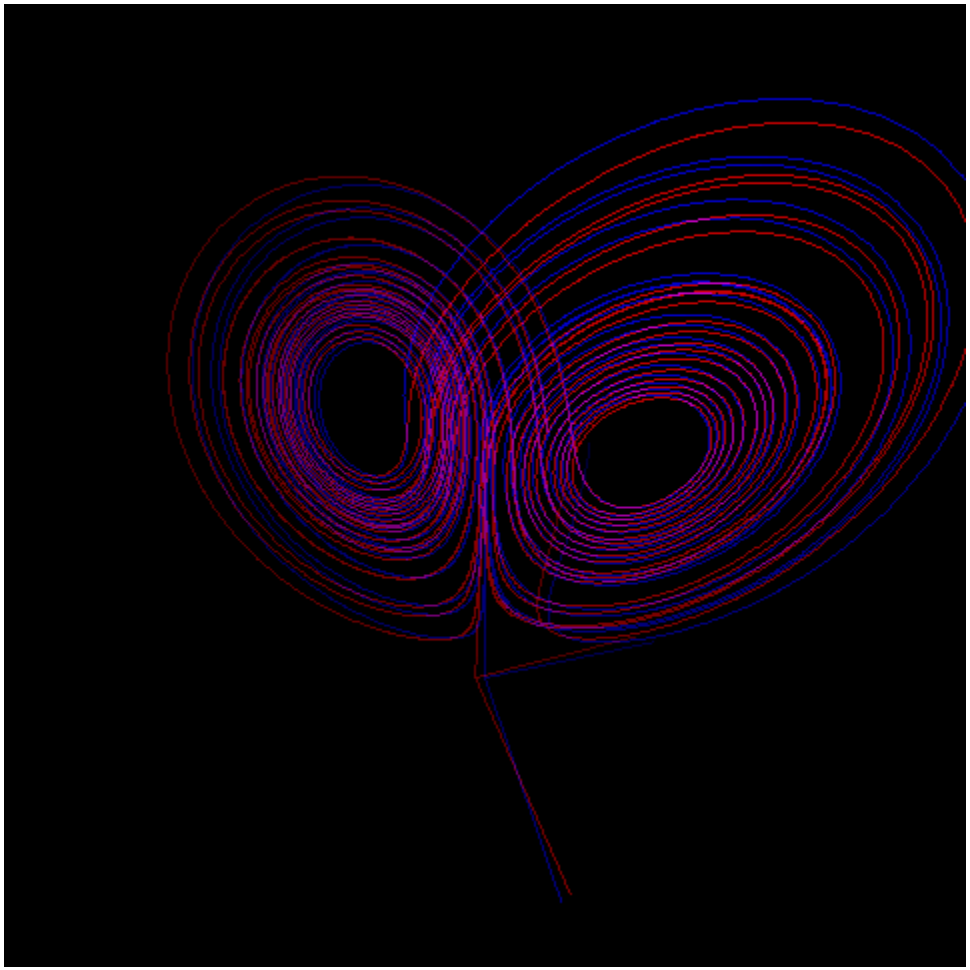
CURSE OF DIMENSIONALITY:

Why do dimension reduction?

DIMENSIONALITY REDUCTION

- **"Curse of dimensionality"**



Lorenz system by Edward Lorenz

"Chaos: When the present determines the future, but the approximate present does not approximately determine the future".

