cake</title>
    ▼<href>
       http://www.recipezaar.com/Better-Than-Grandmas-Pound-Cake-104343
     </href>
    ▼<ingredients>
       crisco, eggs, margarine, milk, flour, flour, sugar, vanilla extract
     </ingredients>
   </recipe>
  ▼<recipe>
    ▼<title>
       Better Than Sex Cake (With Bananas, Coconut, and Pineapple)
     </title>
    ▼<href>
       http://www.recipezaar.com/Better-Than-Sex-Cake-With-Bananas-Coconut-and-Pineapple-163122
     </href>
    ▼<ingredients>
```

Data formats

With the advent of Web 2.0, we have seen the rise of data services or web services like our recipe server -- You can think of these as URL's that return, well, data rather than HTML documents

While this sounds small, let's consider one of the cake recipes from Recipe Puppy -- Here is what the recipe looks like on the screen and then how we'd have to "read" it if we were processing the text

```
<recipes>
   <recipe>
      <title>Best Lemon Blueberry Bundt Cake</title>
      <href>
         http://www.recipezaar.com/Best-Lemon-Blueberry-Bundt-Cake-176927
      </href>
      <ingredients>
         baking powder, baking soda, blueberries, butter, buttermilk, eggs, flour, flour, lemo
      </ingredients>
   </recipe>
   <recipe>
      <title>Best Southern Pound Cake</title>
      <href>
         http://www.recipezaar.com/Best-Southern-Pound-Cake-79972
      </href>
      <ingredients>
         baking powder, butter, shortening, eggs, milk, flour, salt, sugar, vanilla extract
      </ingredients>
   </recipe>

   ...

   <recipe>
      <title>Best Wacky Cake</title>
      <href>http://www.recipezaar.com/Best-Wacky-Cake-125923</href>
      <ingredients>
         flour, baking soda, butter, water, salt, semisweet chocolate, sugar, cocoa powder, va
      </ingredients>
   </recipe>
</recipes>
```

```
{"title":"Recipe Puppy",
 "version":0.1,
 "href":"http://www.recipepuppy.com/",
 "results":[
            {"title":"Best Lemon Blueberry Bundt Cake",
               "href":"http://www.recipezaar.com\/Best-Lemon-Blueberry-Bundt-Cake-176927",
               "ingredients":"baking powder, baking soda, blueberries, butter, buttermilk, blu

               "thumbnail":"http://img.recipepuppy.com/166414.jpg"
            },
            {"title":"Best Southern Pound Cake",
               "href":"http:\/\/www.recipezaar.com\/Best-Southern-Pound-Cake-79972",
               "ingredients":"baking powder, butter, shortening, eggs, milk, flour, salt, suga
               "thumbnail":""
            },
            {"title":"Best Wacky Cake","href":"http://www.recipezaar.com/Best-Wacky-Cake-12592
               "ingredients":"flour, baking soda, butter, water, salt, semisweet chocolate, su
               "thumbnail":"http://img.recipepuppy.com/166944.jpg"
            },
            {"title":"Better Than Grandma's Pound Cake",
               "href":"http://www.recipezaar.com\/Better-Than-Grandmas-Pound-Cake-104343",
               "ingredients":"crisco, eggs, margarine, milk, flour, flour, sugar, vanilla extr
               "thumbnail":""
            },
      ...
            {"title":"Better Than Sex Cake (With Bananas, Coconut, and Pineapple)",
               "href":"http://www.recipezaar.com\/Better-Than-Sex-Cake-With-Bananas-Coconut-an
               "ingredients":"banana, coconut, cool whip, pineapple, pudding, eggs, cream chee
               "thumbnail":""},
           ]
}
```

Ingredients

We'll start by considering just the ingredient lists from 1,000 recipes offered by Recipe Puppy -- In all, there are 381 ingredients, of which these are the most frequent

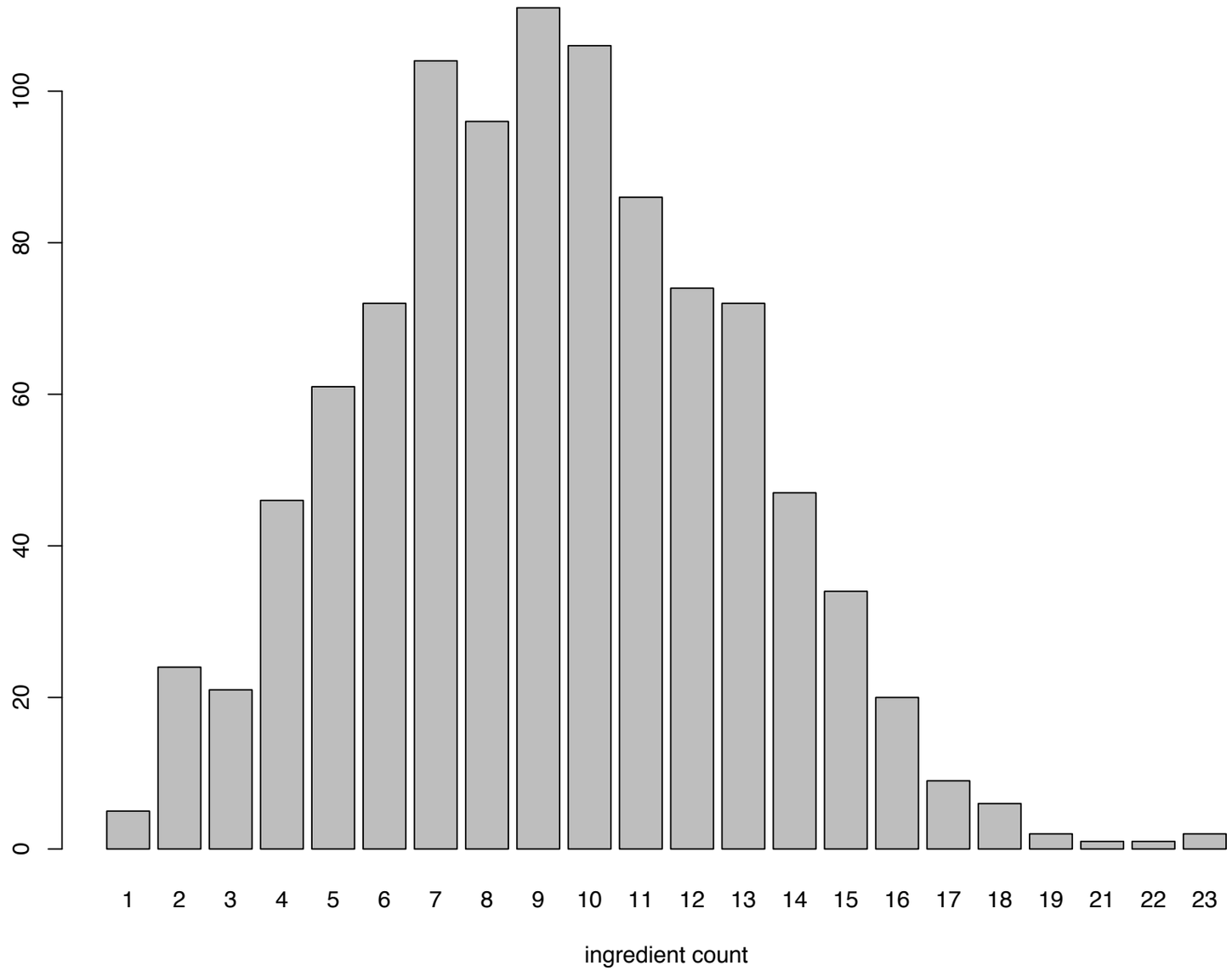
```
      eggs          sugar          flour vanilla.extract           salt
       768            653            653            508            494

    butter   baking.soda vegetable.oil   baking.powder          water
       476            359            342             33            286
```

On the other end of the spectrum, there are 126 ingredients that appear only in one recipe -- Things like Nutella, various liqueurs, garlic and oyster sauce

Here's a breakdown of the number of ingredients per recipe (There are some exceedingly simple and some crazy difficult recipes out there!)

Ingredients

We'll start by considering just the ingredient lists from 1,000 recipes offered by Recipe Puppy -- Here's a breakdown of the number of ingredients per recipe (There are some exceedingly simple and some crazy difficult recipes out there!)



Ingredients

Triple-Chocolate Celebration Cake (23 ingredients): baking powder, baking soda, semisweet chocolate, semisweet chocolate, semisweet chocolate, flour, cherries, cocoa powder, egg yolks, eggs, heavy cream, chocolate, corn syrup, semisweet chocolate chips, strawberries, blackberries, blueberries, raspberries, raspberry jam, salt, sour cream, sugar, cake, vanilla extract, vegetable oil, heavy cream

Spiced Pumpkin Cake with Caramel Icing (22 ingredients): allspice, baking powder, baking soda, flour, vegetable oil, cinnamon, cloves, cream cheese, rum, cranberries, eggs, ginger, heavy cream, orange zest, pumpkin puree, raisins, salt, sugar, sugar, orange zest, vanilla extract, vanilla ice cream, walnut, water"

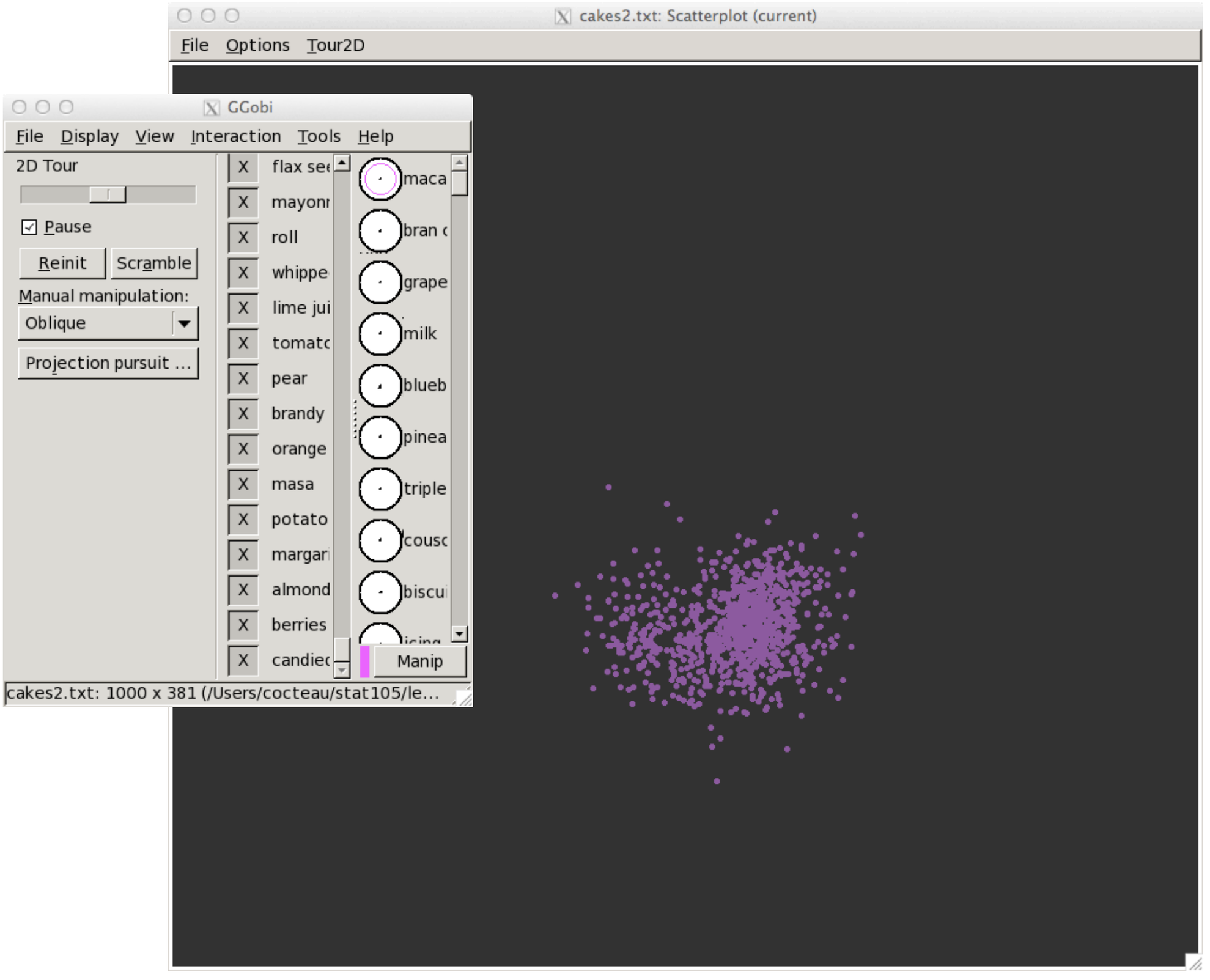
Funny Cake: pie shell

Examining the data

Our data set can be thought of as a 1000 x 381 binary matrix where each row is a recipe and each column represents and ingredient -- This is a “high dimensional” data set in the sense that the number of variables (the vector of unique ingredients) is large

This situation is common when dealing with text data -- The first step is often to reduce the text to a “bag of words” with indicators (or counts or weights, as we’ll see) for each

We can apply the projective methods from earlier in the quarter to have a look at these data -- Let’s fire up GGobi again!


cakes2.txt: Scatterplot (current)
File Options Tour2D
GGobi
File Display View Interaction Tools Help
2D Tour
Pause
Reinit Scramble
Manual manipulation:
Oblique
Projection pursuit ...
flax see
mayonn
roll
whippe
lime jui
tomatc
pear
brandy
orange
masa
potato
margari
almond
berries
candiec
maca
bran c
grape
milk
blueb
pinea
triple
cousc
biscui
Manip
cakes2.txt: 1000 x 381 (/Users/cocteau/stat105/le...

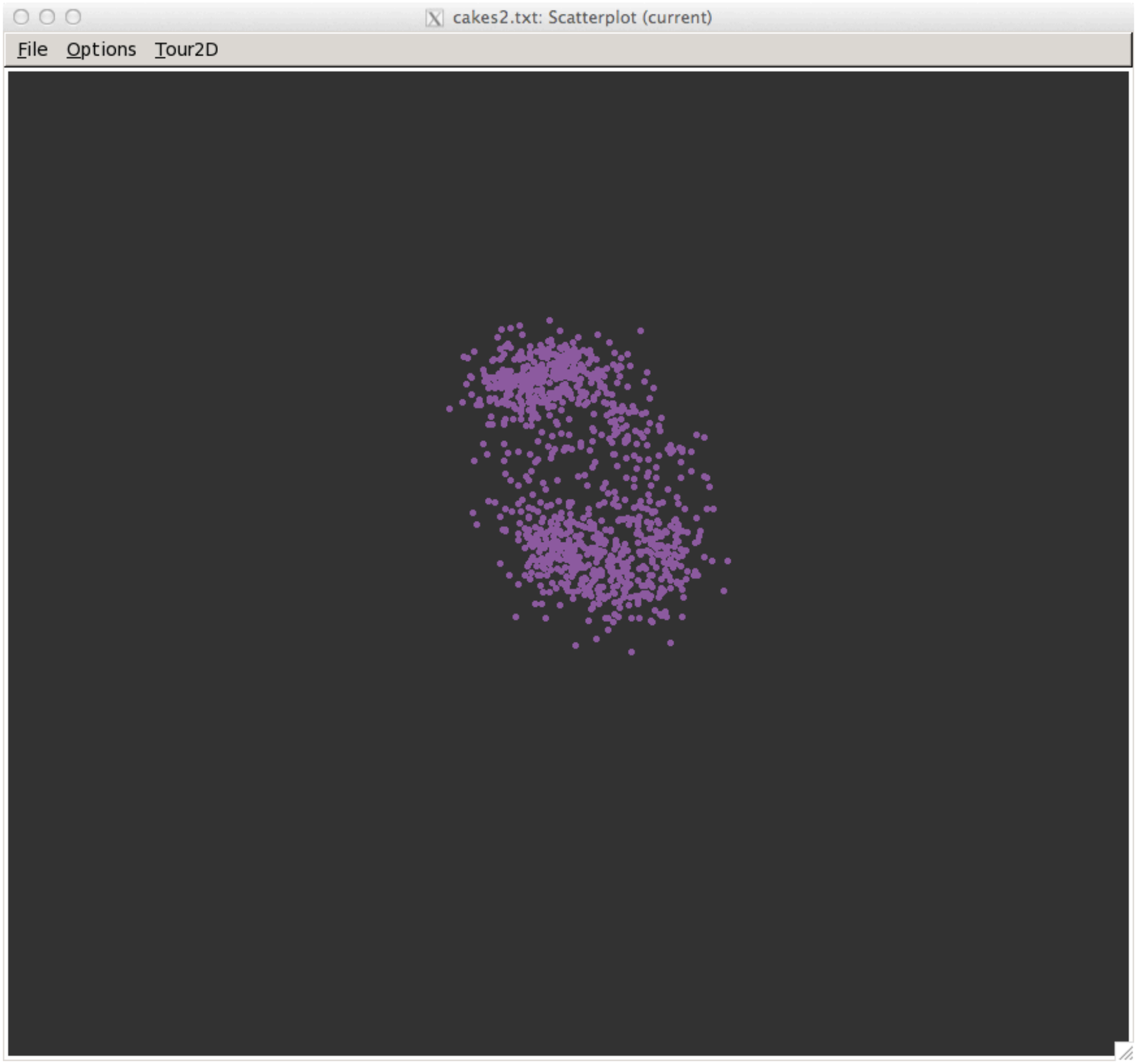
Projections

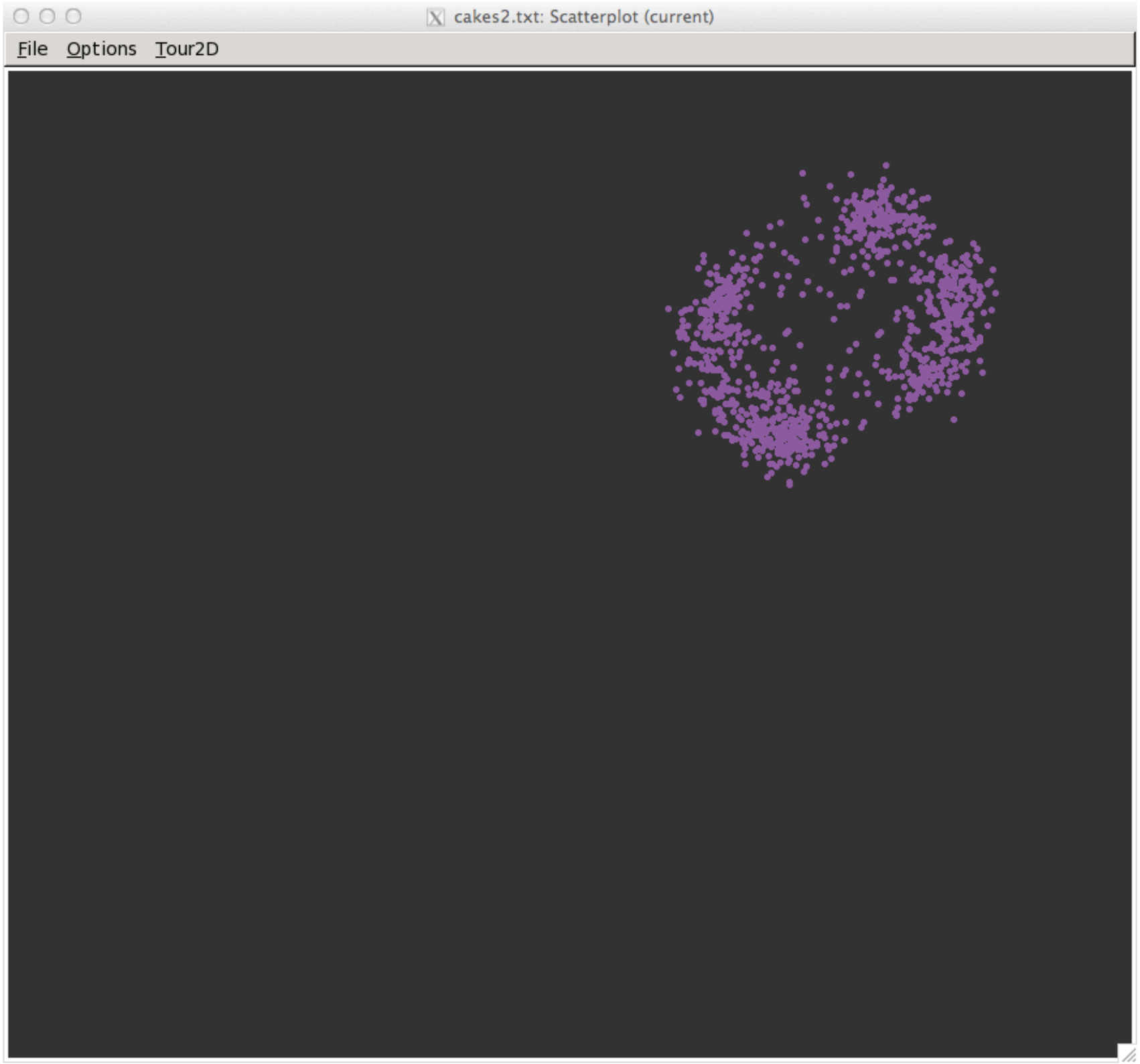
GGobi implements the “Grand Tour,” as smoothly interpolated set of projections of the data -- The Grand Tour selects the directions essentially at random and then moves from one to the next

The Grand Tour is a kind of fishing expedition -- We scan these plots for structure, for **something other than a formless, bivariate “splatter”** (or maybe in technical parlance, something that doesn’t look like observations from a bivariate normal distribution)

In the late 1970s and early 1980s, there was interest in driving the tour to interesting directions, those that don’t appear normal, say -- There were a variety of projection metrics, that scored the projected, bivariate data cloud

For example, one of these responds to “holes” in the data set...

