

# **Biostat 202: Opportunities and Challenges of “Big Data”: Machine Learning 4: Clustering and Data Reduction**

**David Glidden**

**Division of Biostatistics**

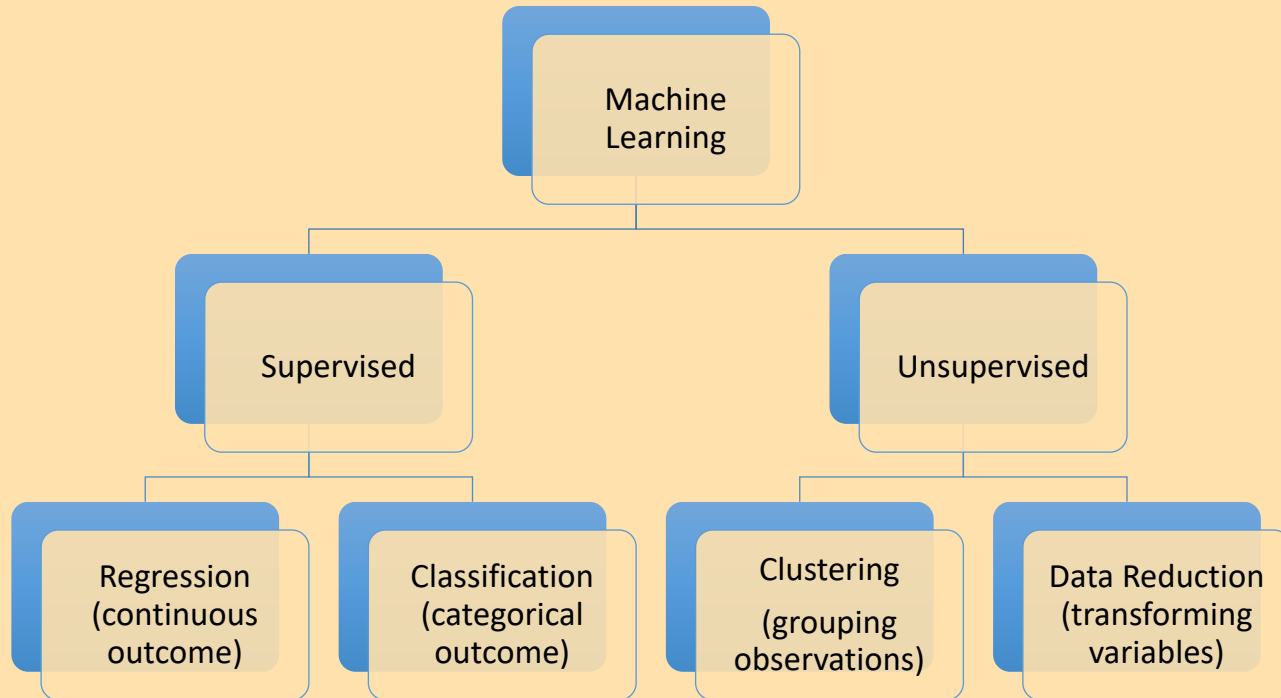
**Department of Epidemiology and Biostatistics**