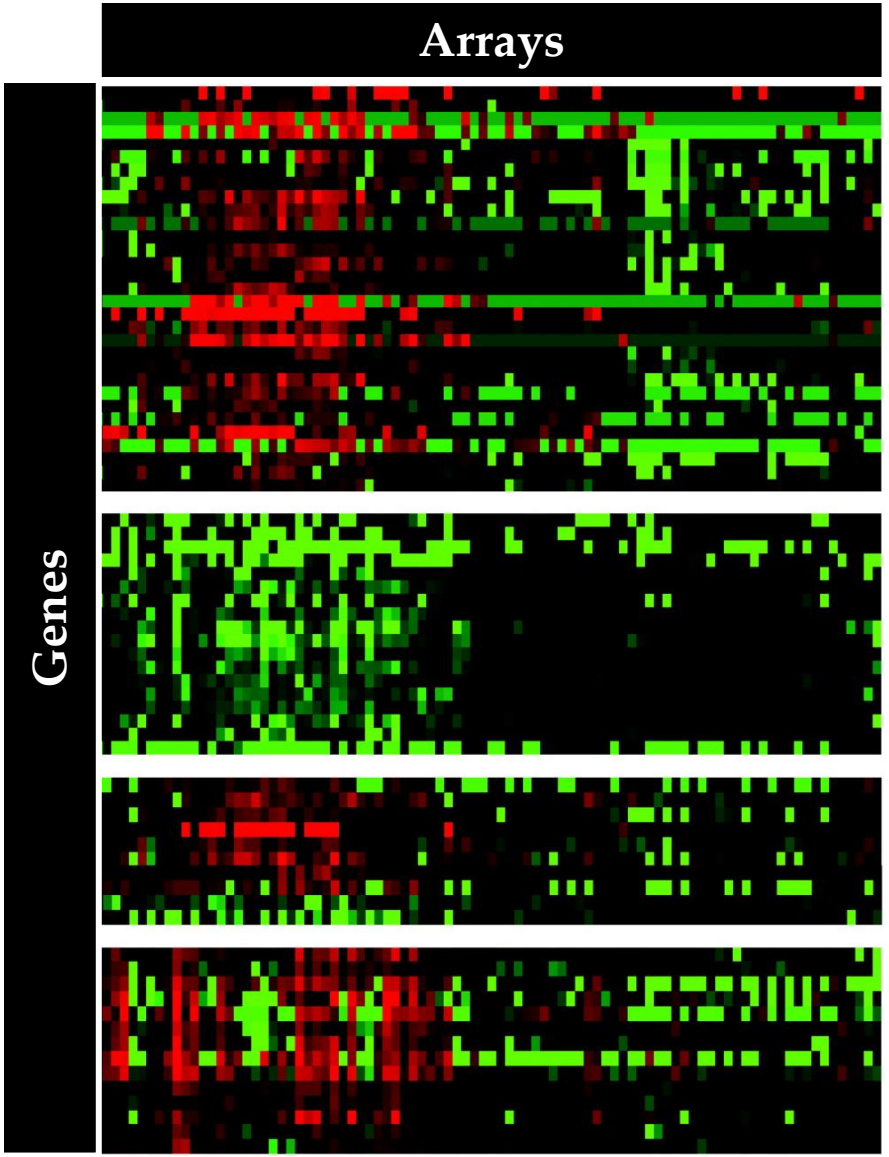
Danforth, Christopher M. (April 2013). "*Chaos in an Atmosphere Hanging on a Wall*"

WHAT ARE THEY

- $X_{n\times p}$ **with large p, small n**
  - Gene expression with
    ~10^4 genes/features
    ~ 10^3 samples