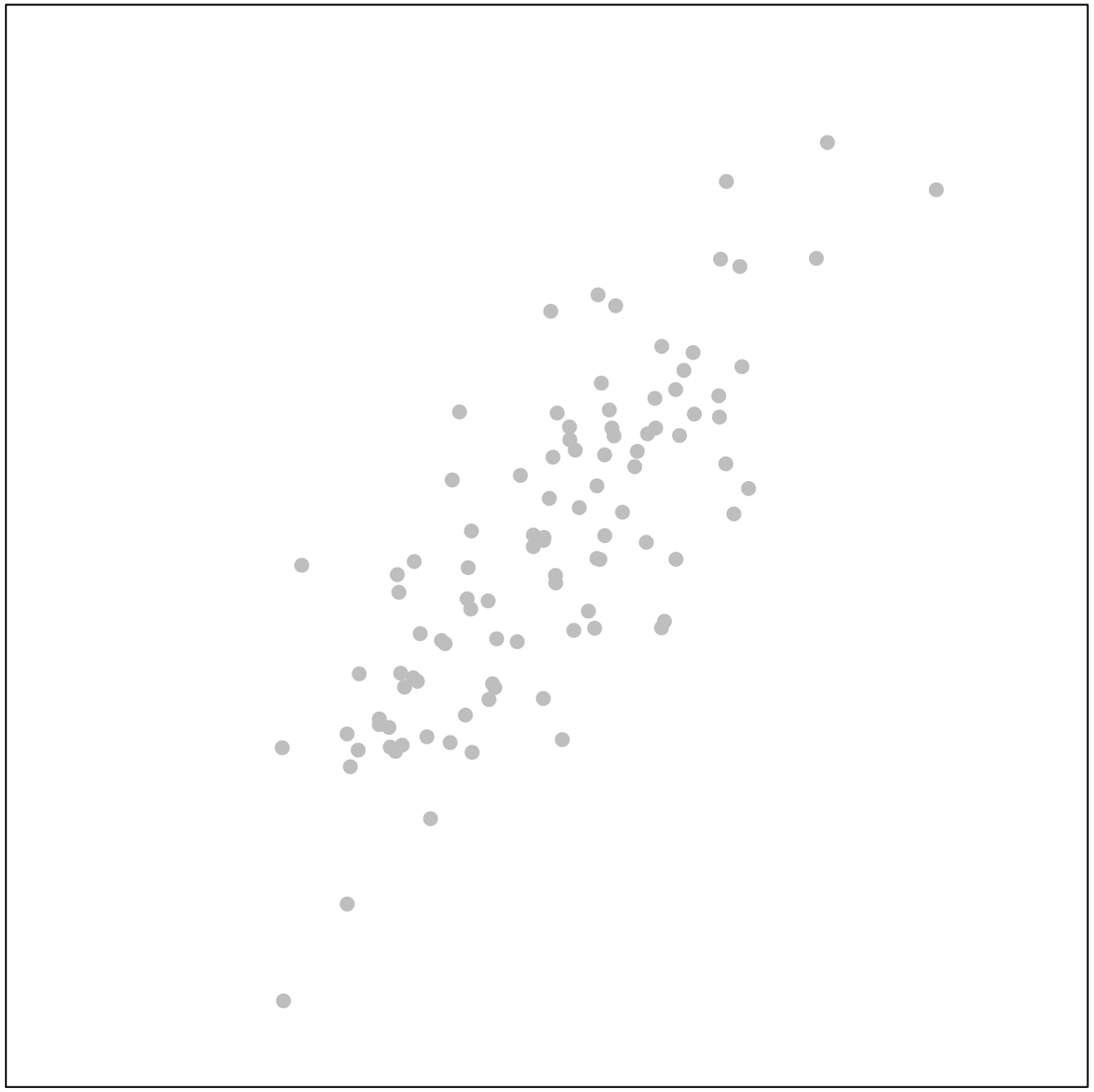

cakes2.txt: Scatterplot (current)
File
Options
Tour2D


cakes2.txt: Scatterplot (current)
File
Options
Tour2D

A distinguished (or at least storied) projection

One particular projection measures “interestingness” in terms of its ability to **represent as much of the variability in the data as possible in a two dimensional display** -- Because of this, the projection is said to be useful for “dimension reduction”

In principal components analysis (PCA) we derive **a new coordinate system** for the data, one that is “aligned” to the shape of the data cloud -- On the next few slides we illustrate the idea with a bivariate data set (we’ll return to our 381-dimensional data in a moment)

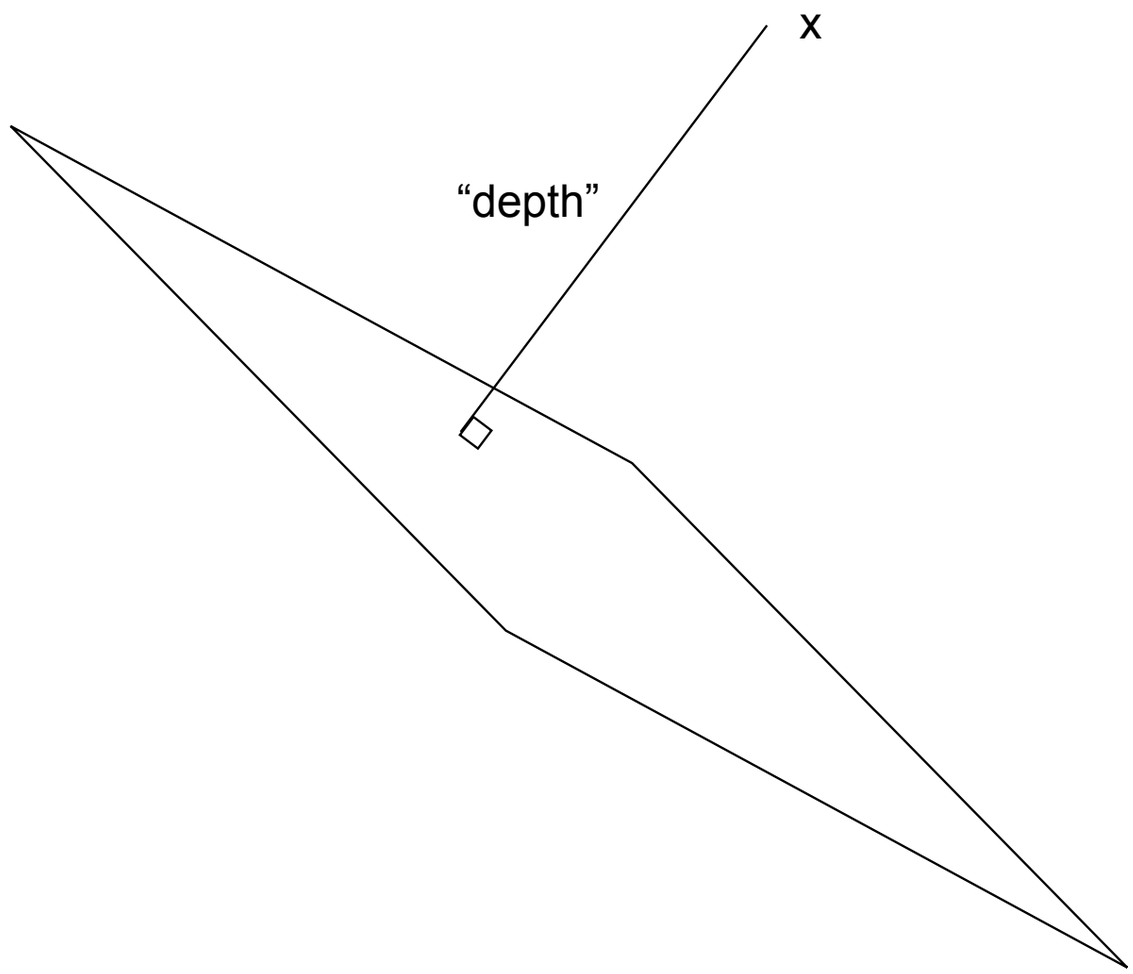


PCA

Now, consider creating **a new coordinate system for the data** (or, rather, come up with a new pair of orthogonal axes) that is “aligned” to the data’s shape -- You can formalize this in two (equivalent ways)

First, given a projection onto a plane, you can think about the distance from that point to the plane as being the “depth” of the point -- We then find the projection that gives us the smallest overall (over the whole data set) depth

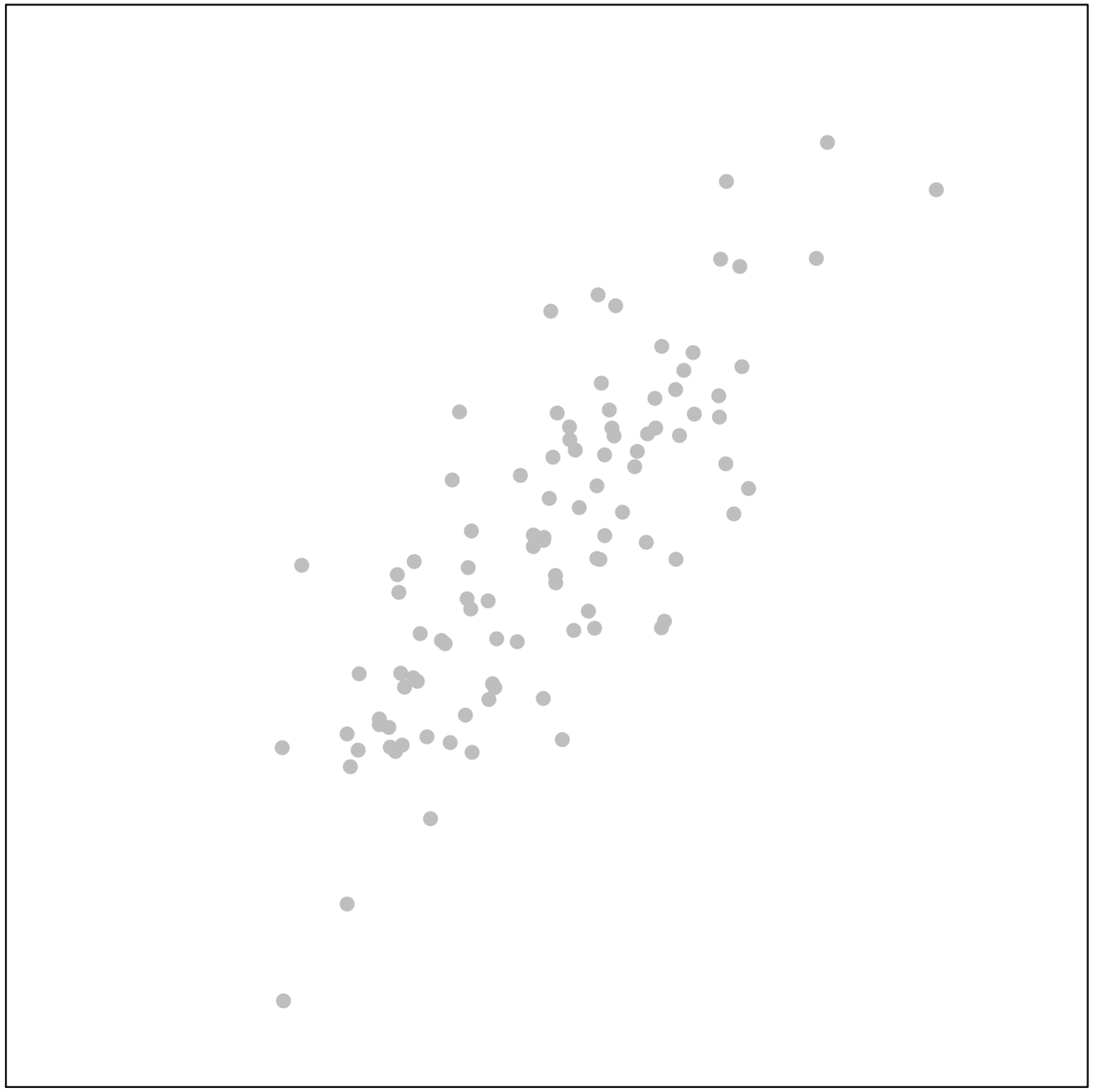
The depth of x when projected on the plane is just its orthogonal distance

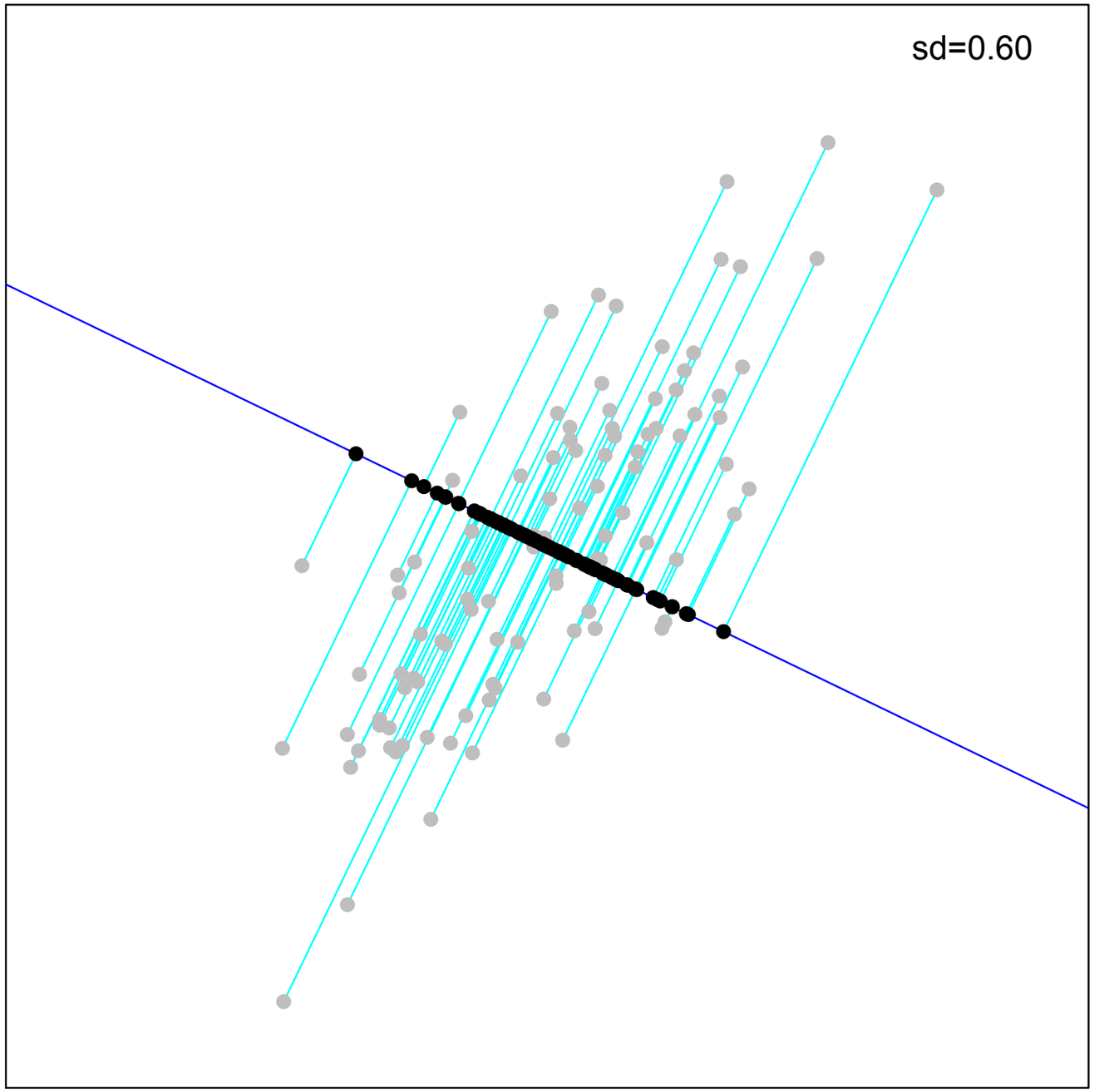


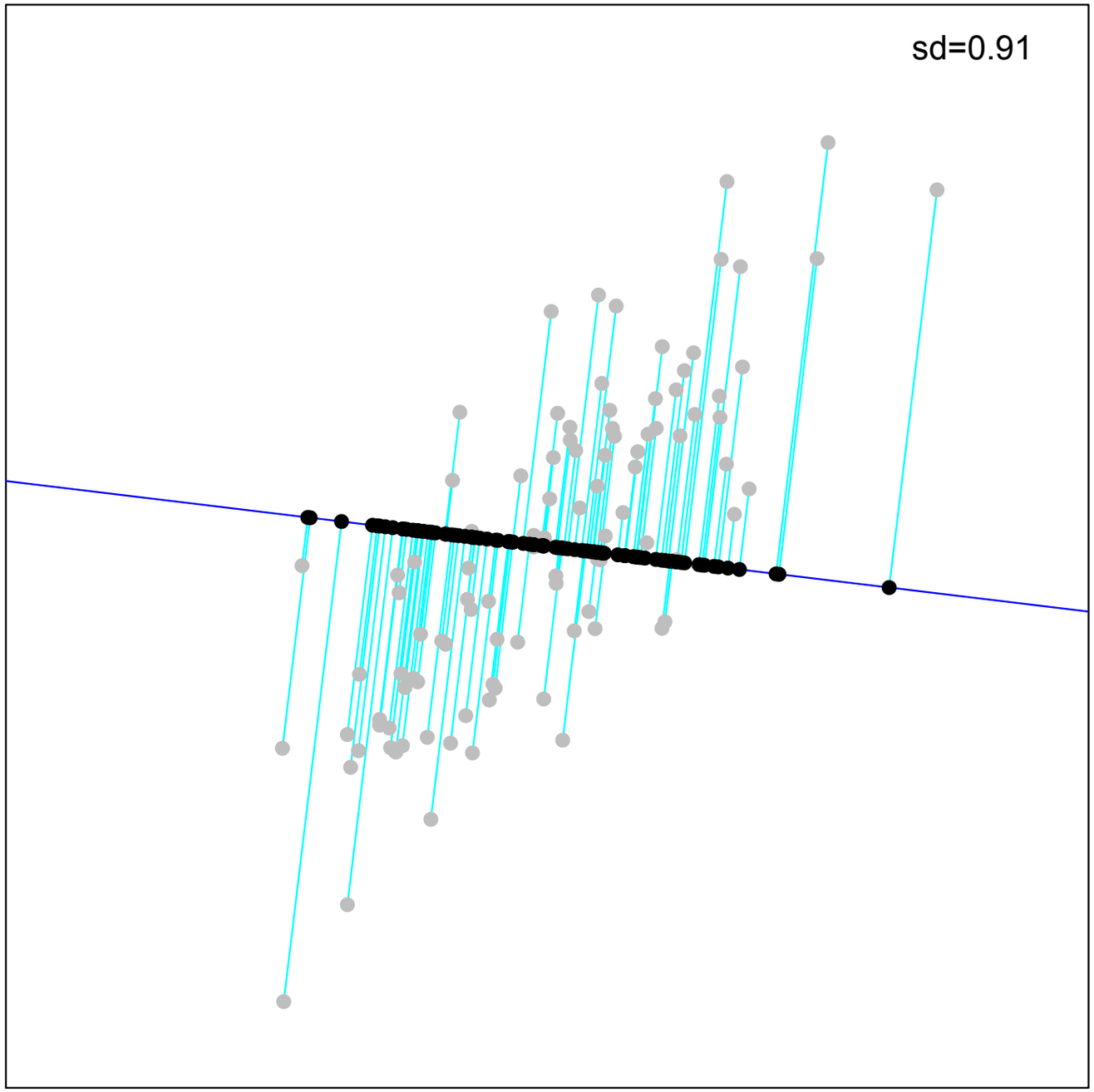
PCA

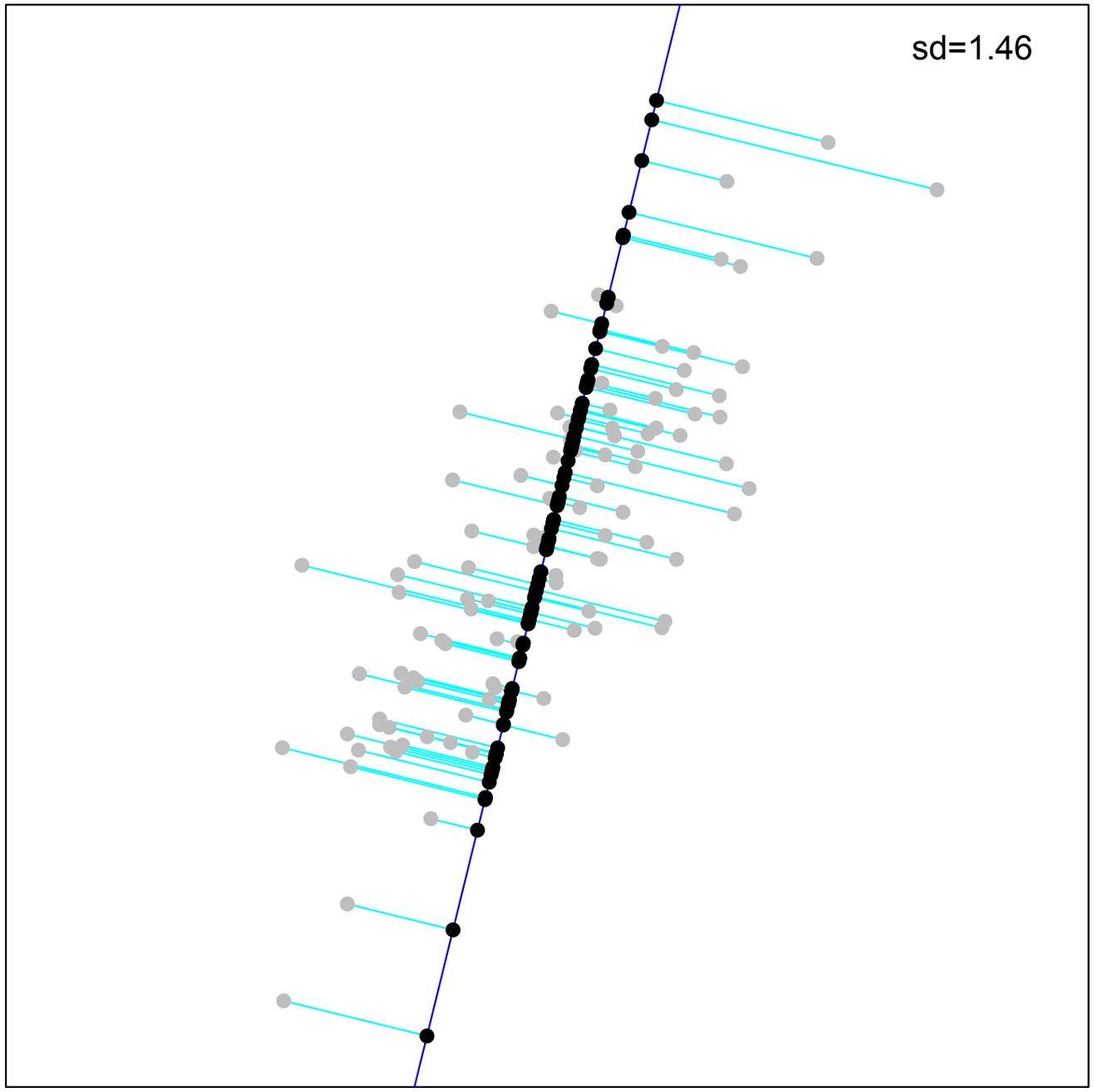
You can also formalize the notion of “aligned” by taking a sequential approach to PCA -- We define a new coordinate system one variable at at time, taking the first coordinate direction to be the one that gives us **the biggest spread** when we project our data along it

On the next three slides we just choose directions at random, project the data and examine its (univariate) spread -- The blue lines indicate the direction, the black dots are the projection of the data, and in the upper righthand corner we present the sample standard deviation of the projected data




sd=0.60


sd=0.91

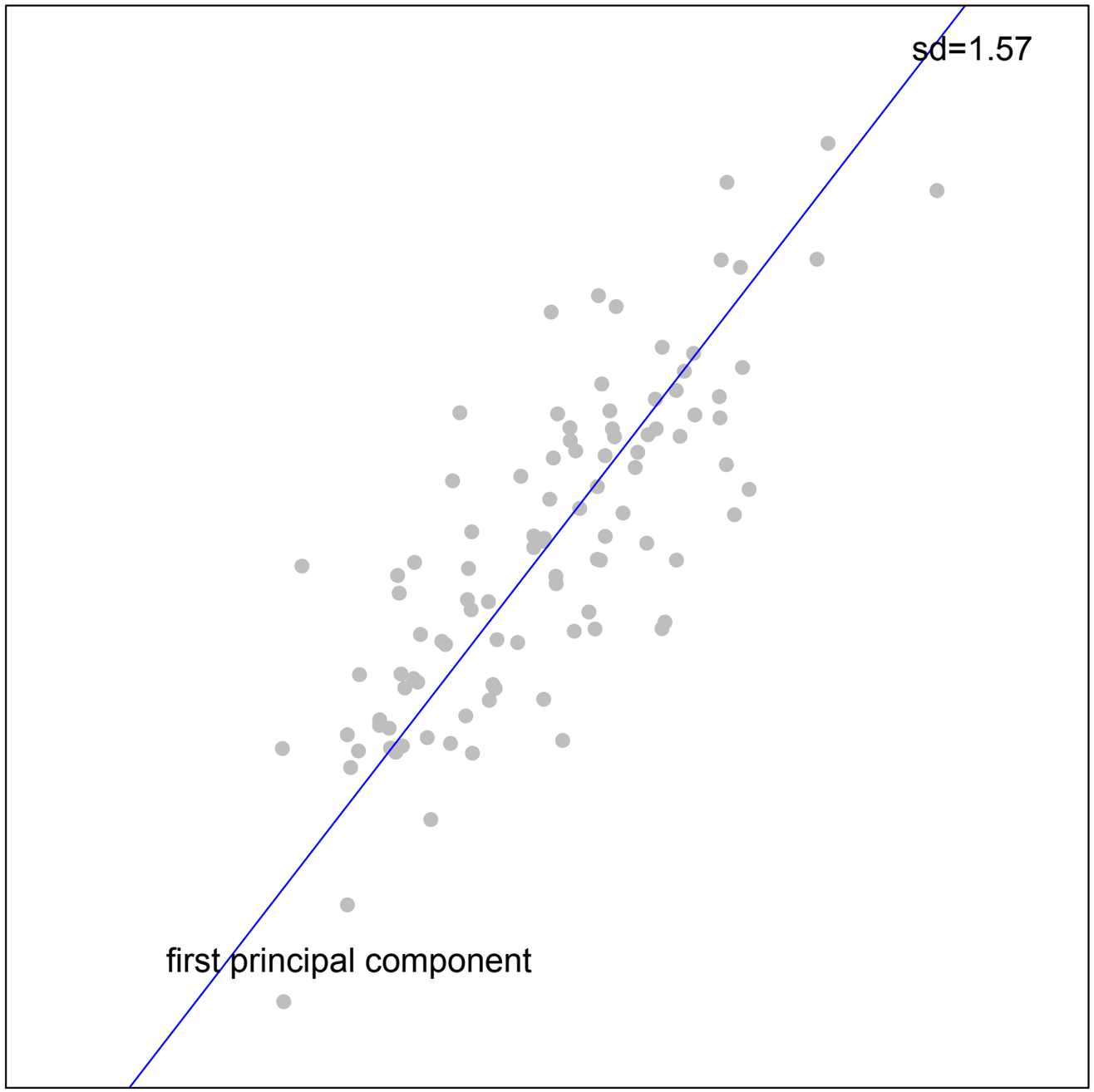

sd=1.46

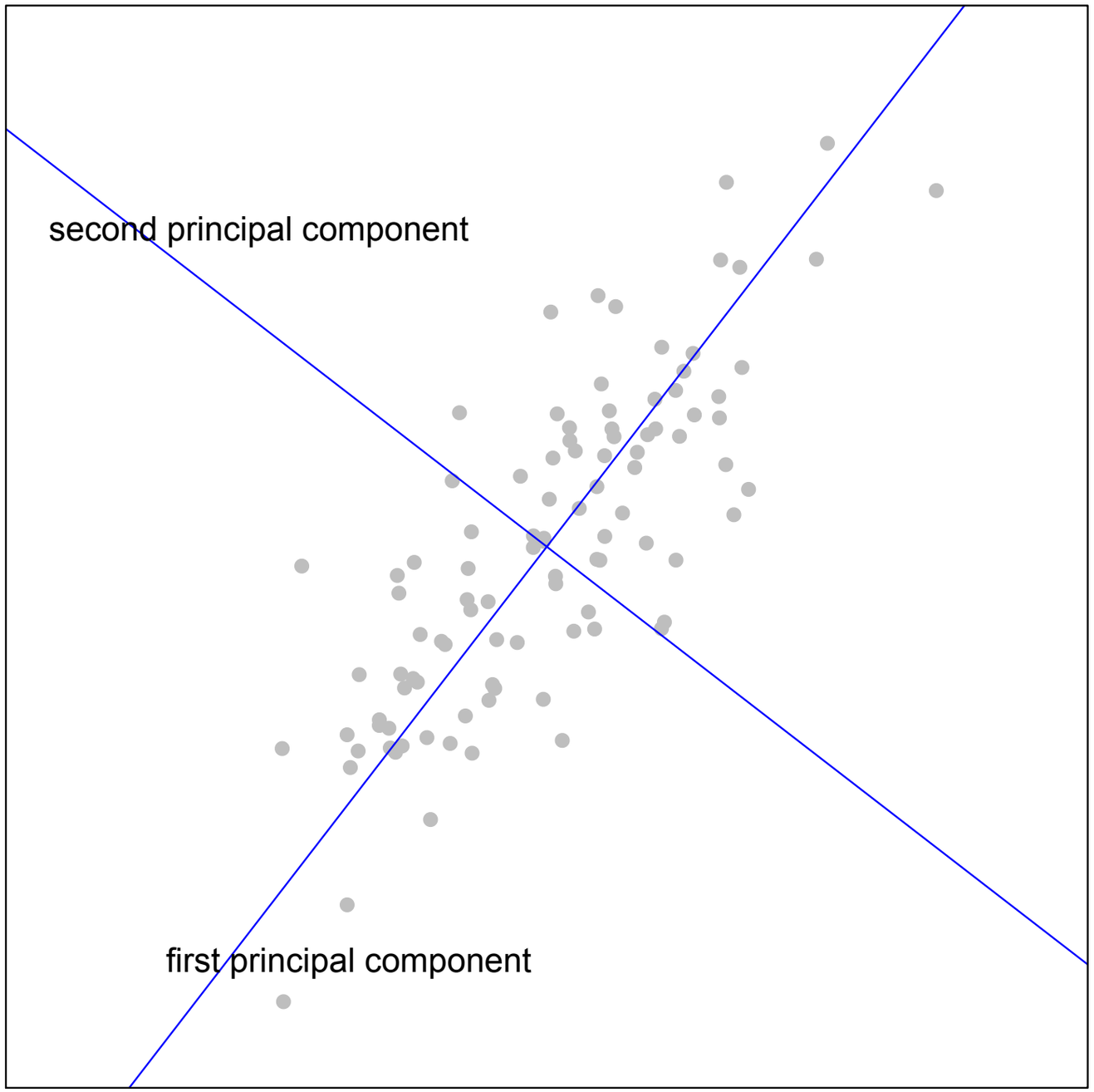
PCA

Clearly, when we select a direction that is “aligned” with the data, the projected values are more spread out -- The first principal component is just **the direction which yields the largest spread**

