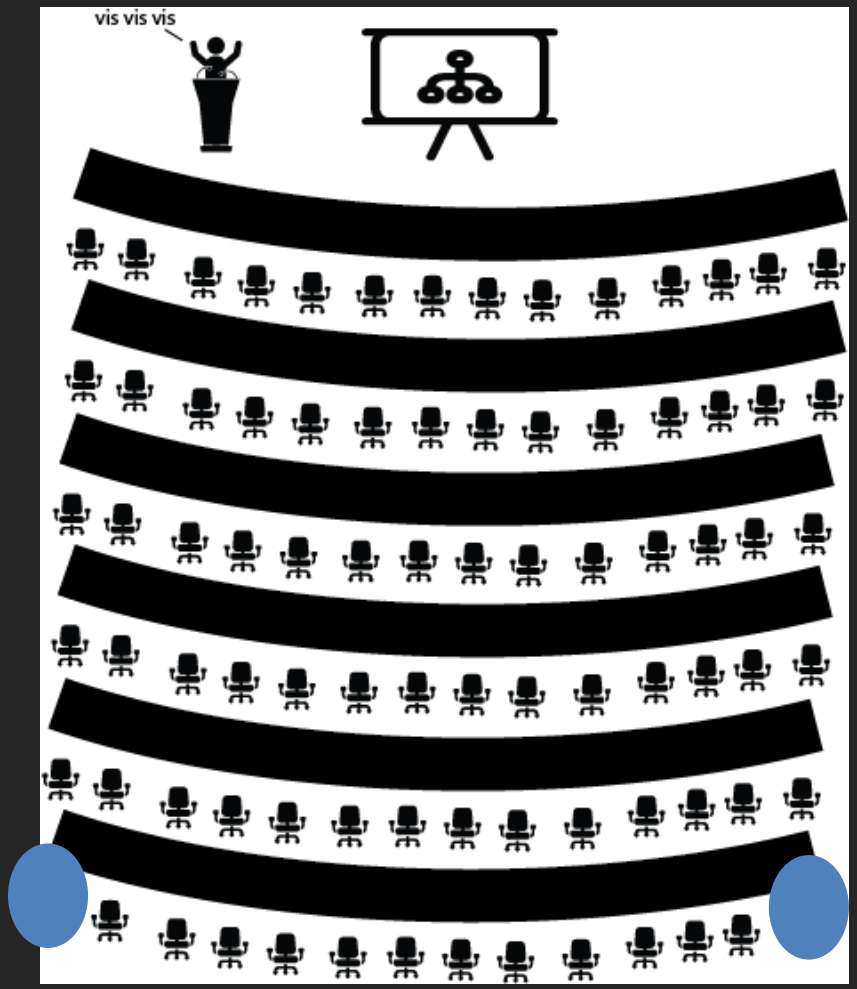
Dimensionality Reduction & Factor Analysis

Some slides from Collins-Thompson and Adar

Dimensions & Factors

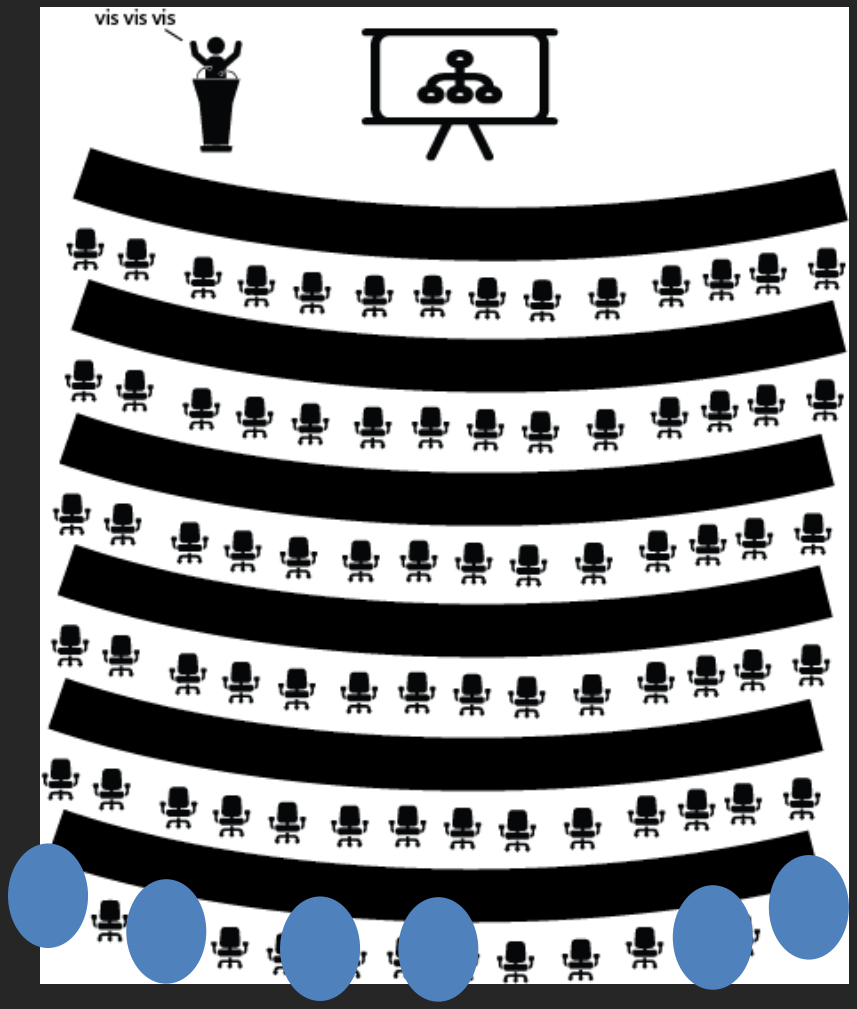
- Dimensionality reduction (MDS)
  - Visual focus (not necessarily meaningful)
- Factor analysis
  - Principal components analysis (PCA)
  - Factor analysis (FA)

Student name	Do you like Pizza House?
Alice	yes
Jake	no
Sam	yes
Beth	yes
Ryan	no
Eric	no
Pat	yes



Do you like Pizza House dimension

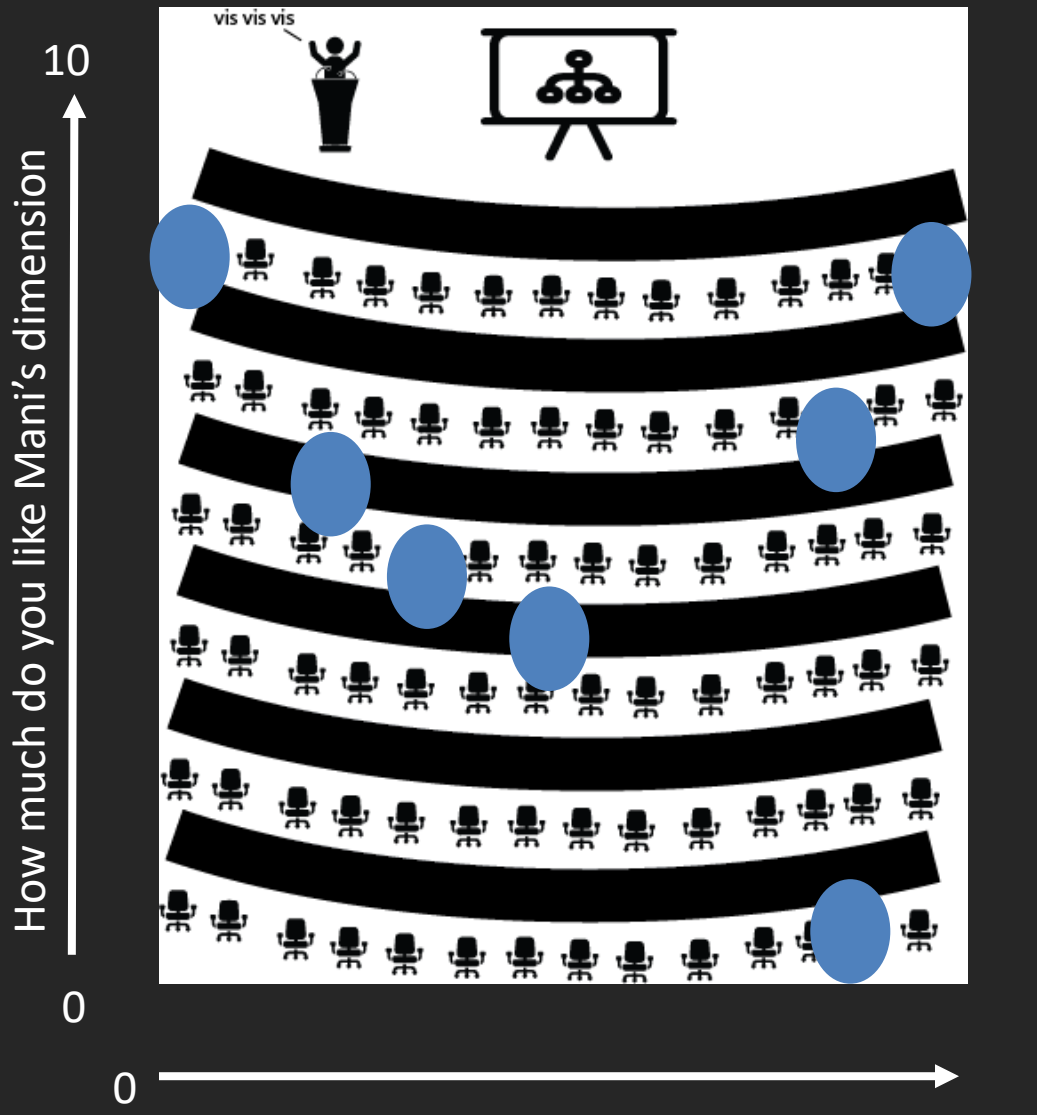
Student name	How much do you like Pizza House?
Alice	10
Jake	8
Sam	8
Beth	6
Ryan	4
Eric	2
Pat	1



0 → 10

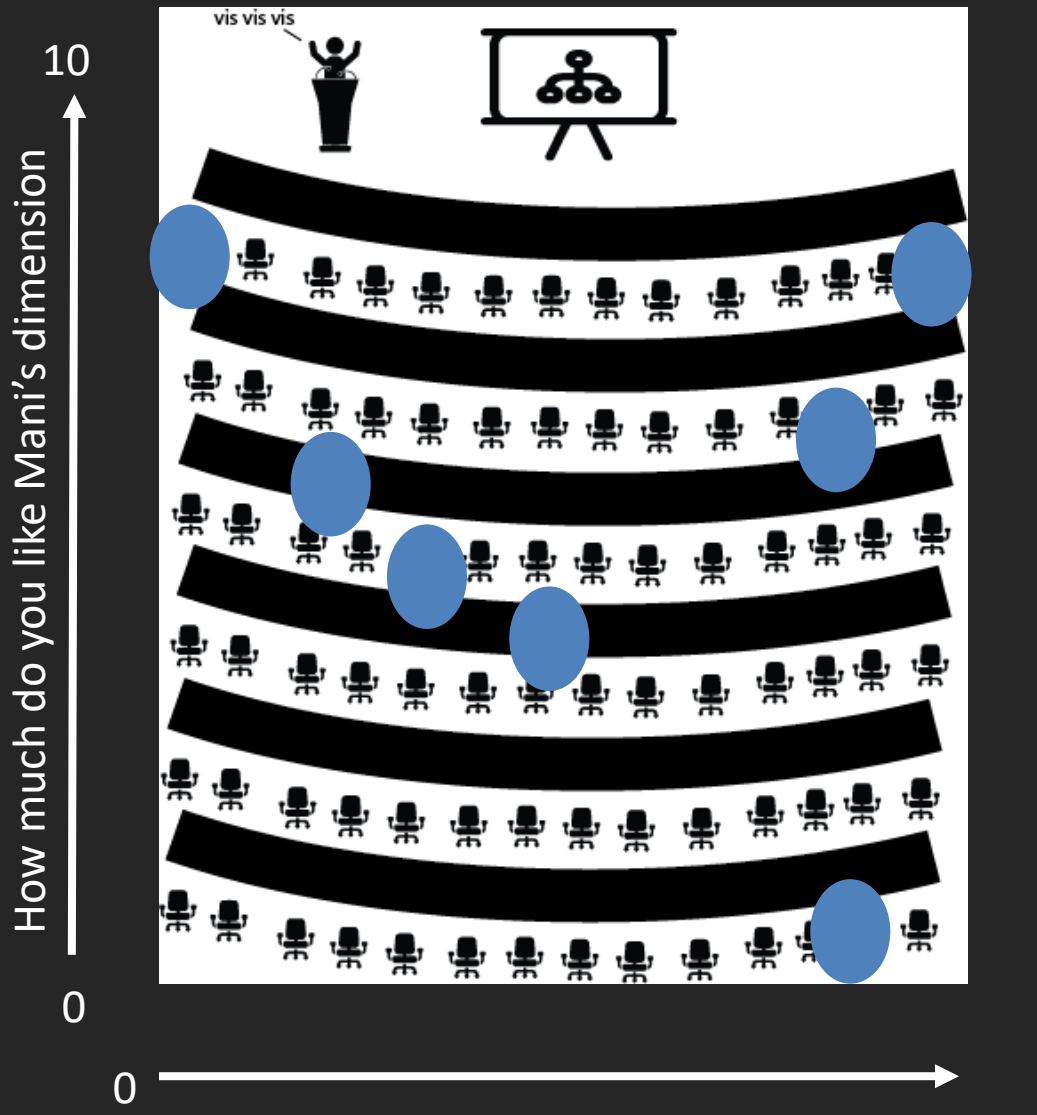
How much do you like Pizza House dimension

name	How much do you like Pizza House?	How much do you like Mani's?
Alice	10	10
Jake	8	1
Sam	8	8
Beth	6	4
Ryan	4	5
Eric	2	6
Pat	1	10



name	PH	Mani's	Dominos
Alice	10	10	10
Jake	8	1	10
Sam	8	8	5
Beth	6	4	6
Ryan	4	5	7
Eric	2	6	10
Pat	1	10	2