**08/23/2018**

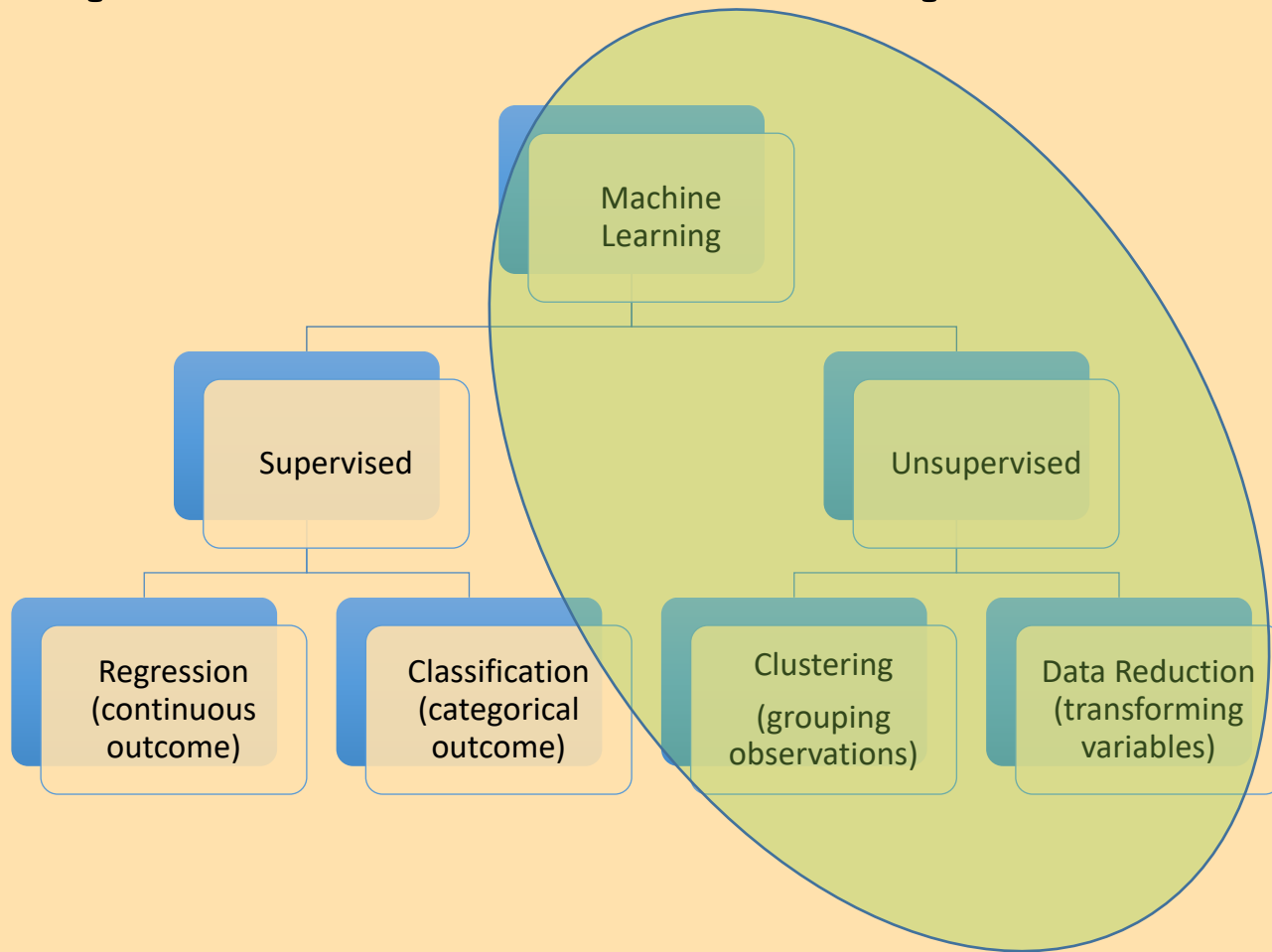
# Overview

- Clustering
  - K-means clustering
  - TwoWay and Kohonen
- Data reduction
  - Principal components analysis (PCA)
- Guidance for choosing between Machine Learning methods
- Ensemble Methods
- More classification techniques in brief (if time)

# Supervised vs. unsupervised



# Supervised vs. unsupervised





# Supervised vs unsupervised

- **Supervised Learning**

- Have data where you know outcome in order to *train* the model
- Train the model to predict future outcomes from sets of predictors

**Prediction: Regression (continuous outcomes) and classification (categorical outcomes)**

- **Unsupervised Learning**

- No desired outcomes
- Separate data points into groups with “similar” properties (clustering)
- Or reduce data to fewer variables in some way that still captures important structure of the data

