Activity #15: Non-metric Multidimensional Scaling & Principal Components Analysis

1) On the (one-dimensional) highway shown below, place 3 cities so that the distances between those cities are:

| | City A | City B | City C |
|---|---|---|---|
| City A | 0 | | |
| City B | 8 | 0 | |
| City C | 14 | 6 | 0 |


2) Use the following distance matrix to place cities D, E, and F along a one-dimensional highway. Create the best map you can (matching the distances as closely as possible), but stay on the highway!

| | D | E | F |
|---|---|---|---|
| D | 0 | | |
| E | 8 | 0 | |
| F | **10** | 6 | 0 |


3) The map you just sketched isn't perfect. Why? What would you have to do to place those cities perfectly on a map?

4) Our map for cities D, E, and F wasn't perfect because we were trying to represent 2-dimensional distances on a 1-dimensional map. Reducing the dimensionality introduced some error (stress) into our map. How could we quantify this stress? In other words, what could we do to calculate how "far-off" our maps are compared to the actual distances?

| Pair of cities | Actual Distance | Distance on your map |
|---|---|---|
| D -> E | 8 | _____ |
| D -> F | 10 | _____ |
| E -> F | 6 | _____ |

5) Let's define stress as:

$$\text{stress} = \sqrt{\frac{\sum_i \sum_j \left(d_{ij} - \hat{d}_{ij}\right)^2}{\sum_i \sum_j d_{ij}^2}}$$ , where $d_{ij}$ represents the actual distance and $\hat{d}_{ij}$ represents the distances on our map

Calculate stress for the one-dimensional map you sketched for cities D, E, and F.

Stress for 1-dimensional map = _____

What would the stress be for the best 2-dimensional model? Stress = _____

6) This time, let's create a 2-dimensional map. I chose 3 actual U.S. cities (labeled X, Y, Z) and found the distances between each pair. How can you create a map that matches all 3 pairwise distances?

Go ahead and sketch a map with these 3 cities. Calculate the stress of your map. Stress = _____.

| | X | Y | Z |
|---|---|---|---|
| X | 0 | | |
| Y | 282 | 0 | |
| Z | 531 | 442 | 0 |

Map

Why are you still not certain your map is correct?

7) If we tried to place those 3 cities on a 1-dimensional map, what would happen to the value of stress?

8) Given all the distances between pairs of objects, we can create a map of those objects. If the map contains a sufficient number of dimensions, we can create a perfect* map of those distances. If we try to reduce the dimensionality, we'll introduce stress.

   * perfect, except for the fact that our map does not show the absolute location of any object and may, in fact, be a rotation or reflection of the actual map.

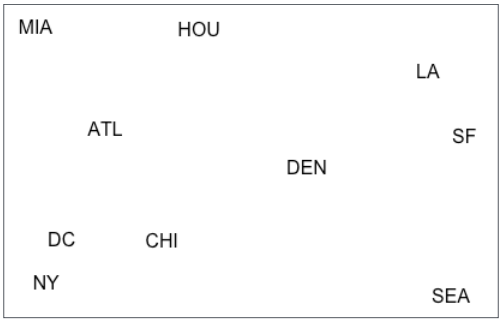
The goal of multidimensional scaling is to represent distances among objects in a simple (lower-dimensional) way.

To use MDS, we need a set of objects and distances. It's easy to imagine this if we're dealing with actual distances.

Example: The matrix to the right displays the distances between pairs of 10 U.S. cities. Can we sketch a map of all 10 cities using only the given distances (and not our prior knowledge of geography)?

| | ATL | CHI | DEN | HOU | LA | MIA | NY | SF | SEA | DC |
|---|---|---|---|---|---|---|---|---|---|---|
| **ATL** | 0 | 587 | 1212 | 701 | 1936 | 604 | 748 | 2139 | 2182 | 543 |
| **CHI** | 587 | 0 | 920 | 940 | 1745 | 1188 | 713 | 1858 | 1737 | 597 |
| **DEN** | 1212 | 920 | 0 | 879 | 831 | 1726 | 1631 | 949 | 1021 | 1494 |
| **HOU** | 701 | 940 | 879 | 0 | 1374 | 968 | 1420 | 1645 | 1891 | 1220 |
| **LA** | 1936 | 1745 | 831 | 1374 | 0 | 2339 | 2451 | 347 | 959 | 2300 |
| **MIA** | 604 | 1188 | 1726 | 968 | 2339 | 0 | 1092 | 2594 | 2734 | 923 |
| **NY** | 748 | 713 | 1631 | 1420 | 2451 | 1092 | 0 | 2571 | 2408 | 205 |
| **SF** | 2139 | 1858 | 949 | 1645 | 347 | 2594 | 2571 | 0 | 678 | 2442 |
| **SEA** | 2182 | 1737 | 1021 | 1891 | 959 | 2734 | 2408 | 678 | 0 | 2329 |
| **DC** | 543 | 597 | 1494 | 1220 | 2300 | 923 | 205 | 2442 | 2329 | 0 |

The hard way: We could arbitrarily place the first city (ATL) in the middle of the map. Then, we could draw a circle with a radius of 587 centered around ATL. Somewhere on that circle, we could place CHI. The location of DEN would have to satisfy two constraints: (1) it would have to be on the edge of a circle with radius 1212 centered at ATL, and (2) it would have to be on the edge of a circle centered at CHI with a radius of 920. There are only two points that satisfy these constraints, so we'd have to choose one of them for CHI. From this point on, the relative locations of all the other cities would be determined exactly (because we know we have a 2-dimensional map).

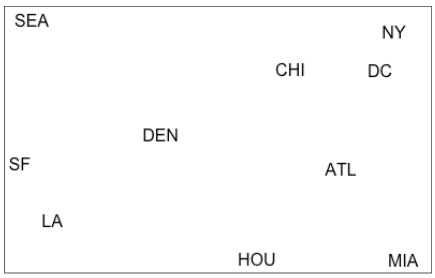



We would find, though, that it's impossible to fit all 10 cities perfectly on a 2-dimensional map. This is either due to elevation of the cities, the curvature of the earth, or the rounding of distances.

This method, while cumbersome, works if we know *a priori* that our map consists of 2 dimensions. Using this method, I obtained the map displayed above (with stress = 0.0099). What's wrong?

To "fix" the map, I rotated it by 180-degrees by multiplying the coordinates of each city by:

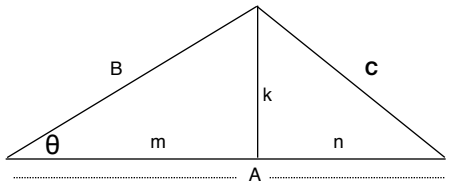
$$\begin{matrix} \text{ATL} \\ \text{CHI} \\ \cdots \\ \text{DC} \end{matrix}\begin{bmatrix} -718.76 & +142.99 \\ -382.06 & -340.84 \\ \vdots & \vdots \\ -979.62 & -335.47 \end{bmatrix}\begin{bmatrix} \cos(\pi) & -\sin(\pi) \\ \sin(\pi) & \cos(\pi) \end{bmatrix} = \begin{bmatrix} +718.76 & -142.99 \\ +382.06 & +340.84 \\ \vdots & \vdots \\ +979.62 & +335.47 \end{bmatrix}$$

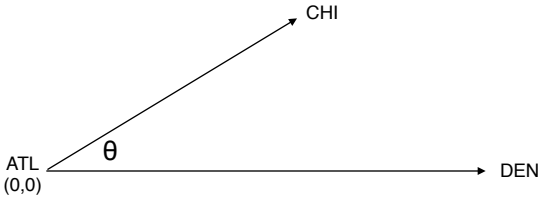


The analytical approach:

(1) Arbitrarily locate the first object (Atlanta) at the origin

(2) Use some fancy math to place the next two objects (Chicago and Denver).




$B^2 = m^2 + k^2$ and $C^2 = n^2 + k^2$

$n = A - m$ so we can substitute

$C^2 = (A-m)^2 + k^2 = A^2 - 2Am + m^2 + k^2$

$= A^2 - 2Am + (m^2 + k^2) = A^2 - 2Am + B^2$

Notice $\cos\theta = \frac{m}{B}$, so $B\cos\theta = m$.

$A^2 - 2A(m) + B^2 = A^2 + B^2 - 2AB\cos\theta$

Law of cosines: $C^2 = A^2 + B^2 - 2AB\cos\theta$

$d_{CD}^2 = d_{AD}^2 + d_{AC}^2 - 2d_{AD}d_{AC}\cos\theta_{CAD}$

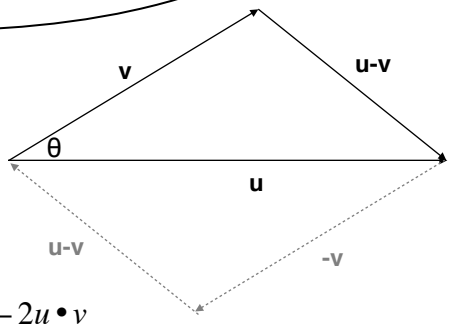
Rearranging terms...