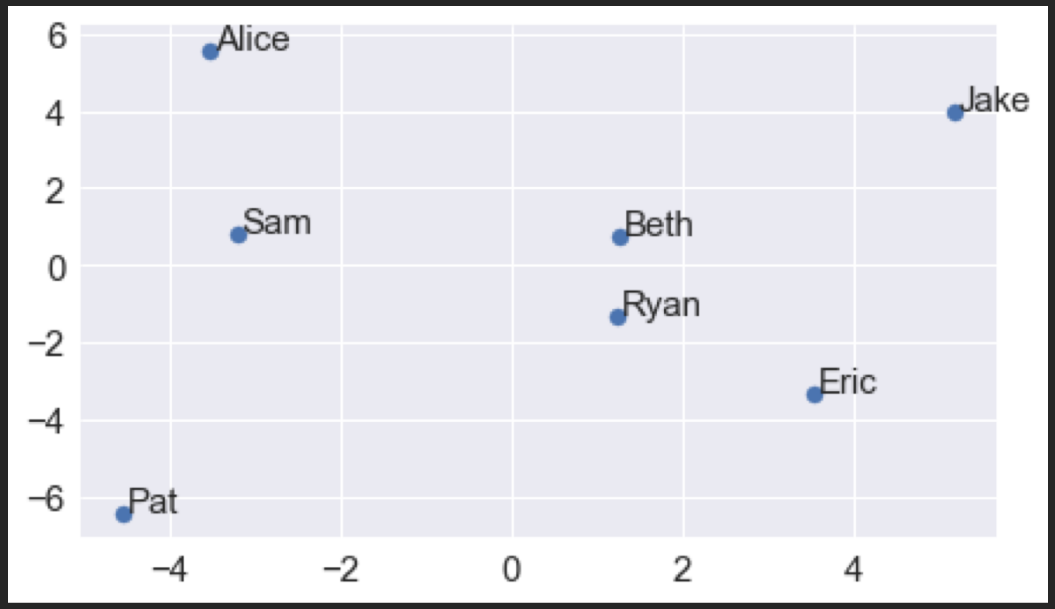
Now what? We're out of dimensions



name	PH	Mani's	Dominos
Alice	10	10	10
Jake	8	1	10
Sam	8	8	5
Beth	6	4	6
Ryan	4	5	7
Eric	2	6	10
Pat	1	10	2

MDS

name	x	y
Alice	-3.5	5.59
Jake	5.18	3.9
Sam	-3.2	.8
Beth	1.2	.72
Ryan	1.2	-1.33
Eric	3.55	-3.33
Pat	-4.52	-6.42



MDS

- What MDS tries to do…
  - Distance in two dimensions is proportional to similarity in the n-dimensions
    - In our case 3
  - How do we measure “similarity”
    - Simplest is Euclidean distance

MDS

- Sim(Alice,Jake) in "original space" =

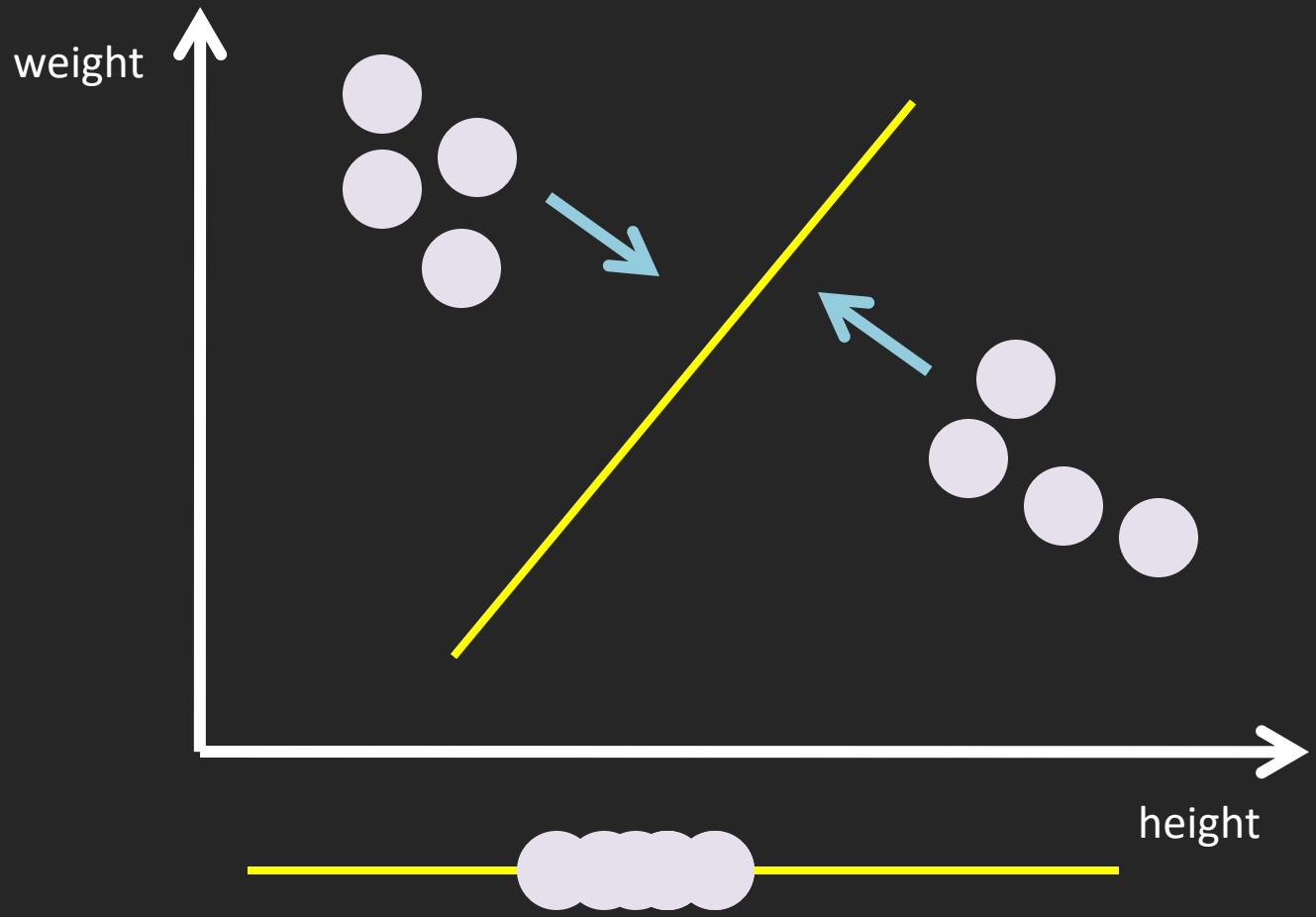
$$\sqrt{(Alice_{PH} - Jake_{PH})^2 + (Alice_{Mani} - Jake_{Mani})^2 + (Alice_{Dom} - Jake_{Dom})^2}$$

- Distance(Alice,Jake) in MDS Space =

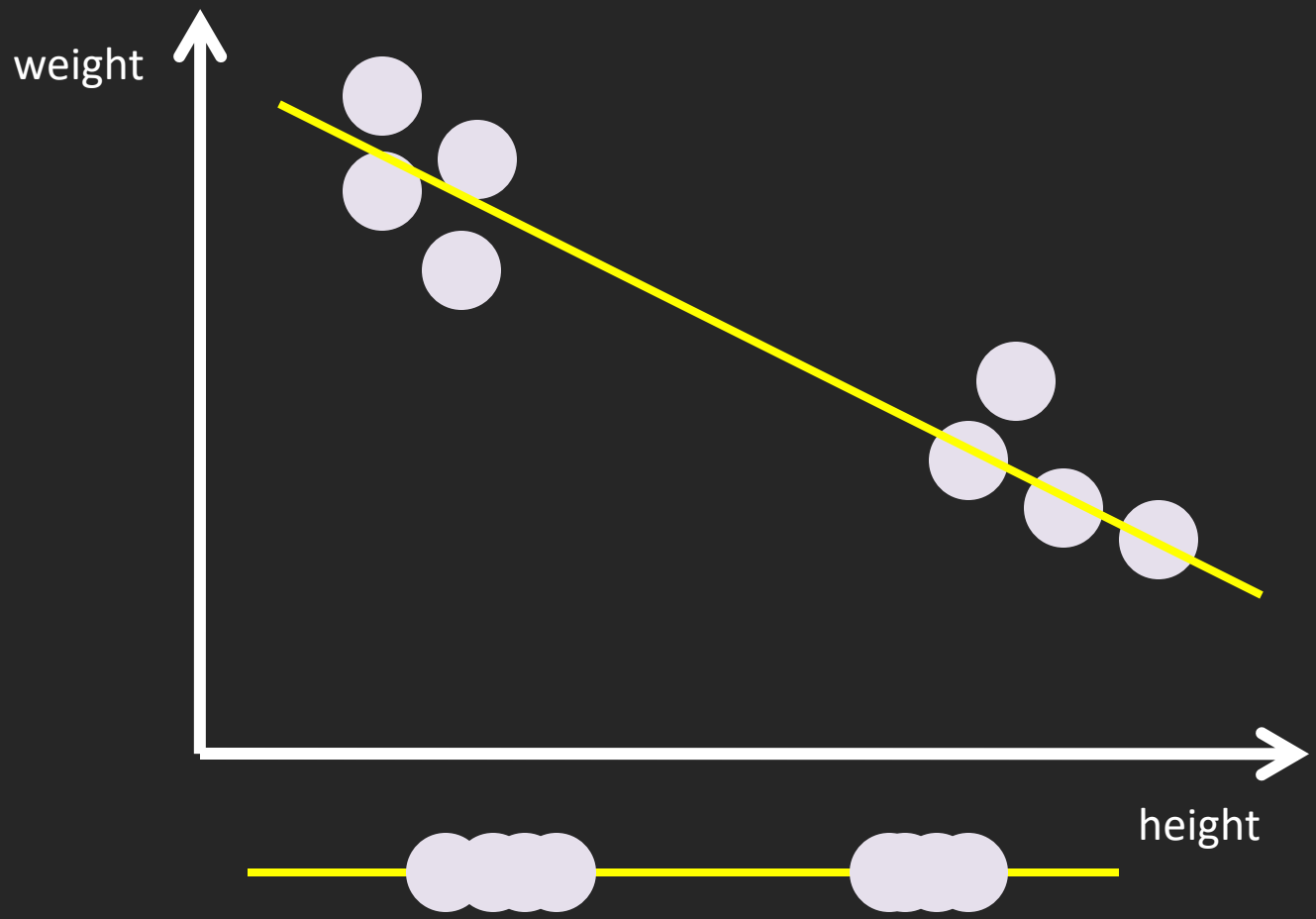
$$\sqrt{(Alice_X - Jake_X)^2 + (Alice_Y - Jake_Y)^2}$$

Sim(Alice,Jake) should be proportional to Distance(Alice,Jake)

name	PH	Mani's	Dominos
Alice	10	10	10
Jake	8	1	10
Sam	8	8	5
Beth	6	4	6
Ryan	4	5	7
Eric	2	6	10
Pat	1	10	2



Goal: draw a line and project the data so that things that are far in 2D are far in 1D



Goal: draw a line and project the data so that things that are far in 2D are far in 1D

More generally…

Multi-dimensional Scaling

- The idea

data x *data* → 2D/3D

MDS

Multi-dimensional Scaling

- Optimization problem
- Pair-wise similarity/dissimilarity of each piece of (multi-dimensional) data
- Tries to plot the n-dimensional data on 2 or 3 dimensions
  - Preserves similarity as well as possible in the reduced space

Multi-dimensional Scaling

- The idea (in practice)

data x *variables* → *data* x *data* → 2D/3D

Pairwise Similarity calculation

MDS

Multi-dimensional Scaling

- Similarity
  - Usually a function
    - Euclidean distance, Cosine, etc.
    - Sim(person1,person2) =

      # of restaurants they like in common
    - sim(fish1,fish2) =

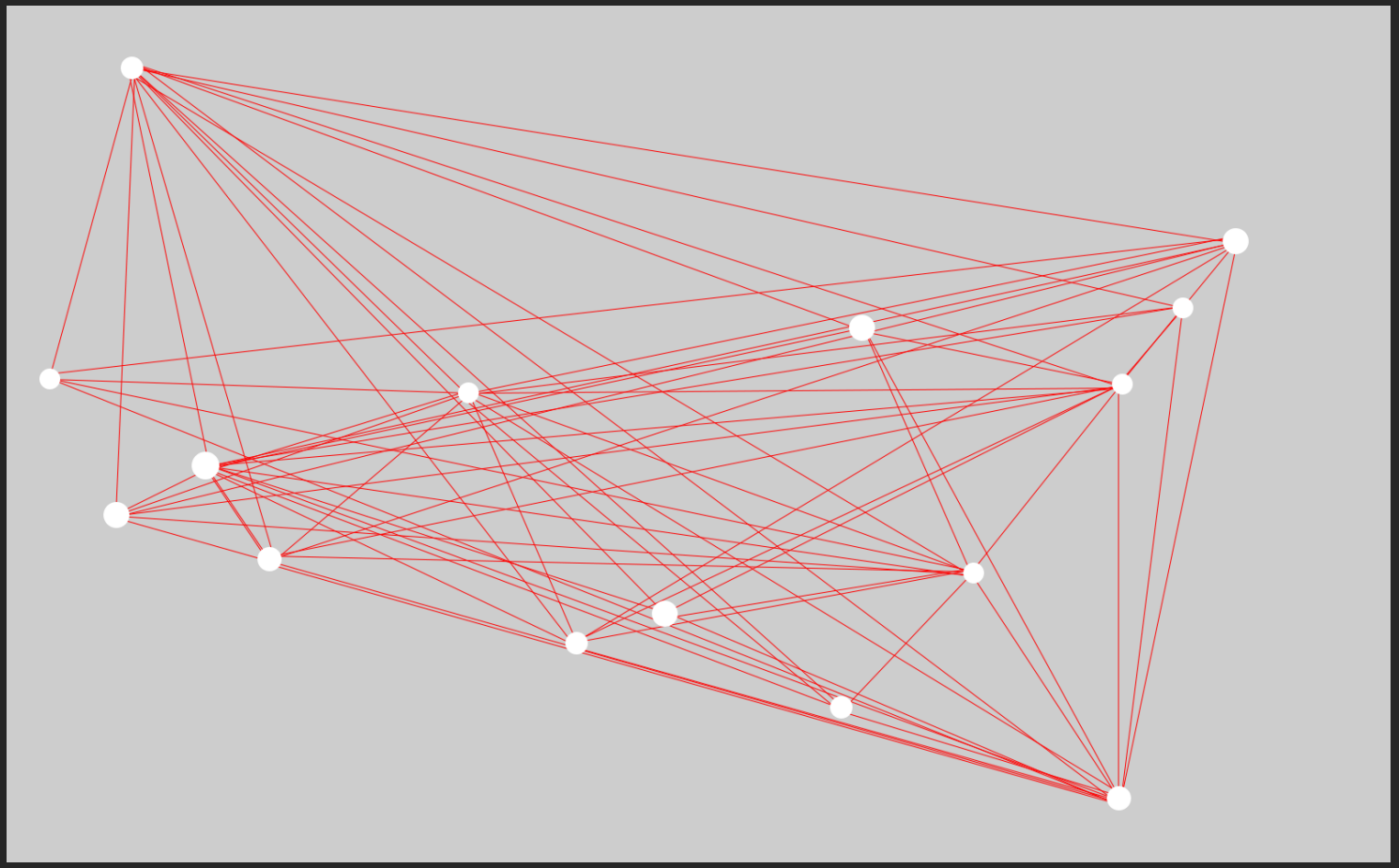
      overlapping genetic markers
    - sim(c1,c2) =