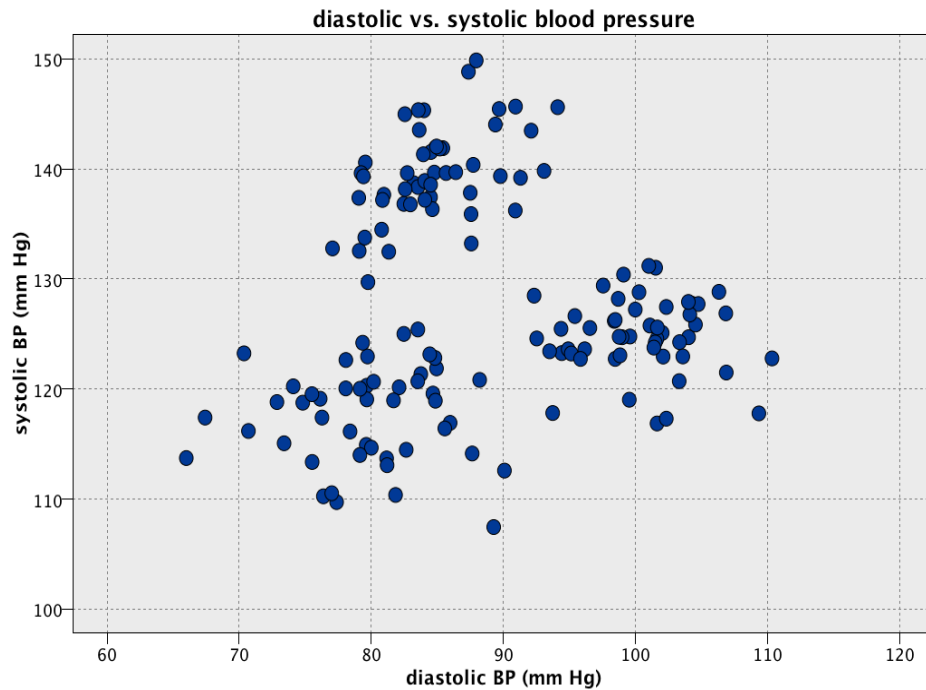
**Clustering and data reduction**

# Clustering

# Clustering problems

- Determine whether there are distinct (unknown) subclasses of brain cancer based on symptomatic and tumor characteristics
- Determine whether there are distinct patterns of gene expression within a patient population

# 2D Clustering example

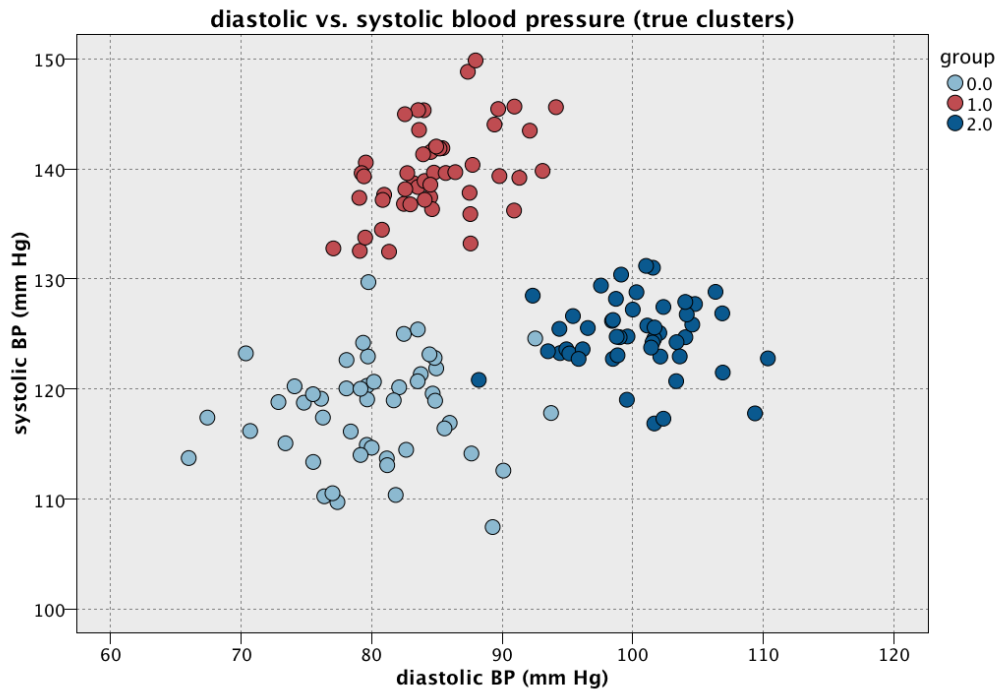


How many clusters?