$-\frac{1}{2}\left(d_{CD}^2 - d_{AD}^2 - d_{AC}^2\right) = d_{AD}d_{AC}\cos\theta_{CAD}$

We know all the distances on the left side

We just don't know this scalar (the coordinates)

Using the "vectorized" diagram to the right, we can restate the Law of Cosines in terms of the lengths of vectors:




$$||u - v||^2 = ||u||^2 + ||v||^2 - 2||u||\,||v||\cos\theta$$

We can rewrite our vectorized Law of Cosines as:

$$||u - v||^2 = (u - v)\bullet(u - v) = (u\bullet u) - (u\bullet v) - (v\bullet u) + (v\bullet v) = ||u||^2 + ||v||^2 - 2u\bullet v$$

The parts underlined in red must be equal, so: $u\bullet v = ||u||\,||v||\cos\theta$

Let's change from this vector notation back to our distance notation: $u\bullet v = ||u||\,||v||\cos\theta = d_{AD}d_{AC}\cos\theta_{CAD}$

Because all the blue underlined parts must be equal, we can write:

$$-\frac{1}{2}\left(d_{CD}^2 - d_{AD}^2 - d_{AC}^2\right) = d_{AD}d_{AC}\cos\theta_{CAD} = ||u||\,||v||\cos\theta = u\bullet v$$

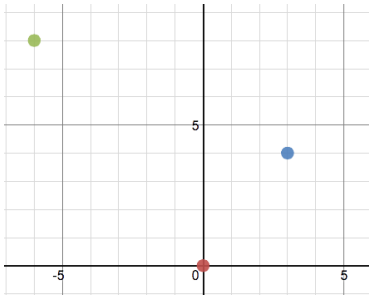
We know all the distances on the left side of this equation; we want to solve for the terms on the right side. To do this, we express the problem in matrix form:

$$_{n-1}B_{n-1} = \begin{array}{c} \\ \text{CHI} \\ \text{DEN} \\ \text{HOU} \\ \dots \end{array} \begin{array}{c} \begin{array}{cccc} \text{CHI} & \text{DEN} & \text{HOU} & \dots \end{array} \\ \begin{bmatrix} 0 & -\frac{1}{2}\left(d_{CD}^2 - d_{AD}^2 - d_{AC}^2\right) & -\frac{1}{2}\left(d_{CH}^2 - d_{AH}^2 - d_{AC}^2\right) & \cdots \\ -\frac{1}{2}\left(d_{CD}^2 - d_{AD}^2 - d_{AC}^2\right) & 0 & -\frac{1}{2}\left(d_{DH}^2 - d_{AH}^2 - d_{AD}^2\right) & \cdots \\ -\frac{1}{2}\left(d_{CH}^2 - d_{AH}^2 - d_{AC}^2\right) & -\frac{1}{2}\left(d_{DH}^2 - d_{AH}^2 - d_{AD}^2\right) & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{array}$$

The matrix form of the above equation becomes: $B = UU'$

The singular value decomposition of B is: $B = V\Lambda V'$ where V is a matrix of mutual orthogonal eigenvectors and Λ is a diagonal matrix of eigenvalues. We can get the following solution: $U = V\Lambda^{1/2}$

As a simple example, suppose we have 3 cities: A, B, and C. We'll place City A at the origin and know the map coordinates for all the other cities: Coordinates for A: (0, 0)
Coordinates for B: (3, 4)
Coordinates for C: (-6, 8)




Let's define *nxq* matrix **U** to represent the coordinates: $U = \begin{bmatrix} 0 & 0 \\ 3 & 4 \\ -6 & 8 \end{bmatrix}$

Remember, in a multidimensional scaling situation, we won't know those coordinates. We'll only know distances between pairs of those objects.

We'll define an *nxn* matrix **D** to represent the known pairwise distances: $D = \begin{bmatrix} 0 & 5 & 10 \\ 5 & 0 & \sqrt{97} \\ 10 & \sqrt{97} & 0 \end{bmatrix}$

We can then define an *nxn* matrix **B** as we derived on the bottom of the previous page:

$$B = UU' = \begin{bmatrix} 0 & 0 \\ 3 & 4 \\ -6 & 8 \end{bmatrix} \begin{bmatrix} 0 & 3 & -6 \\ 0 & 4 & 8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 25 & 14 \\ 0 & 14 & 100 \end{bmatrix}$$

(Note: This calculation assumes we know the coordinates)

We're trying to go from **D** to **B** to **U**. If we can get **B** from **D**, all we'll have to do is factor **D** to get **U**.
On the previous page, we demonstrated how to get the elements of **B**. For example:

For cities B and C: $-\frac{1}{2}\left(d_{BC}^2 - d_{AC}^2 - d_{AB}^2\right) = -\frac{1}{2}\left(\left(\sqrt{97}\right)^2 - 10^2 - 5^2\right) = 14$

For cities B and B: $-\frac{1}{2}\left(d_{BB}^2 - d_{AB}^2 - d_{AB}^2\right) = -\frac{1}{2}\left(0^2 - 5^2 - 5^2\right) = 25$

Now that we can get the elements of **B**, we can factor that matrix to get the coordinates **U**. Using R, I found the eigenvalues and eigenvectors of **B** to be:

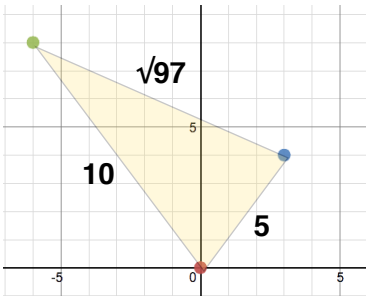
$$V = \begin{bmatrix} 0 & 0 & 1 \\ .1777 & .9841 & 0 \\ .9841 & -.1777 & 0 \end{bmatrix} \qquad \Lambda = \begin{bmatrix} 102.528 & 0 & 0 \\ 0 & 22.472 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The two positive eigenvalues indicate we have a 2-dimensional map. Checking to see that the factorization worked:

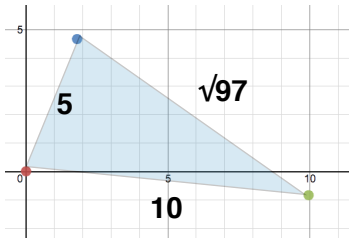
$$B = V\Lambda V' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 25 & 14 \\ 0 & 14 & 100 \end{bmatrix}$$

Finally, I can find the coordinates using $U = V\Lambda^{1/2} = \begin{bmatrix} 0 & 0 & 1 \\ .1777 & .9841 & 0 \\ .9841 & -.1777 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{102.528} & 0 \\ 0 & \sqrt{22.472} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1.799 & 4.665 \\ 9.964 & -.842 \end{bmatrix}$

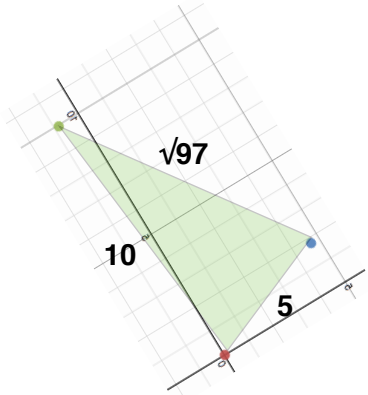
Wait… those aren't the original coordinates! Let's plot them and see what happened…




Original Coordinates




Coordinates from "fancy math"




Rotated and reflected

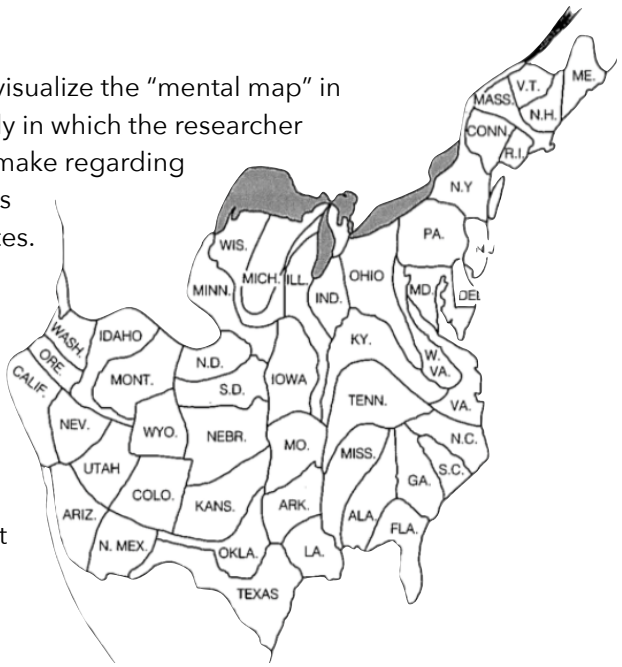
It looks like the fancy math worked, as long as we understand our final map may be a rotation (and/or reflection) of what it "should" be.

9) Ok, so it looks like this MDS method works to place cities on a map. I guess that's cool, but is there anything else MDS can do? I'm glad you asked.

One application of MDS is perceptual mapping – attempting to visualize the "mental map" in the mind of an individual. To the right is a map from a 1962 study in which the researcher was interested in understanding the relative judgments people make regarding the relative distance of U.S. states. The researcher asked subjects to judge the distance between all pairs of the contiguous 48 states.

The map to the right was created using MDS on the data from residents of Boston. From this, we might conclude individuals make clearer perceptual distinctions regarding relative distances that are closer to home.

MDS has also been used to create perceptual maps of physical stimuli, politicians, consumer products, and crimes. If we can get distances between objects, we can use MDS to create a map. We'll see a few examples in just a bit.




Shepard, R.N. (1962). The analysis of proximities: multidimensional scaling with an unknown distance function. Psychometric (27), 125-140.

9) How do we get distances between objects? For geographical examples, it's relatively easy, but how do we get distances between crimes, politicians, or consumer products? What would those "distances" even represent?