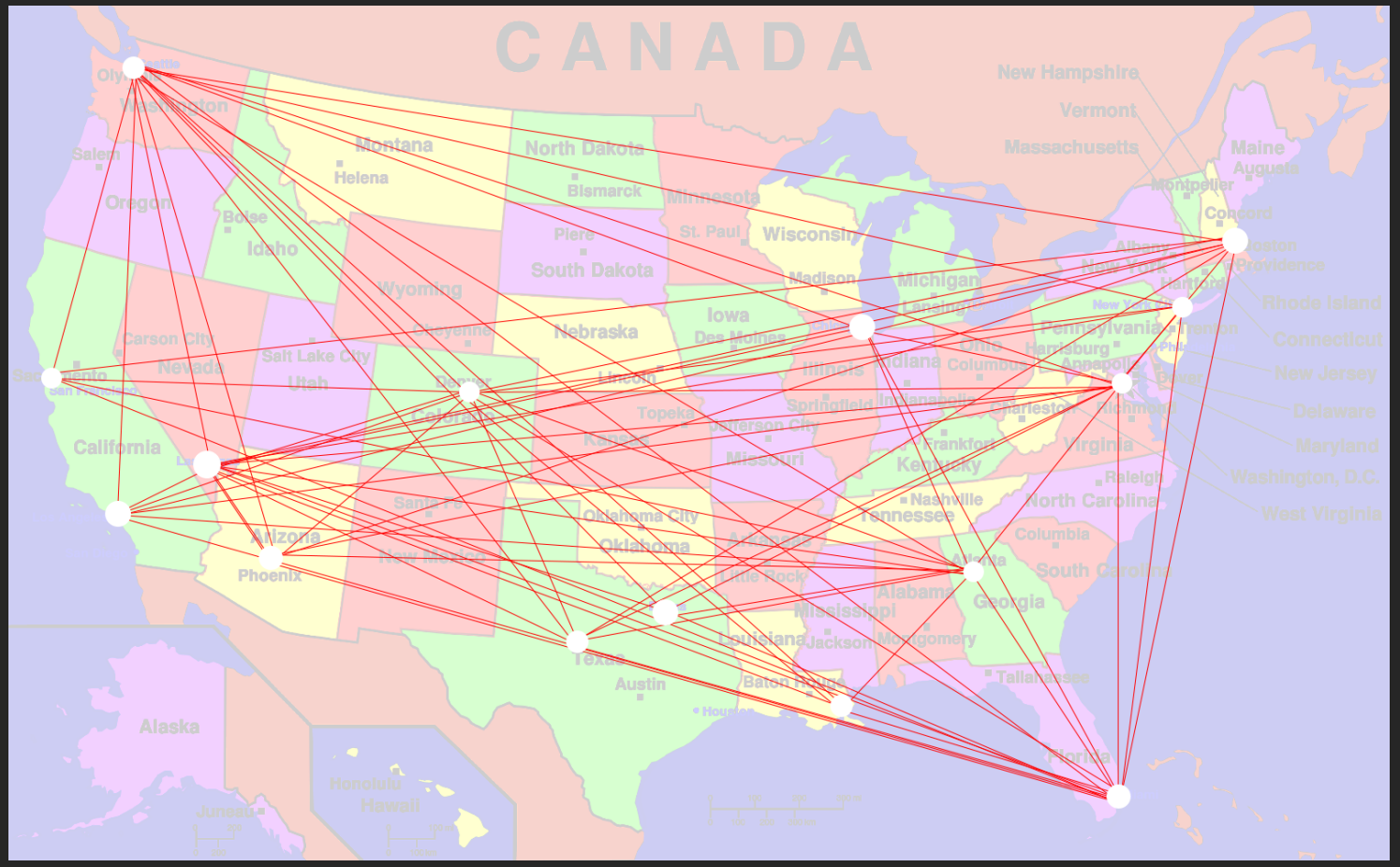
      $(\text{doors1}-\text{doors2})^2 + (\text{cylinders1}-\text{cylinders2})^2 + \ldots$

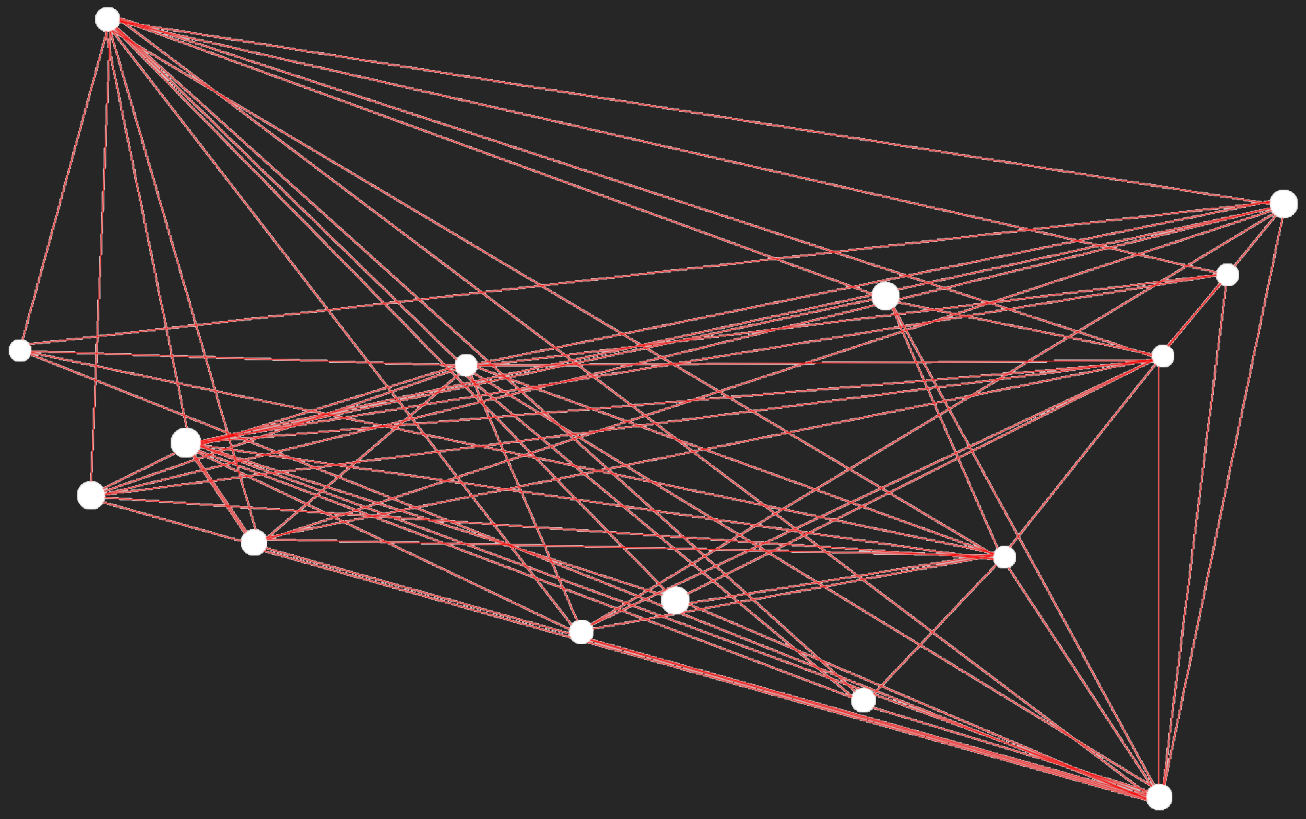
Really ridiculous example

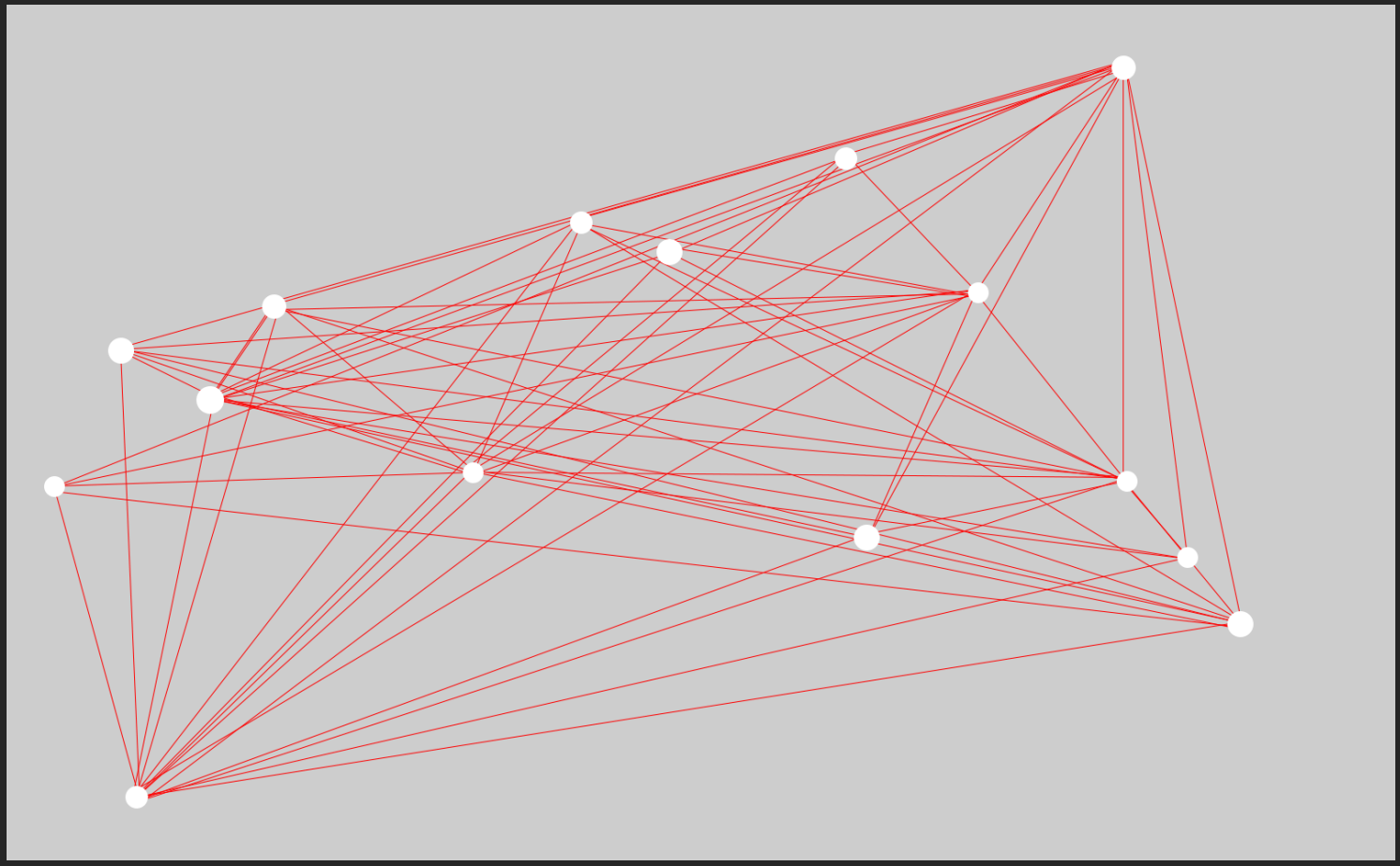
- Have a bunch of cities
- “Similarity” is driving distance

	Atlanta	Boston	Chicago	Dallas	Denver	Houston	Las Vegas	Los Angeles	Miami	New Orleans	New York	Phoenix	San Francisco	Seattle	Washington
Atlanta		1095	715	805	1437	844	1920	2230	675	499	884	1832	2537	2730	657
Boston	1095		983	1815	1991	1886	2500	3036	1539	1541	213	2664	3179	3043	44
Chicago	715	983		931	1050	1092	1500	2112	1390	947	840	1729	2212	2052	695
Dallas	805	1815	931		801	242	1150	1425	1332	504	1604	1027	1765	2122	1372
Denver	1437	1991	1050	801		1032	885	1174	2094	1305	1780	836	1266	1373	1635
Houston	844	1886	1092	242	1032		1525	1556	1237	365	1675	1158	1958	2348	1443
Las Vegas	1920	2500	1500	1150	885	1525		289	2640	1805	2486	294	573	1188	2568
Los Angeles	2230	3036	2112	1425	1174	1556	289		2757	1921	2825	398	403	1150	2680
Miami	675	1539	1390	1332	2094	1237	2640	2757		892	1328	2359	3097	3389	1101
New Orleans	499	1541	947	504	1305	365	1805	1921	892		1330	1523	2269	2626	1098
New York	884	213	840	1604	1780	1675	2486	2825	1328	1330		2442	3036	2900	229
Phoenix	1832	2664	1729	1027	836	1158	294	398	2359	1523	2442		800	1482	2278
San Francisco	2537	3179	2212	1765	1266	1958	573	403	3097	2269	3036	800		817	2864
Seattle	2730	3043	2052	2122	1373	2348	1188	1150	3389	2626	2900	1482	817		2755
Washington D.C.	657	440	695	1372	1635	1443	2568	2680	1101	1098	229	2278	2864	2755	

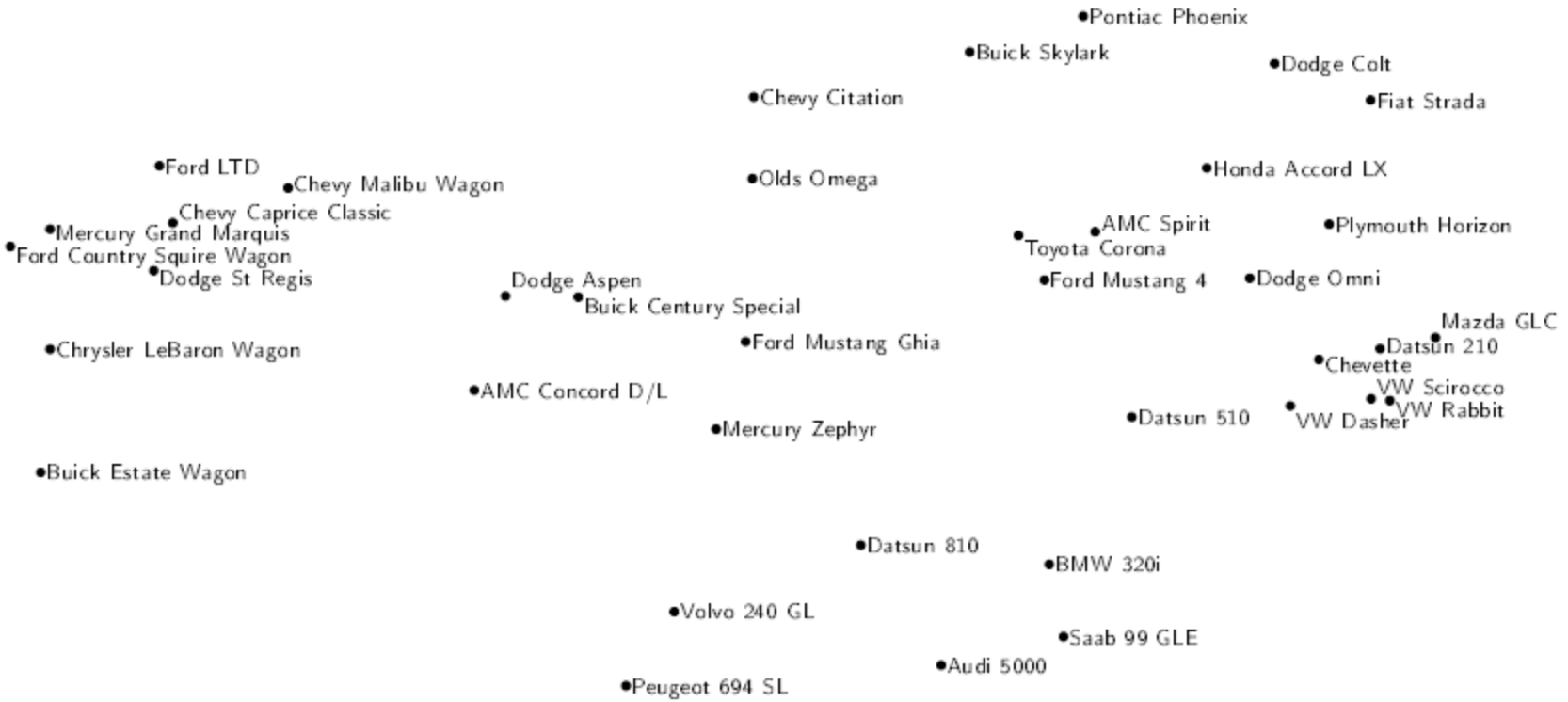



CANADA
New Hampshire
Vermont
Massachusetts
Maine
Augusta
Montpelier
Concord
Boston
Providence
Rhode Island
Connecticut
New Jersey
Delaware
Maryland
Washington, D.C.
West Virginia
Virginia
Raleigh
North Carolina
Columbia
South Carolina
Georgia
Alabama
Montgomery
Tallahassee
Florida
Mississippi
Jackson
Louisiana
Baton Rouge
Austin
Texas
Houston
Arkansas
Little Rock
Oklahoma
Oklahoma City
Nashville
Tennessee
Kentucky
Frankfort
Missouri
Jefferson City
Topeka
Kansas
Springfield
Illinois
Indianapolis
Indiana
Columbus
Ohio
Harrisburg
Pennsylvania
Lansing
Michigan
Madison
Wisconsin
St. Paul
Minnesota
Des Moines
Iowa
Lincoln
Nebraska
Pierre
South Dakota
Bismarck
North Dakota
Helena
Montana
Cheyenne
Wyoming
Salt Lake City
Utah
Carson City
Nevada
Boise
Idaho
Salem
Oregon
Phoenix
Arizona
Santa Fe
New Mexico
California
Alaska
Juneau
Honolulu
Hawaii