We could use BOOTSTRAP

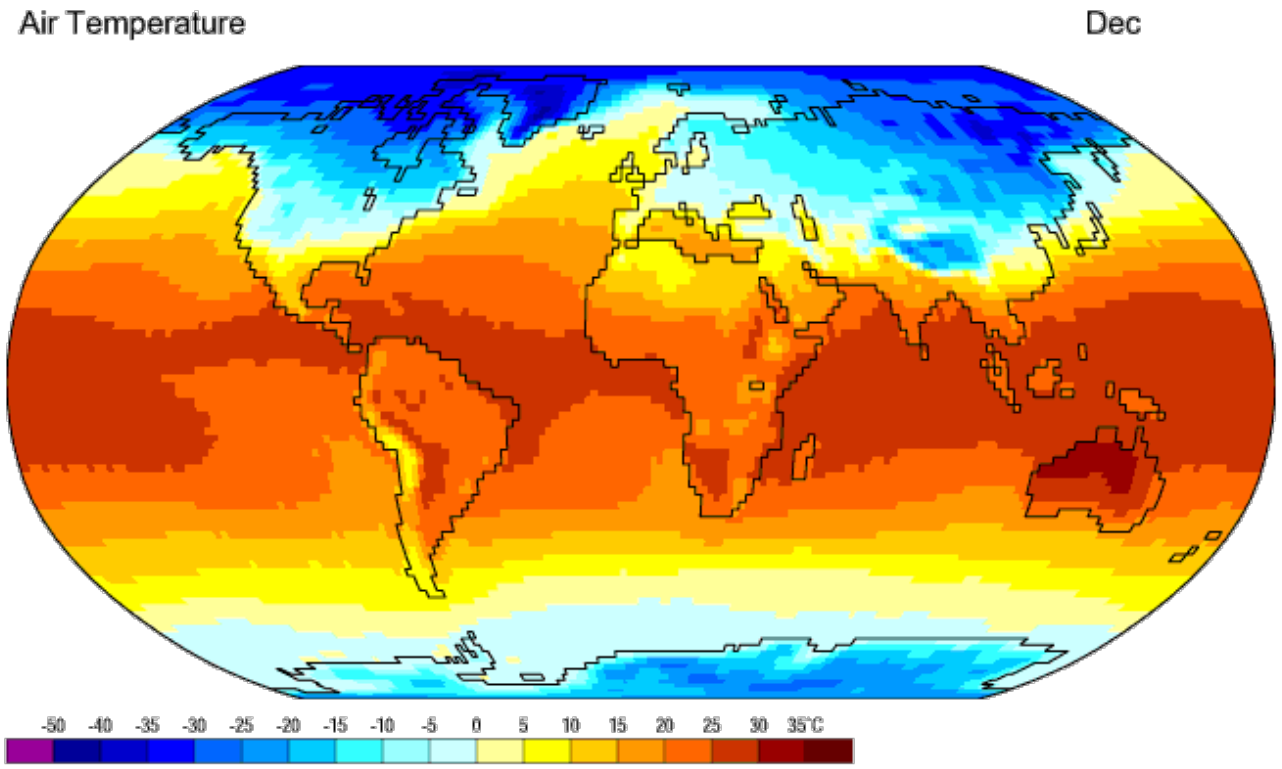
考



WHAT ARE THEY

This looks like a PDE

- $X_{n\times p}$ **with large p, small n**
  - Global Climate Data (NCEP/NCAR Reanalysis 1959-1997)



DATA IS REORGANIZED

- **The New Components:**
  - Are Independent, orthogonal, uncorrelated
  - Decrease in the amount of variance

Thus, only some will be retained for further study – Dimension Reduction

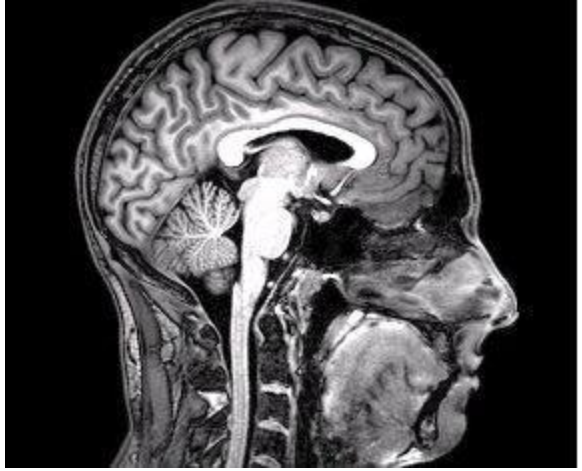
APPROACH AND CHALLENGES

- Data pre-processing
  - Normalization
  - Missing values (Imputation or Not)

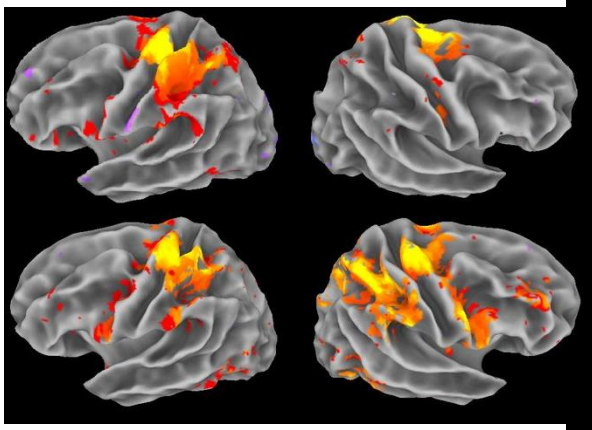
1. Del; 2. AVG; 3. KNN; 4. Predict with classification