**Properties of distances**. Given three objects *a, b,* and *c:* $d_{ab} > 0$

$$d_{ab} = d_{ba}$$

$$d_{ac} \leq d_{ab} + d_{cb}$$

**Methods to get distances:**

**Direct methods:** Data = a matrix of distances between pairs of objects

- Distances = physical distances from a map; number of interactions between people (social distance)
- Judgments or interval estimates = Ask subjects to rate the dissimilarity (on a scale from 0 to 10) between all pairs of objects
- Clusters = Ask subjects to sort objects into piles; or examine naturally occurring groups, such as paragraphs, communities, and associations. Record 0 if two objects occur in the same group and 1 if they do not. Sum these counts over replications or judges.
- Triads = Ask subjects to compare 3 objects at a time and report which two are most similar (or which one is odd). Do this over all possible triads of objects. To compute dissimilarities, sum over all triads. There are more triads than pairs of objects, so this method is more tedious.
- Tetrads = Ask subjects to compare 2 pairs of objects and report which pair is most similar. Do this over all possible tetrads of objects. To compute dissimilarities, sum over all tetrads.

**Indirect methods:** Data = a matrix in which rows represent objects and columns represent attributes (variables)

- Computed distances = Attributes for each row represent a vector; calculate distance between vectors for pairs of objects.
- Correlations (inverted somehow) = Normally, we're interested in calculating correlations between attributes (variables or columns). In this method, calculate correlations between pairs of objects (pairs of rows). Since correlations represent similarities, we need to convert them to distances. To do this, we could (a) multiply the correlations by -1, (b) take one minus each correlation, or (c) take the square root of one minus the correlation-squared
- Counts of discrepancies = Counting discrepancies between columns (if columns are binary measures).

**Metric distances?** While the MDS method to create maps from physical distances is kinda neat, most of the interesting scaling problems do not involve actual (metric) distances. In these cases, we might assume our distances are *ordinal* (that the rank order of the distances between objects is meaningful, but the actual distances may not be). To analyze this non-metric data, we can use *non-metric multidimensional scaling (NMDS)*.

10) Let's see an example of NMDS in action. Below, I've pasted data from a 1978 study on the perceived similarities among 12 nations. The average <u>similarity</u> ratings are shown, so we'll need to first convert them to dissimilarities (distances).

It looks like a 7-point scale was used, so I'll use: Distance = 7 – similarity rating.

| | Brazil | Congo | Cuba | Egypt | France | India | Israel | Japan | China | Russia | USA | Yugoslavia |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Brazil | 7.00 | 4.83 | 5.28 | 3.44 | 4.72 | 4.50 | 3.83 | 3.50 | 2.39 | 3.06 | 5.39 | 3.17 |
| Congo | 4.83 | 7.00 | 4.56 | 5.00 | 4.00 | 4.83 | 3.33 | 3.39 | 4.00 | 3.39 | 2.39 | 3.50 |
| Cuba | 5.28 | 4.56 | 7.00 | 5.17 | 4.11 | 4.00 | 3.61 | 2.94 | 5.50 | 5.44 | 3.17 | 5.11 |
| Egypt | 3.44 | 5.00 | 5.17 | 7.00 | 4.78 | 5.83 | 4.67 | 3.83 | 4.39 | 4.39 | 3.33 | 4.28 |
| France | 4.72 | 4.00 | 4.11 | 4.78 | 7.00 | 3.44 | 4.00 | 4.22 | 3.67 | 5.06 | 5.94 | 4.72 |
| India | 4.50 | 4.83 | 4.00 | 5.83 | 3.44 | 7.00 | 4.11 | 4.50 | 4.11 | 4.50 | 4.28 | 4.00 |
| Israel | 3.83 | 3.33 | 3.61 | 4.67 | 4.00 | 4.11 | 7.00 | 4.83 | 3.00 | 4.17 | 5.94 | 4.44 |
| Japan | 3.50 | 3.39 | 2.94 | 3.83 | 4.22 | 4.50 | 4.83 | 7.00 | 4.17 | 4.61 | 6.06 | 4.28 |
| China | 2.39 | 4.00 | 5.50 | 4.39 | 3.67 | 4.11 | 3.00 | 4.17 | 7.00 | 5.72 | 2.56 | 5.06 |
| Russia | 3.06 | 3.39 | 5.44 | 4.39 | 5.06 | 4.50 | 4.17 | 4.61 | 5.72 | 7.00 | 5.00 | 6.67 |
| USA | 5.39 | 2.39 | 3.17 | 3.33 | 5.94 | 4.28 | 5.94 | 6.06 | 2.56 | 5.00 | 7.00 | 3.56 |
| Yugoslavia | 3.17 | 3.50 | 5.11 | 4.28 | 4.72 | 4.00 | 4.44 | 4.28 | 5.06 | 6.67 | 3.56 | 7.00 |

| | Brazil | Congo | Cuba | Egypt | France | India | Israel | Japan | China | Russia | USA | Yugoslavia |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Brazil | 0.00 | 2.17 | 1.72 | 3.56 | 2.28 | 2.50 | 3.17 | 3.50 | 4.61 | 3.94 | 1.61 | 3.83 |
| Congo | 2.17 | 0.00 | 2.44 | 2.00 | 3.00 | 2.17 | 3.67 | 3.61 | 3.00 | 3.61 | 4.61 | 3.50 |
| Cuba | 1.72 | 2.44 | 0.00 | 1.83 | 2.89 | 3.00 | 3.39 | 4.06 | 1.50 | 1.56 | 3.83 | 1.89 |
| Egypt | 3.56 | 2.00 | 1.83 | 0.00 | 2.22 | 1.17 | 2.33 | 3.17 | 2.61 | 2.61 | 3.67 | 2.72 |
| France | 2.28 | 3.00 | 2.89 | 2.22 | 0.00 | 3.56 | 3.00 | 2.78 | 3.33 | 1.94 | 1.06 | 2.28 |
| India | 2.50 | 2.17 | 3.00 | 1.17 | 3.56 | 0.00 | 2.89 | 2.50 | 2.89 | 2.50 | 2.72 | 3.00 |
| Israel | 3.17 | 3.67 | 3.39 | 2.33 | 3.00 | 2.89 | 0.00 | 2.17 | 4.00 | 2.83 | 1.06 | 2.56 |
| Japan | 3.50 | 3.61 | 4.06 | 3.17 | 2.78 | 2.50 | 2.17 | 0.00 | 2.83 | 2.39 | 0.94 | 2.72 |
| China | 4.61 | 3.00 | 1.50 | 2.61 | 3.33 | 2.89 | 4.00 | 2.83 | 0.00 | 1.28 | 4.44 | 1.94 |
| Russia | 3.94 | 3.61 | 1.56 | 2.61 | 1.94 | 2.50 | 2.83 | 2.39 | 1.28 | 0.00 | 2.00 | 0.33 |
| USA | 1.61 | 4.61 | 3.83 | 3.67 | 1.06 | 2.72 | 1.06 | 0.94 | 4.44 | 2.00 | 0.00 | 3.44 |
| Yugoslavia | 3.83 | 3.50 | 1.89 | 2.72 | 2.28 | 3.00 | 2.56 | 2.72 | 1.94 | 0.33 | 3.44 | 0.00 |

Think about what these distances represent. If we had metric distance data, we could say the distance between China and Israel (4.00) is twice as large as the distance between Russia and the U.S. With our ordinal scale data (non-metric distances), we can only say that in our map, China and Israel should be farther apart from each other than Russia and the U.S. (but we cannot say how much farther).

In NMDS, our goal is to create a map that is consistent with the rank orderings of the distances between pairs of objects. In other words, we want to achieve a monotone relationship between the distances in our data and the distances on our map. To do this, we use an iterative process:

1. Choose the dimensionality of the map. The choice of the "best" number of dimensions is subjective, but can be based on the stress of the map (or the objectives of the analysis). We'll use 2 dimensions in this example.

2. Choose an initial configuration for the map (i.e., place each object somewhere on the map). You could use metric MDS or arbitrarily choose the initial configuration. Either way, this iterative process will continue to move the objects around the map until the fit is no longer improved.

3. Calculate the distances between all pairs of points on the map. We'll compare these distances to the "actual" distances in our data.

4. Assess the correspondence between the distances on the map and the distances in the data. To do this, we can calculate stress use a Shepard's plot. Stress informs us of the error between our map and the actual distances. A Shepard's plot shows the ordinal relationship between the pairwise distances on the map and the actual pairwise distances.

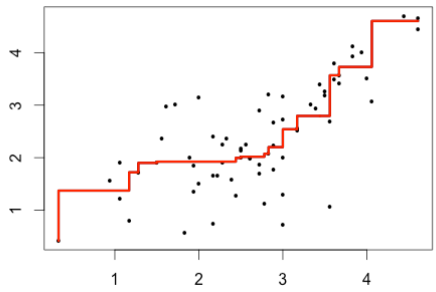
5. Using a numerical optimization method, move the points around the map to reduce its stress. One way to do this would be to use a *gradient search method*. We could take the derivative of stress with respect to the coordinate locations for each object on our map – for example, we could calculate the partial derivative $\frac{\partial \text{stress}}{\partial x_{11}}$ – to find the rate of change of stress with respect to a change in the position of object #1 on the 1st dimension of the map. Since we're trying to minimize stress, we change the coordinate location of each object in the *negative* direction of the vector of partial derivatives.