Car Example



Basalaj, http://www.pavis.org/essay/pavis.pdf

Car Example

Basalaj, http://www.pavis.org/essay/pavis.pdf

Challenges of dimensionality reduction

- Dimensions of visualizations not directly related to original dimensions
  - May be hard to interpret or explain
- Different slices yield potentially different views
  - Some randomization may be involved

Factor Analysis (a close relation)

Consider a heptathlon participant…

Athlete

100m sprint

800m run

110m hurdles

Long jump

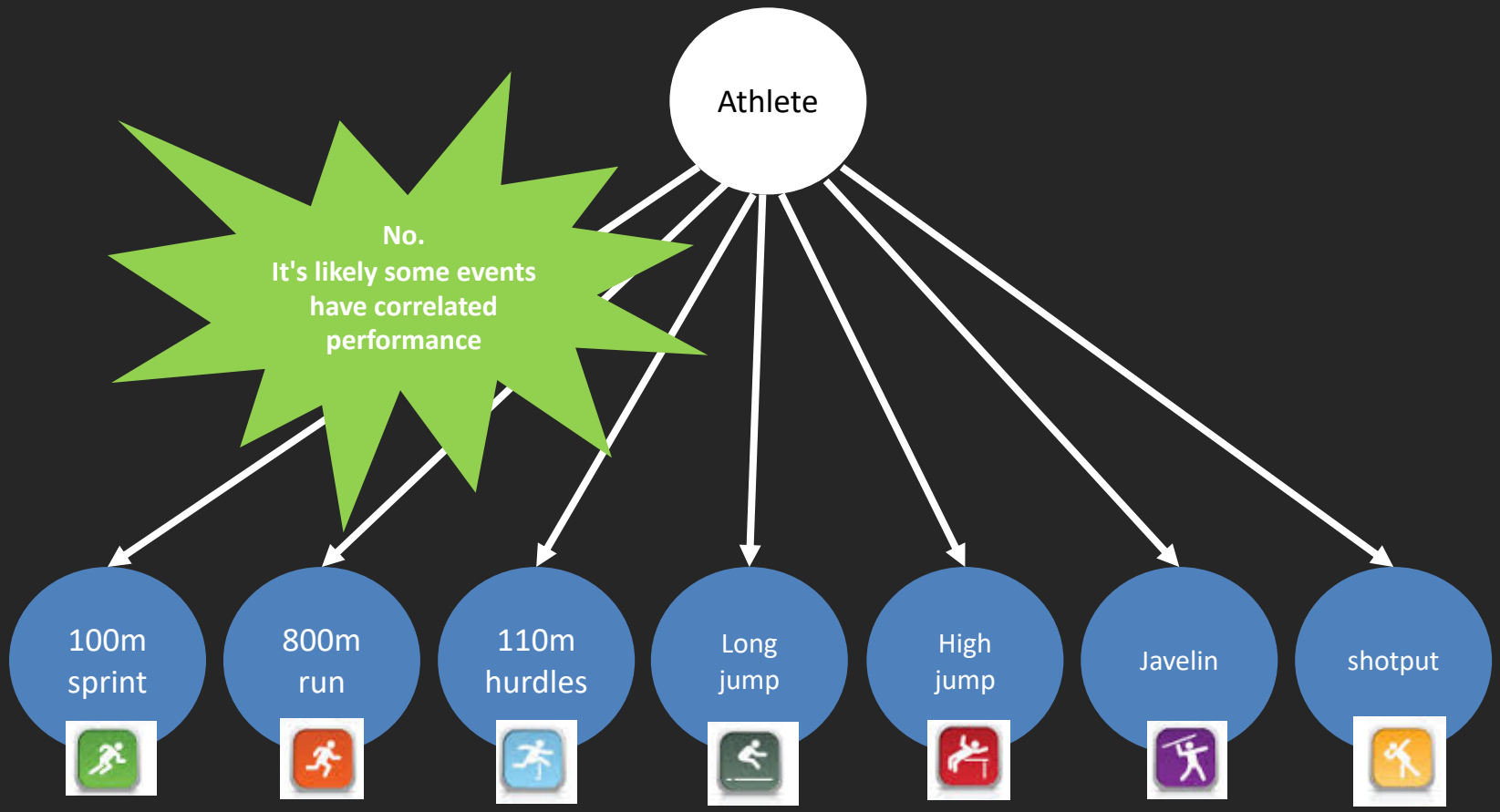
High jump

Javelin

shotput

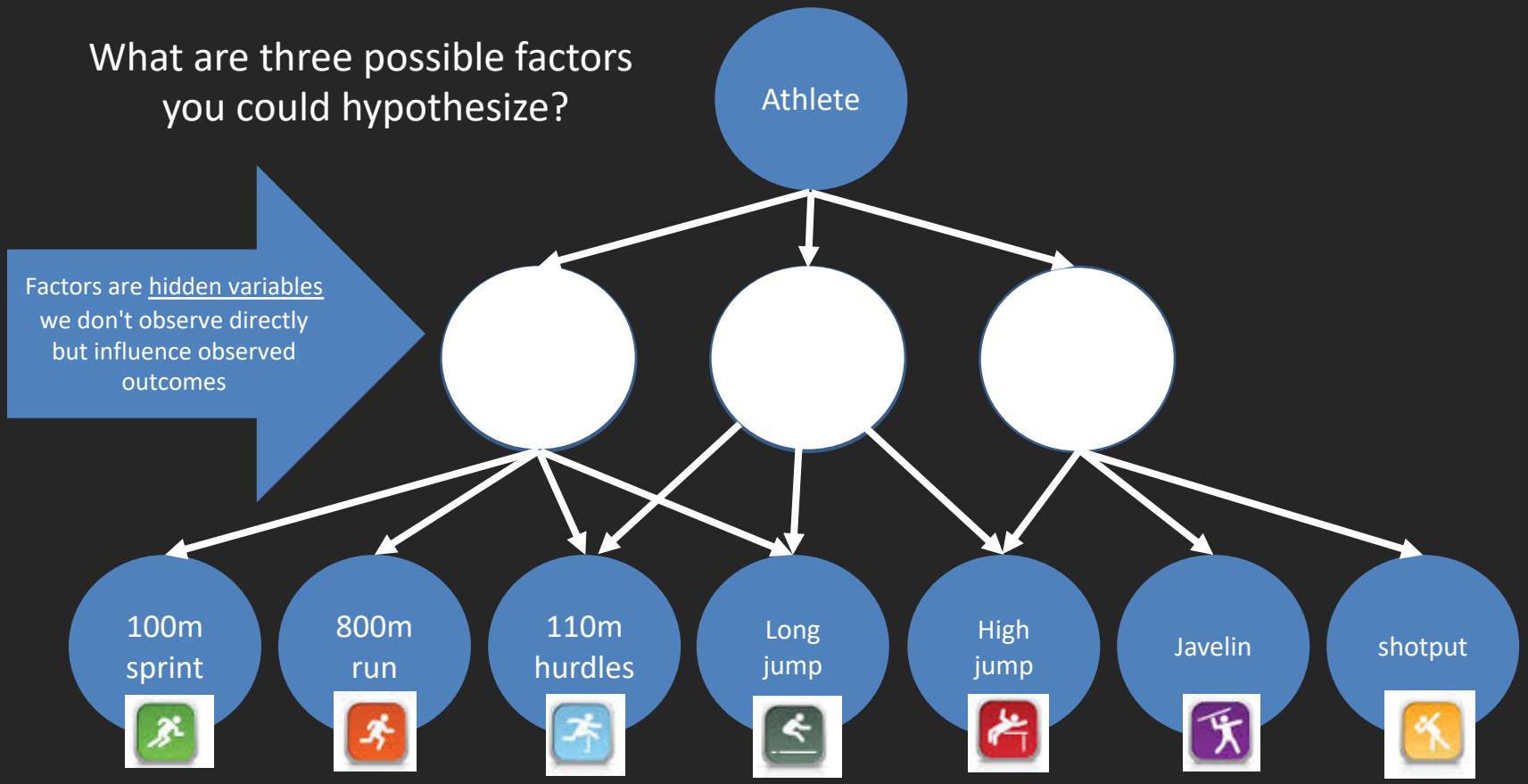
We observe their score in each event = variable

Group question: Are the athlete's scores independent?



Each event score = variable

We might be able to describe an athlete's abilities in terms of a smaller number of factors that strongly influence their scores in all events.



Each event score = variable

Factor analysis

Goal: "explain" many observed variables in terms of a much smaller number of unobserved variables (factors)

Factor analysis

- A form of dimensionality reduction
  - Compress/reduce huge # of variables to essential subset
  - Create interpretable models of observed phenomena in terms of relatively independent factors
  - Create a summary representation of an object
    - E.g. topic vector for a document
  - Address sparsity problems by finding groupings of similar data
    - E.g. find groups of users
  - Find representative samples from a much larger set

Factors and factor loadings

- Factor variables:
  - F_{RUN} : Running ability (-1 to 1)
  - F_{JUMP} : Jumping ability (-1 to 1)
  - F_{THROW} : Throwing ability (-1 to 1)

- Goal: Rewrite our data in terms of a linear combination of the factor variables
- The factor loadings are the weights that relate the *p* observed variables to the *k* factors in a linear model

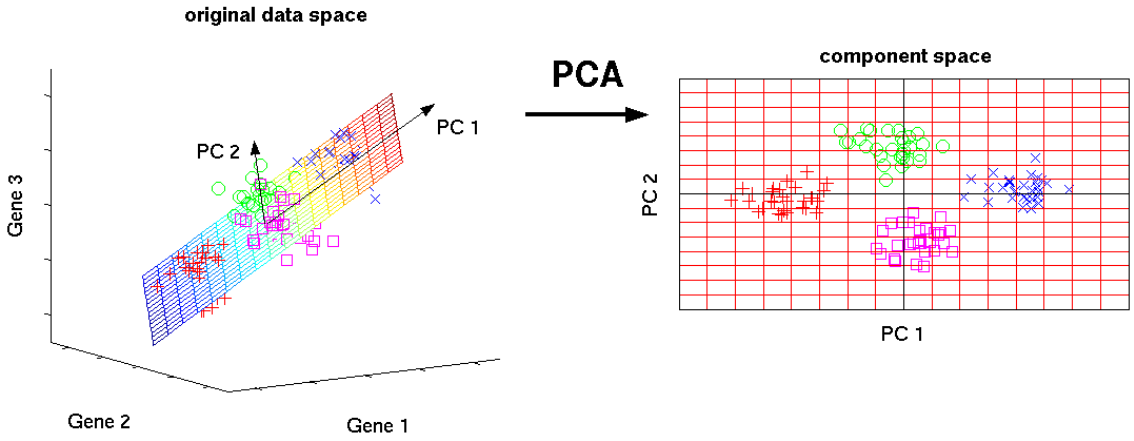
$$\text{Score}(A_E) = w_{RUN}F_{RUN}(A_E) + w_{JUMP}F_{JUMP}(A_E) + w_{THROW}F_{THROW}(A_E)$$

A few types

- Principal Components Analysis (PCA)
  - Projects data to lower dimensions
  - Extracts all variance
- Factor Analysis (FA)
  - Seeks small number of unobserved underlying variables that might explain the common variance (not all variance) in the data.