The second principal component is the direction, **orthogonal to the first, giving the next largest spread** -- For our little two dimensional data set, once we picked one direction, the second is set by orthogonality

For data that live in higher dimensions (like our 381-dimensional space of ingredients), we can continue adding directions, each time making sure **the new addition is orthogonal to the previous ones** and that, subject to this constraint, **the spread of the projected data is as large as possible**


sd=1.57
first principal component

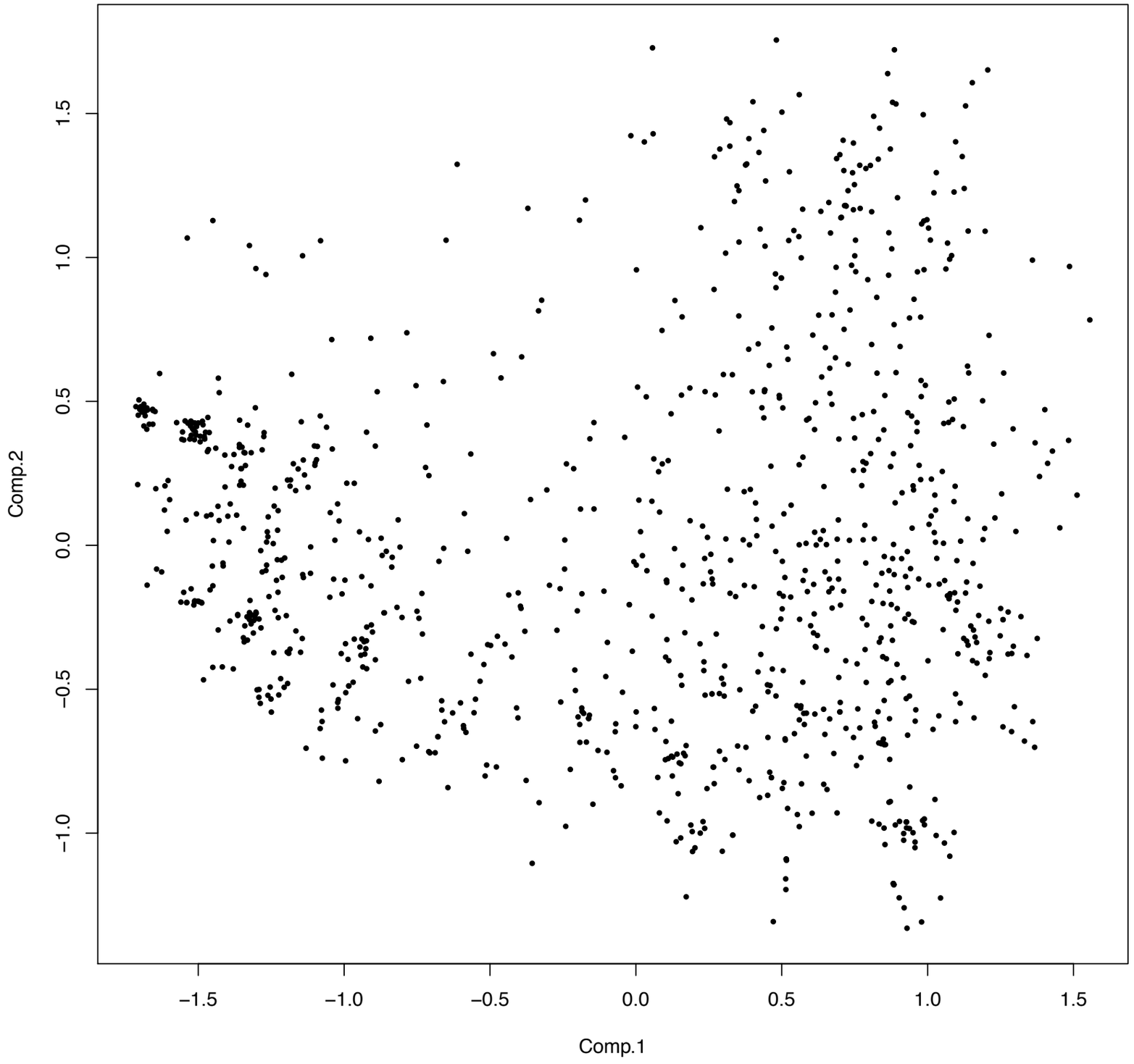

second principal component
first principal component

PCA

Principal components provide a new coordinate system that is aligned to the features of the data -- It is often used as **a tool for dimension reduction**, focusing our attention on the first few components as they capture as much of the variation in the data as possible

In this sense, then, we can **plot the data projected onto the first two principal components**, say, to generate the view from the Grand Tour that will account for the greatest variation in the data

Let’s now apply the projection to our 381-dimensional recipe data and plot the “scores” for the first two components, in effect plotting our data in this new coordinate system...


Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

Naming

The columns in our original data set are meaningful, they represent the presence or absence of an ingredient -- When faced with the new coordinates of PCA, we should try to understand **what the coordinates represent**

Each principal component is just **a linear combination of our original variables** -- We have taken our 381 ingredient variables and have replaced them with 381 principal components, new variables that represent decreasing variation

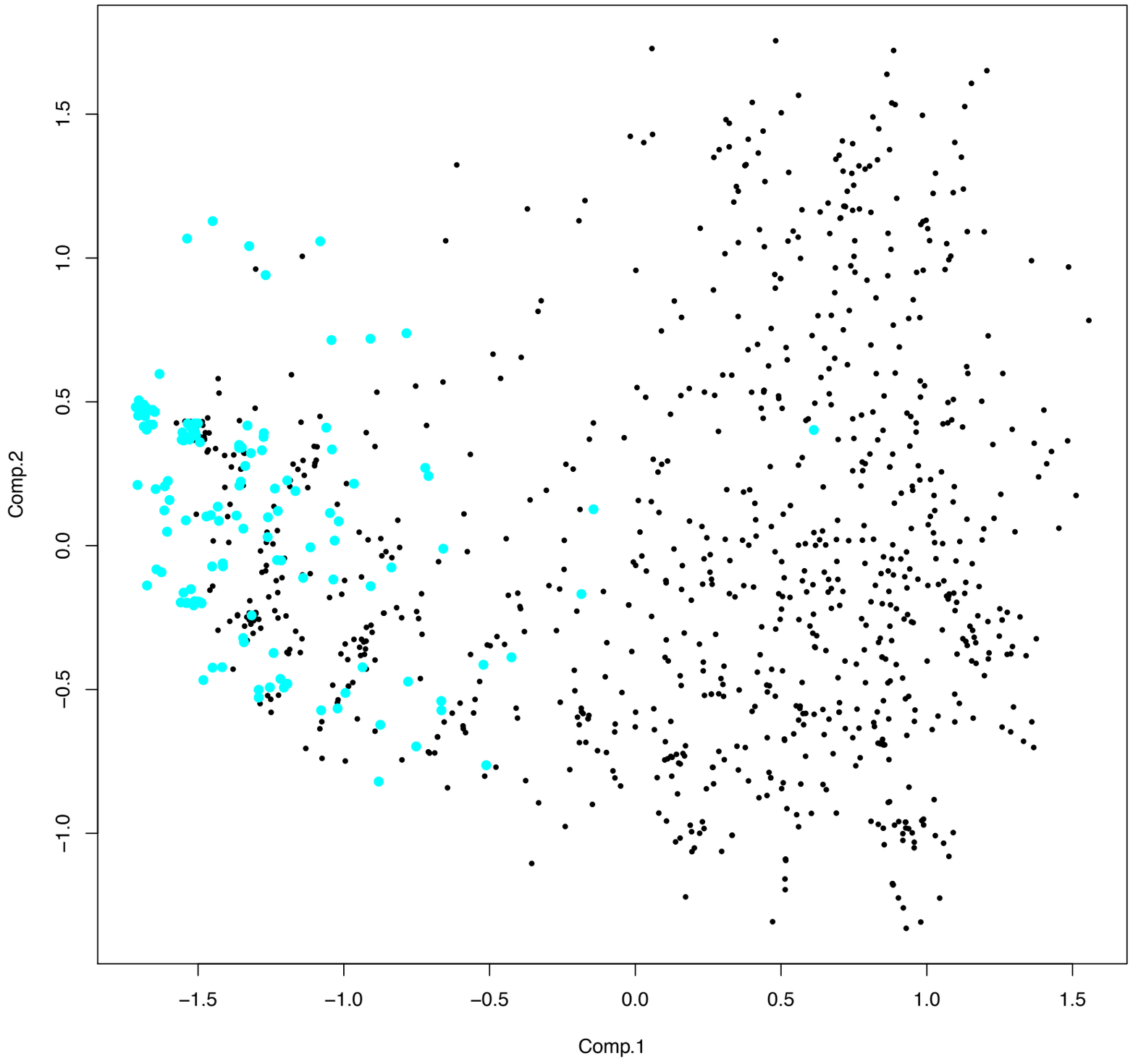
Naming

The factor loadings associated with each principal component are just the coefficients assigned to each of the original ingredients to make the new variables -- Here is what we get for the first, looking at the largest in absolute value

PC1 = -0.19 **cake.mix** - 0.13 **water** - 0.11 **vegetable.oil** - ...