Where are the clusters?

# 2D clustering example

The truth



Truth has 3 clusters  
(kind of obvious)

# Cluster Analysis (Data Segmentation)

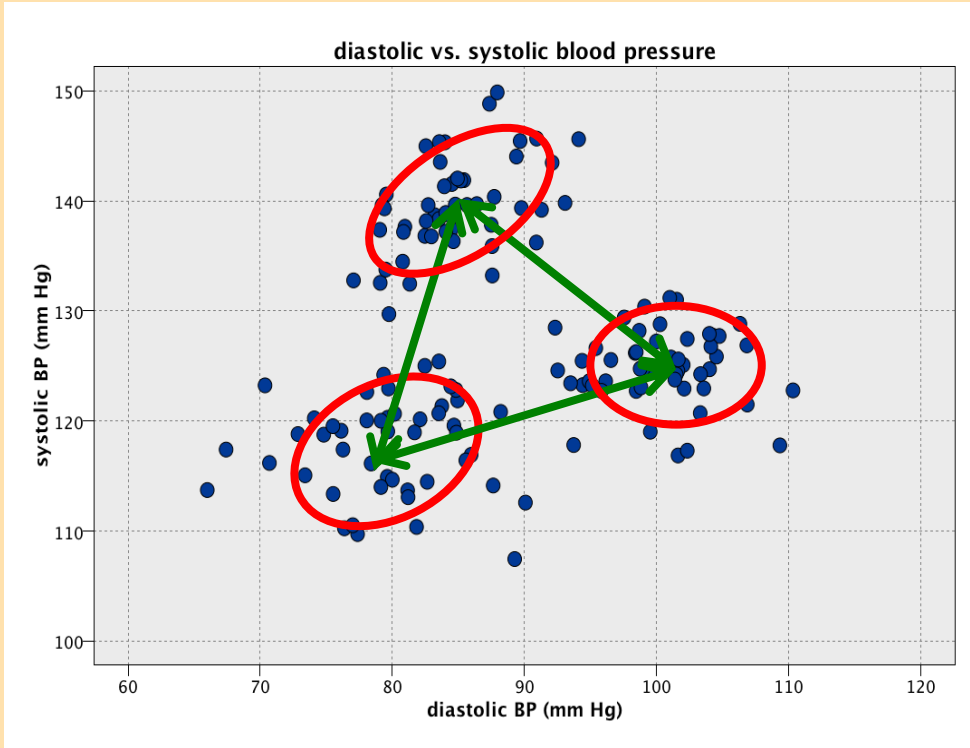
- Objective is to group (or segment) collection of objects into subsets (or *clusters*)
- There is no “Target” in cluster analysis (Unsupervised)
- Objects within a cluster are more similar (or less dissimilar) than objects in different clusters

# 2D clustering example

When determining clusters

We want long green lines  
(large distance between  
cluster centers)

We want small ellipses  
(points within clusters  
grouped tightly together)



Maximize similarity within clusters and minimize similarity  
between clusters

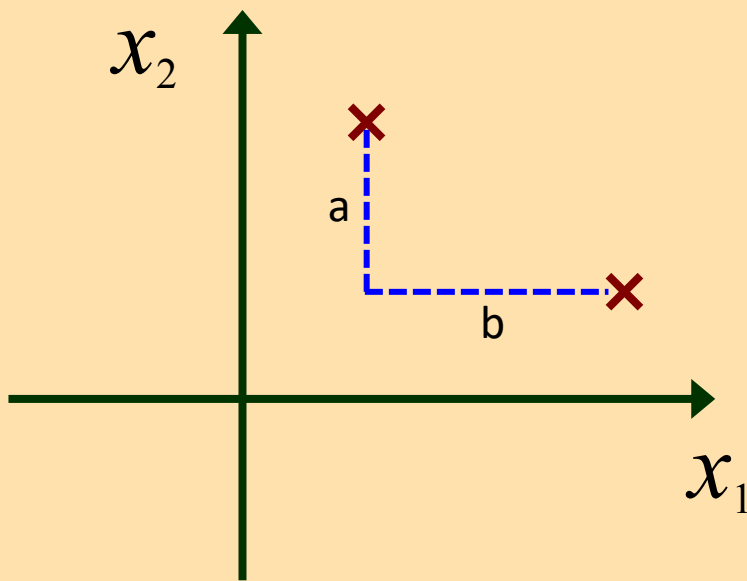
# Cluster Analysis (Data Segmentation)