PCA Intuition

- Project k-dimensional cloud onto a 2-d surface
- Finds the most "informative" projection (in terms of characterizing variation)
- Another view:
  - Fit k-dimensional ellipsoid to the cloud
    - Each axis represents a principal component
    - Axis is "small" = variance along that axis is small
    - Omitting that axis only loses "small" amount of information

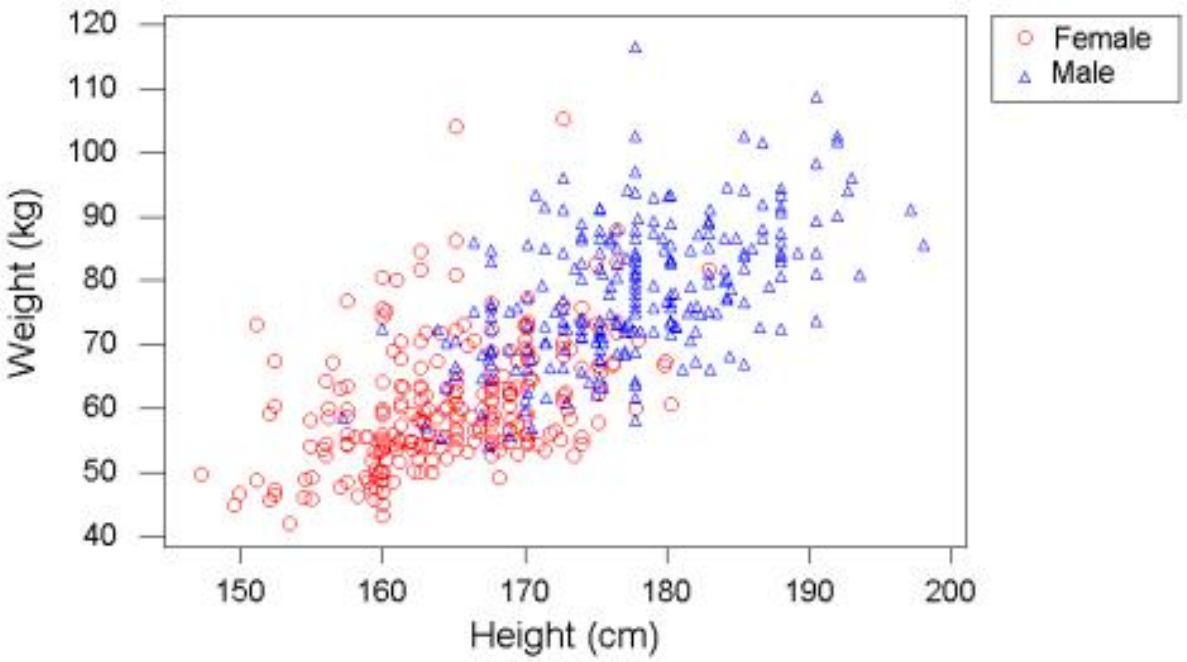


Use of PCA

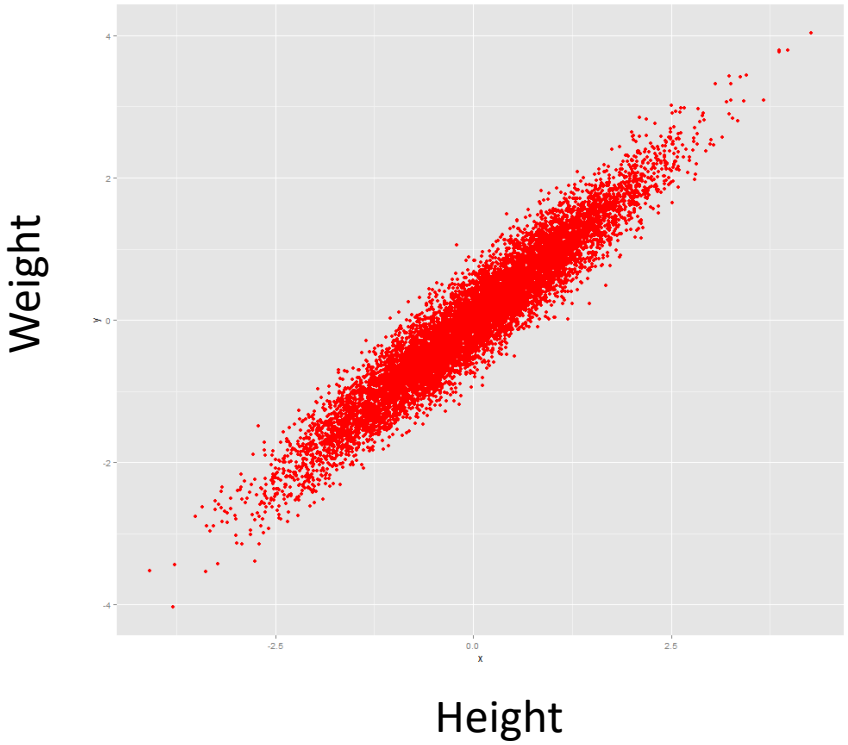
- Identify combinations of variables that explain most of the variation in the data
- Compress high-dimensional datasets to a few dimensions
- Filter noise from data (as a result of the approximation)

Human weight and height have a 2-d distribution (not quite Gaussian but we'll assume they are)

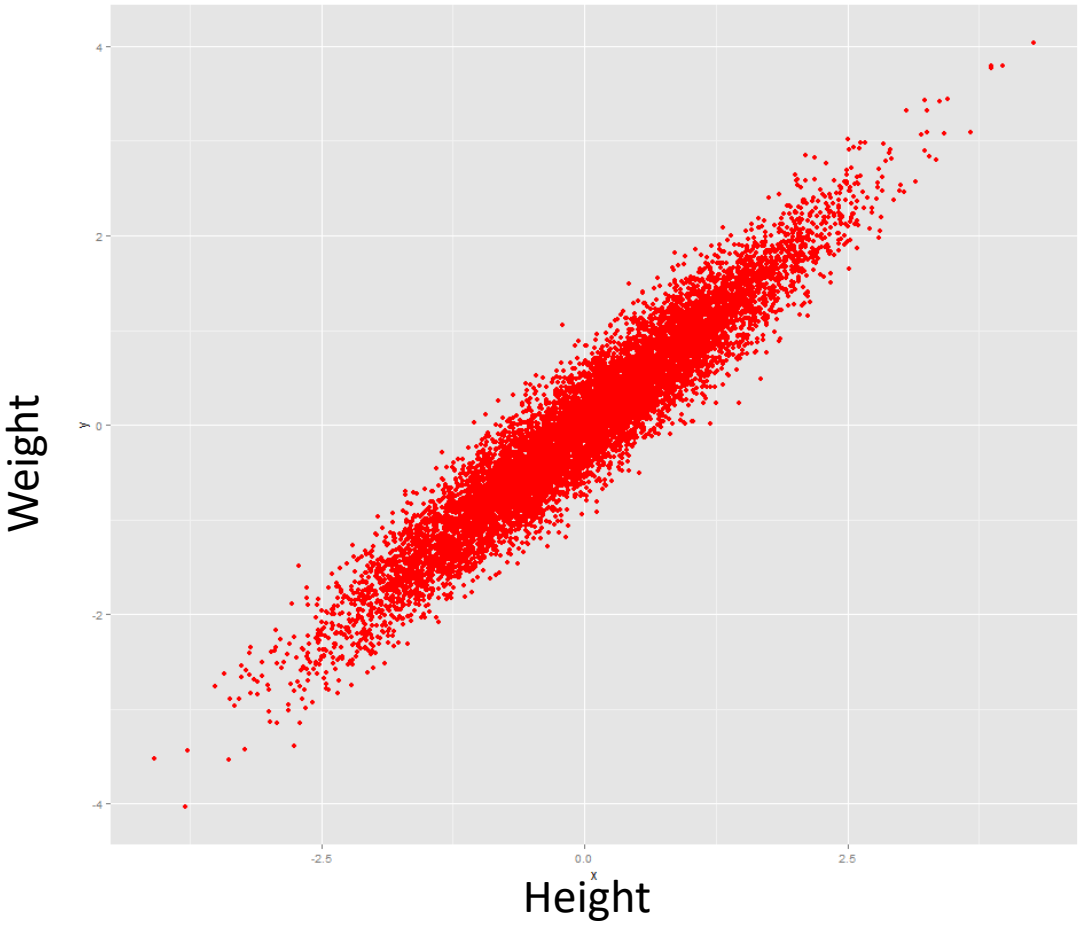
Source: http://www.amstat.org/publications/jse/v11n2/datasets.heinz.html



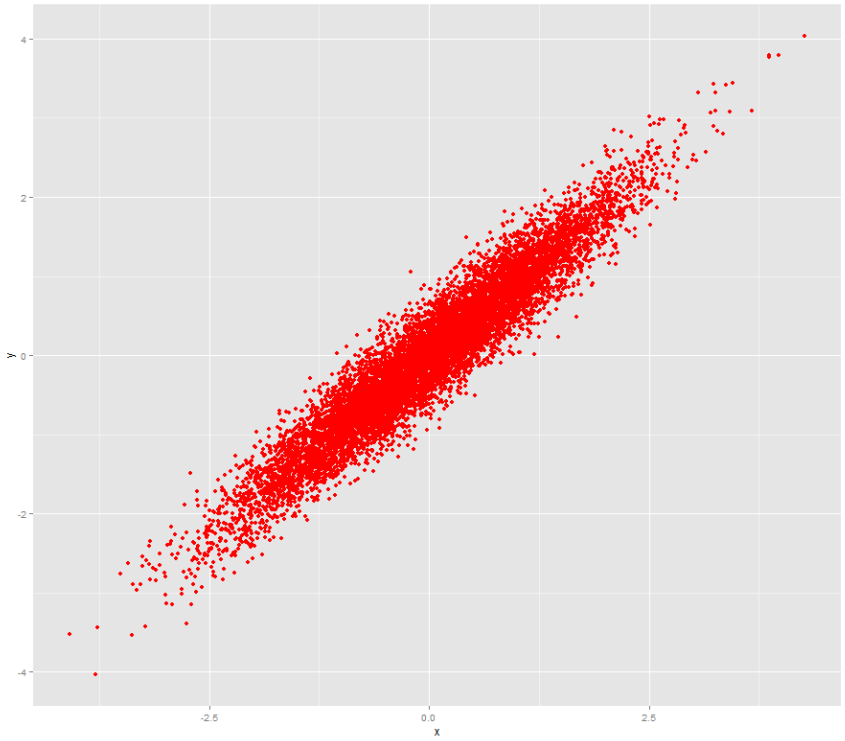
Let’s assume this is our data…



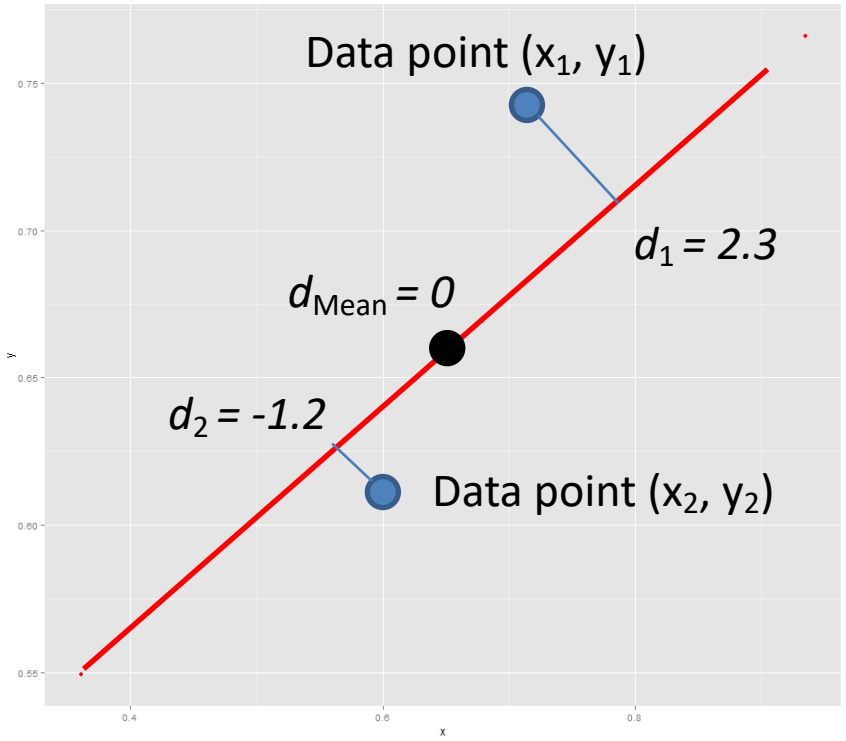
- Each point is 2D (x,y) → needs 2 numbers
- Say you only have space for 1 number
- What number would you store to best approximate a point's position?



Idea: Collapse the cloud to 1 dimension, then approximate a point (x_i, y_i) by its projected position d_i onto the line.

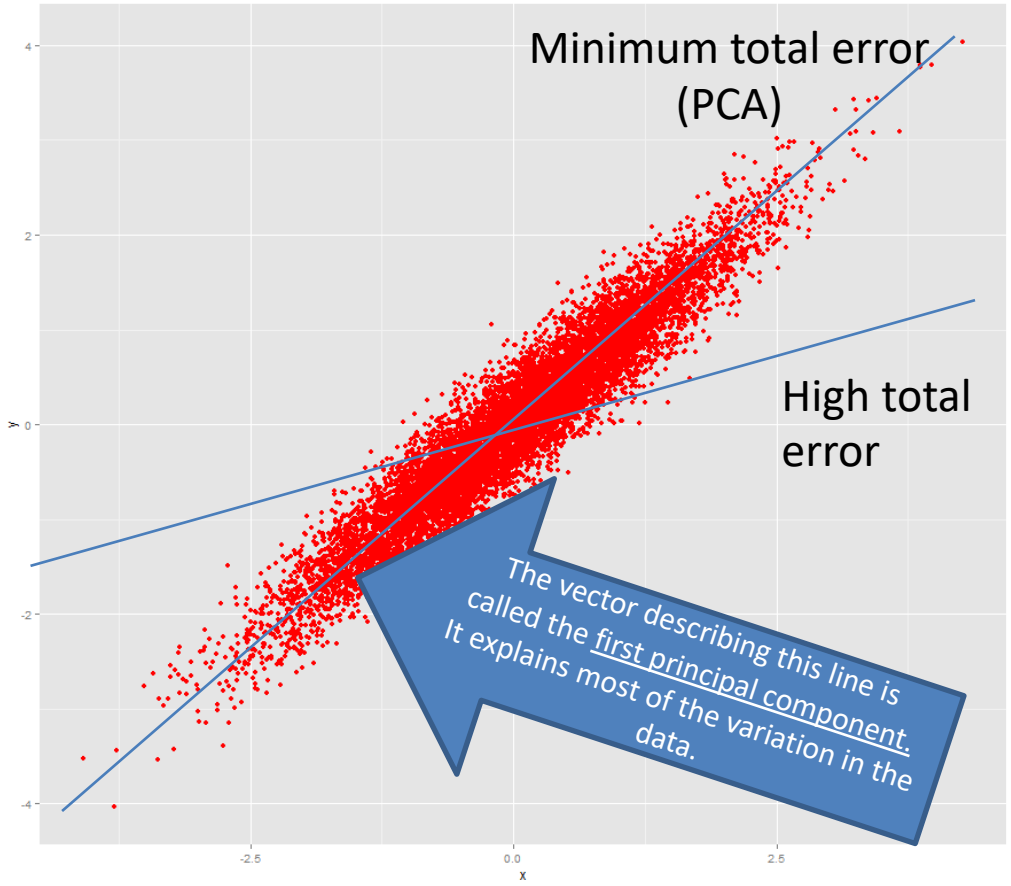


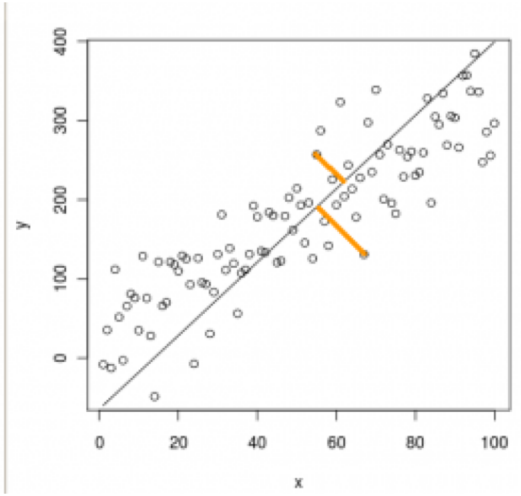
2-dimensional cloud



A 1-dimensional approximation to the cloud (using a linear model)

How do we pick the 'best' linear approximation? PCA!