+ 0.23 **butter** + 0.29 **baking.powder** + 0.31 **vanilla.extract** + 0.37 **sugar** + 0.42 **salt** + 0.44 **flour**

Those recipes scoring low in this first principal component direction involve ingredients like **cake mix, water and vegetable oil** -- Those scoring high involve **flour, sugar and baking powder**, ingredients that you need when you make a cake from “scratch”

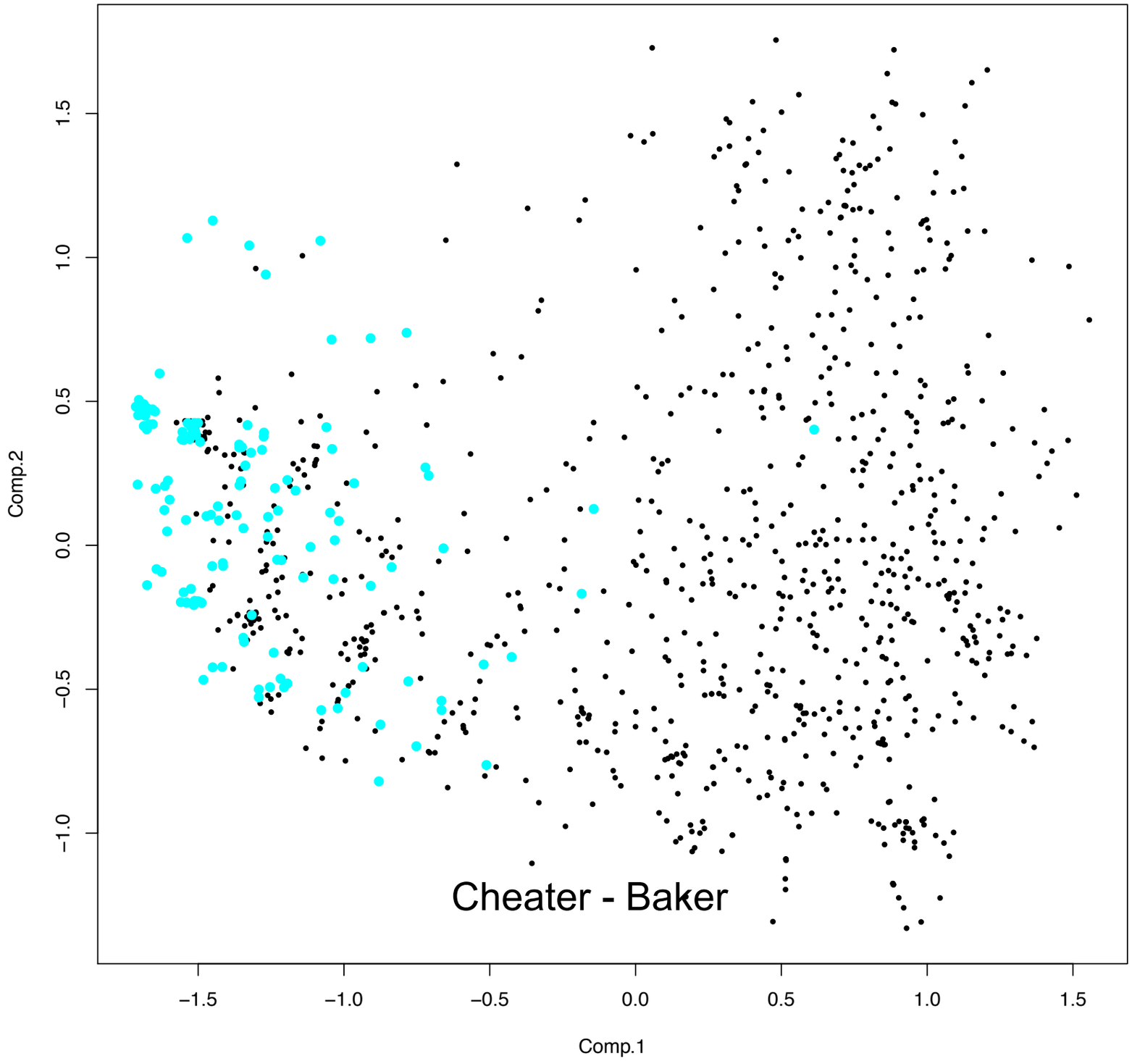
In the next slide we color recipes cyan that involve cake mix as an ingredient...


Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

Naming

With this in mind, the first principal component tends to divide recipes based on whether someone needs to **bake from scratch or bootstrap with a cake mix** (or actual cake in some cases!)

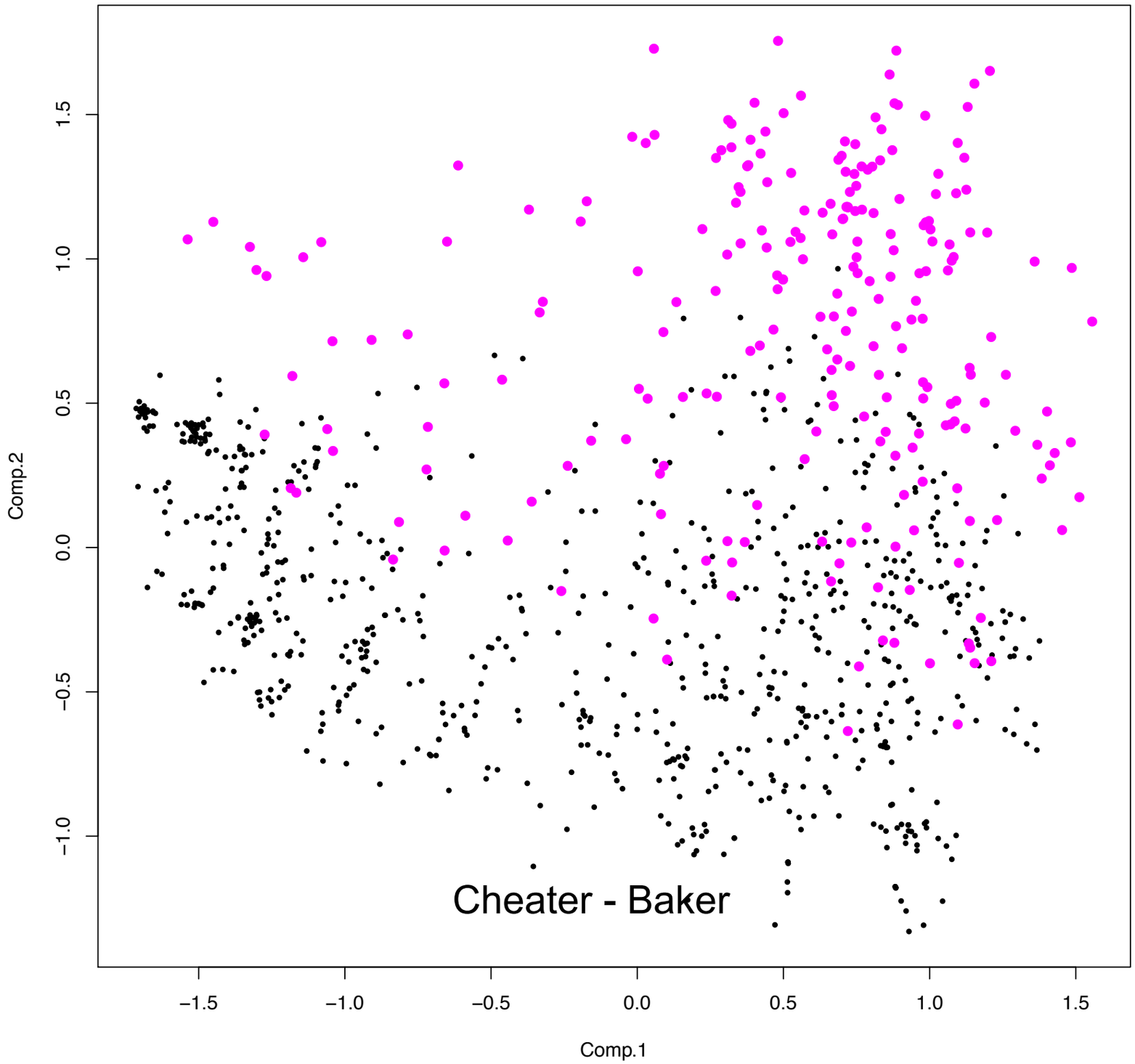
We’ll call this the Cheater-Baker axis...

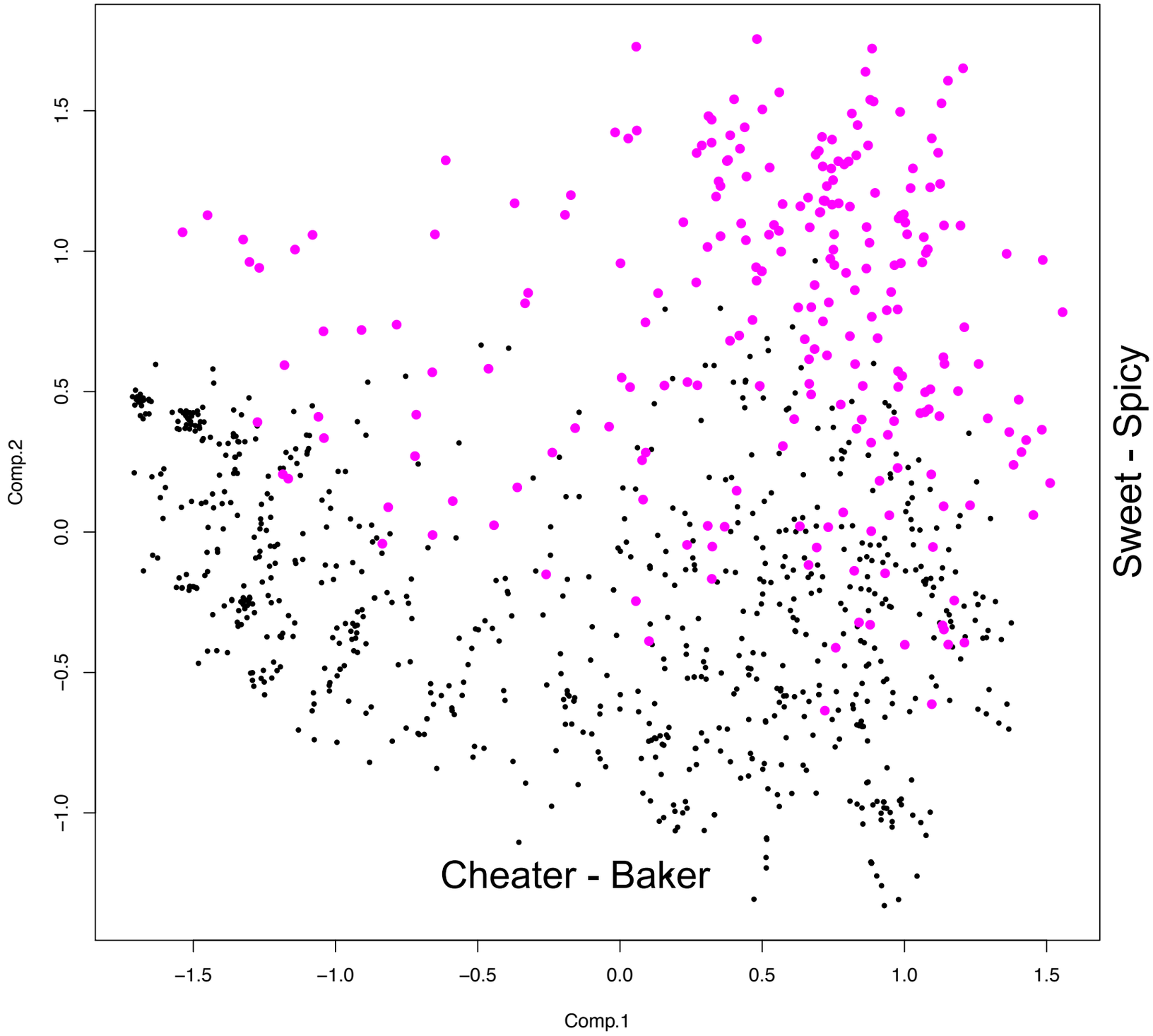

Cheater - Baker
Comp.1
Comp.2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

Naming

Looking at the loadings for the second principal component we see high scores for **cinnamon, nutmeg, raisins and carrots** -- On the low end we see **butter, powdered sugar and lemon zest**

We'll call the second principal component the Sweet-Spicy axis -- On the next slide, recipes containing cinnamon are highlighted in magenta


Cheater - Baker
Comp.1
Comp.2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Cheater - Baker
Sweet - Spicy
Comp.1
Comp.2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

Under the hood

Principal components, with their focus on variability, operate on the sample variance-covariance matrix of our data -- Specifically a 381-by-381 matrix of covariances between the ingredient variables

After a little (linear) algebra, we find that the principal components are really just the eigenvectors of this matrix, and the variation they capture the eigenvalues -- So computationally, this technique is pretty straightforward

A general methodology

Using understanding or “reading” large quantities of text as an application, we introduced a number of methodological tools for viewing and structuring high-dimensional data