Let's see this iterative process in action…

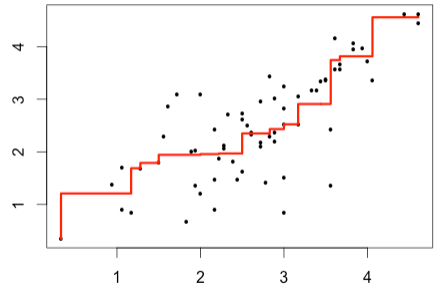
1. I choose a 2-dimensional map

2. I use metric MDS to create an initial configuration of nations on my 2D map

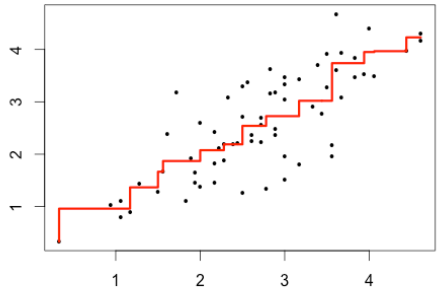
3-4: From this initial configuration, STRESS = 0.0942. Shepard's plot




5. After one iteration of this optimization method, the map was updated. One highlighted difference is that Egypt moved down on this new map. STRESS = 0.0738.




5. I continued this iterative process until stress was no longer shrinking (at a tolerance of 0.00001). The process took 30 iterations. STRESS = 0.05530



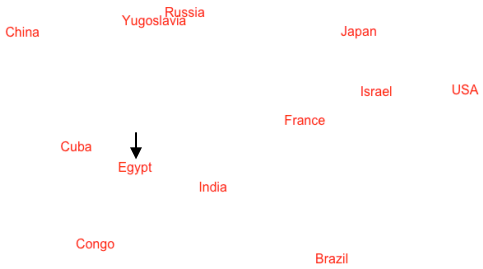

To the right, I've superimposed the final configuration on top of the initial configuration. You can see how the nations moved through the iterative process. Japan, for instance, moved to the right as the distances were optimized.

At this point, we could try to fit orthogonal axes on the graph and interpret the dimensions.

Can you interpret either of the dimensions on this map?

**Initial configuration**


**One iteration**



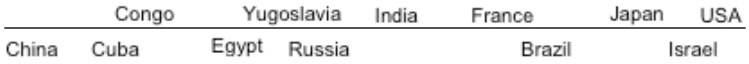

**30 iterations**


**Change from initial to final configuration**

11) What would happen if we tried to create a map with a higher number of dimensions? To see, I ran an NMDS on this example data for 1-6 dimensions and recorded the stress of each map:

| Dimensions: | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Stress: | 0.21084 | 0.05530 | 0.02610 | 0.02128 | 0.02042 | 0.02042 |

From this, how many dimensions would you choose? Explain. The one dimension map is displayed below.




12) Let's look at some other examples of NMDS analyses. Interpret the results and try to explain what the dimensions might represent.

**Example: Fruits**

Data: Subjects were presented with samples of 16 fruits and were asked to rate how well they liked each fruit on a scale from 0-100. The subjects were <u>not</u> told what factors should influence their judgments.

Sample of data:

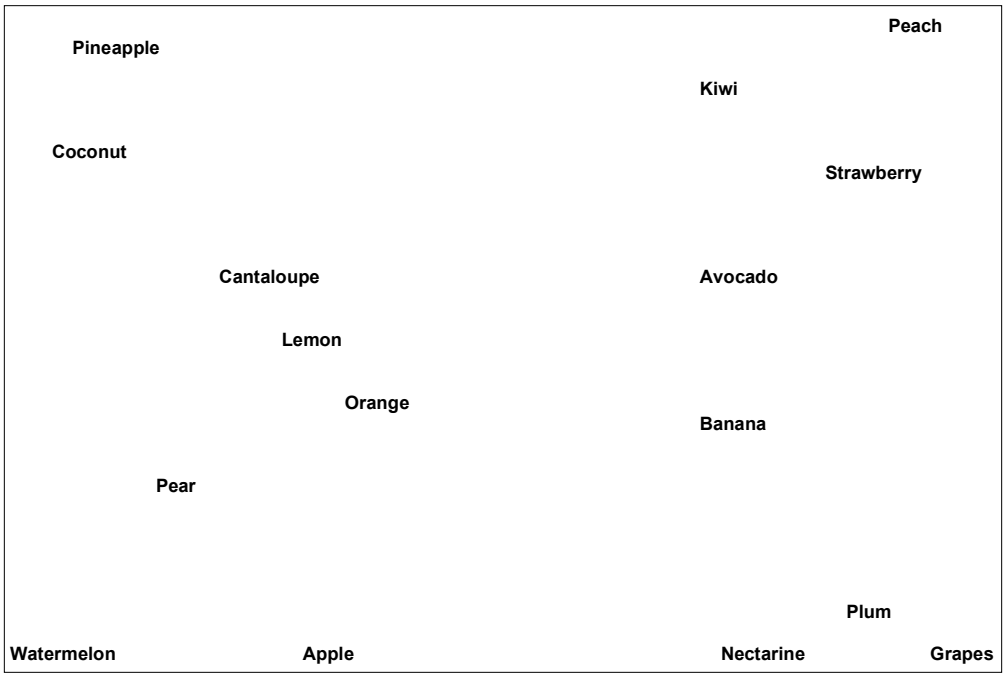
| | Pineapple | Coconut | Strawberry | Banana | Plum | Grapes | Blueberry | Peach |
|---|---|---|---|---|---|---|---|---|
| Subject #1 | 64 | 47 | 80 | 25 | 16 | 54 | 8 | 78 |
| Subject #2 | 100 | 20 | 75 | 68 | 11 | 50 | 60 | 90 |

Correlations between each pair of fruits were then calculated. These correlations were converted to distances by taking 1 – r.

Goal: Determine what factors influence preferences.

Results: Let's ignore stress and the Shepard's plot. The 2-dimensional map is displayed below.

Can you draw and interpret a pair of orthogonal axes? Why are some fruits nearby and others are far apart?



## Example: Crime Rates

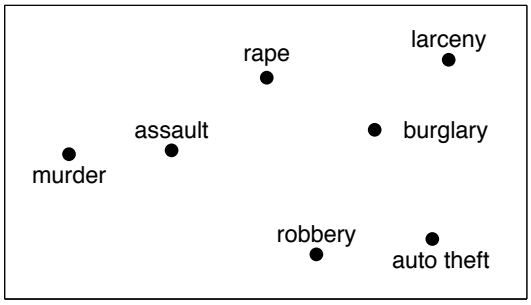
Data: Correlations between crime rates in the U.S.
The correlations were converted to distances.

Goal: Detect patterns within the correlation matrix.

Result: A 2-dimensional map is displayed below.

| crime | no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| murder | 1 | 1.00 | 0.52 | 0.34 | 0.81 | 0.28 | 0.06 | 0.11 |
| rape | 2 | 0.52 | 1.00 | 0.55 | 0.70 | 0.68 | 0.60 | 0.44 |
| robbery | 3 | 0.34 | 0.55 | 1.00 | 0.56 | 0.62 | 0.44 | 0.62 |
| assault | 4 | 0.81 | 0.70 | 0.56 | 1.00 | 0.52 | 0.32 | 0.33 |
| burglary | 5 | 0.28 | 0.68 | 0.62 | 0.52 | 1.00 | 0.80 | 0.70 |
| larceny | 6 | 0.06 | 0.60 | 0.44 | 0.32 | 0.80 | 1.00 | 0.55 |
| auto theft | 7 | 0.11 | 0.44 | 0.62 | 0.33 | 0.70 | 0.55 | 1.00 |

Table 1: Correlations of crime rates over 50 U.S. states.




Can you interpret a set of orthogonal axes?

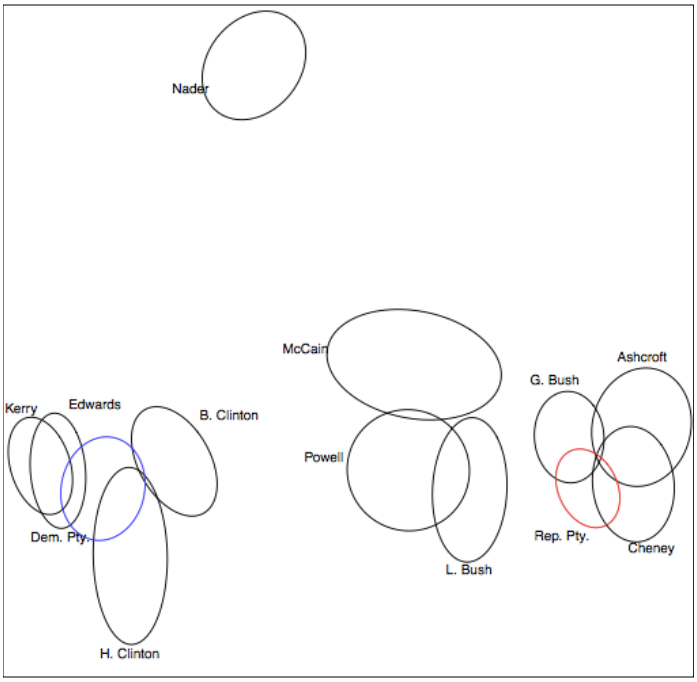
Source: Borg, I., & Groenen, P. (1997). Modern multidimensional scaling: theory and applications. New York: Springer.