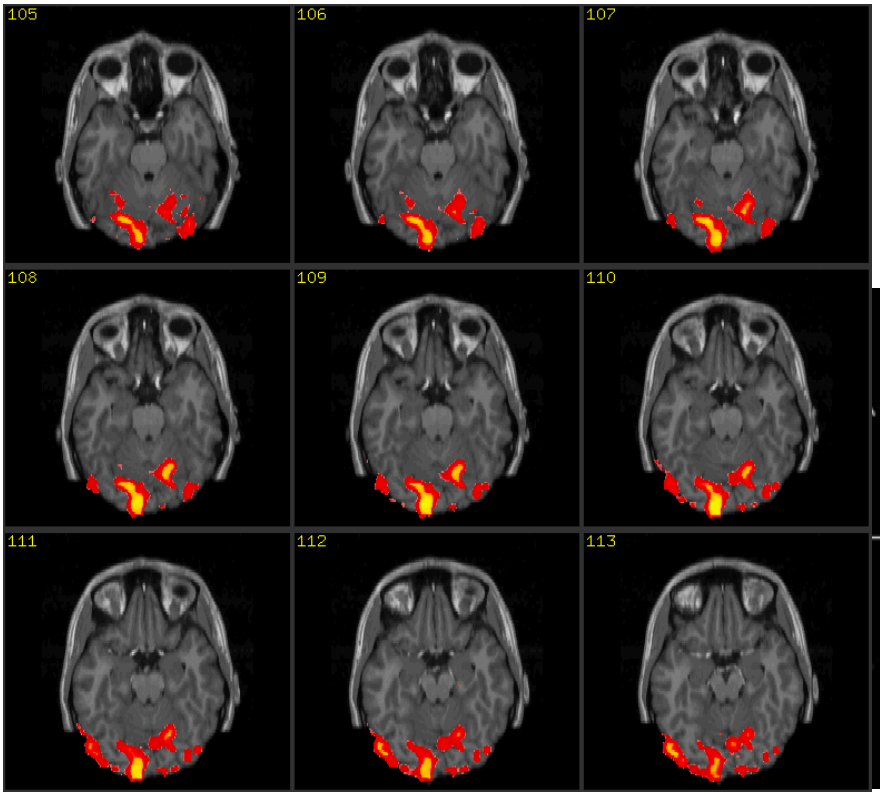
- Inference
  - Feature selection
  - Dimension reduction
  - Signal detection
- Clustering
  - Distance
  - Features

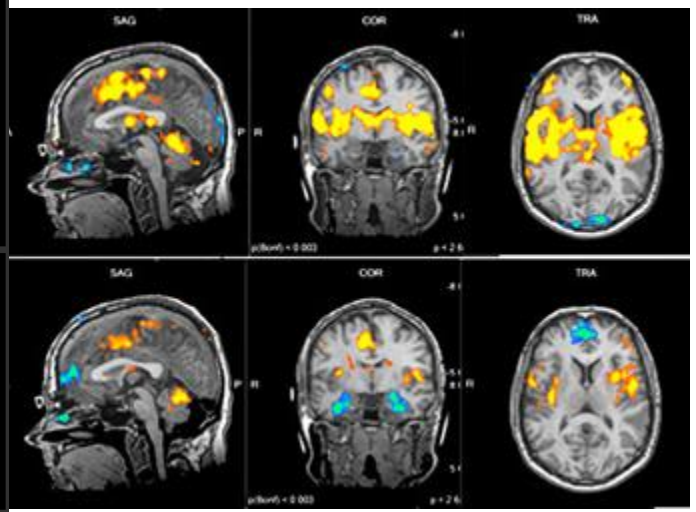
WHAT ARE THEY



- **Random Field Data**
  - Functional MRI scan







APPROACH AND CHALLENGES

- Data pre-processing
- Inference

- Clustering
- Network (Connectivity)
  - Dependence or Covariance in Space-Time

SPARSITY 稀疏性

- Lots of ‘NA’s, who still take up storage
- Problematic for statistical significance
- Demanding even more observations
- Make organizing and searching data hard
- With almost all high dimensional data

SPARSITY + HIGH DIMENSIONALITY

- Example: 10 Years' Data on National Supermarket Chains

Chain	Location	Time	Customer	Product
10	2800	520	50Millions	1Millions

Albertsons
Aldi
Kmart
Kroger
SuperTarget
Trader Joe's
Walmart
…

NYC
Jersey City
San Francisco
Los Angeles
Seattle
…

Cereals
Bread
Potato Chips
Apples
…

Store in an array of 10 X 2800 X 520 X 50,000,000 X 1,000,000

SPARSITY + HIGH DIMENSIONALITY

- Example: 10 Years' Data on National Supermarket Chains

Chain	Location	Time	Customer	Product
10	2800	520	50Millions	1Millions

1. compute into categorical variable;
2. A Priori

Store in an array of

10-by-2800-by-520-by-50,000,000-by-1,000,000

But we might only have $2.9E^{15}$ empty (or <10) cells,
Sparsity of the array ~0.0004%