While pairwise scatterplots can gives us a sense of the (marginal) distribution of our data, **projection and the visualization of a series of (random or guided) projections** is an effective approach for uncovering more complex structure

Principal components analysis focuses on one particular projection of the data -- It is a technique for **reducing the dimension of a data set** by compacting its major variation into (typically) a few derived variables

It has a huge number of applications...

Document clustering

This simple investigation suggests that recipes fall into **natural groups depending on their ingredient lists** -- This is probably obvious, but to see it in the data is always a satisfying moment

We are interested in these groups or “clusters” not only because they seem to agree with our intuition, but also for a variety of practical reasons -- In web searching, for example, it might be useful to not just return a long list of pages containing a key word, but instead **organize things by “topic” or cluster**

Google News, for example, does this for news stories...

Google News

news.google.com/topstories?hl=en-US&gl=US&ceid=US:en

Google News

Search for topics, locations & sources

M

- Top stories
- For you
- Following
- Saved searches
- COVID-19
- U.S.
- World
- Your local news
- Business
- Technology
- Entertainment
- Sports

https://news.google.com/foryou?hl=en-US&gl=US&ceid=US%3...

Headlines

More Headlines

COVID-19 news: See the latest coverage of the coronavirus (COVID-19)

DOJ launches inquiry into Minneapolis police operations, a day after Chauvin guilty verdicts

USA TODAY · 2 hours ago

- **Derek Chauvin wrote attorney's number on hand in case of guilty verdict: report**
  Fox News · 18 hours ago
- **Chauvin verdict: Seeing is believing | COMMENTARY**
  Baltimore Sun · 5 hours ago · **Opinion**
- **Medcalf: Chauvin trial verdict offers glimmer of hope of what Minnesota can become**
  Minneapolis Star Tribune · 16 hours ago · **Local coverage**
- **In Chauvin verdict, Black and white Americans catch glimpse of one another's worlds | Editorial**
  Detroit Free Press · 1 hour ago · **Opinion**

View Full Coverage

New York

Thunderstorms

67°F

Today	Thu	Fri	Sat	Sun
68°F 36°F	52°F 40°F	65°F 49°F	68°F 53°F	62°F 45°F