## Example: 2004 Presidential Election

Data: Affective ratings of political figures from 711 respondents to the CPS American National Election Survey in 2004.

Result: A 2-dimensional map is displayed to the right.
Stress = 0.04.
(includes 95% bootstrap confidence ellipsoids)

Can you interpret a pair of orthogonal axes?

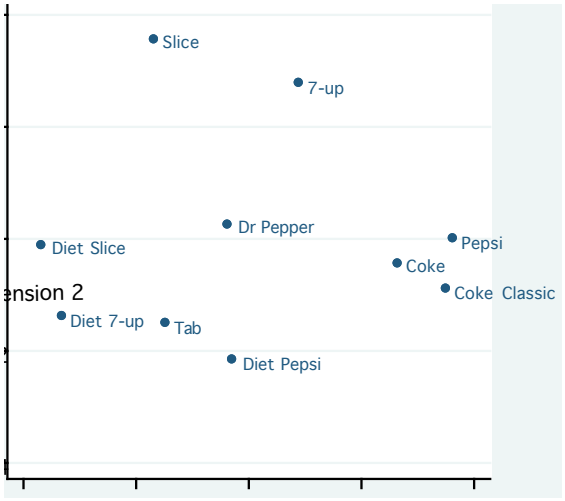



Source: http://www.quantoid.net/papers/jacoby_armstrong_bsmds.pdf

## Example: Sodas

| | Coke | Coke Cl. | Diet Pepsi | Diet Slice | Diet 7-up | Dr Pepper | Pepsi | Slice | Tab | 7-up |
|---|---|---|---|---|---|---|---|---|---|---|
| Fruity | 5.79 | 6.49 | 5.8 | 2.91 | 4.29 | 4.03 | 5.73 | 1.38 | 5.22 | 2.86 |
| Carbonation | 3.42 | 3.89 | 4.87 | 5.66 | 4.93 | 4.36 | 3.14 | 5.18 | 5.24 | 3.89 |
| Calories | 4.68 | 5.57 | 3.36 | 3.47 | 3.63 | 5.4 | 4.61 | 4.84 | 3.8 | 4.5 |
| Tart | 3.32 | 4.24 | 5.01 | 6.08 | 6.22 | 4.47 | 2.71 | 3.73 | 5.35 | 3.52 |
| Thirst | 4.56 | 4.19 | 5.56 | 5.08 | 5.52 | 4.77 | 4.15 | 2.77 | 5.24 | 2.78 |
| Popularity | 3.35 | 2.21 | 4.05 | 5.86 | 6.31 | 5.1 | 2.24 | 5.63 | 5.35 | 3.98 |
| Aftertaste | 3.95 | 3.7 | 5.28 | 5.21 | 5.61 | 4.89 | 3.71 | 4.03 | 5.17 | 2.98 |
| Pick-up | 3.07 | 2.71 | 4.73 | 6.33 | 6.31 | 4.24 | 3.08 | 5.07 | 5.12 | 4.15 |

Can you interpret a pair of orthogonal axes?



**Example: Marvel® Super Heroes**

| | CptAmerica | Spiderman | Wolverine | IronMan | Thor | Hulk | Hawkeye | Black Widow | Ant Man | MrFantastic | DrStrange | ProfessorX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IQ | 3 | 4 | 2 | 6 | 2 | 6 | 3 | 3 | 4 | 6 | 4 | 5 |
| Strength | 3 | 4 | 4 | 6 | 7 | 7 | 2 | 3 | 2 | 2 | 2 | 2 |
| Speed | 2 | 3 | 2 | 5 | 7 | 3 | 2 | 2 | 2 | 2 | 2 | 2 |
| Durability | 3 | 3 | 4 | 6 | 6 | 7 | 2 | 3 | 2 | 5 | 2 | 2 |
| EnergyProj | 1 | 1 | 1 | 6 | 6 | 1 | 1 | 3 | 3 | 1 | 6 | 5 |
| Fighting | 6 | 4 | 7 | 4 | 4 | 4 | 6 | 6 | 2 | 3 | 6 | 3 |

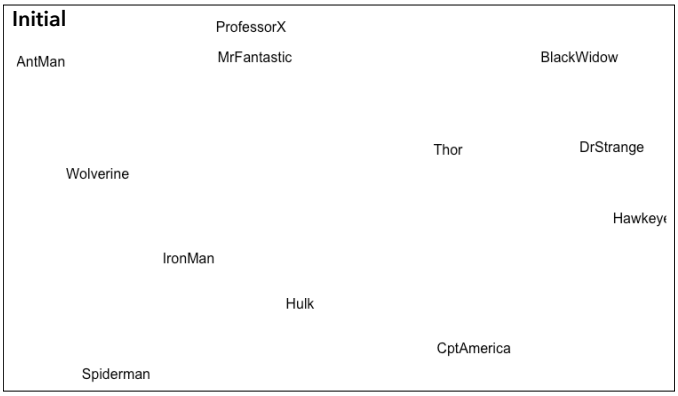
Data: Power grid scores (7-dimensions)

Source: http://marvel.com/universe/OHOTMU:Power_Grids and http://marvel.wikia.com/Power_Grid

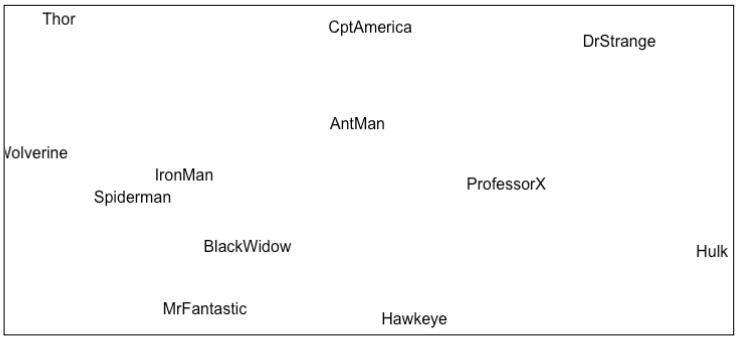
Goal: Reduce dimensionality; explain similarities

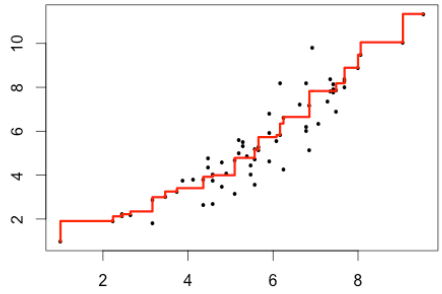
Distances: Calculated between row vectors

Maps: Initial: STRESS = 0.289
1-dimension: STRESS = 0.163
2-dimensions: STRESS = 0.029
3-dimensions: STRESS = 0.007




**2-D**



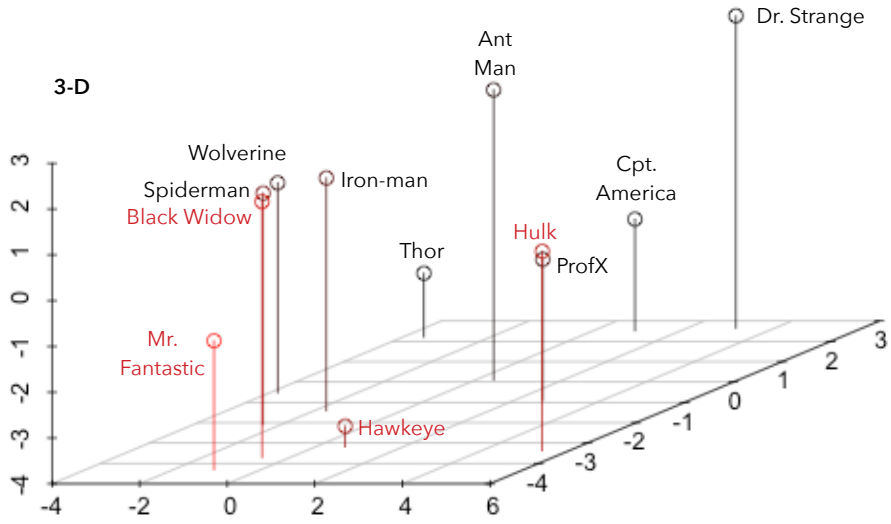



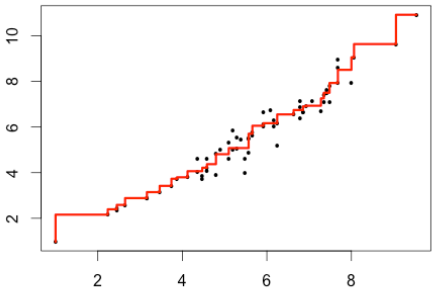

Initial map: It looks like some nearby heroes make sense (e.g., Professor X, Mr. Fantastic, and Ant-Man are intelligent).

2-D map: I'm not sure I see what the 2 dimensions represent.

3-D map: I'm still not sure I have a great interpretation, but the Shepard's plot shows a much better correspondence.