1 Yottabyte=1e+15G

Storage needed: 0.6 yottabytes (8 bytes/cell)

SPARSITY + HIGH DIMENSIONALITY

- Example: 10 Years' Data on National Supermarket Chains

Chain	Location	Time	Customer	Product
10	2800	520	50Millions	1Millions

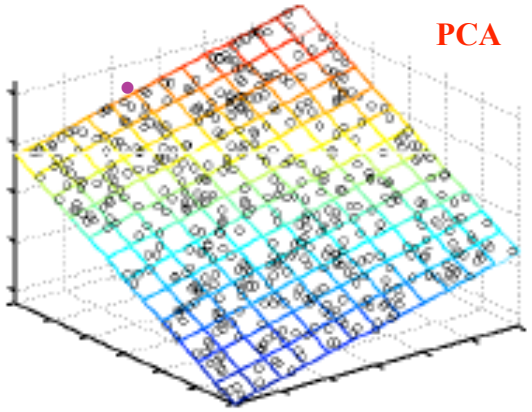
"Composite"

- Reduce sparse cells and "stretch" the sparse dimensions
- "Tuples" for sparse dimensions (location, products, customers) with Time being dense
- A great reduction on storage requirement and make further analysis possible

STRUCTURE IN HIGH DIMENSIONS

- **Types of structures**
  1. Clustered or sparse?
     - Clustering analysis
     - Density estimation (local regression)

     Future study
  2. Low dimensional manifolds?
     - Linear – which low dimensional subspace the data lives?
     - Nonlinear – which low dimensional submanifold the data lives?



or

考

DIMENSIONAL REDUCTION 考

- **Linear methods: PCA and MDS**
  - PCA is to seek best data representation in a lower dimensional space, with the criterion for "best" as "the best subspace to use for projection lies in the direction of maximal variance"
  - PCA works on covariance matrix! – rely on your assumption that covariance matrix describes the features or contains the structure of your data
  - Checklist for PCA: centering, scaling, maximizing projected variance
  - Interpreting PCA: eigenvectors: principal axes of maximum variance subspace; eigenvalues: variance of projected inputs along principal axes; reduced dimensionality: number of significant eigenvalues

PCA DIMENSION REDUCTION

Input (high dimensional)

$x_1, x_2, \ldots x_n$ points in R^p

Output (low dimensional)

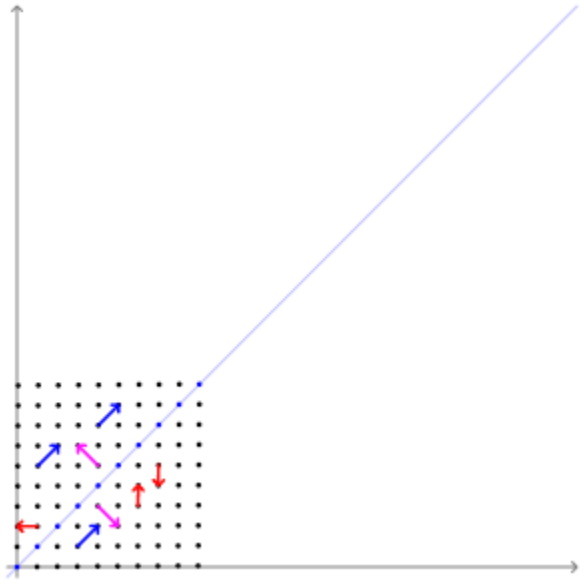
$y_1, y_2, \ldots y_n$ points in R^q $(q<<P)$