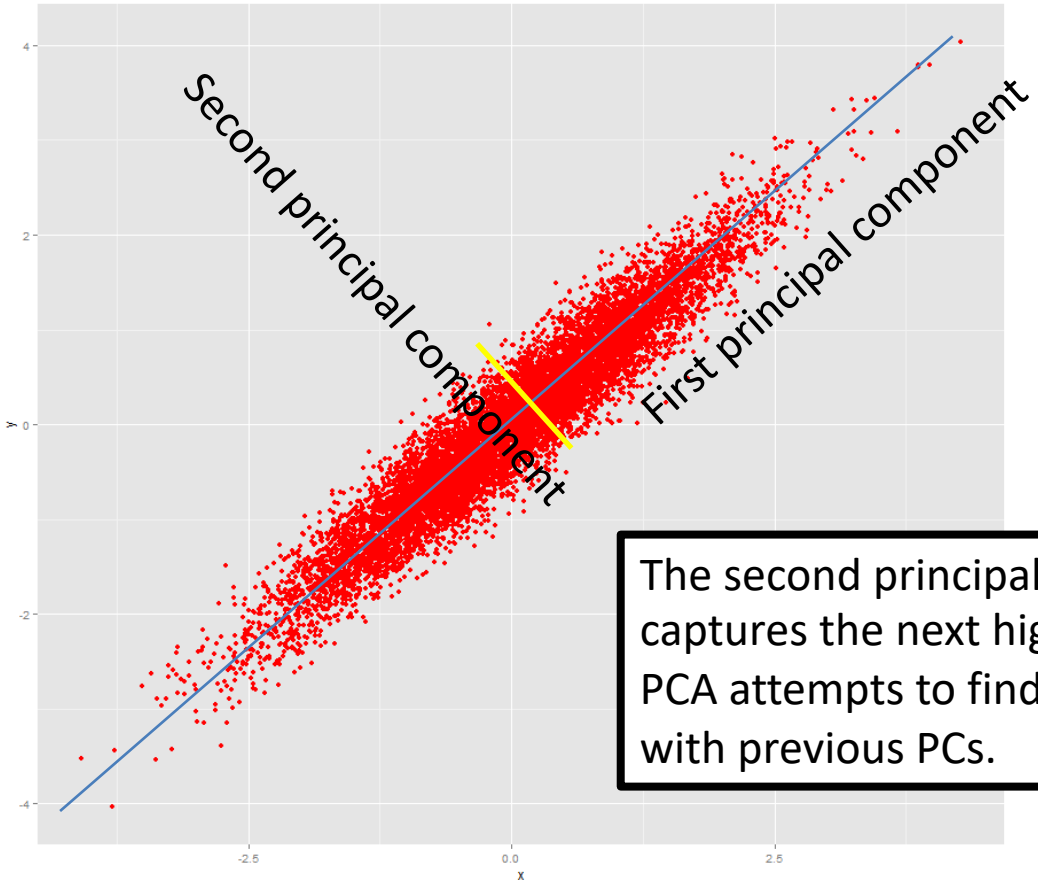


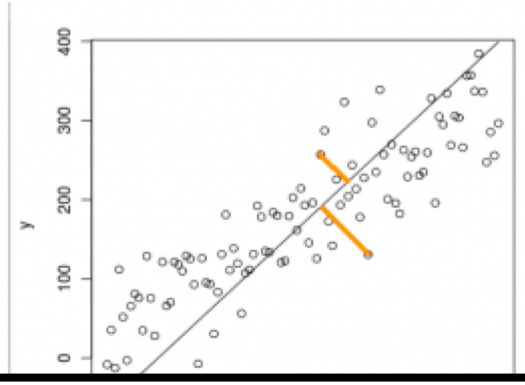


PCA finds the unique model line that minimizes error orthogonal (perpendicular) to the model line.

Source: http://www.cerebralmastication.com/2010/09/principal-component-analysis-pca-vs-ordinary-least-squares-ols-a-visual-explination/

We can repeat this process to improve the approximation.
This will give us the second principal component.





The second principal component (yellow) captures the next highest orthogonal direction of variance. PCA attempts to find the next PC that is uncorrelated with previous PCs.

The principal component vectors have an origin that is the mean (centroid) of the cloud

Source: http://www.cerebralmastication.com/2010/09/principal-component-analysis-pca-vs-ordinary-least-squares-ols-a-visual-explination/

Heptathlon dataset
Scores in 7 events: Seoul 1988 Olympics

Athlete	Hurdles	HighJump	Shot	Run200m	LongJump	Javelin	Run800m	Score
Joyner-Kersee (USA)	12.69	1.86	15.80	22.56	7.27	45.66	128.51	7291
John (GDR)	12.85	1.80	16.23	23.65	6.71	42.56	126.12	6897
Behmer (GDR)	13.20	1.83	14.20	23.10	6.68	44.54	124.20	6858
Sablovskaite (URS)	13.61	1.80	15.23	23.92	6.25	42.78	132.24	6540
Choubenkova (URS)	13.51	1.74	14.76	23.93	6.32	47.46	127.90	6540
Schulz (GDR)	13.75	1.83	13.50	24.65	6.33	42.82	125.79	6411
Fleming (AUS)	13.38	1.80	12.88	23.59	6.37	40.28	132.54	6351
Greiner (USA)	13.55	1.80	14.13	24.48	6.47	38.00	133.65	6297
Lajbnerova (CZE)	13.63	1.83	14.28	24.86	6.11	42.20	136.05	6252
Bouraga (URS)	13.25	1.77	12.62	23.59	6.28	39.06	134.74	6252
Wijnsma (HOL)	13.75	1.86	13.01	25.03	6.34	37.86	131.49	6205
Dimitrova (BUL)	13.24	1.80	12.88	23.59	6.37	40.28	132.54	6171
Scheider (SWI)	13.85	1.86	11.58	24.87	6.05	47.50	134.93	6137
Braun (FRG)	13.71	1.83	13.16	24.78	6.12	44.58	142.82	6109

25 observations (athletes), 8 variables

Step 1: Transform the data

- Seven events have very different variances
  - Standard deviation for the 800m is 8.29 (sec)
  - For high jump it's only 0.078 (m)

  Group question: What happens if we work with non-normalized data?

Step 1: Transform the data

- Seven events have very different variances
  - Standard deviation for the 800m is 8.29 (sec)
  - For high jump it's only 0.078 (m)
  - If unscaled scores, the 800m results will have a disproportionate effect.
- Also: Some results are measured in seconds (lower numbers better), others in scores, or meters (higher numbers better)
- Normalization is important
- Clean it up!

Scaling is important for PCA

- PCA is sensitive to the scaling of the variables
  - (k-means and other algorithms are as well)
- Variables in different units *must* be scaled (e.g. temperature vs mass).
- Variables in the same units but having wildly different variances *should* be scaled.

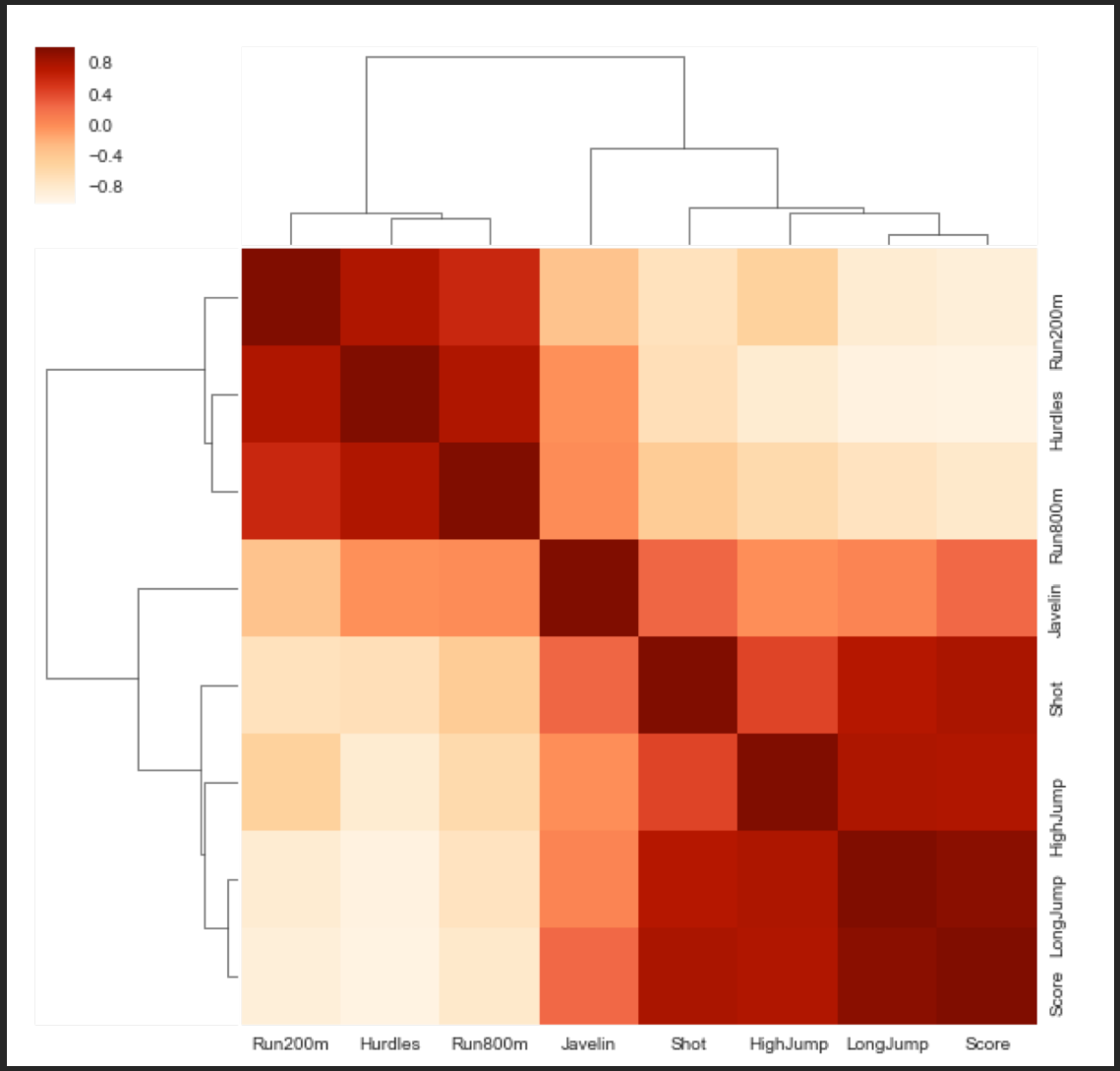
Scaling is important for PCA

- Typical pre-processing steps for PCA:
  - Mean subtraction (a.k.a. "mean centering")
    - Ensures that the first principal component describes the direction of maximum variance
    - Subtract mean from every point
  - Variable scaling
    - Divide each variable by its standard deviation
  - Do both
    - “z-score”

Step 2: Check for correlations

	Hurdles	HighJump	Shot	Run200m	LongJump	Javelin	Run800m	Score
Hurdles	1.000000	-0.811403	-0.651335	0.773721	-0.912134	-0.007763	0.779257	-0.923198
HighJump	-0.811403	1.000000	0.440786	-0.487664	0.782442	0.002153	-0.591163	0.767359
Shot	-0.651335	0.440786	1.000000	-0.682670	0.743073	0.268989	-0.419620	0.799699
Run200m	0.773721	-0.487664	-0.682670	1.000000	-0.817205	-0.333043	0.616810	-0.864883
LongJump	-0.912134	0.782442	0.743073	-0.817205	1.000000	0.067108	-0.699511	0.950437
Javelin	-0.007763	0.002153	0.268989	-0.333043	0.067108	1.000000	0.020049	0.253147
Run800m	0.779257	-0.591163	-0.419620	0.616810	-0.699511	0.020049	1.000000	-0.772776
Score	-0.923198	0.767359	0.799699	-0.864883	0.950437	0.253147	-0.772776	1.000000

Can you see any structure in the correlation matrix? Are there groups of related events (variables)?

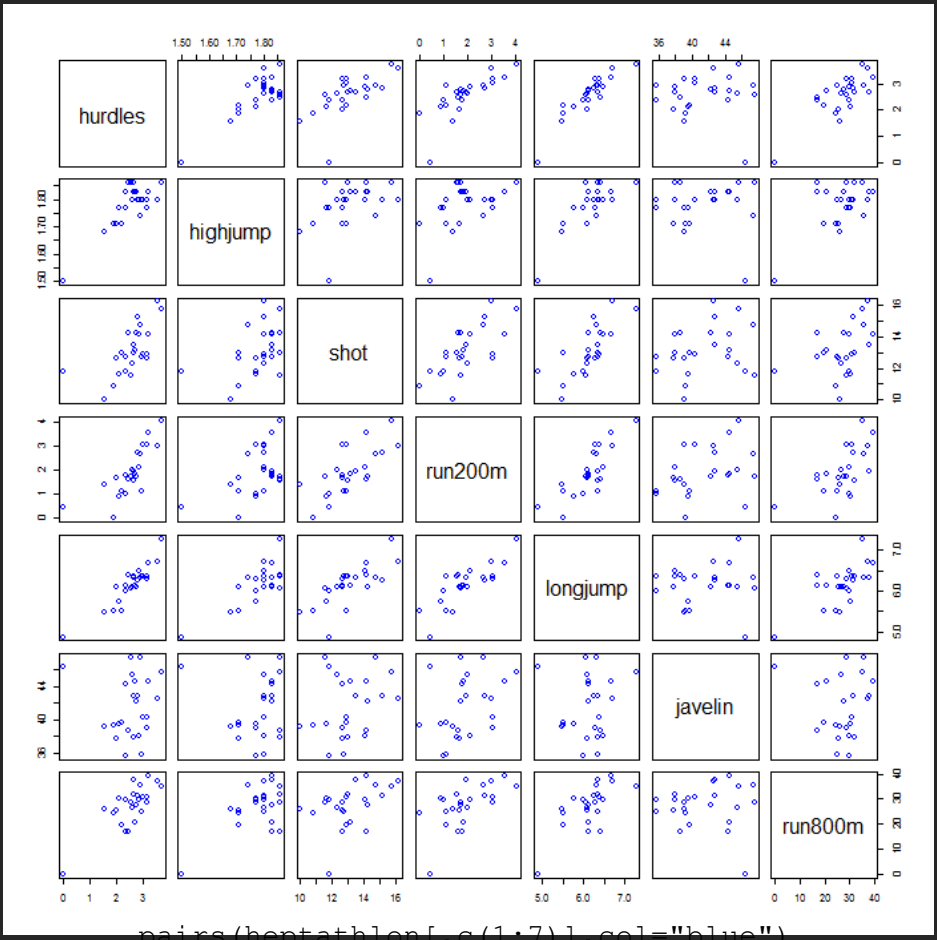


Dark = high correlation
Light = low correlation

Could we find a small set of core athletic abilities (factors) that mostly explain the structure of correlated results for a large set of event scores (variables)?

- Is there a "running factor" that would lead to good results across multiple running events?
- A "jumping factor" ?
- A "throwing factor" (arm strength) ?
- Endurance?
- Seven variables is not a large number
  - PCA comes into its own in larger data sets

Scatterplots confirm what we observed in the correlation matrix




```
pairs(heptathlon[,c(1:7)],col="blue")
```

Observations?