**3-D**



**Final Example**: **Prestigious Jobs**

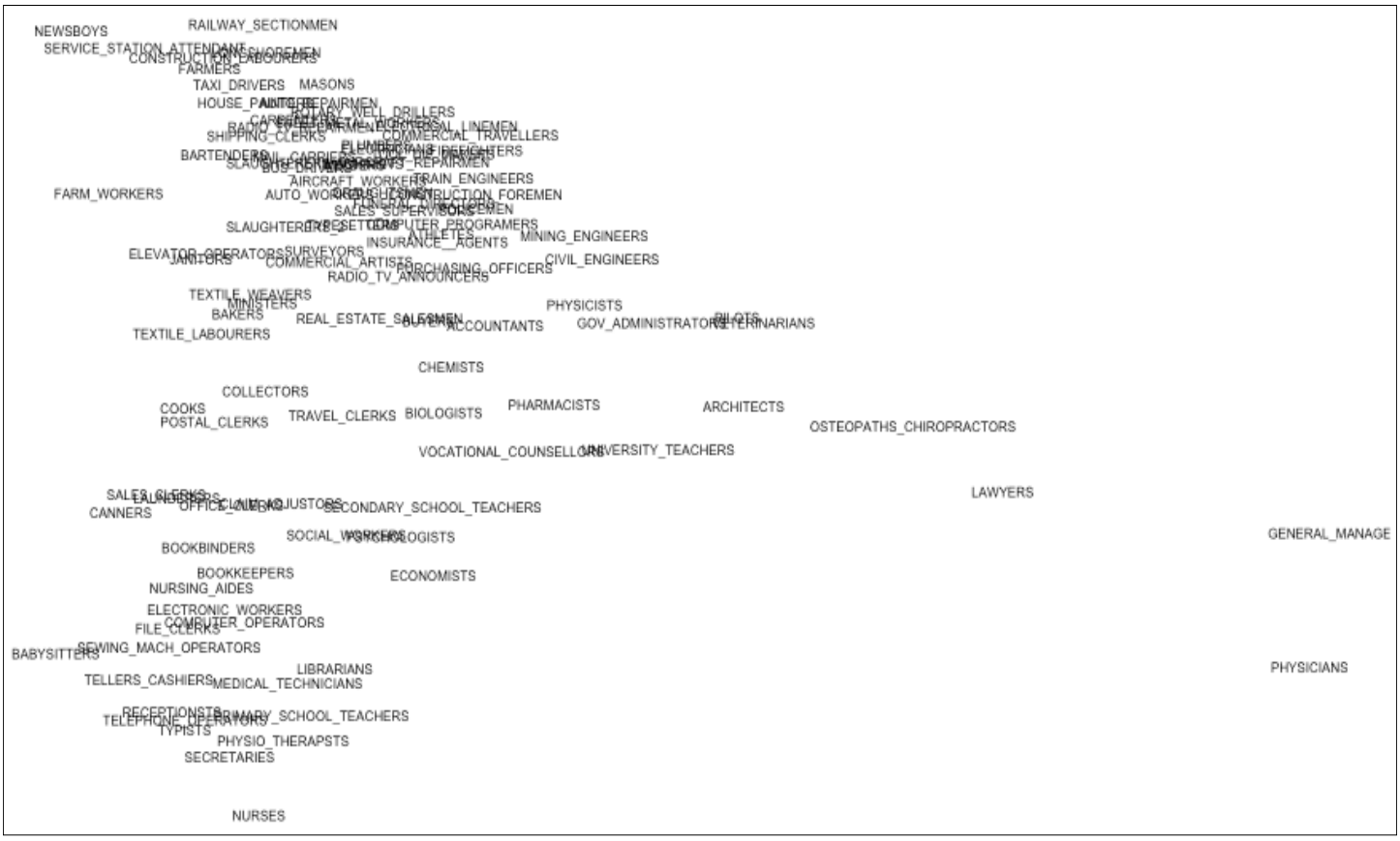
Data: The prestige dataset we've used in class.

Distances: Calculated between row vectors

Maps: 1-dimension: STRESS = 0.00002569

2-dimensions: STRESS = 0.0000006135

3-dimensions: STRESS = 0.000000000061



13) Describe what a surfboard looks like to someone who has never seen one before. Then, describe it to someone who can only understand 2-dimensions. Finally, describe it to someone who only understands one dimension. How many dimensions do you need to give a "good enough" description of a surfboard?

Recall the example where we tried to predict student retention at St. Ambrose from a long list of potential predictors, including: ACT scores (English, math, reading, science), high school GPA, demographic variables (gender, race, mother/father level of education, financial variables), first semester credits attempted, student major, and responses to several dozen survey questions.

In that example, I chose a small subset of predictors and had you fit a logistic regression model. We then compared that model to other models with other predictors. The choice of predictors for the first model was somewhat arbitrary – the only concern I had when I chose those predictors was to make sure I avoided multicollinearity. I did not want the predictor variables to be strongly correlated with each other.

It would be nice if we could choose a subset of variables that were guaranteed to be uncorrelated with each other. If these variables were linear combinations of our predictors, a small subset of these uncorrelated linear transformations might retain most of the information in all our predictors.

The goal of *Principal Components Analysis (PCA)* is to find a set of principal components (linear combinations of predictors) that: (1) is much smaller than the original set of predictor variables
(2) accounts for nearly all the information (variance) in our data

In predicting student retention, a PCA might find a set of principal components (PCs) that could be interpreted as:
a) Academic ability (some combination of ACT scores and high school GPA)
b) Economic status (some combination of mother/father level of education and financial predictors)
c) (more principal components that include a bunch of other stuff)

We might also find the first 2 PCs account for 80% of the variation in the data. That would mean we could use those 2 PCs in our regression analysis (instead of using all the correlated predictors).

In addition to being useful for data reduction, PCA gives us insight into the structure of a dataset. If we think of variables as dimensions and observations as a swarm of points scattered throughout those dimensions, PCA gives us insight into the structure of the swarm of data points.

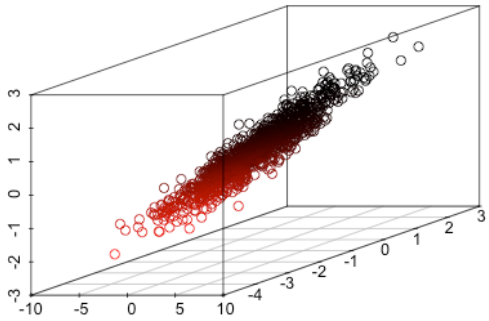
To use PCA, we'll need to know how to find the principal components and how to interpret them.

Finding PCs can be accomplished via:
- spectral (eigenvalue) decomposition of a variance-covariance (or correlation) matrix, or
- singular value decomposition (SVD) of a data matrix

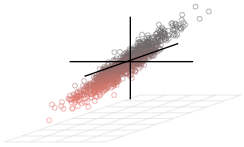
Interpretation of PCs can be with respect to:
- The importance of each PC measured by the proportion of total sample variance accounted for by the PC
- The importance of each predictor variable within each PC, as measured by:
  - The weight of the PC for each variable
  - The correlation between each predictor and the PC

Linear algebra is not a prerequisite for this course, so I'll give my best shot at explaining eigenvalue decomposition and SVD without going into the matrix algebra.

14) As a quick example, take a look at the scatterplot of variables X1, X2, & X3 displayed to the right. I simulated this data so it would look something like a surfboard (if you really squint).




If I draw the axes at the origin, you can see this data has been centered. I converted each variable by subtracting its mean.



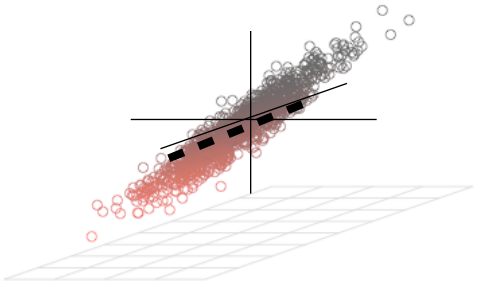
Imagine each point on the scatterplot is pulling in an effort to stretch the data from the origin. The points aren't trying to pull each other; they're trying to stretch the data out from the origin. The farther a point is from the origin, the harder it's pulling to stretch the data.

The forces of all this pulling and stretching can be summarized in what's called a covariance matrix. A covariance matrix shows all the variances and covariances for a set of data. For this dataset, the covariance matrix is

| | x1 | x2 | x3 |
|---|---|---|---|
| x1 | $\sigma_{11} = \sigma^2_1 = 5.00142$ | $\sigma_{12} = 0.61214$ | $\sigma_{13} = 0.67859$ |
| x2 | $\sigma_{21} = 0.61214$ | $\sigma_{22} = \sigma^2_2 = 1.10395$ | $\sigma_{23} = 0.66672$ |
| x3 | $\sigma_{31} = 0.67859$ | $\sigma_{32} = 0.66672$ | $\sigma_{33} = \sigma^2_3 = 0.42313$ |

The sum of all the variances represents the total sample variance in the dataset: 5.001 + 1.104 + 0.423 = 6.528. The 3 variables in this dataset contain a total variance of 6.528.

Remember, we want to find a smaller number of principal components (PCs) that will account for much of this variation. Think again about all the points pulling and stretching our data. The first PC (which happens to be the eigenvector of the covariance matrix) is a line pointing in the direction in which the points are pulling the hardest. To the right, I've attempted to sketch this PC (think of the dotted line as cutting right through the center of the points or through the length of the surfboard).



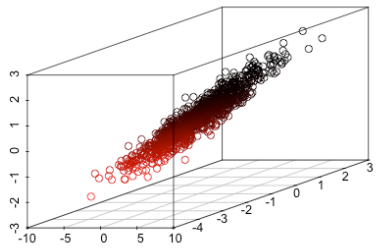
The length of the PC will tell us how strongly the data are pulling in that direction and, therefore, the amount of variation the PC accounts for. In this case, as we'll see, the PC has a variance of 5.22138 (in other words, it accounts for 5.22138 / 6.528 = 80% of the sample variance in our data). The length of the surfboard gives us about 80% of the information about the surfboard.