1. Assume inputs are centered: $\sum_i^n x_i = 0$ $(x_i$ *is a vector)*
2. Given a <u>unit vector</u> u and a point x, the projection of x onto u is given by $X^T u$
3. Maximize projected variance:

$$\text{var}(y) = \frac{1}{n}\sum_i (x_i^T u)^2 = \frac{1}{n}\sum_i u^T x_i x_i^T u$$

$$= u^T (\frac{1}{n}\sum_i x_i x_i^T) u = u^T C u$$

EIGENVECTORS



Power Method and Inverse Power Method
Jacobian Method

Graph source: wikipedia

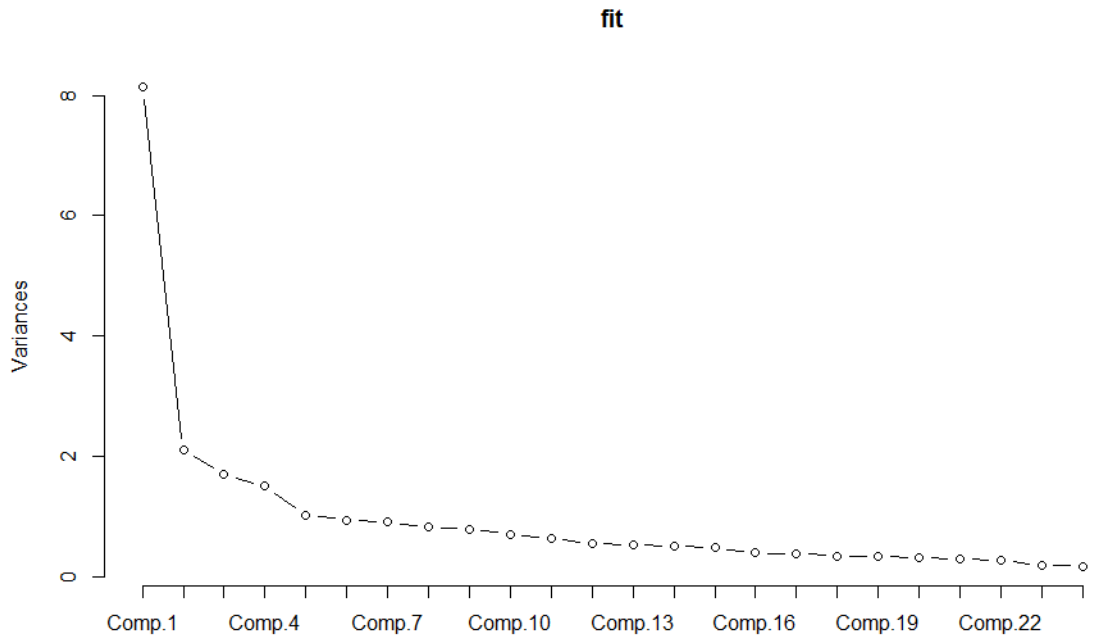
How to find eigenvectors and eigenvalues?

PCA

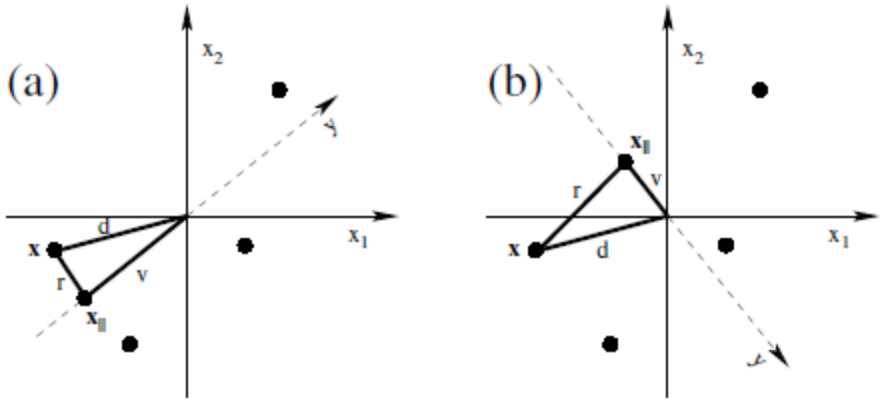
- First q Principal Component:
  - projected our p-dimensional data into a q-dimensional sub-space
  - New sets of variables are essentially linear combinations of your observed variables
  - We use the ratio of variance "explained" by the projected data to help us decide how many (q) PCs to retain → this can also be done/assisted with a Scree plot (next slide)