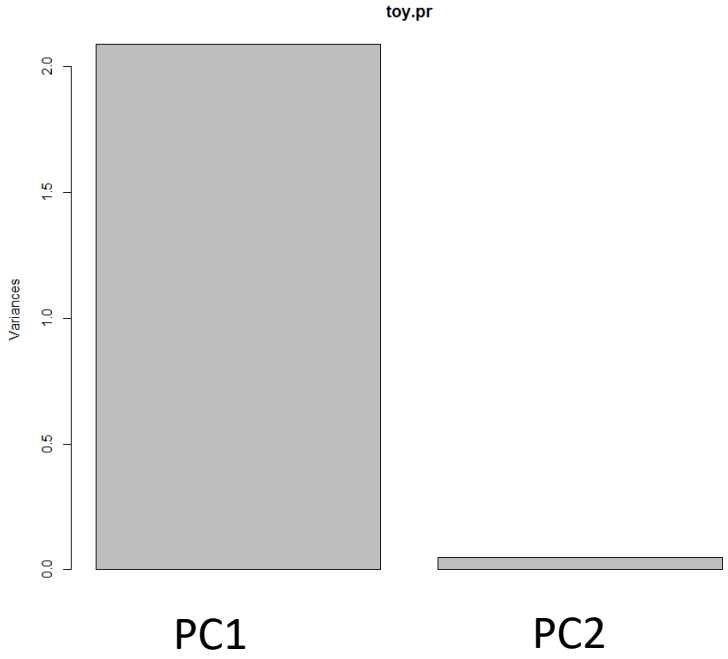
1. Clear linear relationships between hurdles, high jump, shot put, 200m, and long jump.

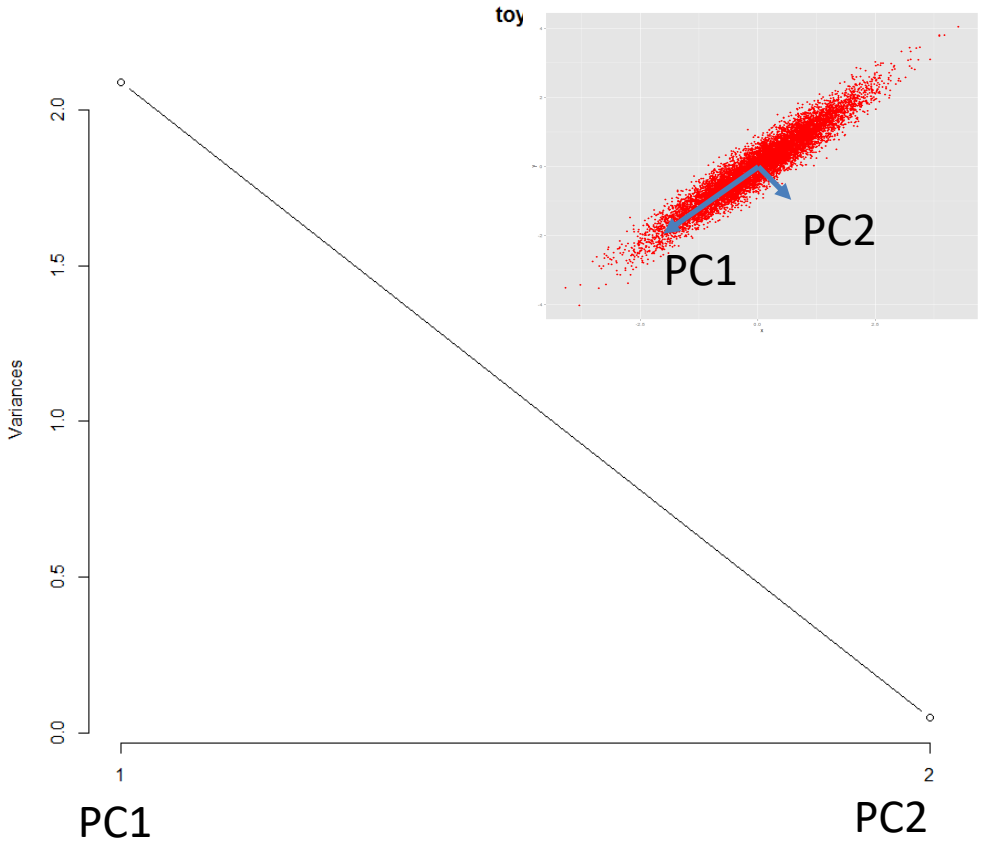
2. Javelin and, to some extent, 800m results are less correlated with the other events.

Step 3: Run PCA

- (we'll do this in a few minutes)

Screeplots show how much variance is explained by each principal component

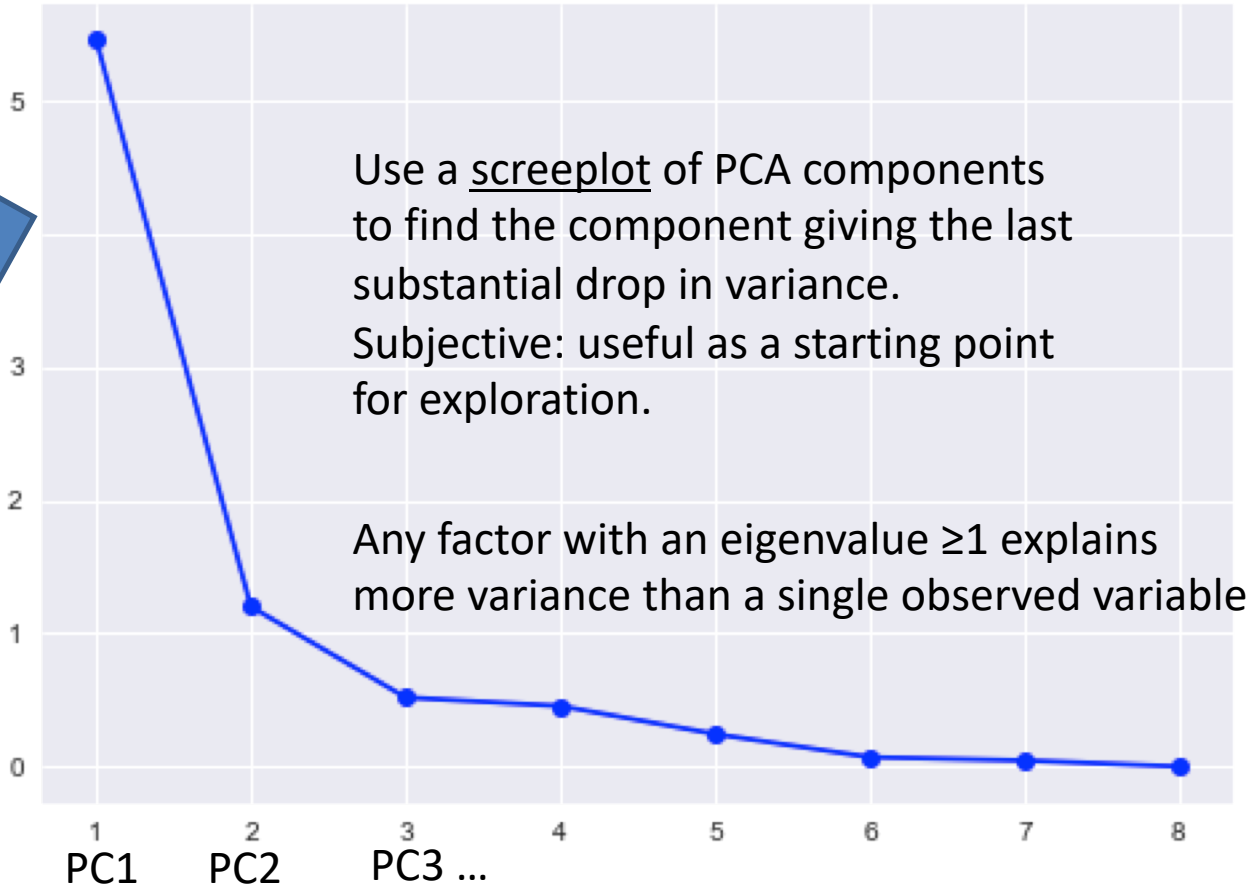




How to pick the "right" number of factors?

Suggests the first PC captures most of the interesting variation in the data.

Implies that a single factor (skill?) may "explain" performance across multiple events



Use a screeplot of PCA components to find the component giving the last substantial drop in variance.
Subjective: useful as a starting point for exploration.

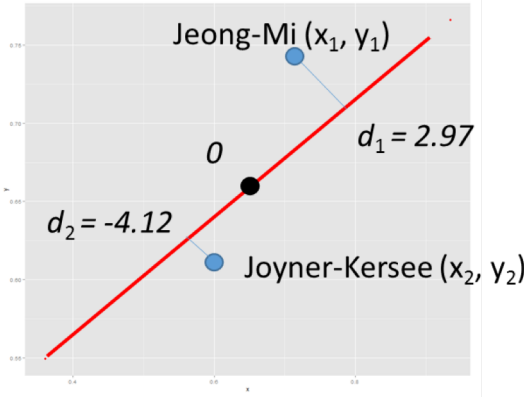
Any factor with an eigenvalue ≥1 explains more variance than a single observed variable.

Can test this: Project each athlete onto PC1

Compare to ranking based on original data:

	hurdles	highjump	shot	run200m	longjump	javelin	run800m	score
Joyner-Kersee (USA)	3.73	1.86	15.80	4.05	7.27	45.66	34.92	7291
John (GDR)	3.57	1.80	16.23	2.96	6.71	42.56	37.31	6897
Behmer (GDR)	3.22	1.83	14.20	3.51	6.68	44.54	39.23	6858
Sablovskaite (URS)	2.81	1.80	15.23	2.69	6.25	42.78	31.19	6540
Choubenkova (URS)	2.91	1.74	14.76	2.68	6.32	47.46	35.53	6540
Schulz (GDR)	2.67	1.83	13.50	1.96	6.33	42.82	37.64	6411
Fleming (AUS)	3.04	1.80	12.88	3.02	6.37	40.28	30.89	6351
Greiner (USA)	2.87	1.80	14.13	2.13	6.47	38.00	29.78	6297
Lajbnerova (CZE)	2.79	1.83	14.28	1.75	6.11	42.20	27.38	6252
Bouraga (URS)	3.17	1.77	12.62	3.02	6.28	39.06	28.69	6252
Wijnsma (HOL)	2.67	1.86	13.01	1.58	6.34	37.86	31.94	6205
Dimitrova (BUL)	3.18	1.80	12.88	3.02	6.37	40.28	30.89	6171
Scheider (SWI)	2.57	1.86	11.58	1.74	6.05	47.50	28.50	6137
Braun (FRG)	2.71	1.83	13.16	1.83	6.12	44.58	20.61	6109

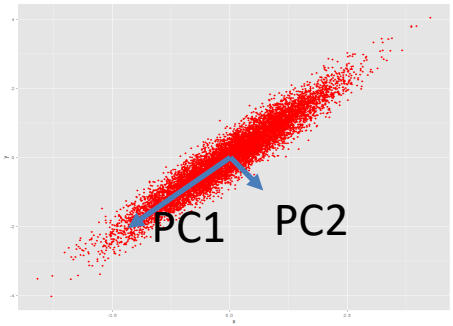
The PCA-based score captures the actual score ranking well.

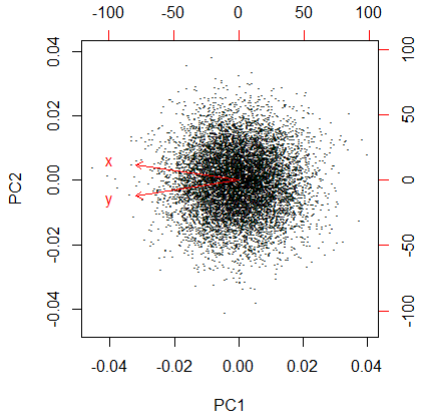


Visualizing PCA results with biplots

- A biplot shows:
  - The data points (top/right scale)
  - The variables (bottom/left scale)

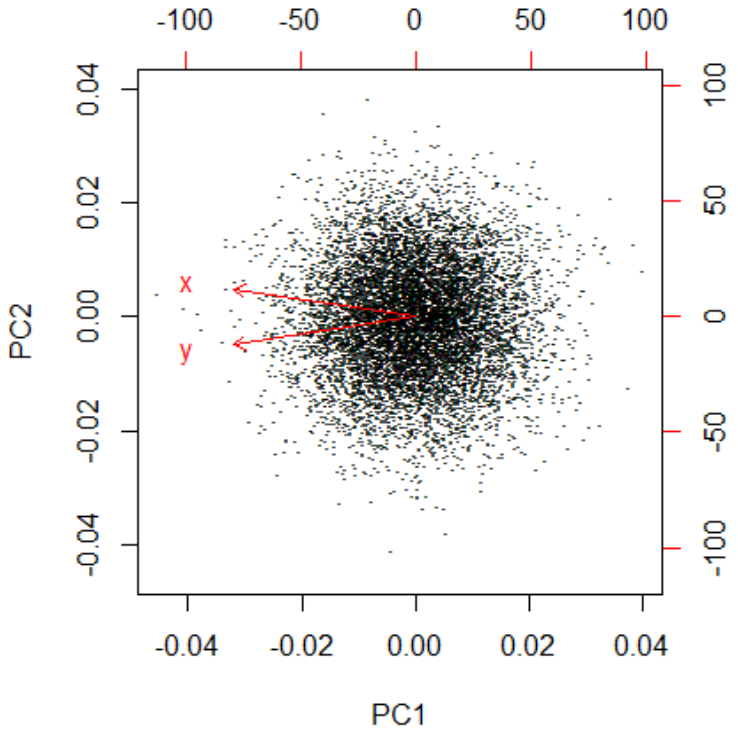
- Data are plotted as "seen" from the PC axes
- Variables are plotted by their PC loadings
  - The angle formed by the vectors for any two variables reflects their actual pairwise correlation



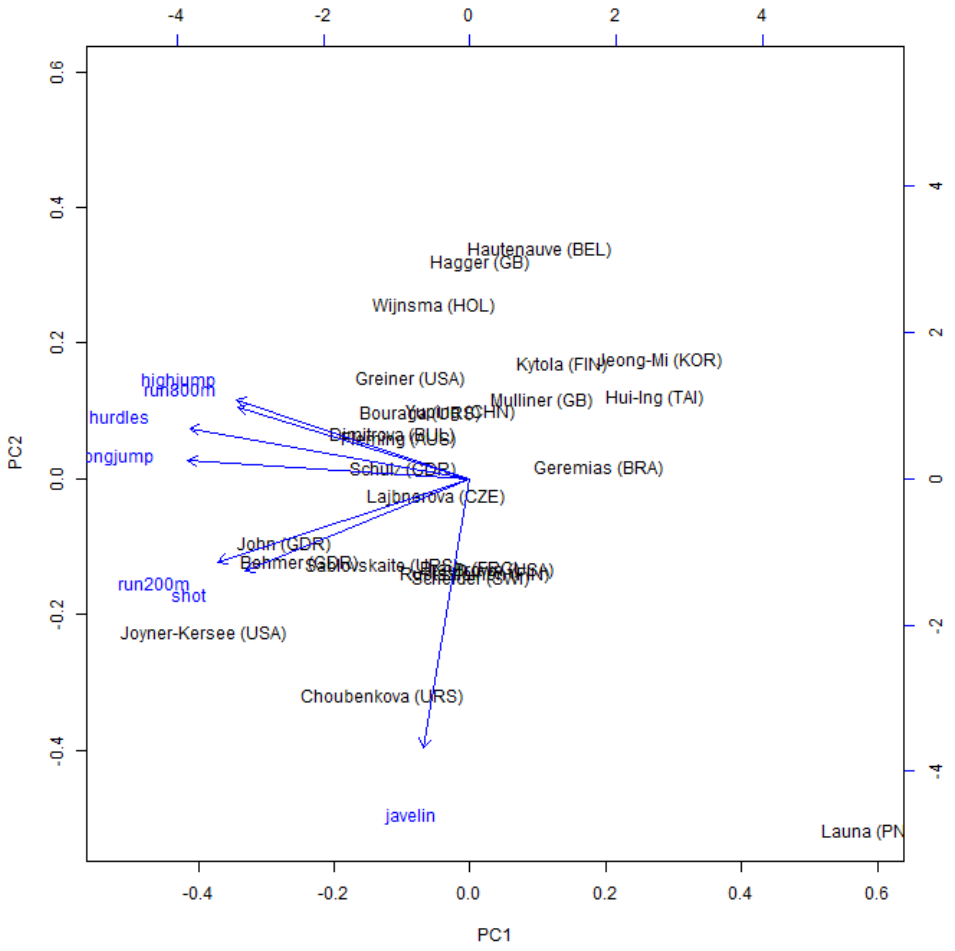


Biplots

- Points close = Observations with similar component projections
- Vectors close = Variables that are correlated
- Observations whose points project furthest in the direction of a variable have the most of whatever the variable measures.