C | **F** | K More on weather.com

Severe Thunderstorm Warning

New Jersey

At 211 PM EDT, a severe thunderstorm was located over Passaic, moving east at 40 mph ... This severe thunderstorm will be near ... Lyndhurst and Rutherford around 215 PM EDT ...

18 minutes ago ·

Clustering

Clustering or, rather, dividing the data into natural groups, is also known as "unsupervised learning" -- It is unsupervised in the sense that we don't have any tags telling us that a document begins to one class or another (these problems are referred to as "supervised learning")

Clustering, like regression, is a statistical idea that has been around for **a long, long time** -- Rather than review the history, we'll simply note that there are hundreds of different approaches to identifying groups in data

K-means clustering

K-means clustering divides our data $X_1, \ldots, X_n$ into K groups $G_1, \ldots, G_K$ (you specify K), where the groups are associated with cluster centers $\mu_1, \ldots, \mu_K$ (points in the d-dimensional space along with the data)

A point X_i is associated with group G_j if its nearest center (distance again!) is μ_j -- So the cluster centers are attractors, and each group is defined as the data points closest to the given center

K-means clustering

Given data $X_1, \ldots, X_n$ and a pre-specified value of K, we "learn" this model by finding values for $\mu_1, \ldots, \mu_K$ so as to "minimize" the overall loss

$$V = \sum_{k=1}^{K} \sum_{X_i \in G_k} \|X_i - \mu_k\|^2$$

We solve this problem iteratively...

K-means clustering

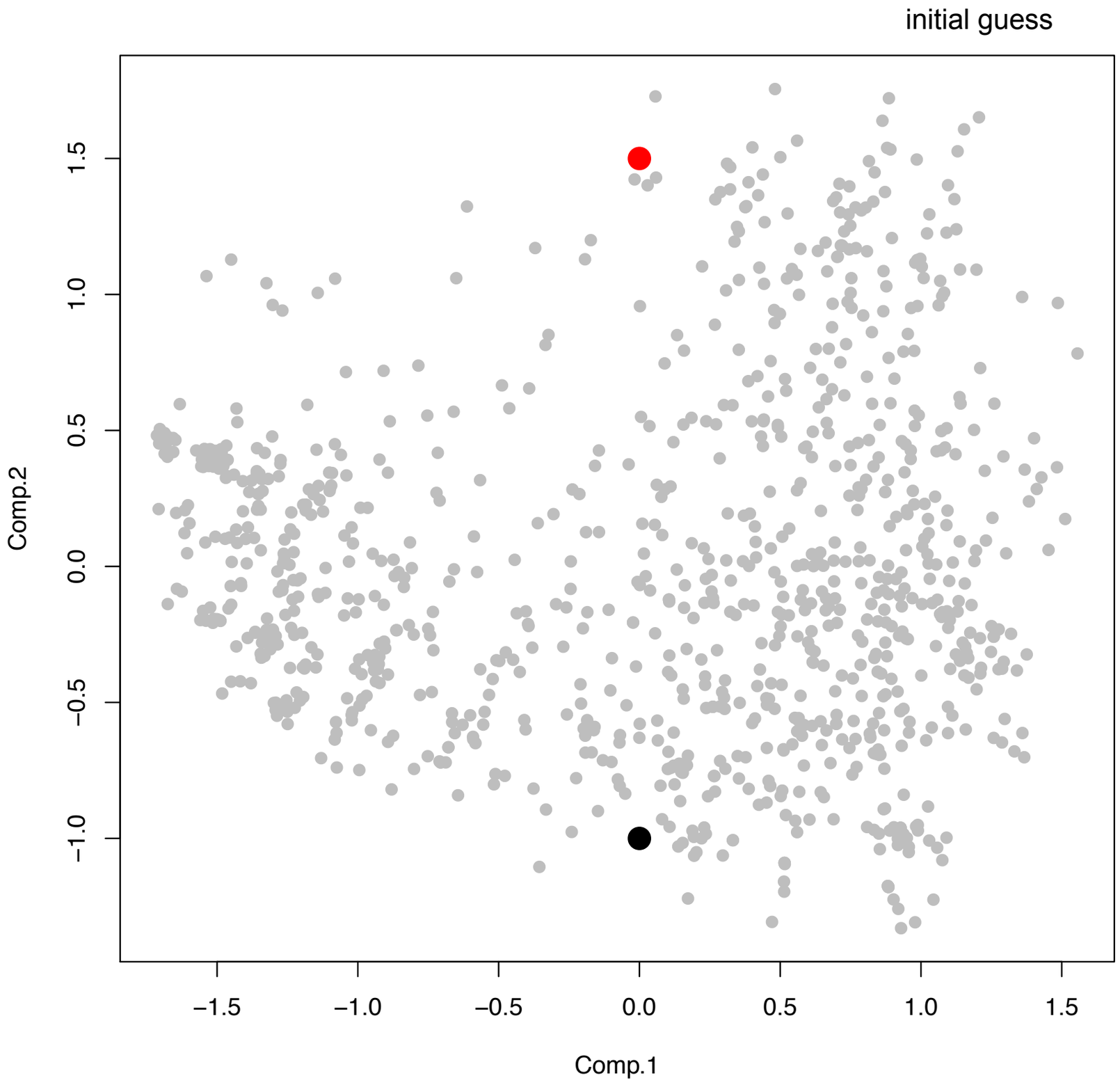
With K-means, we want to divide our data $X_1, \ldots, X_n$ into, well, K groups or clusters
-- The algorithm is pretty simple

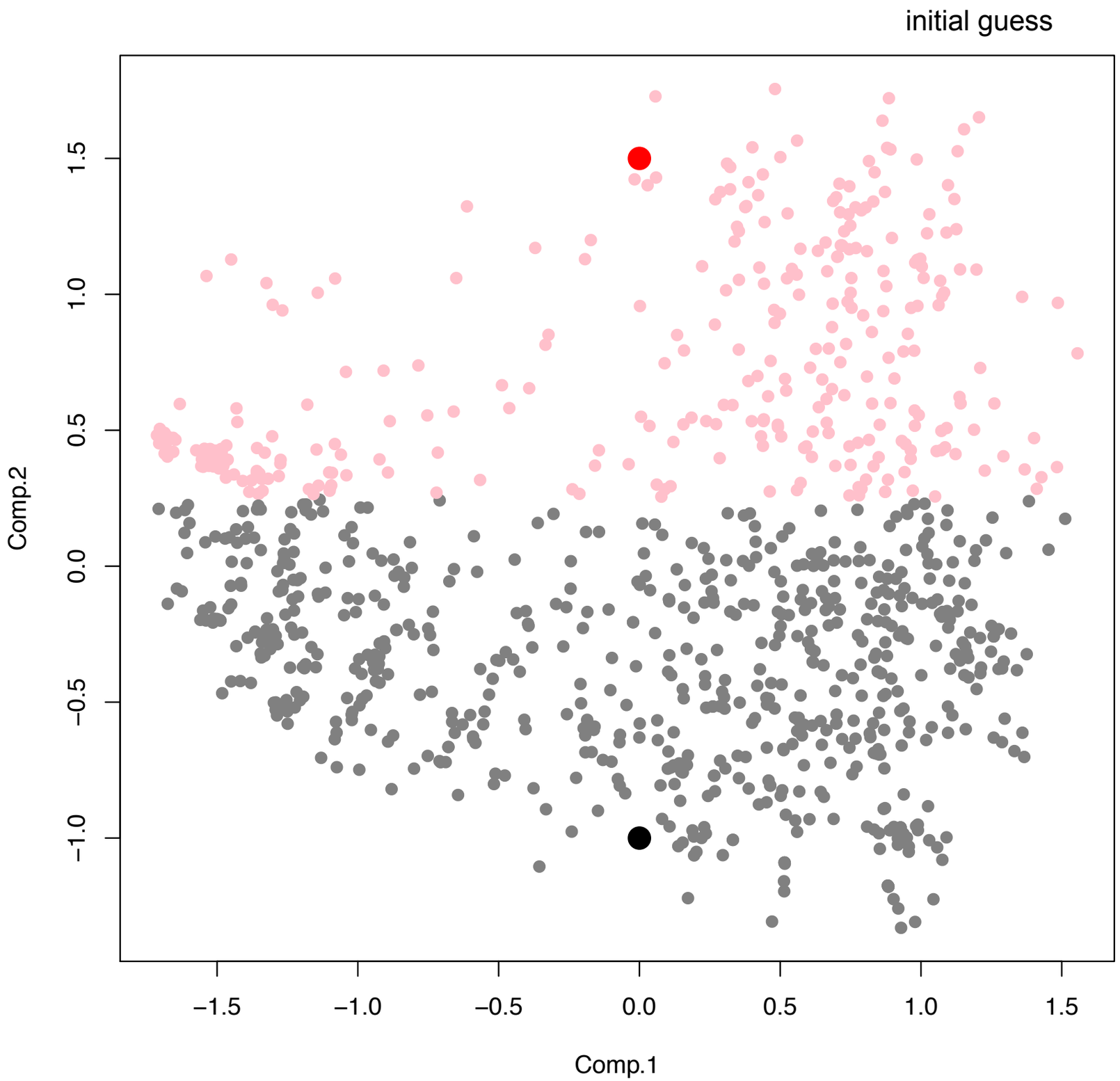
Make an initial guess for the centers $\mu_1^0, \ldots, \mu_K^0$

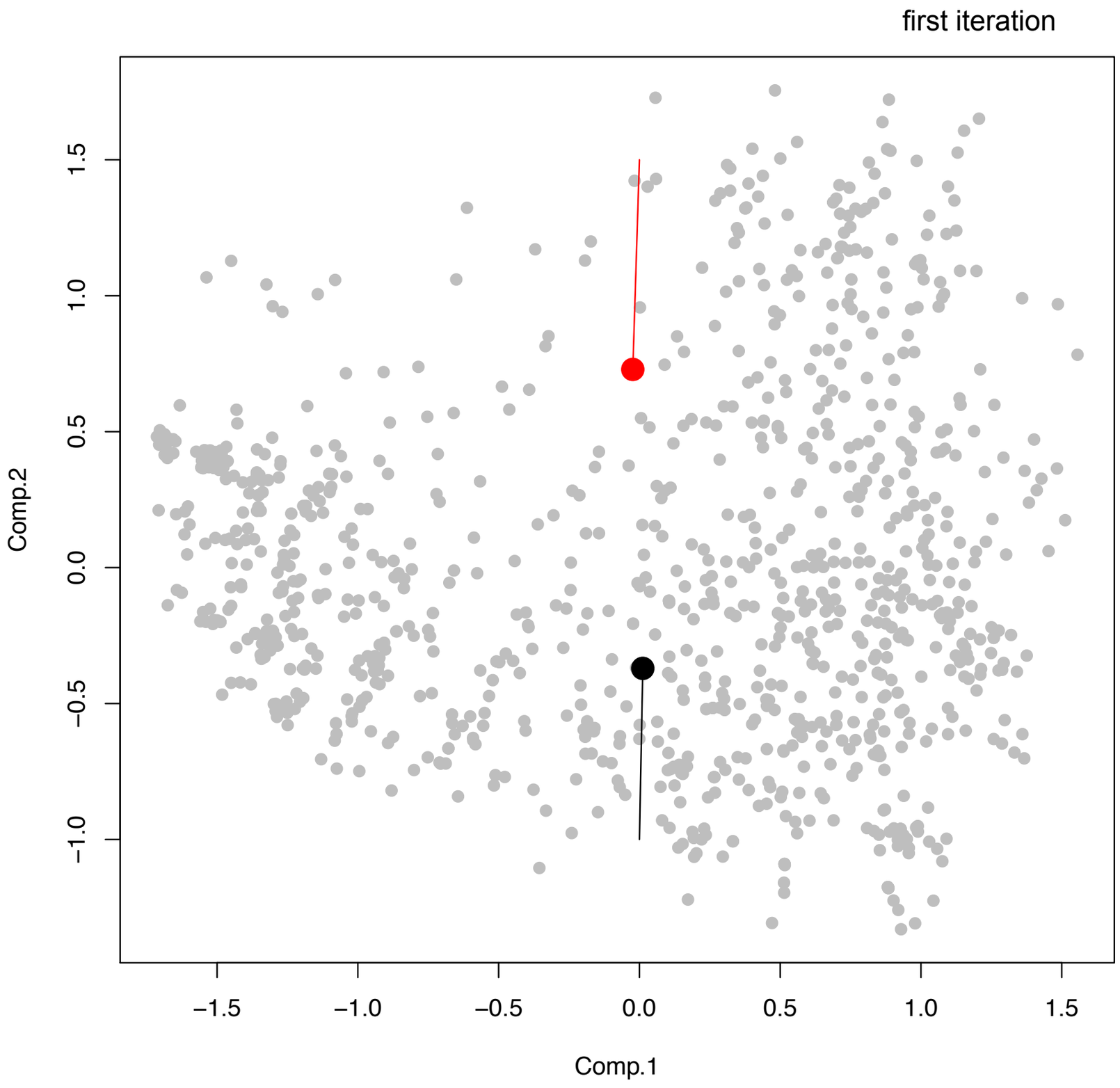
Until there's no change in these values do:

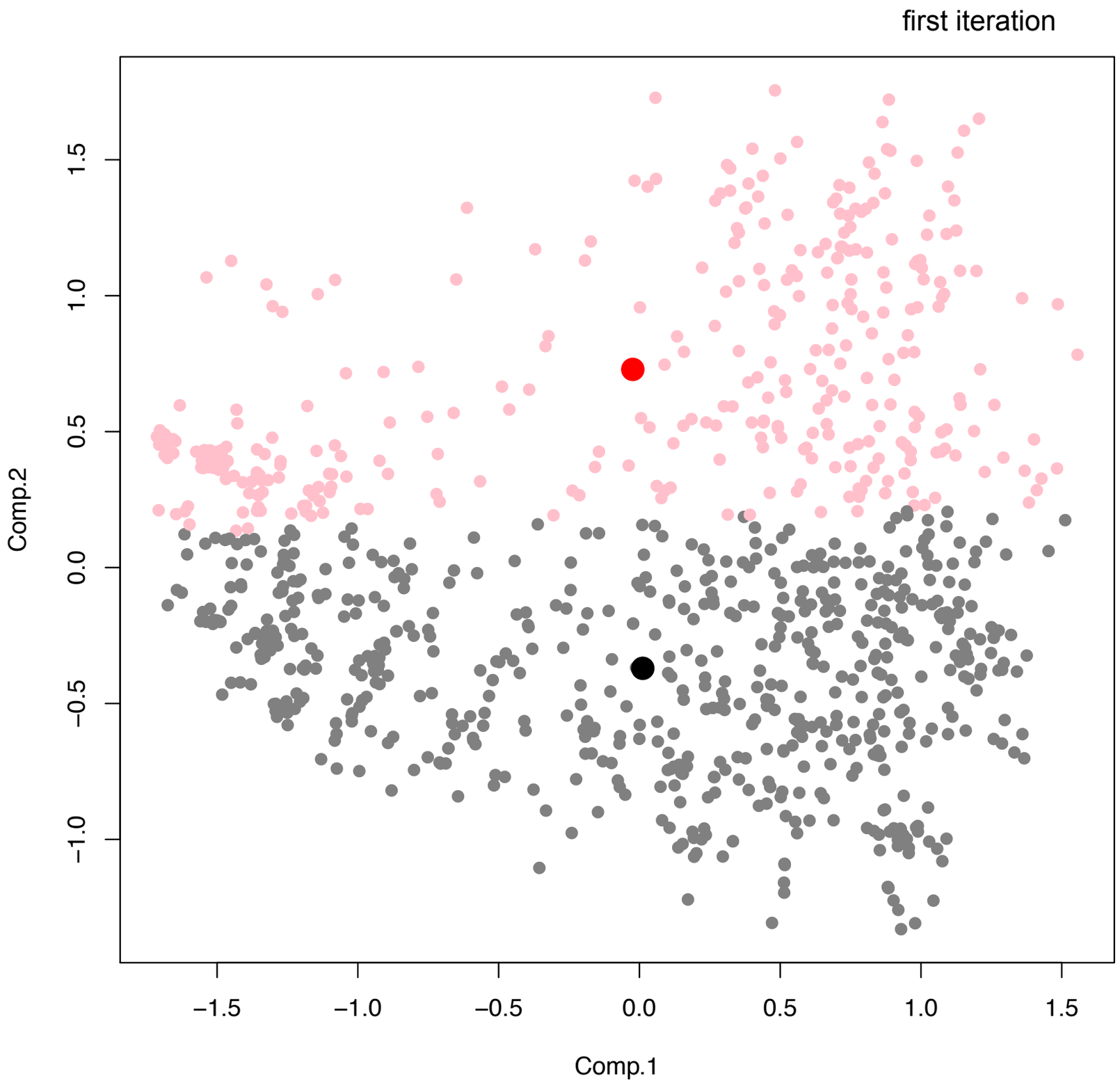
1. Assign each data point X_i to the nearest cluster center using simple Euclidean distance

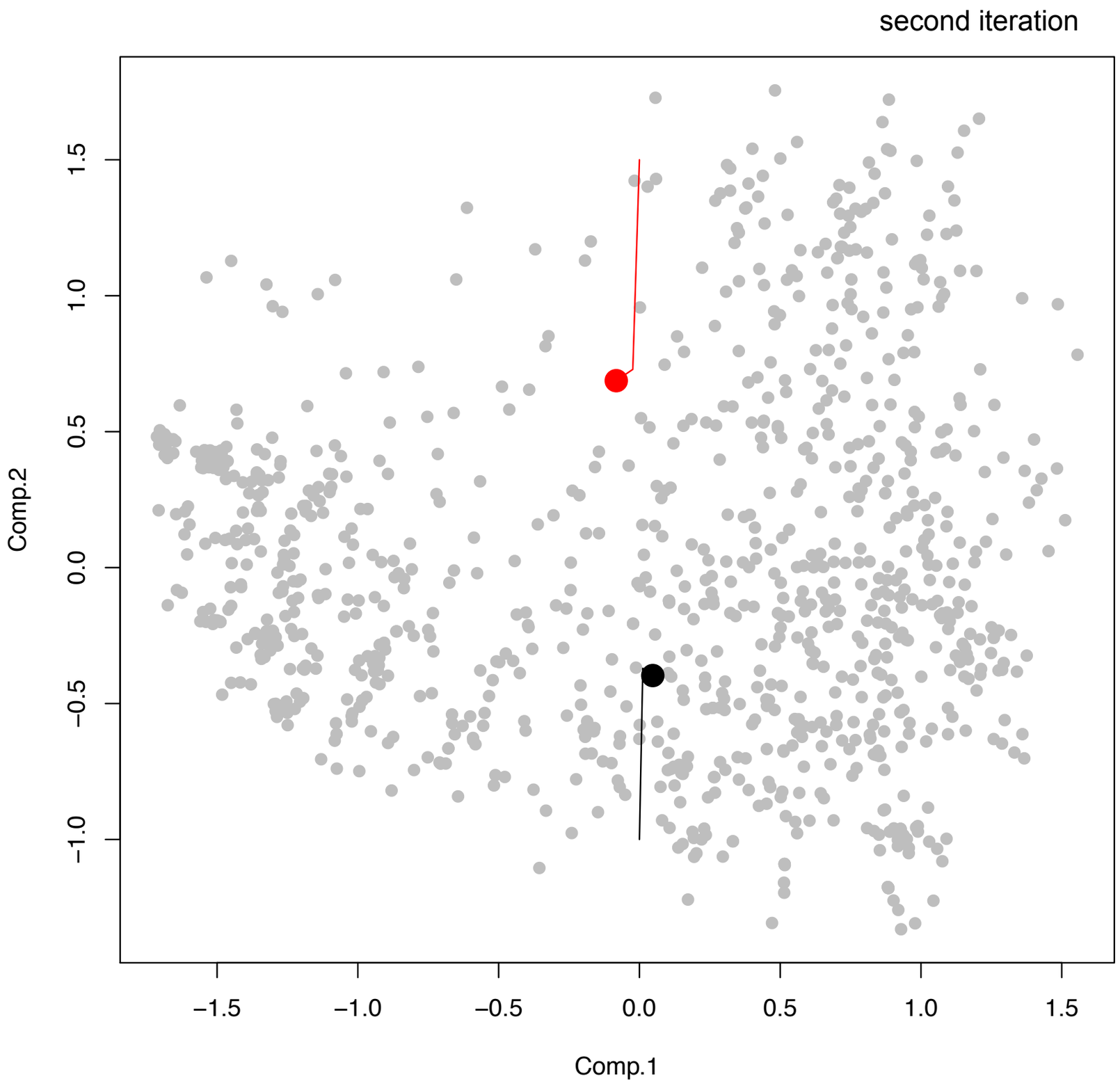
2. For each cluster k, update μ_k^1 to be the mean of all the points associated with the group