HOW DO WE CHOOSE *Q*? - **VISUALIZATION**

- Screeplot – help to find the cutting point of choosing the number of PCs



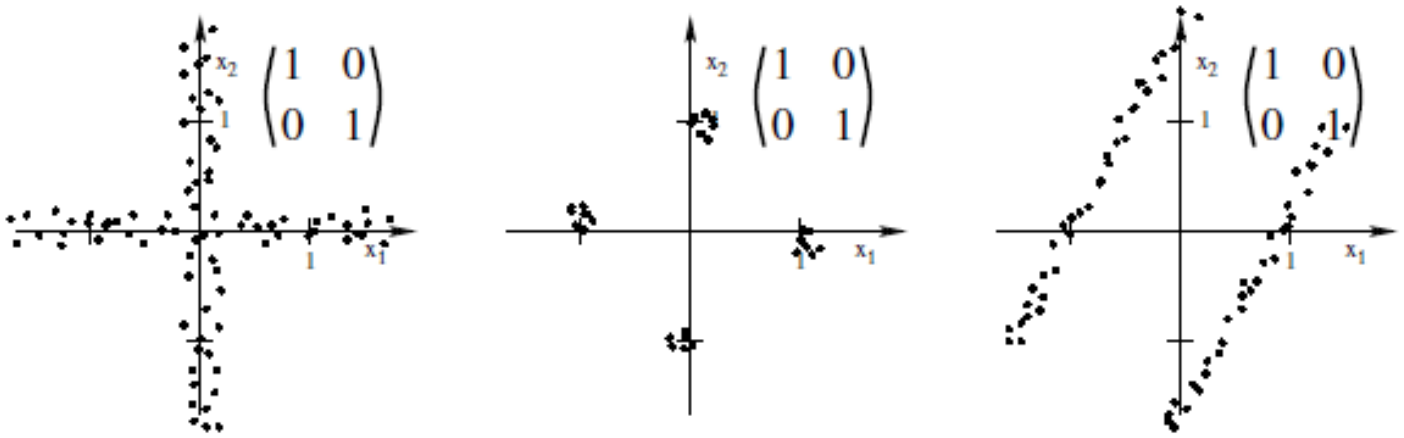
RECONSTRUCTION ERROR & VARIANCE



$$r^2 + v^2 = d^2$$ **FIXED**

Reconstruction Error	Variance of the PC	Variance of the Data
Minimized	Maximized	Constant

COVARIANCE ≠ DATA STRUCTURE



- The covariance matrix only gives you information about this general extent of the data, no higher-order structure of the data.

SUMMARY: PROPERTIES OF **PCA**

- **Strengths**
  - Spectral decomposition (eigen method) and singular value decomposition
  - Without estimation of parameters
  - No iterations – direct decomposition
  - No local optima (especially winning over MDS, right?!)
- **Weaknesses**
  - Rely on covariance matrix – analysis limited to second order statistics, high-order structure will fail PCA –> MDS?
  - Limited to linear projections – linear combination of observed variables

DIMENSIONAL REDUCTION

- **Linear methods: PCA and MDS**

Beyond Linearity

In PCA we believe the underlying structure is linear;
Else we use MDS.

  - MDS is to seek best data representation in a lower dimensional space, with the criterion for "best" as "the best subspace to preserve pairwise distances/similarities"
  - MDS works on similarity matrix – measured by Euclidean distance/correlation matrix
  - Checklist for MDS: centering, scaling, minimizing STRESS
  - Objective of MDS: to construct a configuration of n points in Euclidian space by using the information about the "distances" between the n patterns
  - Interpreting MDS: eigenvectors: ordered, scaled and truncated to yield low dimensional embedding; eigenvalues: measure how each dimension contributes to dot products; estimated dimensionality: number of significant eigenvalues

MDS

- **A n-by-n distance/similarity matrix**

$$D^{(X)} = \{d_{ij}^{(X)}\}, d_{ii}^{(X)} = 0 \text{ and } d_{ij}^{(X)} > 0$$

- **MDS attempts to find *n* data points $\{y_1, y_2, \ldots, y_n\}$ in d dimensions, such that $d_{ij}^{(Y)}$ is similar to $d_{ij}^{(X)}$, i.e., $D^{(Y)} \approx D^{(X)}$**
- **Metric MDS minimizes** $\min_Y \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{ij}^{(X)} - d_{ij}^{(Y)})^2$
- **Non-Metric MDS: transform pairwise distances:** $\delta_{ij} \rightarrow g(\delta_{ij})$
  - Transformation: nonlinear but monotonic
  - Preserves rank order of distances
  - Find vectors $\{y_i\}$ such that $\left\| y_i - y_j \right\| \approx g(\delta_{ij})$
  - Cost = $\min_Y \sum_{ij} (g(\delta_{ij}) - \left\| y_i - y_j \right\|)^2$

MEASURE THE "DISTANCE"

Euclidean distance:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ip} - x_{jp})^2}$$

Use dist() in R

What if variables are on different scales?