Biplot of heptathlon PCA components

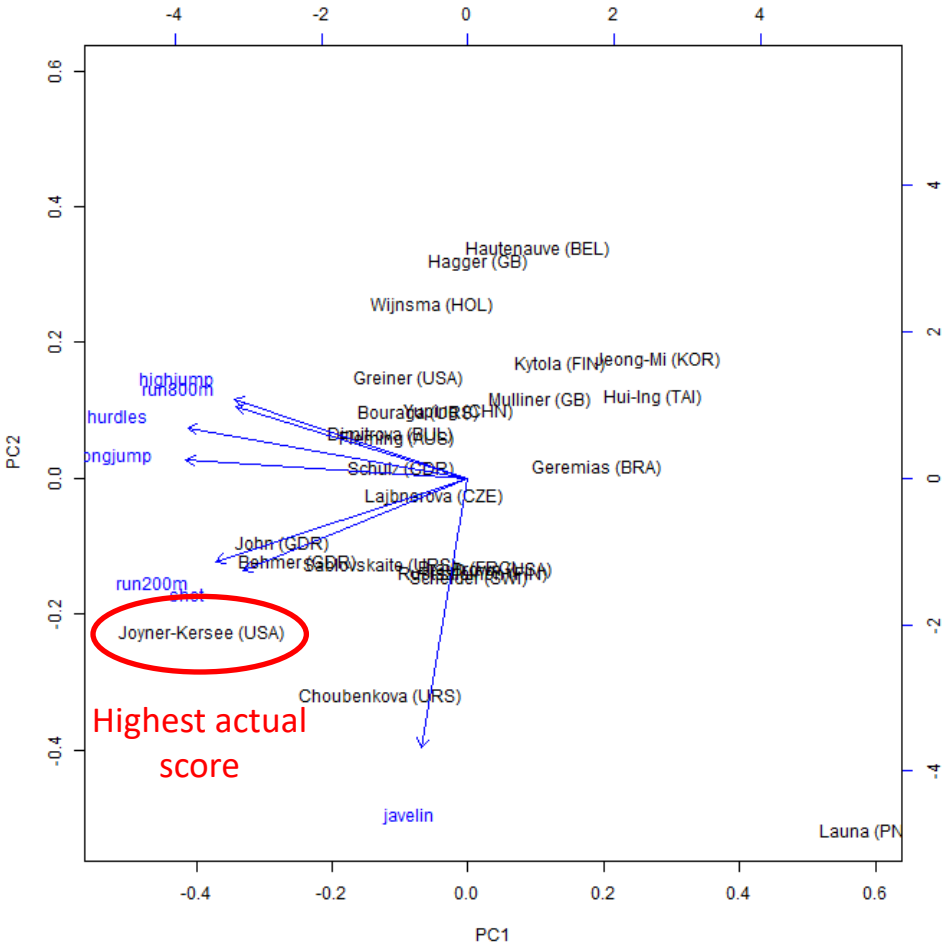


Remember:
For any pair of variables, the angle between their biplot vectors reflects their actual correlation

Athletes good at one event are likely to be good at similar events

Interpreting heptathlon PCA results

- Can reverse:
  - Find location in 7-d space given score on PC1
  - If we know PC2 could make a better guess
  - Having all seven components, we could completely reconstruct their scores
    - But there isn't much point in this since we had the scores already!



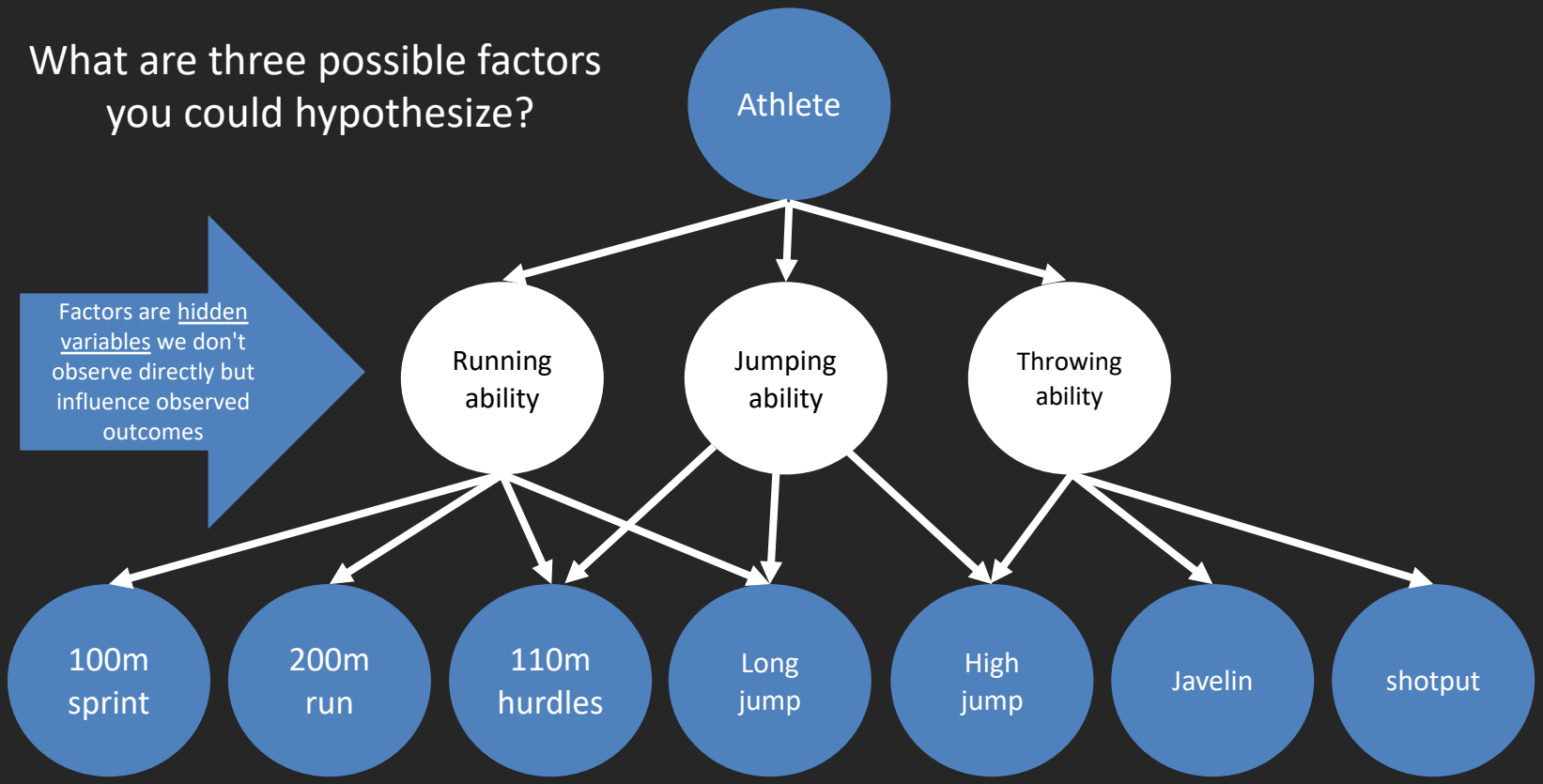
Highest actual score

Factor Analysis

- PCA tries to map *n* dimensions to *m*
- Factor Analysis
  - assumes *m* important dimensions
  - when linearly combined (with noise) produce our *n* dimensions
  - Orthogonality assumptions go away

Factor Analysis

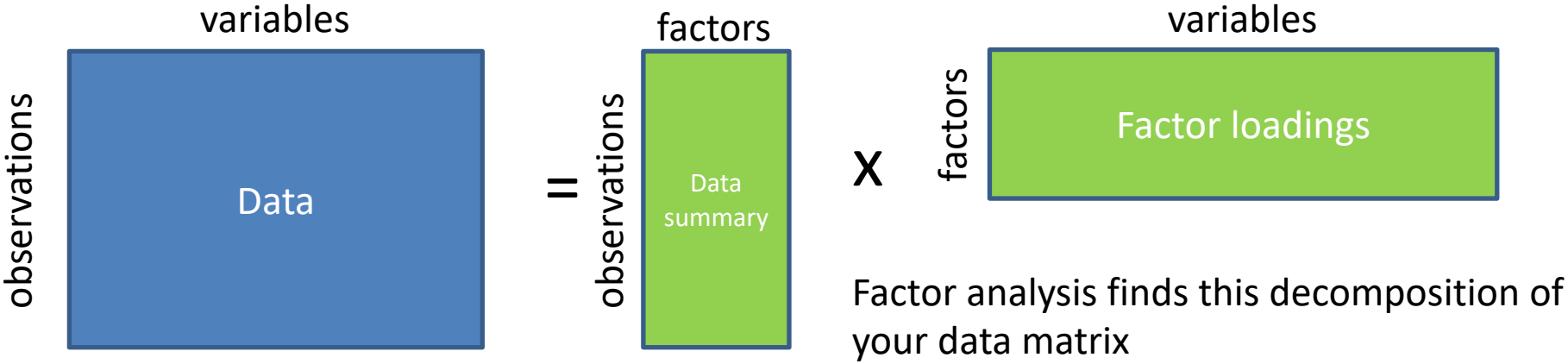
What are three possible factors you could hypothesize?



Each event score = variable

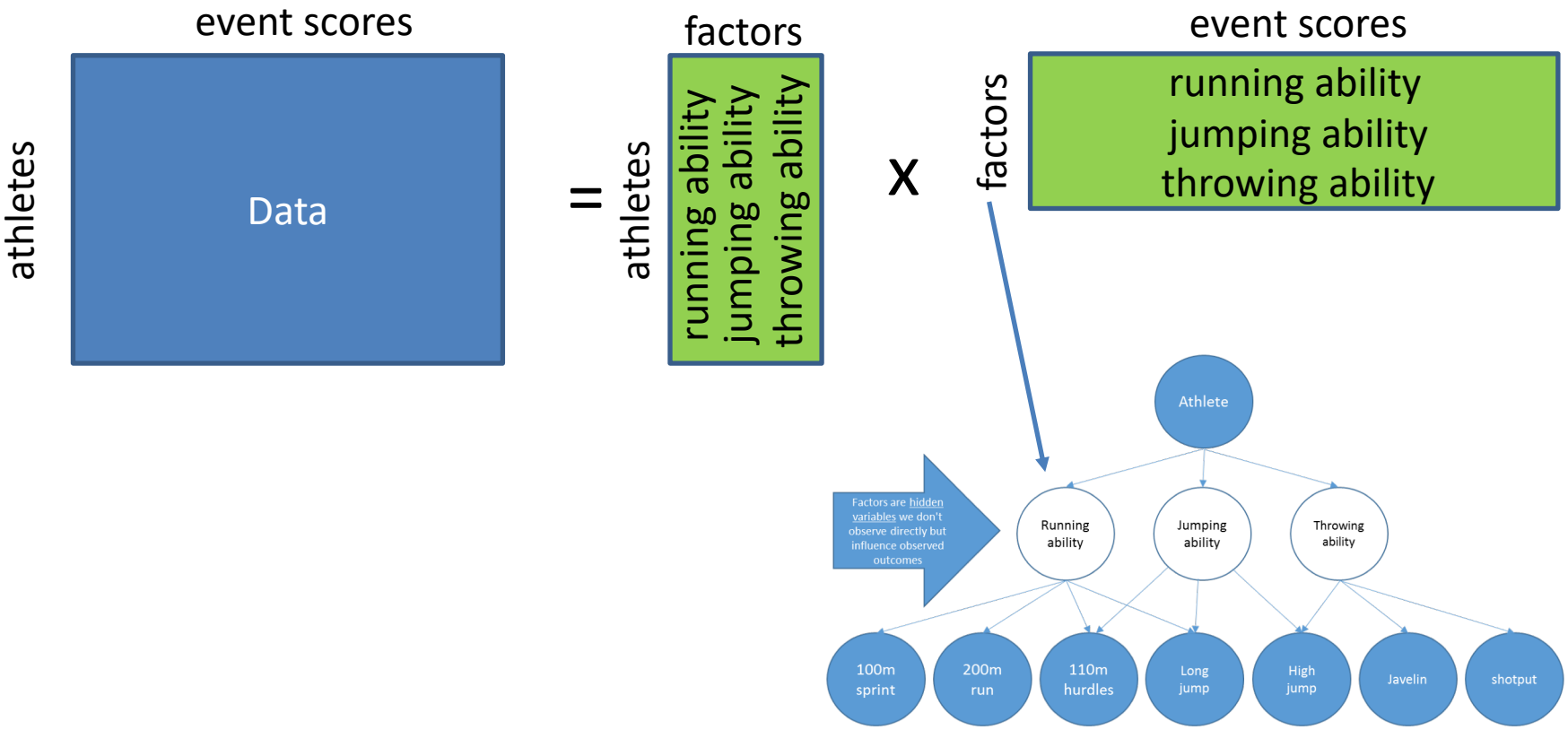
Factor analysis

- Finds *k* factors that "explain" the correlation structure in the observed variables
- Traditionally specify *k* in advance
- You can specify whether the factors can be correlated or uncorrelated
- More flexible and general than PCA



Factor analysis finds this decomposition of your data matrix

Factors in the hepthalon



Summary

- Evaluating clustering
- Dimensionality reduction
  - When we have similarity (or weird spaces)
    - MDS
  - When we can assume orthogonal components
    - PCA
  - When we can assume features of latent variables
    - Factor Analysis