Now that we have this first PC, we try to find the second by looking at all vectors that are perpendicular (orthogonal) to the first. We choose the direction among these that has the strongest pulling force. This represents our second PC which, in this case, accounts for just under 20% of the variance (virtually all the variance remaining). This means the length and width of our surfboard tell us virtually everything we need to know.

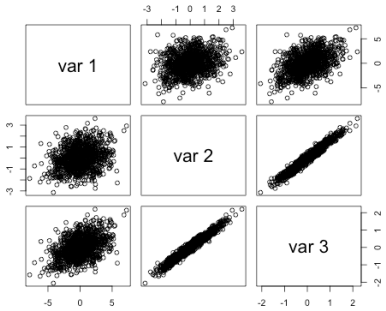
The 3rd PC, which is the last orthogonal vector for a dataset with 3 variables, represents the 3rd direction of force. It's the depth of the surfboard, which really doesn't matter a whole lot.

In this example, 2 PCs retain almost all the variance of our original 3 variables. We could, then, use these uncorrelated PCs as predictors in a regression model. We could also use these PCs to gain an understanding about the relationships among our variables (which, in the case of the physical dimensions of a surfboard, don't make much sense).

15) The previous page attempted to give you an intuitive sense of PCA. Let's try to gain some understanding about the underlying mathematics.



Before we get to real data, let's take one more look at the dataset I created for the previous example. To the right, you can see a 3-dimensional scatterplot and all the pairwise 2-D scatterplots showing the relationships among X1, X2, & X3.




Notice (again) that since X2 and X3 are so highly correlated, perhaps one linear combination of these variables will contain most of the information contained within each of those variables.

The data in this example look like an nxp matrix, with n=1000 observations and p=3 variables:

| | X1 | X2 | x3 |
|---|---|---|---|
| [1,] | -3.6150820 | -2.447474 | -1.6277340 |
| [2,] | 1.3277630 | -0.355969 | -0.1139310 |
| [3,] | 4.6904700 | 2.279994 | 1.6039920 |
| [4,] | -0.0214397 | 2.194254 | 1.2431600 |
| | … | … | … |
| [1000] | 2.2214370 | 0.700732 | 0.5446426 |

You can visualize this matrix **X** as the swarm of points in the scatterplot. We're going to decompose this matrix **X** into three components: a) An *nxp* matrix, **U**, of uncorrelated variables with unit variances
b) A *pxp* diagonal scaling matrix, **Λ**, that stretches or scales the uncorrelated variables
c) A pxp rotation matrix, **V**, that rotates the points

A singular value decomposition simply states: $X = U\Lambda^{1/2}V^T$

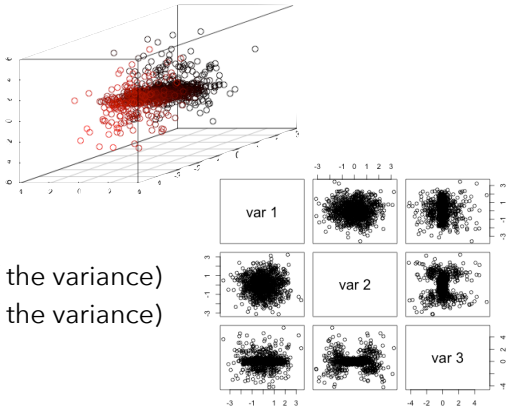
The columns of our rotation matrix (eigenvectors of the covariance matrix) represent our principal components. The elements along the diagonal of our scaling matrix, **Λ**, represent the variance accounted for by each PC.

For the data in this example, I had R find the following SVD:

$$X = U \quad \Lambda^{1/2} \quad V^T$$

$$\begin{bmatrix} -3.615 & -2.447 & -1.628 \\ +1.328 & -0.356 & -0.114 \\ \vdots & \vdots & \vdots \\ +2.221 & +0.701 & +0.545 \end{bmatrix} = \begin{bmatrix} -1.835 & +1.775 & +1.773 \\ +0.530 & +0.577 & +0.472 \\ \vdots & \vdots & \vdots \\ +1.331 & -0.231 & +0.013 \end{bmatrix} \begin{bmatrix} 2.285 & 0 & 0 \\ 0 & 1.143 & 0 \\ 0 & 0 & (1.2\times10^{-7}) \end{bmatrix} \begin{bmatrix} +0.972 & +0.171 & +0.161 \\ +0.228 & -0.853 & -0.469 \\ -0.058 & -0.493 & +0.868 \end{bmatrix}$$

As can be seen to the right, the columns of the **U** matrix are uncorrelated (r = 0 for each pair of columns).




The diagonal elements of **Λ** represent the variance accounted for by each PC. As was mentioned on the previous page, the variance accounted for by each PC is:

- PC 1 accounts for 2.285^2 = 5.221 (or 80% of the variance)
- PC 2 accounts for 1.143^2 = 1.307 (or 20% of the variance)
- PC 3 accounts for virtually no variance

The columns of the **V** matrix (or rows of the $\mathbf{V}^T$ matrix) represent our principal components.

- PC #1: [0.972, 0.171, 0.161] (mostly represents X1)
- PC #2: [0.228, -0.853, -0.469] (mostly represents X2 and X3)
- PC #3: [-0.058, -0.493, 0.868] (mostly represents X2 and X3)

If we, for some reason, decided to use just one PC, we could convert our data into principal component scores:

| | | | |
|---|---|---|---|
| Observation #1 from our dataset: | x1 = -3.615 | x2 = -2.447 | x3 = -1.628 |
| Principal Component #1: | 0.972 | 0.171 | 0.161 |
| Principal Component Score: | (-3.615 x 0.972) + (-2.447 x 0.171) + (-1.628 x 0.161) = -4.194 | | |

We could do this to get PC scores for all 1000 observations in this dataset and use those PC scores in our regression analysis.

Now that we have our PCs and our PC scores, we might be interested in looking at the correlations between our original predictor variables and our PC scores:

| | $x1$ | $x2$ | $x3$ |
|---|---|---|---|
| $PC\ 1$ | 0.993 | 0.371 | 0.566 |
| $PC\ 2$ | 0.116 | –0.929 | –0.824 |
| $PC\ 3$ | 0.000 | 0.000 | 0.000 |

With a fictitious example like this, there's not much to interpret here.

**Example: Government Survey**

Data: 100 subjects answered the following 6 questions about their political opinions:

- X1: Government should spend more money on schools
- X2: Government should spend more money to reduce unemployment
- X3: Government should control big business
- X4: Government should expedite desegregation through busing
- X5: Government should see to it that minorities get their respective job quotas
- X6: Government should expand the Head Start program

Unfortunately, I don't have the actual data from this survey. Instead, I have the following correlation matrix:

| | Schools | Emply | CntrlBus | DesegBus | Quotas | HeadStart |
|---|---|---|---|---|---|---|
| Schools | 1 | 0.6008 | 0.4984 | 0.1920 | 0.1959 | 0.3466 |
| Emply | 0.6008 | 1 | 0.4749 | 0.2196 | 0.1912 | 0.2979 |
| CntrlBus | 0.4984 | 0.4749 | 1 | 0.2079 | 0.201 | 0.2445 |
| DesegBus | 0.1920 | 0.2196 | 0.2079 | 1 | 0.4334 | 0.3197 |
| Quotas | 0.1959 | 0.1912 | 0.201 | 0.4334 | 1 | 0.4207 |
| HeadStart | 0.3466 | 0.2979 | 0.2445 | 0.3197 | 0.4207 | 1 |

The eigenvectors of this correlation matrix were found to be:

```
           [,1]       [,2]        [,3]       [,4]       [,5]        [,6]
[1,] -0.4628634  0.3700542  0.12155806 -0.1969581  0.1690869  0.75276697
[2,] -0.4531557  0.3647420 -0.03560602 -0.3489640  0.3787645 -0.62857532
[3,] -0.4202459  0.3334449 -0.30752665  0.6479136 -0.4319532 -0.10611156
[4,] -0.3415384 -0.4845598 -0.61456832 -0.4351027 -0.2754094  0.07546109
[5,] -0.3544231 -0.5463162  0.03334405  0.4691417  0.5945844  0.03444451
[6,] -0.4019161 -0.2925373  0.71454682 -0.1011496 -0.4604156 -0.14175504
```

with associated eigenvalues: `2.6333490 1.2266525 0.6835548 0.5604686 0.5049686 0.3910066`

a) How much total sample variance do we have? What proportion of variation is accounted for by the first two PCs?

b) How would you Interpret the first two PCs?

c) How could we use these PCs in future analyses?

**Example: Need for cognition**

Data: 201 subjects responded to an 18-item survey designed to measure "need for cognition" (the extent to which the subject enjoys and engages in thinking and problem solving). Subjects rated their agreement with each of the following items on a 9-point scale (ranging from –4 = very strongly disagree to 0 = neutral to +4 = very strongly agree):

1 I would prefer complex to simple problems.
2 I like to have the responsibility of handling a situation that requires a lot of thinking.
3 Thinking is not my idea of fun.*
4 I would rather do something that requires little thought than something that is sure to challenge my thinking abilities.*
5 I try to anticipate and avoid situations where there is likely a chance I will have to think in depth about something.*
6 I find satisfaction in deliberating hard and for long hours.
7 I only think as hard as I have to.*
8 I prefer to think about small, daily projects to long-term ones.*
9 I like tasks that require little thought once I've learned them.*
10 The idea of relying on thought to make my way to the top appeals to me.
11 I really enjoy a task that involves coming up with new solutions to problems.
12 Learning new ways to think doesn't excite me very much.*
13 I prefer my life to be filled with puzzles that I must solve.
14 The notion of thinking abstractly is appealing to me.
15 I prefer a task that is intellectual, difficult, & important to one that is important but does not require much thought
16 I feel relief rather than satisfaction after completing a task that required a lot of mental effort.*
17 It's enough for me that something gets the job done; I don't care how or why it works.*
18 I usually end up deliberating about issues even when they do not affect me personally.