initial guess
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


initial guess
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


first iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


first iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


second iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

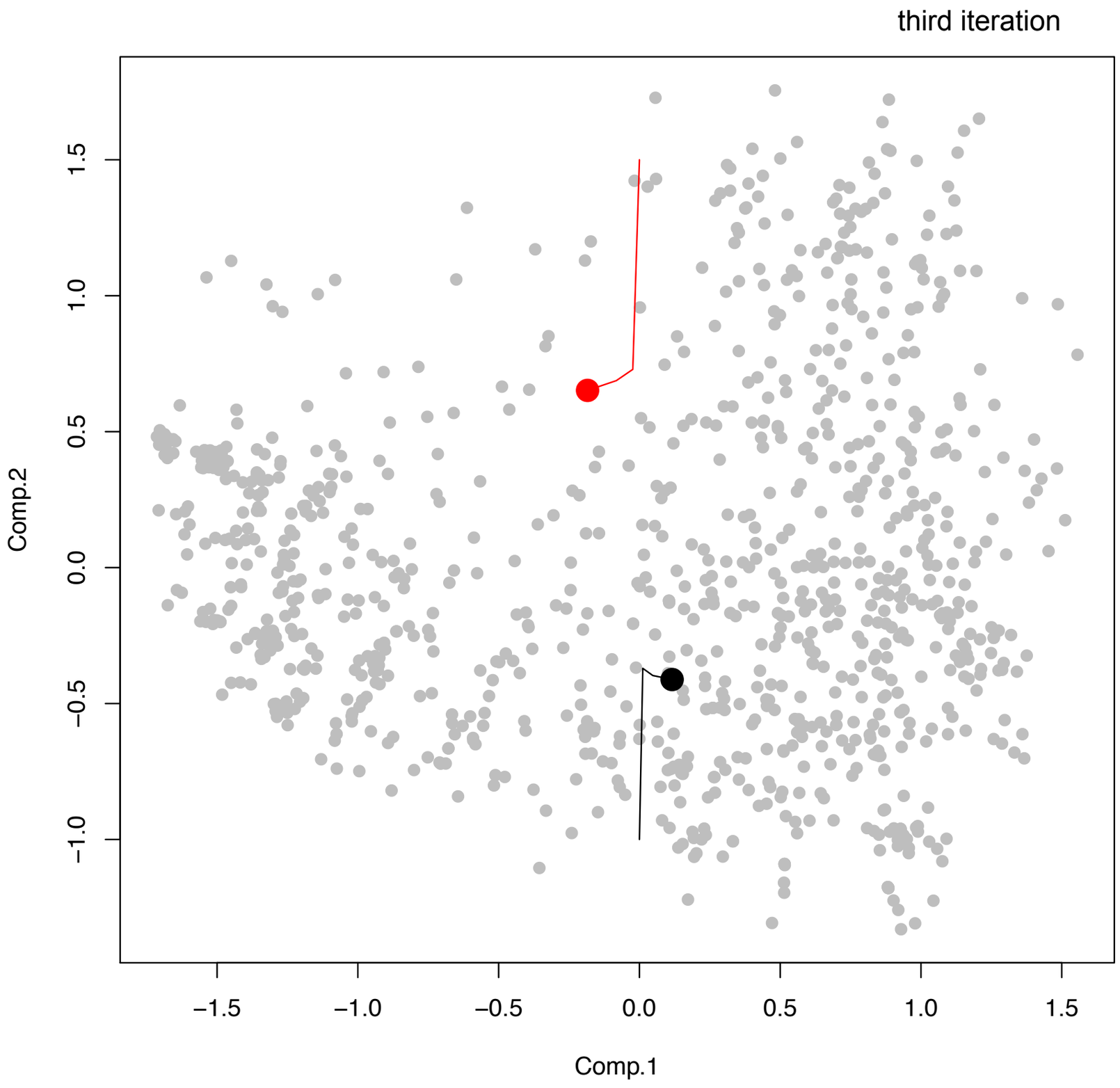
second iteration

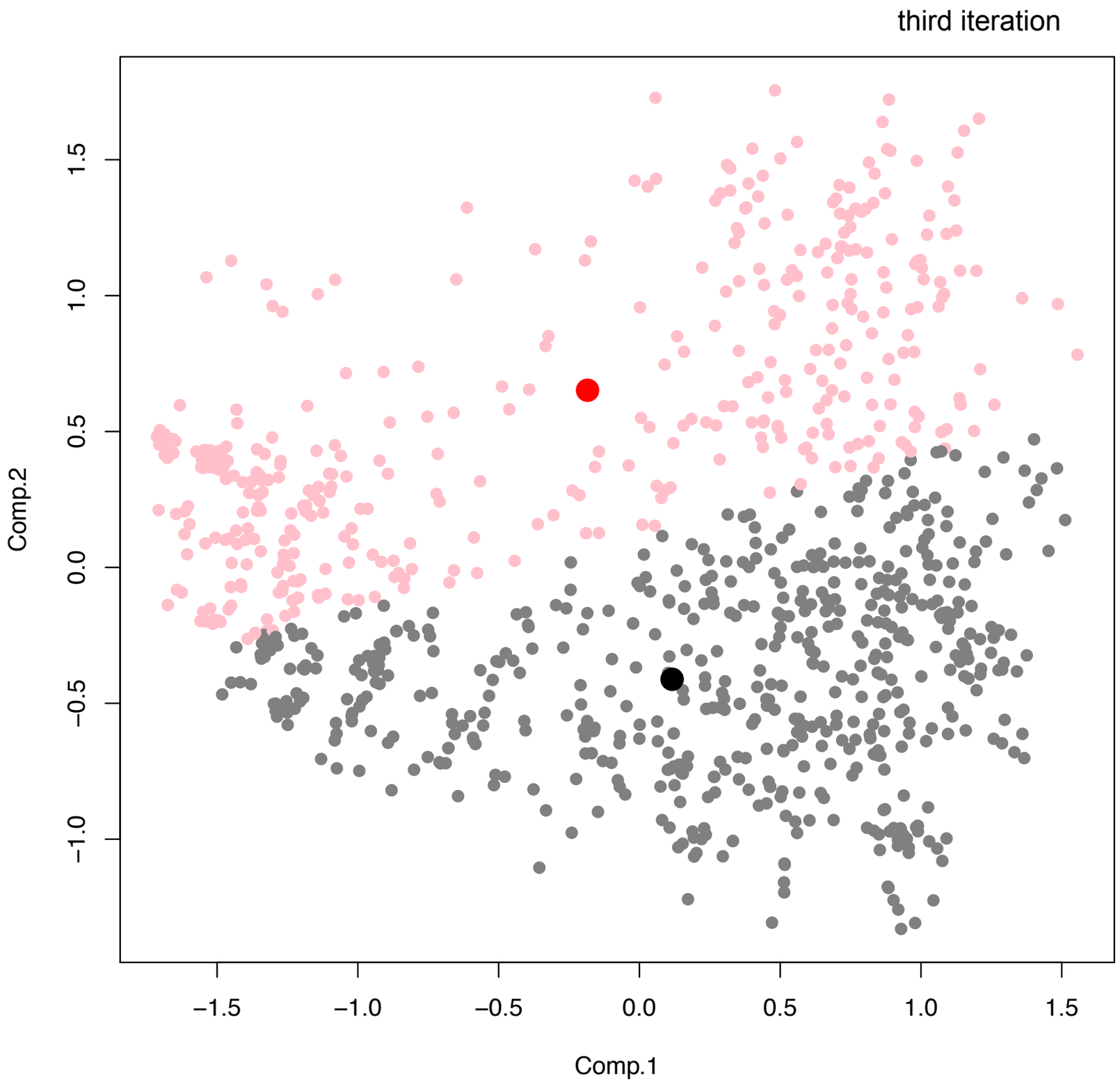
Comp.2

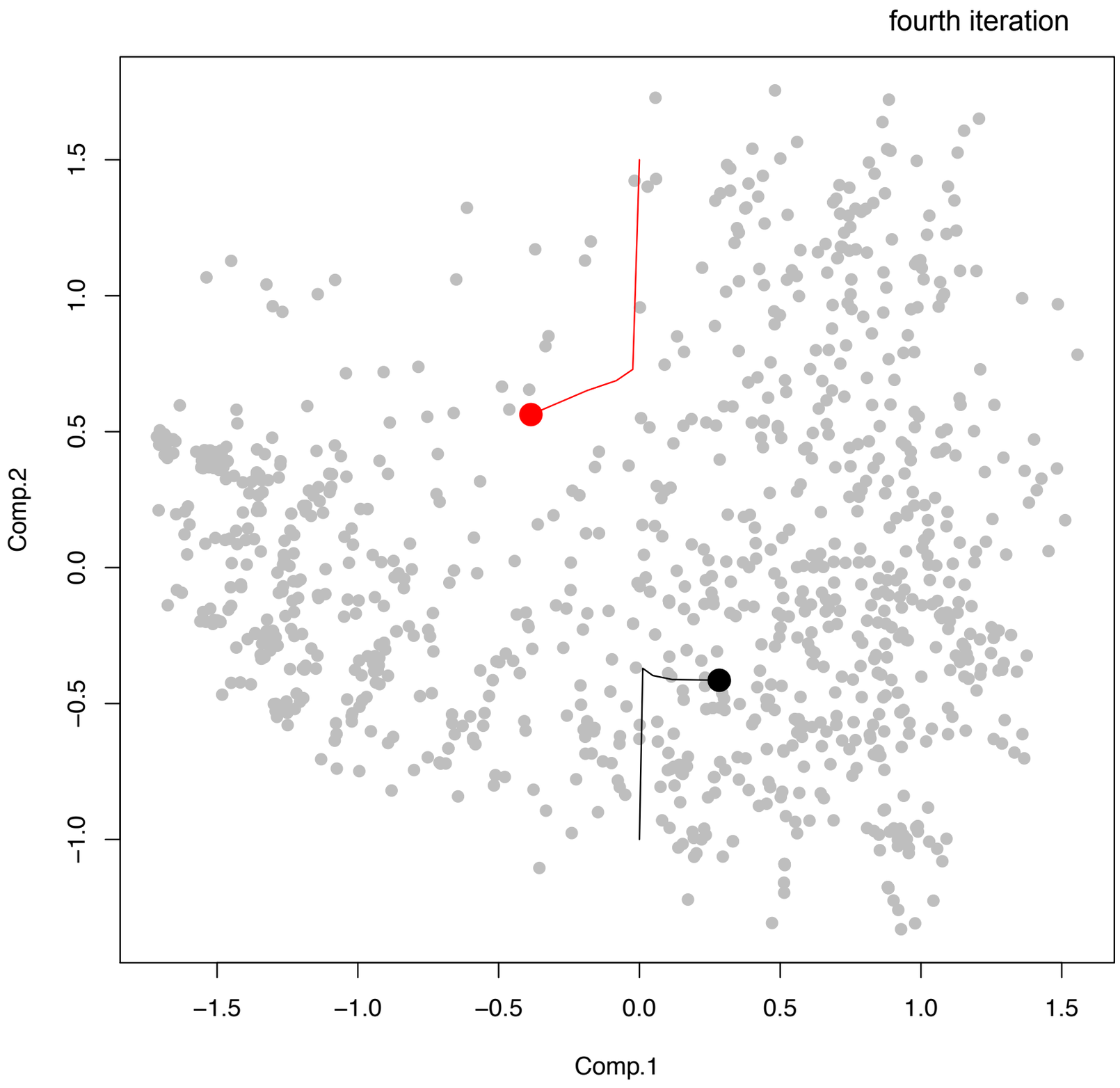
1.5
1.0
0.5
0.0
−0.5
−1.0

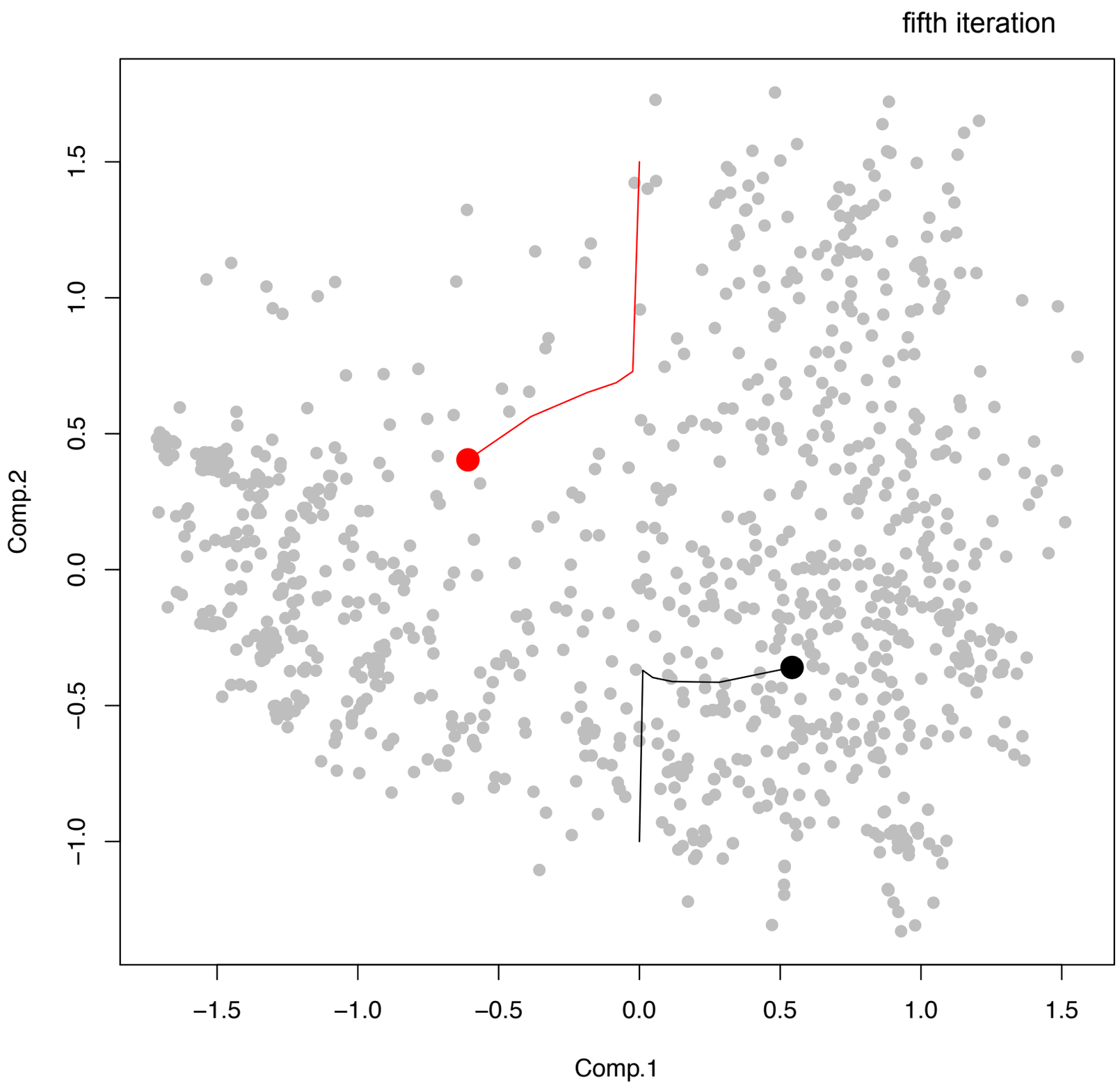
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

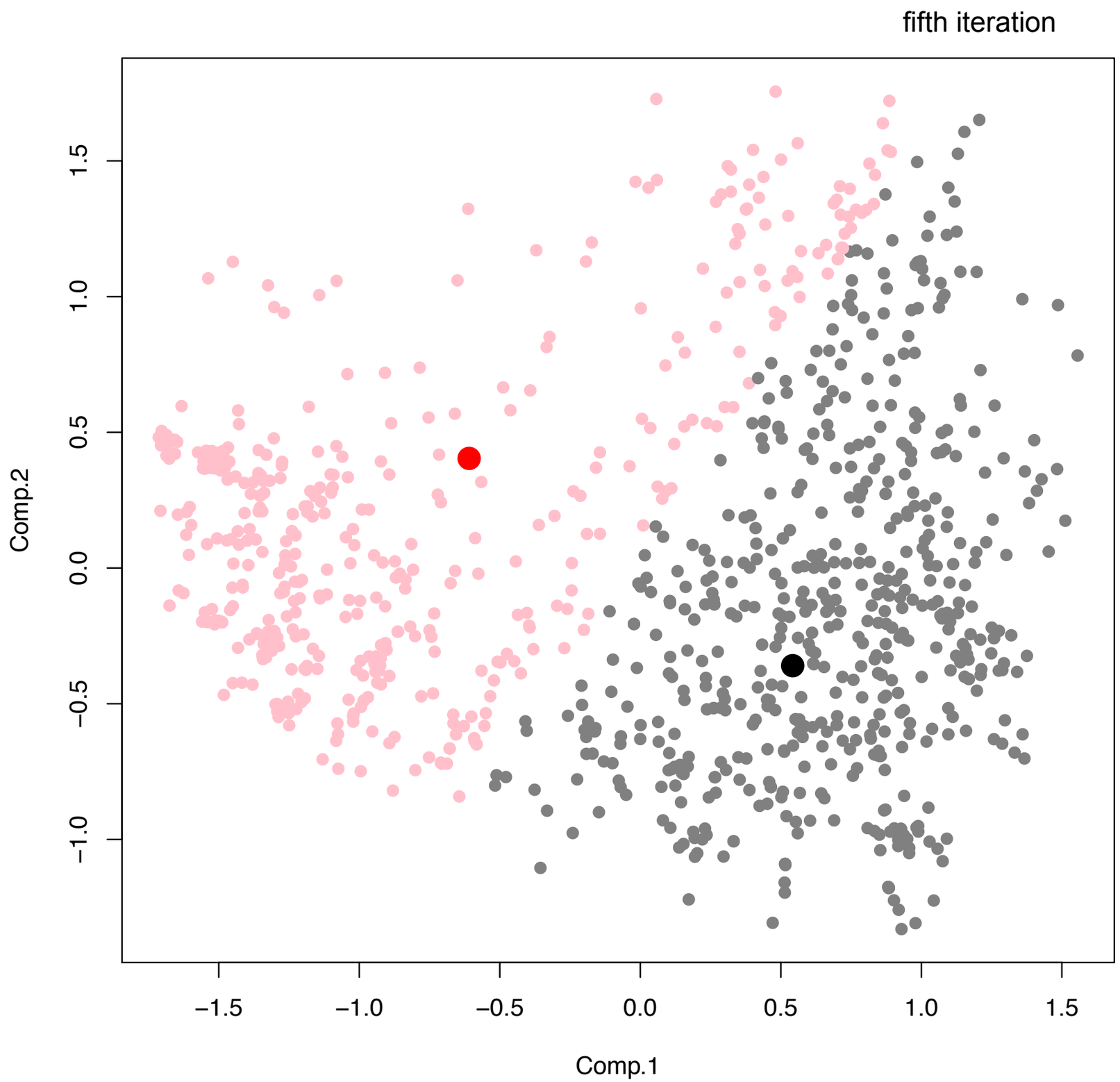
Comp.1

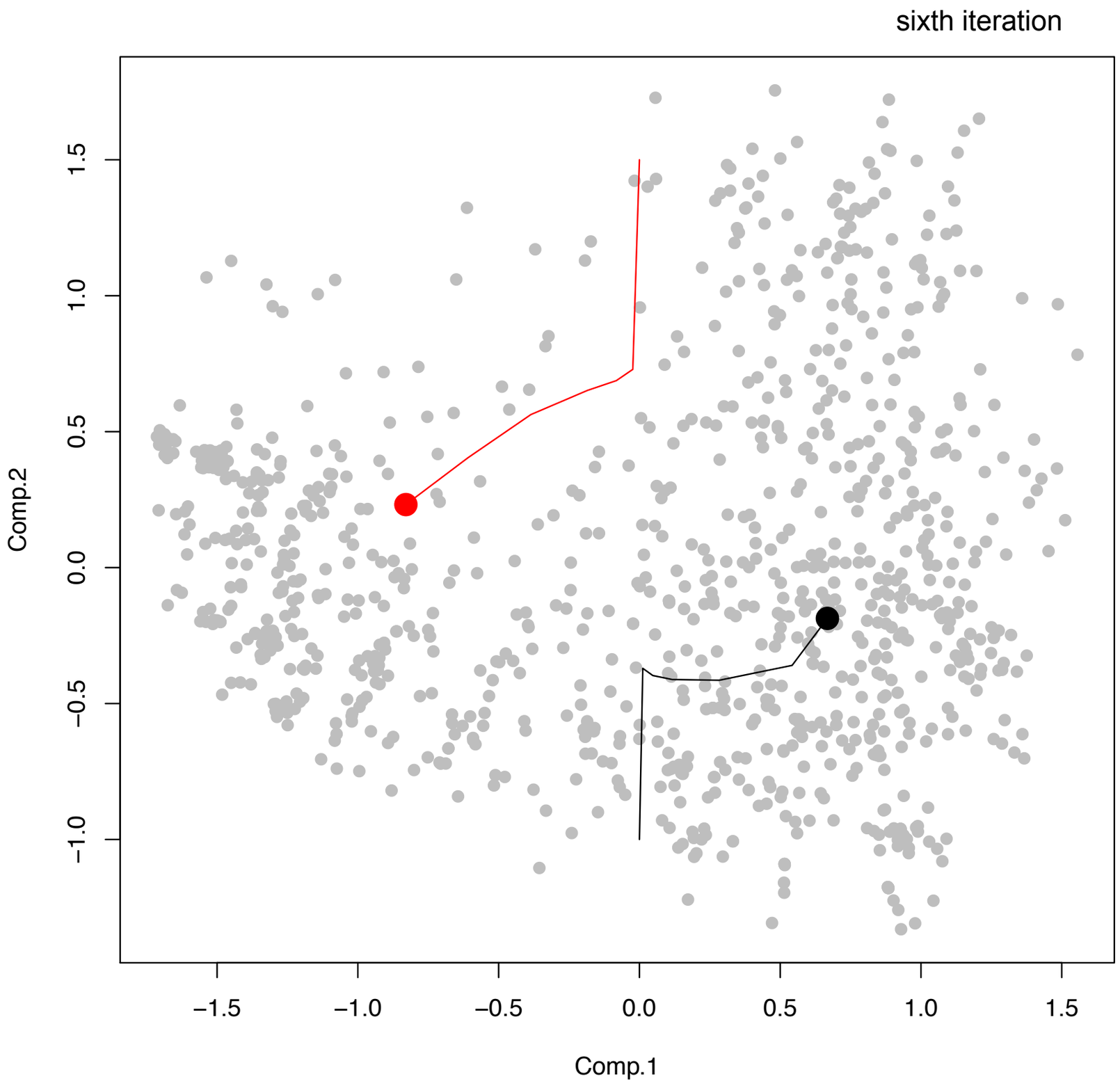

third iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

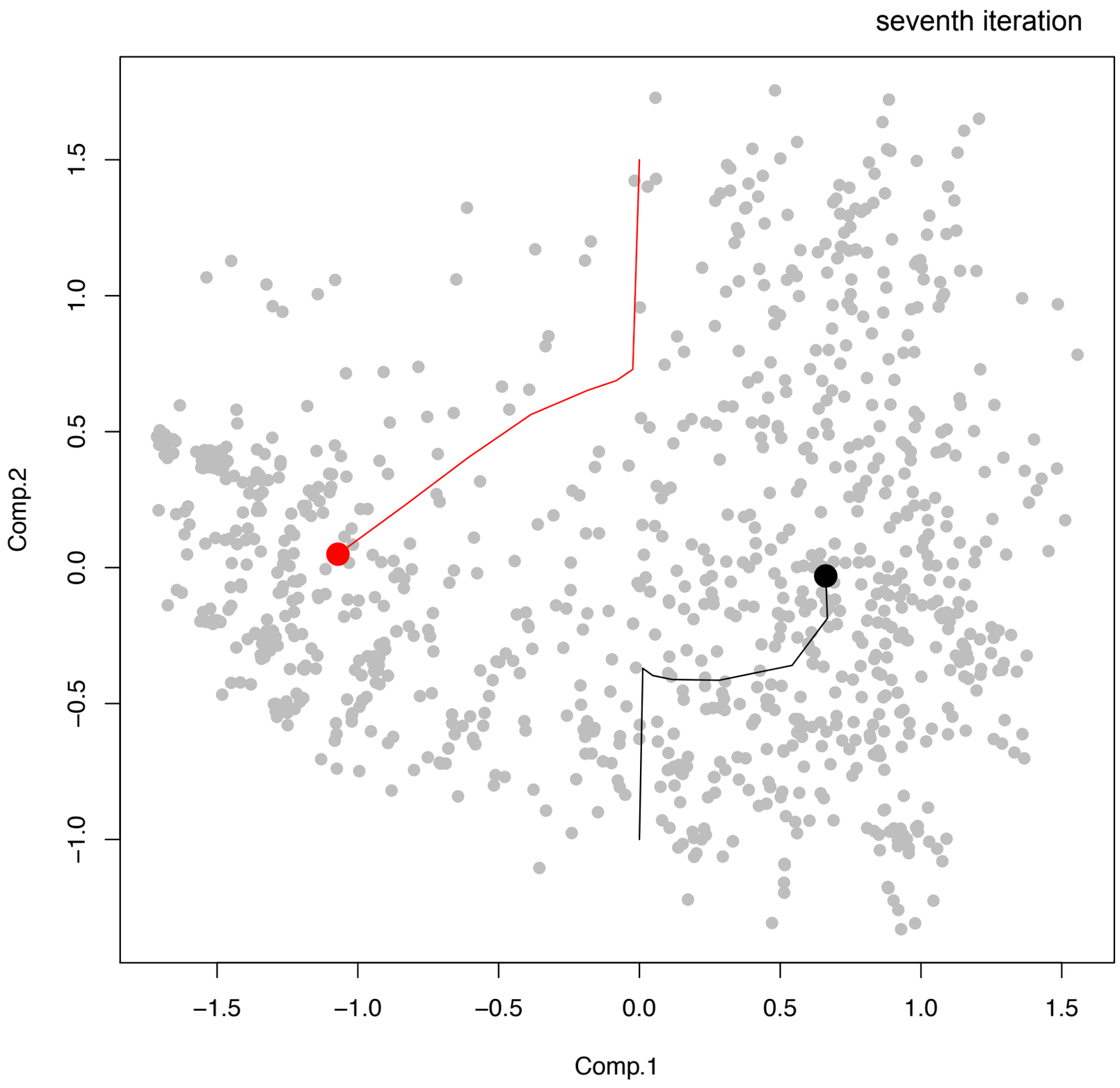

third iteration
Comp.2
Comp.1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

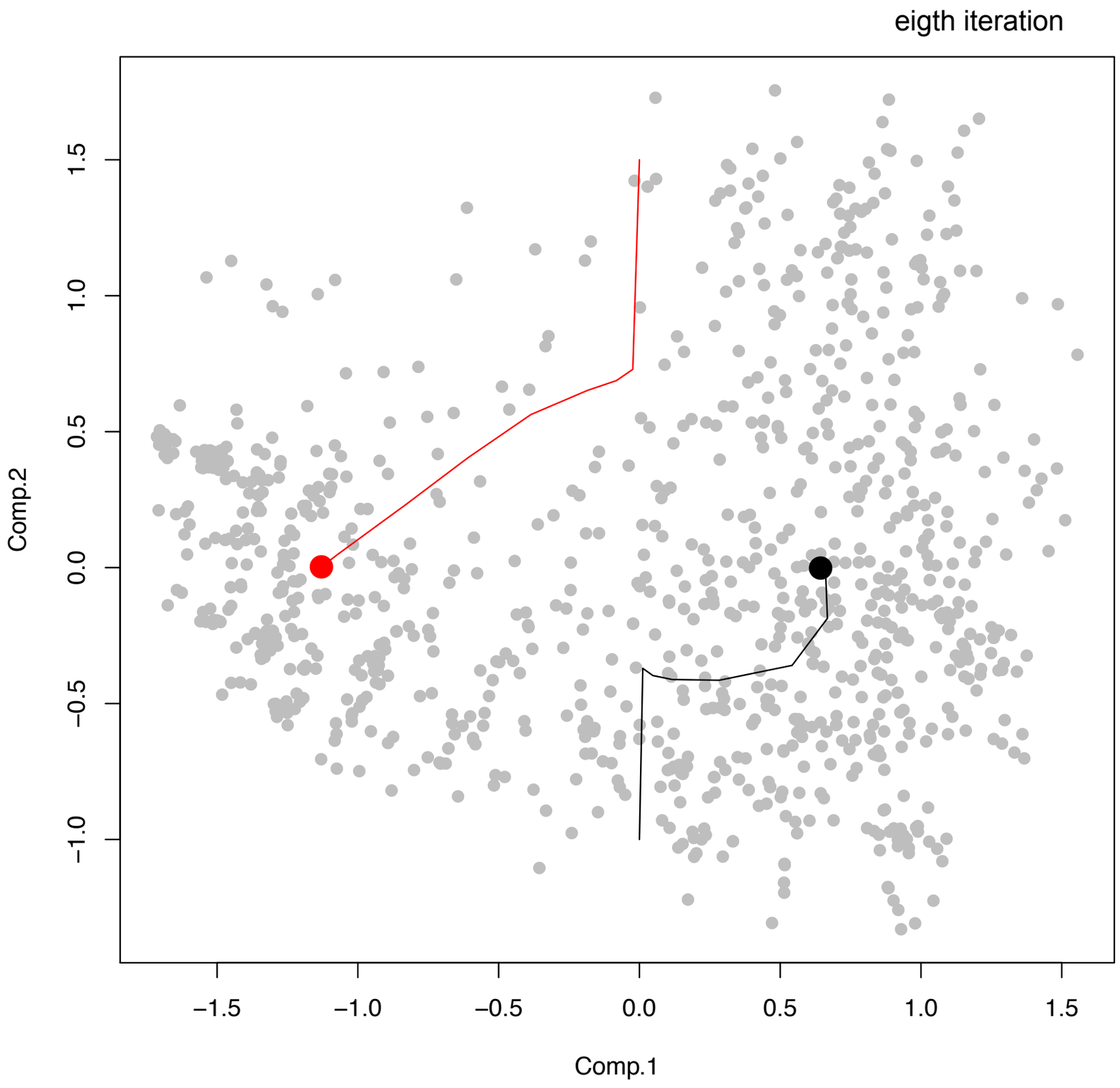

fourth iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

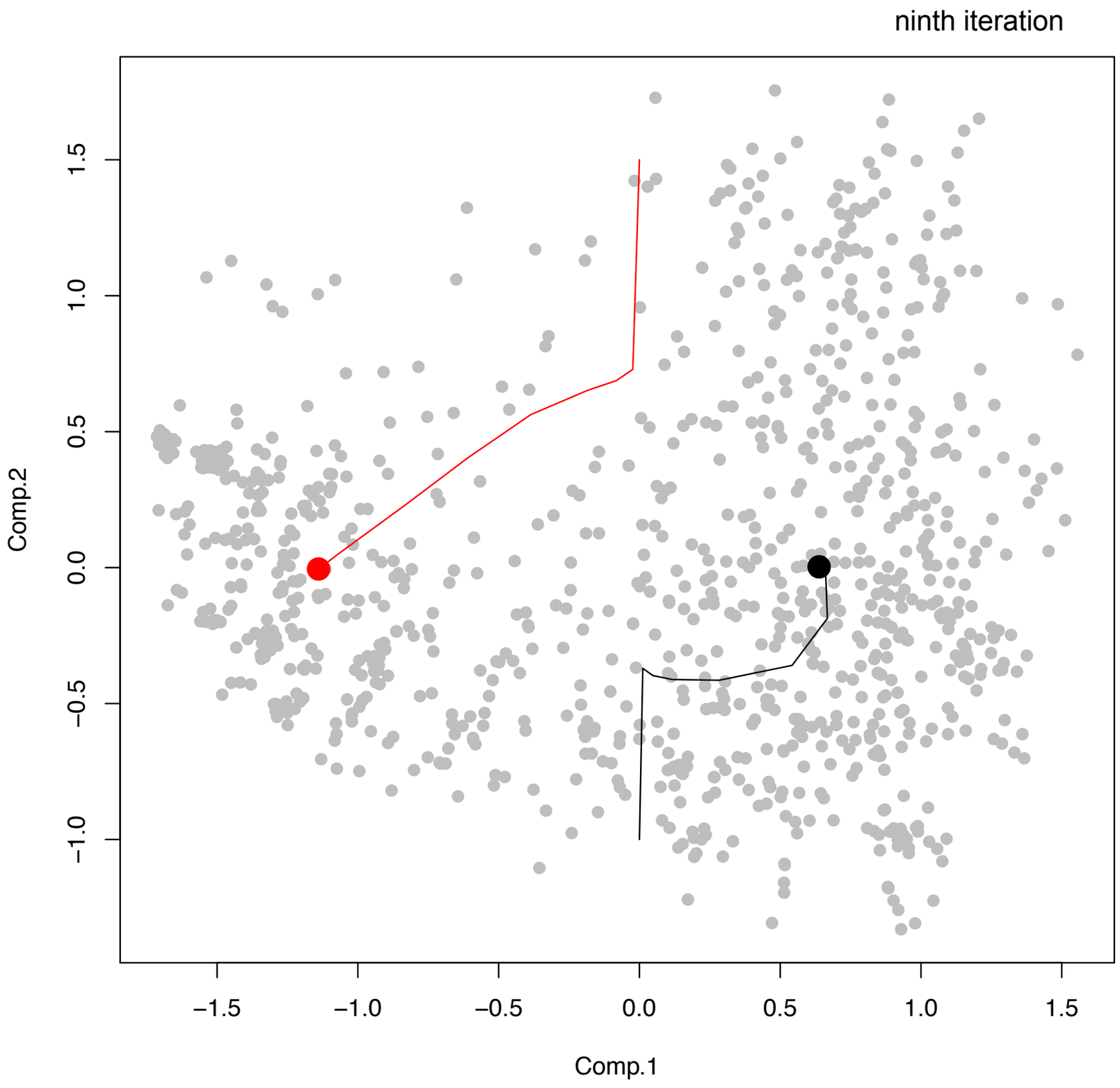

fifth iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

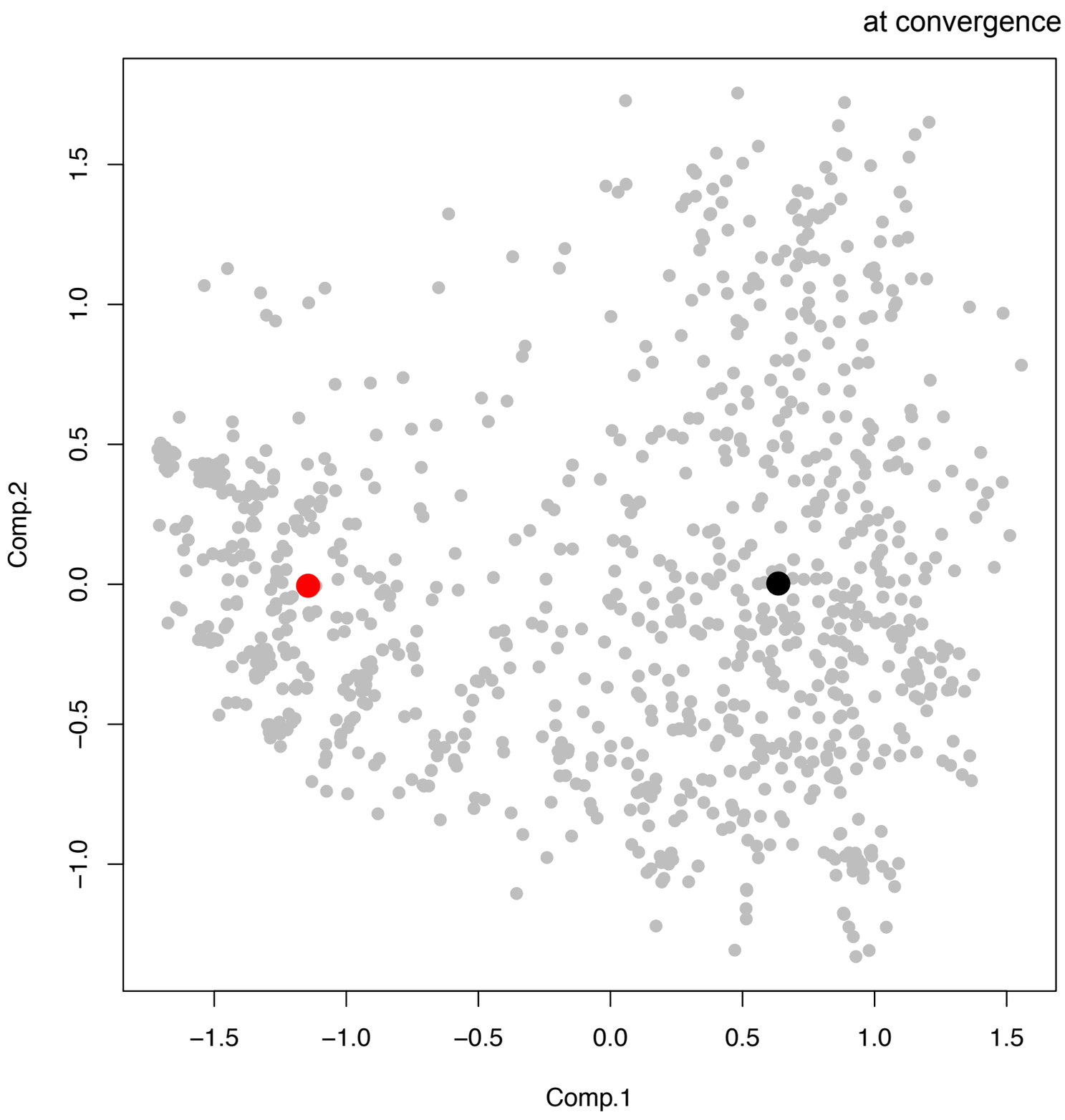

fifth iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

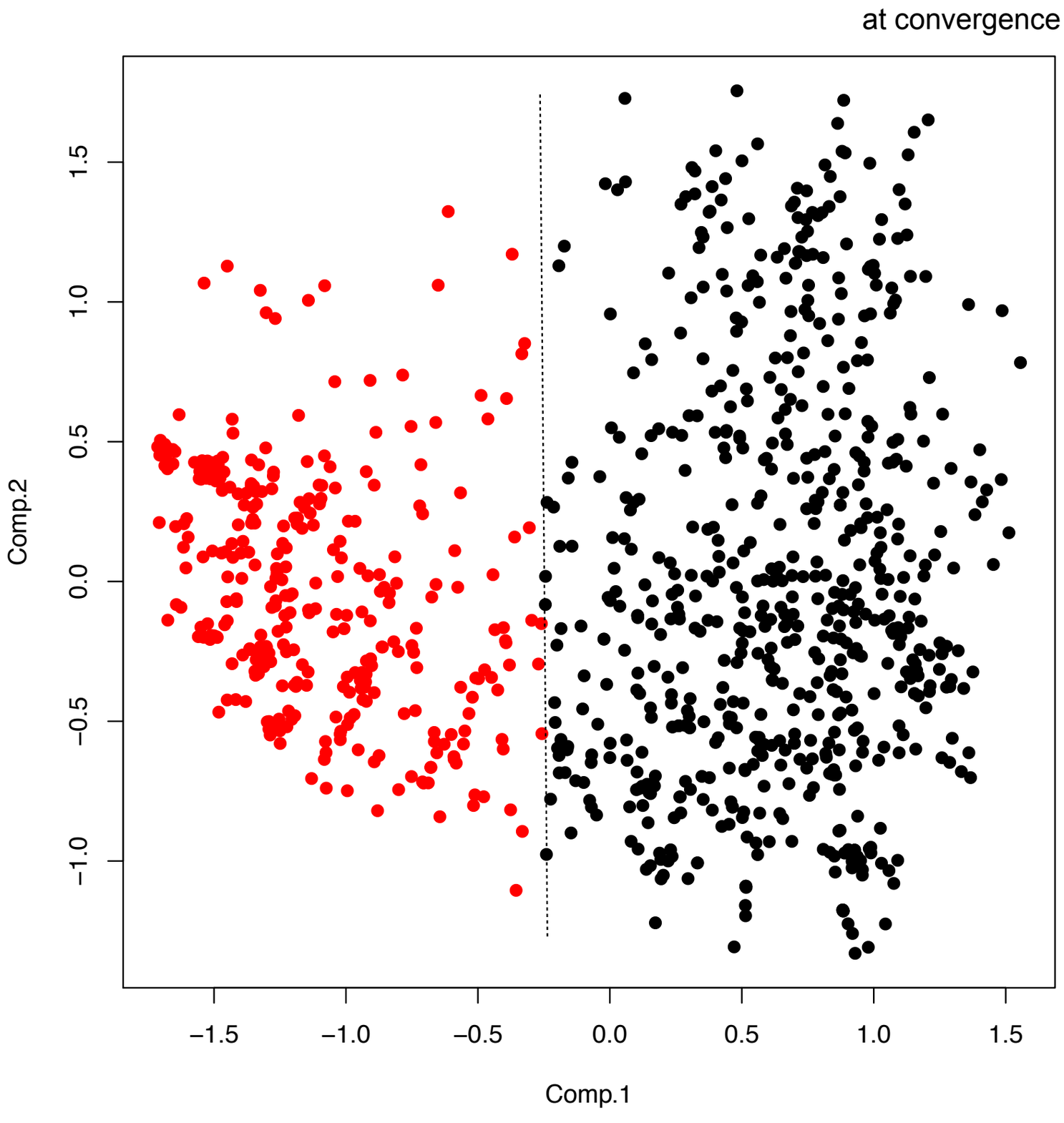

sixth iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


seventh iteration
1.5
1.0
0.5
0.0
−0.5
−1.0
Comp.2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


eigth iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


ninth iteration
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

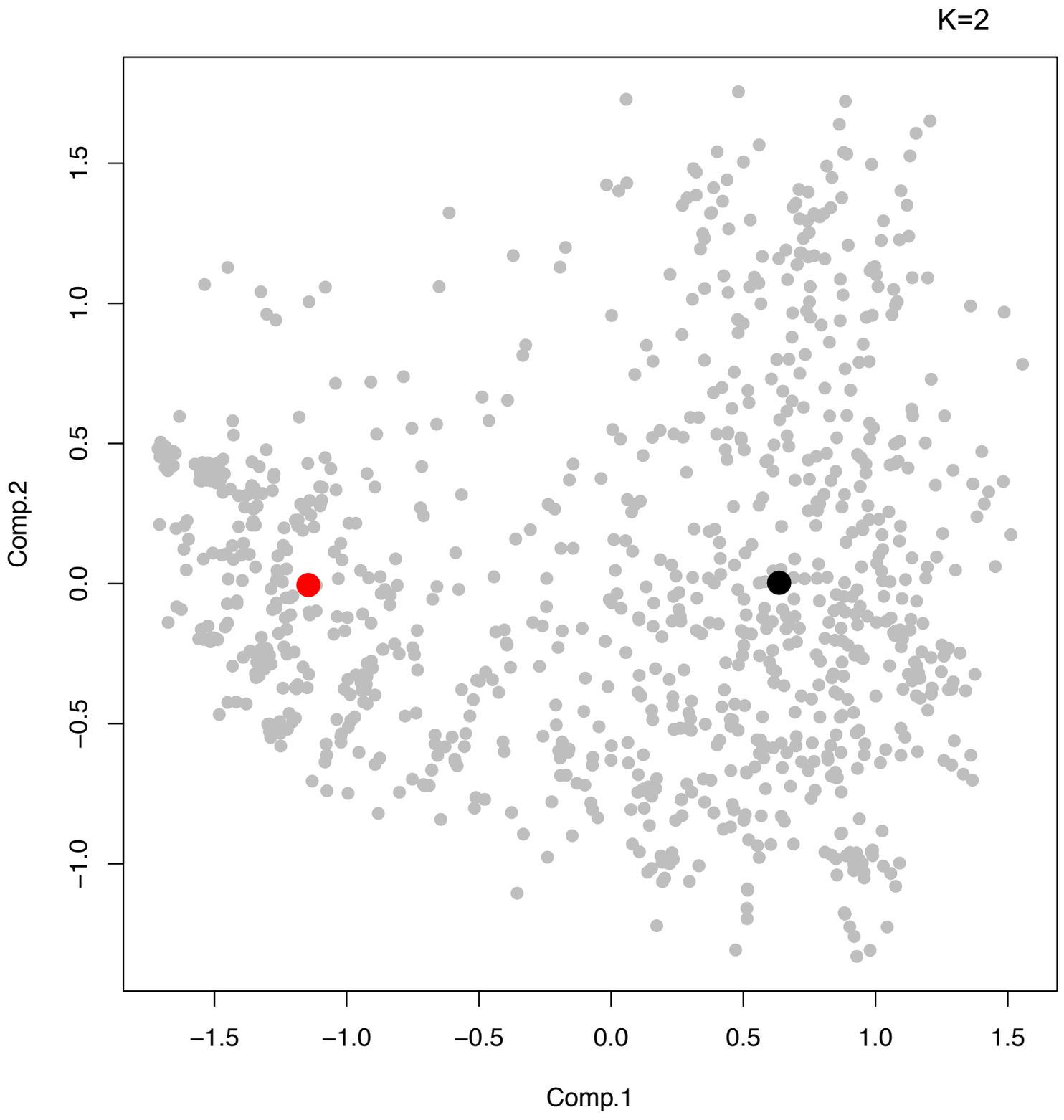

at convergence
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

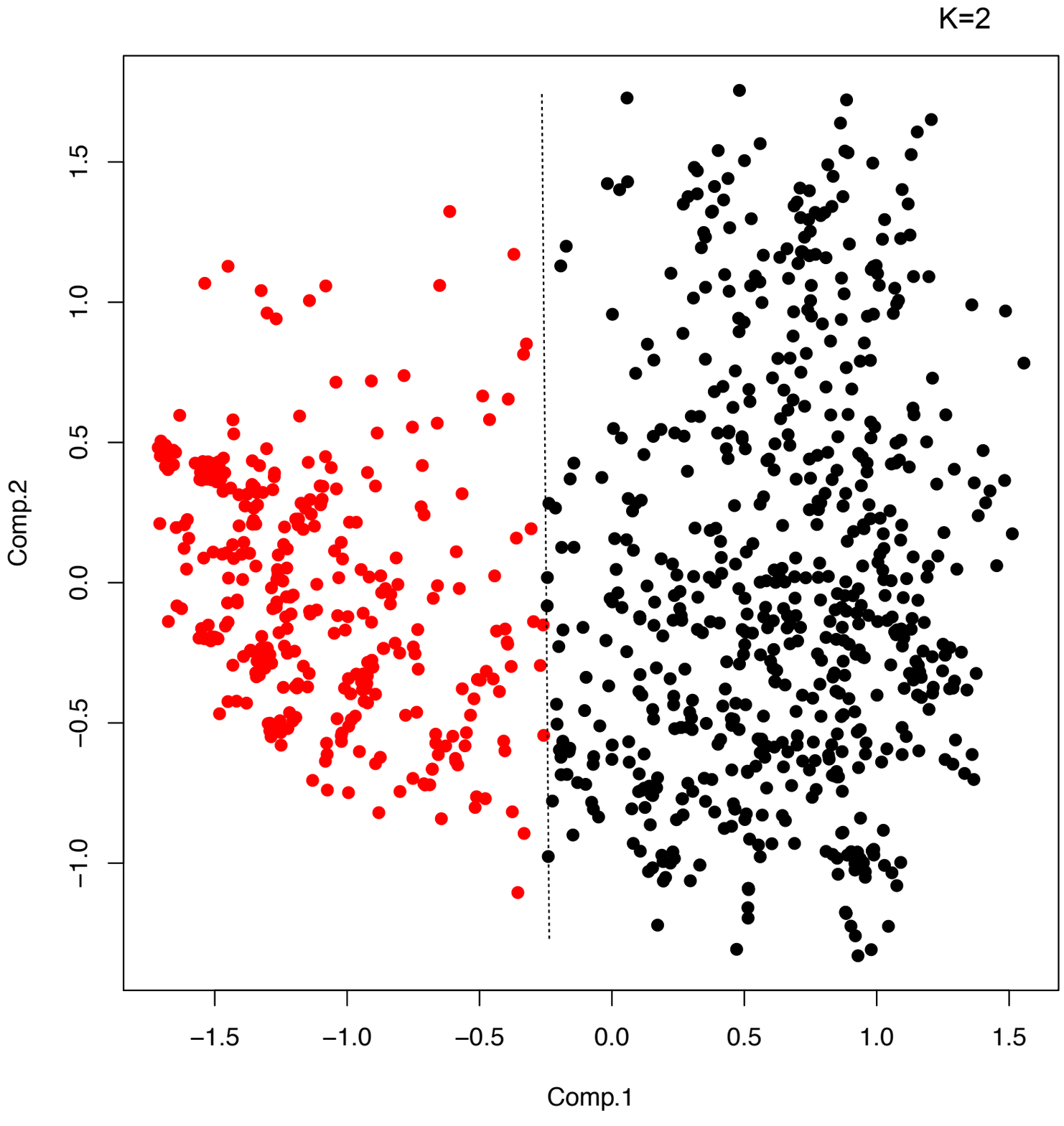

at convergence
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

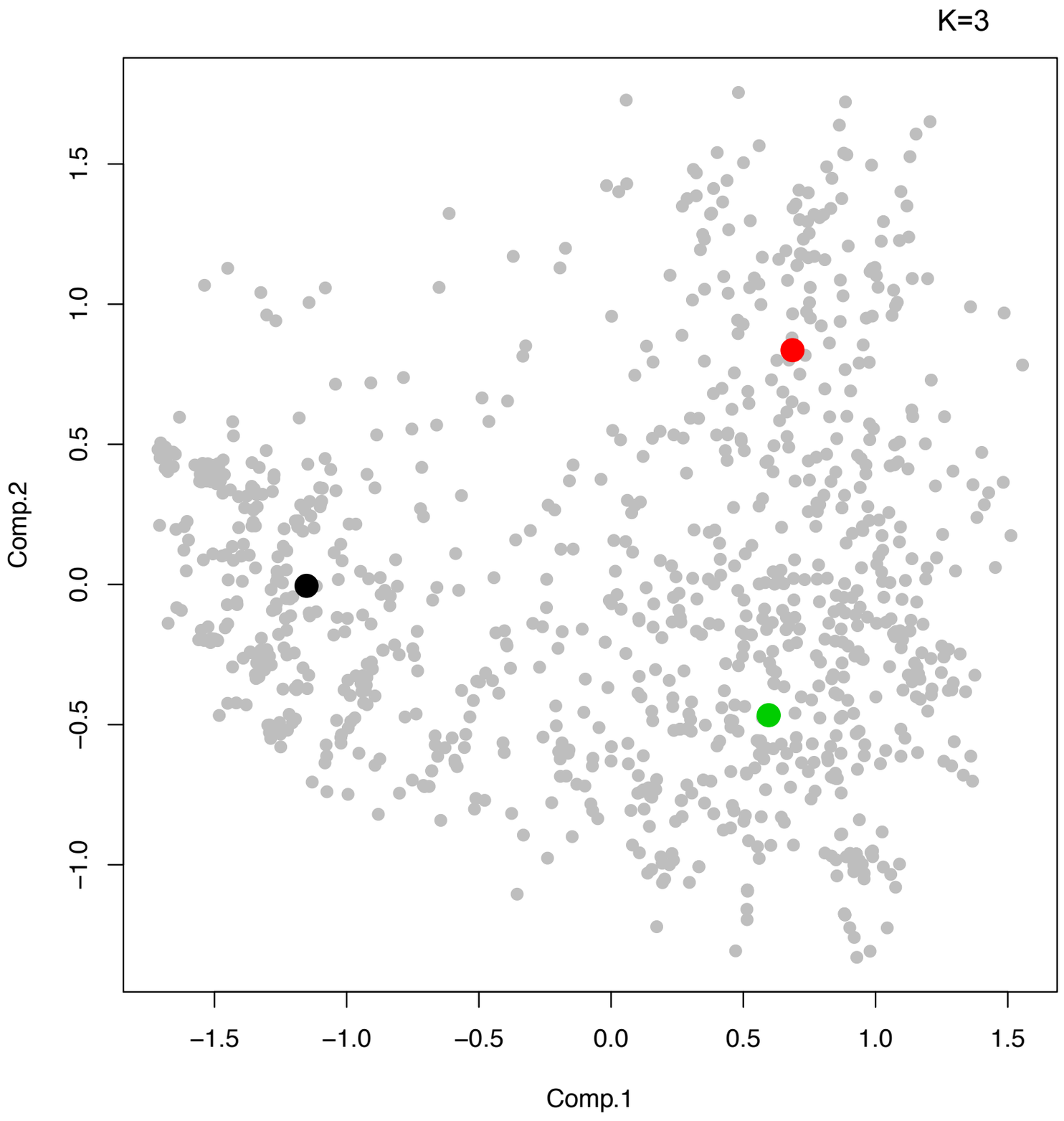
K-means

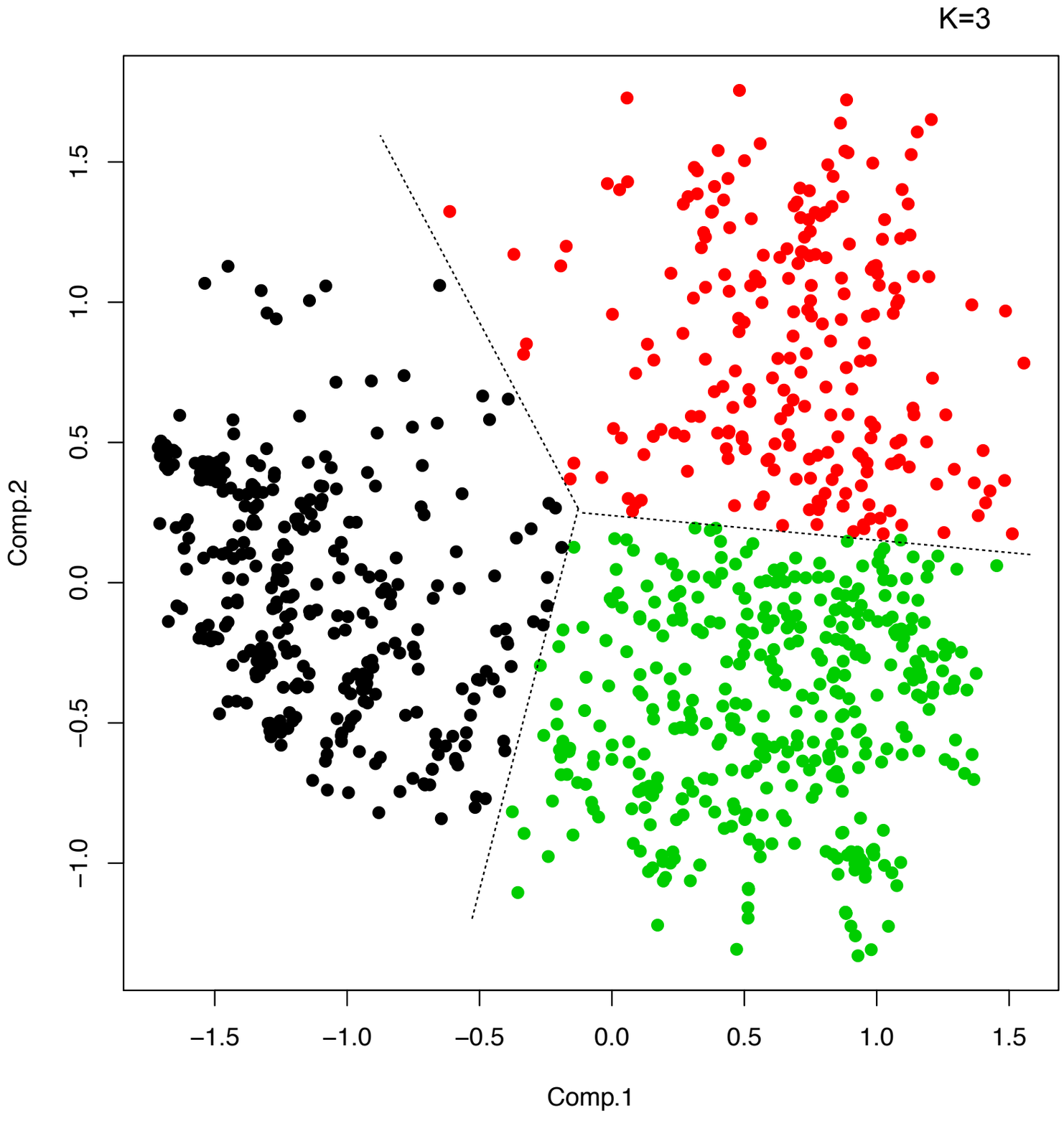
Through this simple iterative scheme, we construct a so-called **“hard” clustering**, in that each data point is associated with a single cluster (here K=2) -- There are **“soft” clustering schemes** that assign weights or probabilities that each point belongs to the different clusters

Here is how the algorithm behaves as we increase from K=2 to K=3, 4, 5...