- Normalization or Standardization

```
R> dist(scale(measure[, c("chest", "waist", "hips")],
+        center = FALSE))

      1    2    3    4    5    6    7    8    9   10   11
2  0.17
3  0.15 0.08
4  0.22 0.07 0.14
5  0.11 0.15 0.09 0.22
6  0.29 0.16 0.16 0.19 0.21
7  0.32 0.16 0.20 0.13 0.28 0.14
8  0.23 0.11 0.11 0.12 0.19 0.16 0.13
```

MEASURE THE "DISTANCE" (CONT'D)

Euclidean distance: $d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ip} - x_{jp})^2}$

Manhattan distance: $d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + ... + |x_{ip} - x_{jp}|$

Maximum distance: $d(i,j) = max_{k=1}^{p} |x_{ik} - x_{jk}|$

Less common used one:

Minkowski distance: (The p norm)

$$d(i,j) = (|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + ... + |x_{ip} - x_{jp}|^q)^{\frac{1}{q}}$$

Canberra: $d(i,j) = \sum_{p=1}^{p=P} \frac{|x_{ip} - x_{jp}|}{|x_{ip} + x_{jp}|}$ Weighted Manhattan

Use `dist()` in R

MDS ISSUE: INITIAL CONFIGURATION

```
> Lawler
        T1M1 T2M1 T3M1 T1M2 T2M2 T3M2 T1M3
T2M3 T2M1 0.53
T3M1 0.56 0.44
T1M2 0.65 0.38 0.40
T2M2 0.42 0.52 0.30 0.56
T3M2 0.40 0.31 0.53 0.56 0.40
T1M3 0.01 0.01 0.09 0.01 0.17 0.10
T2M3 0.03 0.13 0.03 0.04 0.09 0.02 0.43
T3M3 0.06 0.01 0.30 0.02 0.01 0.30 0.40 0.40
```

About the matrix:
Criteria:
T1 = quality of output,
T2 = ability to generate output,
T3 = demonstrated effort to perform
Evaluation:
M1 = rating by superior
M2 = peer rating
M3 = self-rating

Steps:

1. Convert similarities to dissimilarities (**smacof** only works with dissimilarities)
2. (1) Classical MDS – Check STRESS
   (2) Random initialization – Check STRESS
3. Compare & Assess

KKT 条件

FIND BENCHMARK FOR STRESS

1. Random dissimilarities

```
set.seed(429)
stressvec = randomstress(n = 9, ndim = 2, nrep = 500)
mean(stressvec) # to get a benchmark
## [1] 0.3065868

fit = mds(LawlerD)
fit$stress
# [1] 0.2414665 # 0.24<0.3 but not a guarantee of a
```

2. Modern approaches focus on **permutations** of dissimilarity matrix

```
set.seed(429)
res.perm = permtest(fit, nrep = 1000, verbose =
FALSE)
```

```
> set.seed(429)
> res.perm = permtest(fit, nrep = 1000, verbose = FALSE)
> res.perm
Call: permtest.smacof(object = fit, nrep = 1000, verbose = FALSE)

SMACOF Permutation Test
Number of objects: 9
Number of replications (permutations): 1000

Observed stress value: 0.241
p-value: 0.325
```

0.325 is the new benchmark to judge whether our MDS is sound

- Permutation tests provide more useful null distribution (or critical criterion) than random dissimilarity approach.

SUMMARY: PROPERTIES OF **MDS**

- **Strengths**
  - Relaxes distance constraints
  - Yield nonlinear embedding
- **Weaknesses**
  - Highly nonlinear, iterative optimization with local minima → potential solutions: (1) classical MDS as a starting point (2) random initialization; – STRESS benchmark: (1) random dissimilarity; (2) permutation
  - Sometimes, it is subjective or unclear how to choose distance transformation