The highlighted items* were intended to be reversed-coded so that higher scores represent lower need for cognition.

Suppose we want to create a single index for "need for cognition" (instead of looking at the 18 questions separately). One way to create an index would be to simply take the sum of all the item scores (subtracting the reverse-coded items). An alternative approach would be to use PCA.

Data source: http://www.bradthiessen.com/html5/data/cognition.csv
Survey source: http://www.liberalarts.wabash.edu/ncs/

a) The SVD of the data matrix (after converting all data to z-scores) yielded 18 PCs and the amount of sample variance accounted for by each. The table shows this information for the first 4 PCs. Below, I've pasted a Scree plot showing the variance accounted for by each PC.

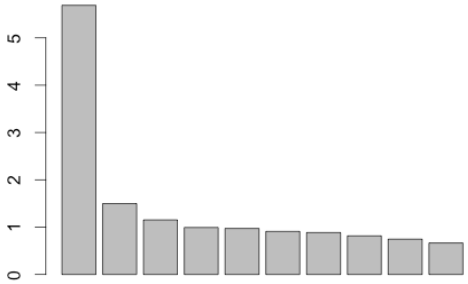
Based on the loadings in the table, interpret the first PC. How does this PC compare to an index in which all the item scores are summed (after reverse-coding some items)?

b) Based on the proportion of variance explained by each PC, how many PCs would you choose to keep?

| | PC1 | PC2 | PC3 | PC4 |
|---|---|---|---|---|
| c1 | 0.2598 | -0.1136 | 0.0585 | -0.1212 |
| c2 | 0.3063 | -0.0275 | 0.0059 | -0.1612 |
| c3 | -0.2488 | -0.1144 | 0.1148 | 0.1330 |
| c4 | -0.2735 | -0.2621 | 0.0435 | -0.1566 |
| c5 | -0.2980 | -0.0959 | 0.0021 | -0.3395 |
| c6 | 0.1917 | -0.1898 | 0.2207 | 0.0684 |
| c7 | -0.2150 | -0.1899 | -0.1505 | -0.1380 |
| c8 | -0.2205 | -0.0747 | 0.1583 | 0.2547 |
| c9 | -0.2144 | -0.3187 | 0.3103 | 0.2047 |
| c10 | 0.2625 | -0.1918 | -0.2399 | 0.0640 |
| c11 | 0.2864 | -0.1828 | -0.0351 | 0.1348 |
| c12 | -0.2629 | 0.0481 | -0.1013 | -0.4137 |
| c13 | 0.2340 | -0.2776 | -0.2549 | 0.0639 |
| c14 | 0.2425 | -0.4093 | 0.0856 | 0.0354 |
| c15 | 0.1395 | -0.2478 | -0.0427 | -0.6145 |
| c16 | -0.1534 | -0.3723 | -0.4022 | 0.0327 |
| c17 | -0.2215 | -0.4181 | -0.0483 | 0.2164 |
| c18 | 0.1004 | -0.1560 | 0.6939 | -0.2170 |
| **Var** | **5.688** | **1.499** | **1.153** | **0.993** |
| **Prop** | **31.5%** | **8.3%** | **6.4%** | **5.5%** |

c) How do we decide on the number of principal components to keep? Several criteria or methods have been proposed to determine the *optimal* number of PCs to retain in an analysis:

- Scree plots. Wikipedia tells me, "Scree is a collection of broken rock fragments at the base of crags, mountain cliffs, volcanoes or valley shoulders that has accumulated through periodic rockfall from adjacent cliff faces." With regards to PCA, a scree plot shows the proportion of total sample variance explained by each PC.




To the left, I've sketched a scree plot for the first 10 PCs in this example. To choose the number of PCs to retain, we typically look for the bend (elbow) in the plot. Based on this scree plot, how many PCs would you choose to retain?

- Proportion of variance explained. We could somewhat arbitrarily choose to keep the PCs that account for at least a certain threshold of the variance in our data. For example, we could choose to keep PCs until we account for 50% of the variance. If that's our criterion, how many PCs would we keep?

- Kaiser's rule, which is really just a recommendation, calls for keeping all PCs that account for at least a variance of 1.0 (assuming the variables have been standardized, as they have in this example). Based on this recommendation, how many PCs should we keep?

- Horn's method. Rather than use a fixed value of 1.0 (like Kaiser), Horn suggested that we generate random data (with the same number of observations and variables) and conduct a PCA. We then compare the amount of variance explained by the PCs in our actual data to the amount of variance explained by the corresponding PCs generated from the random data. Each "actual data PC" that accounts for more variance than the corresponding "random data PC" should be retained.

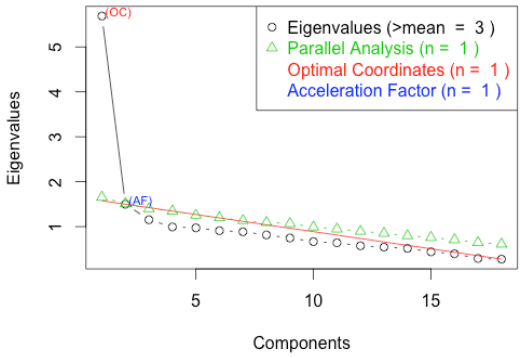
  Below, I've pasted the amount of sample variance explained by 12 of our actual data PCs and random data PCs:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Actual data: | 5.688 | 1.499 | 1.153 | 0.993 | 0.973 | 0.909 | 0.885 | 0.815 | 0.746 | 0.666 | 0.640 | 0.573 |
| Random data: | 1.504 | 1.471 | 1.313 | 1.272 | 1.211 | 1.182 | 1.121 | 1.079 | 0.999 | 0.949 | 0.879 | 0.813 |

  Based on this, how many PCs would you choose to retain?

- Bootstrap method. Instead of generating a single sample of random data (as in Horn's method), we could take a large number of bootstrap samples from our data. Remember, a bootstrap sample is a sample of N observations taken with replacement. For each bootstrap sample, we could calculate the amount of variance accounted for by each PC.

  If we generate enough bootstrap samples, we'd get a distribution of the variance explained by each PC. We could, then, determine the 95th percentile of this distribution. Then, we could compare the amount of variance explained by our actual data PCs to the amount explained by the 95th percentile of our bootstrap PCs.




To the left, I've sketched a plot of the variance explained by the actual data PCs (the black line) to the 95th percentiles of our bootstrap PCs (the green line). From this, how many PCs would you choose to retain?

d) Let's see what kind of scores we get if we use only the first PC. To get the scores, we simply need to multiply our data matrix by our vector of PC loadings.

Here are the PC scores for the first 5 subjects in the dataset: -3.158 -0.977 2.056 -6.222 1.330

We could compare these to the scores we'd get by summing the items (after reverse-coding the appropriate items):

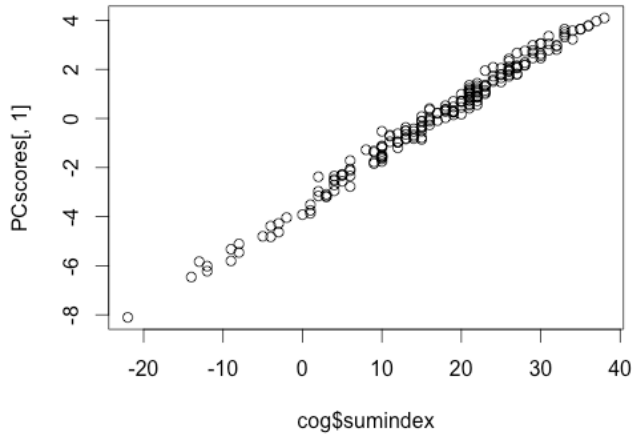
Here are the summed index scores for the first 5 subjects: 2 12 27 -12 23

Those don't look similar at all, but they're on different scales (with different means and standard deviations). Let's standardize these scores and compare them…

| | | | | | |
|---|---|---|---|---|---|
| Standardized PC scores: | -1.30 | -0.42 | 0.91 | -2.54 | 0.55 |
| Standardized summed index scores: | -1.32 | -0.41 | 0.86 | -2.61 | 0.56 |

For our single "need for cognition" index, should we use the simple sum of the item scores or should we use the PC scores? Explain.

Finally, here's a plot showing the relationship between the PC scores and the summed index scores (both unstandardized):




Other interesting examples:
Image compression: http://jackman.stanford.edu/classes/350B/10/prImage.pdf
Connection to Fourier Transform: http://luthuli.cs.uiuc.edu/~daf/courses/CS-498-DAF-PS/Lecture%209%20-%20PCA.pdf
State data example with biplots: http://www.stat.cmu.edu/~cshalizi/350/2008/lectures/14/lecture-14.pdf
Eigenfaces: http://www.cs.princeton.edu/~cdecoro/eigenfaces/