K=2
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


K=2
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

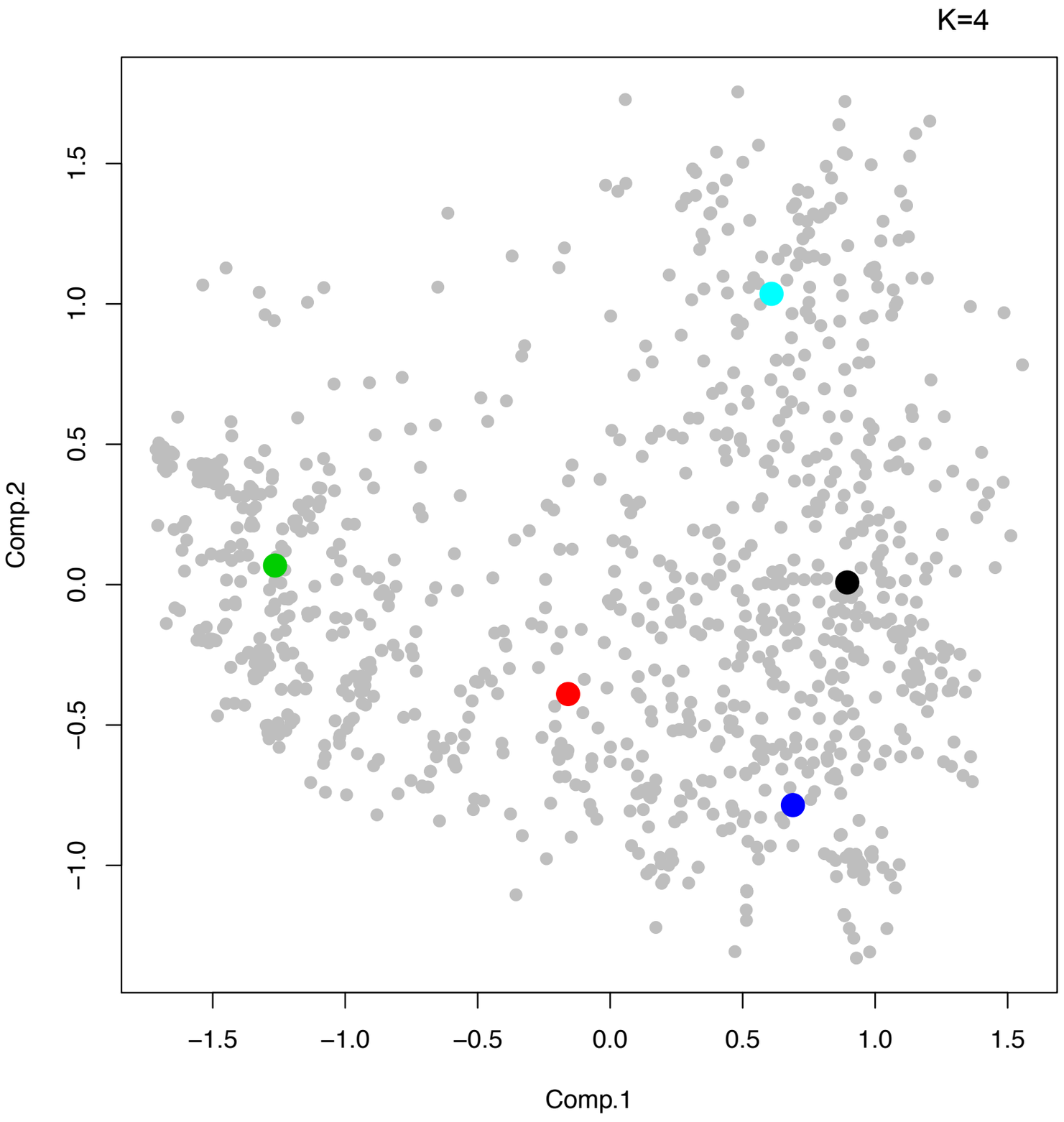

K=3
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

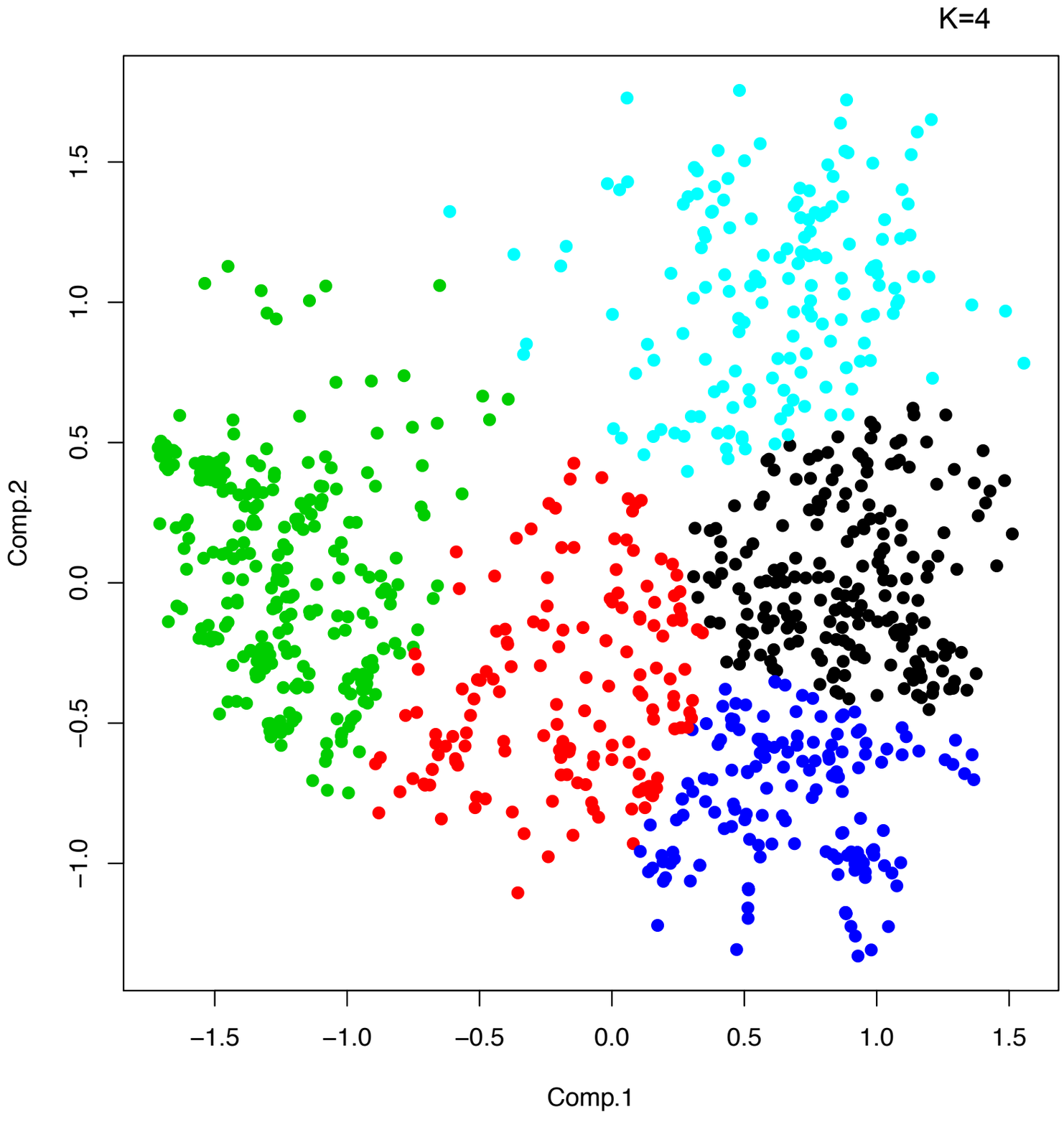

K=3
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1

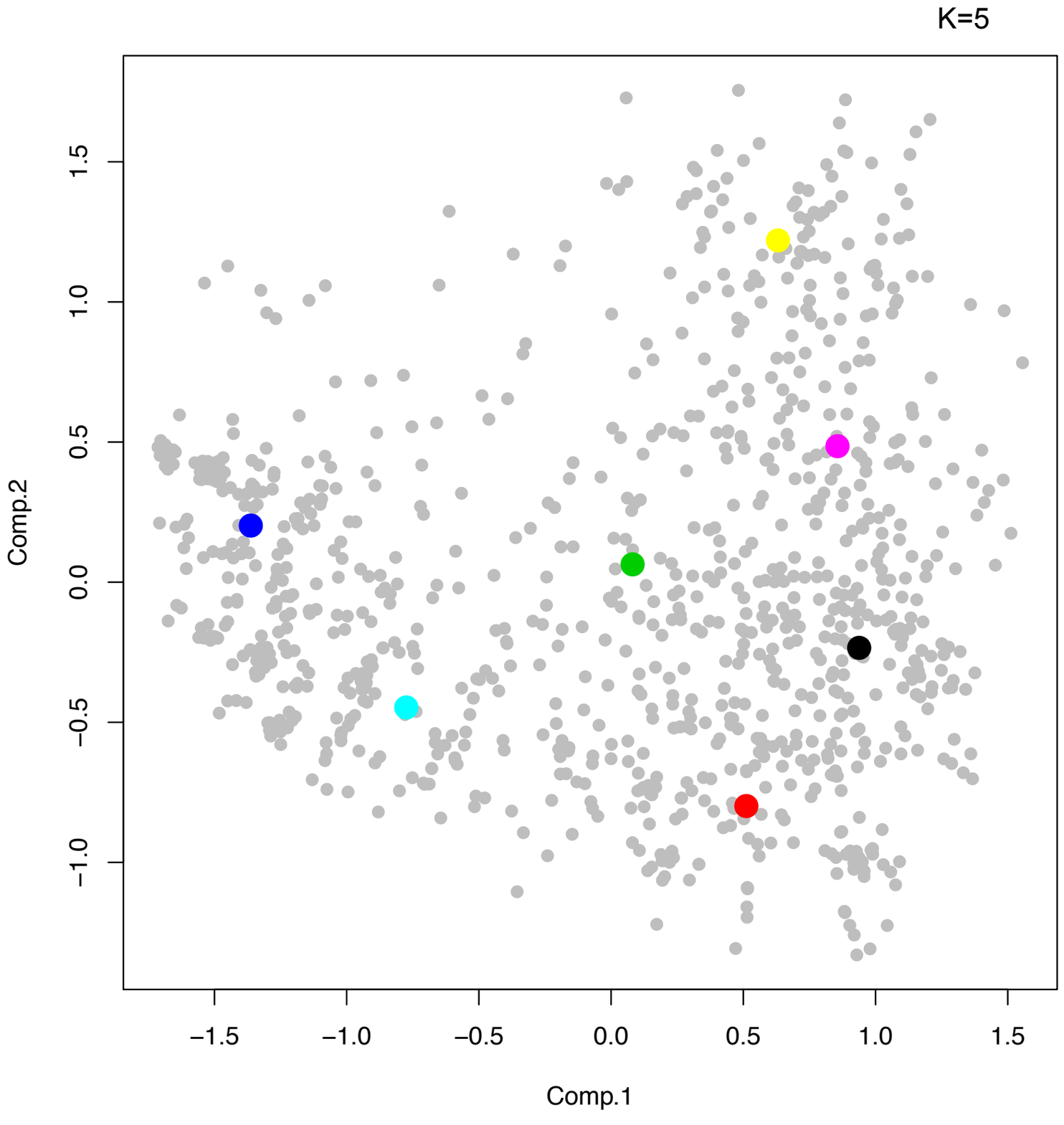
Clustering

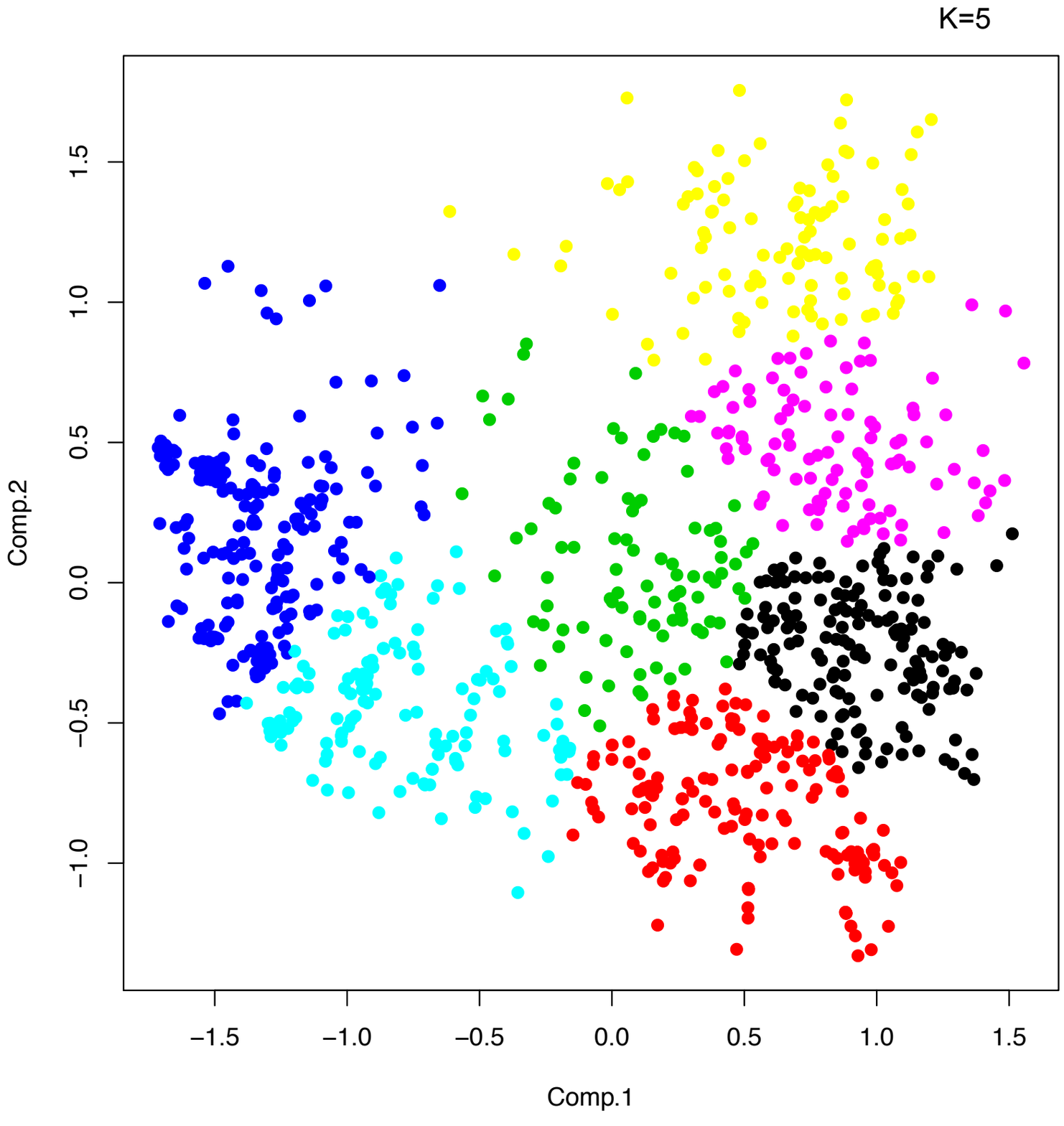
On the previous two slides, we asked for three clusters and displayed the three cluster centers (the three means) and indicated cluster membership

Note that the algorithm assigns points according to the nearest group mean and so in the end we have divisions based on the **Voronoi tessellation of these center points**


K=4
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


K=4
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


K=5
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1


K=5
Comp.2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Comp.1