- Need to define similarity (or dissimilarity) and we need to be able to measure it
- Want to maximize similarity (or minimize dissimilarity) within clusters and minimize similarity (maximize dissimilarity) between clusters



# Measures of similarity/dissimilarity

Most common choice for dissimilarity between 2 instances is the squared difference summed over all attributes (higher values = more dissimilar)



Consider 2 observations with only 2 variables  $x_1$  and  $x_2$  – dissimilarities could be

- Square diff =  $a^2 + b^2$
- Absolute diff =  $|a| + |b|$
- Can also differentially weight attributes e.g.  $ka^2 + mb^2$

Extension to 3D would be e.g.  $a^2 + b^2 + c^2$ , and so on for higher dimensions

# Measures of similarity/dissimilarity

- Generally need to “standardize variables”
- Ordinal variables are treated the same as continuous variables
- Nominal variable differences between groups need to be explicitly given. A common choice is difference 1 if the groups are the same and 0 if they are different. (This then needs to be standardized if any non-categorical variables are also considered.)
- Remember, we want to maximize similarity (or minimize dissimilarity) within clusters and minimize similarity (maximize dissimilarity) between clusters

# **K-means clustering**

# K-means clustering overview

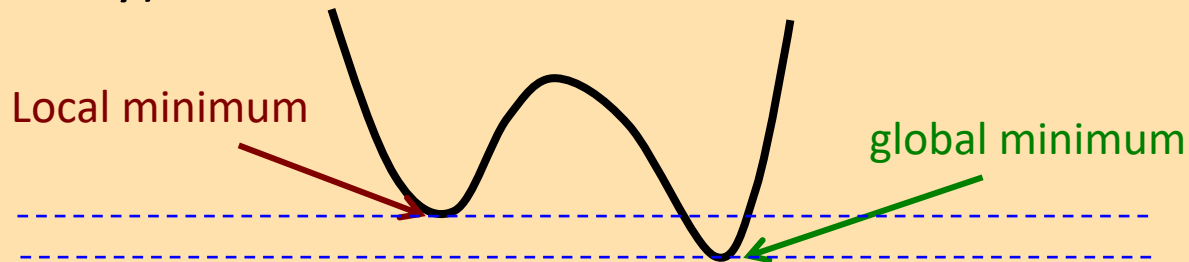
- Algorithm to split set of observations into K clusters, where K can equal 2, 3, 4, ...
- K is pre-chosen (though often choose a set of different K's to try)
- Uses square difference dissimilarity measure  
 $a^2 + b^2 + c^2 + \dots$
- Procedure: starts from a random cluster assignment of observations, and then iterates toward increased similarity (reduced dissimilarity) – estimating cluster centers on each iteration
- Common practice to try a bunch of different values for K and use some kind of measure to determine the best K after the fact. Modeler uses the “Silhouette measure” for this purpose.

# K-means algorithm

- Algorithm proceeds as follows:
  1. Randomly assign a cluster number (between 1 and K) to each of the observations/instances in the dataset. These are the *initial* cluster assignments
  2. Iterate between the following until the cluster assignments no longer change:
    - a. For each cluster determine the center (centroid) of the cluster
    - b. Assign each observation to the cluster whose center is *closest* to them in terms of squared differences (i.e., the center that is the most *similar* to the observation)
- The algorithm increases the within-cluster similarity relative to the between cluster similarity at every step

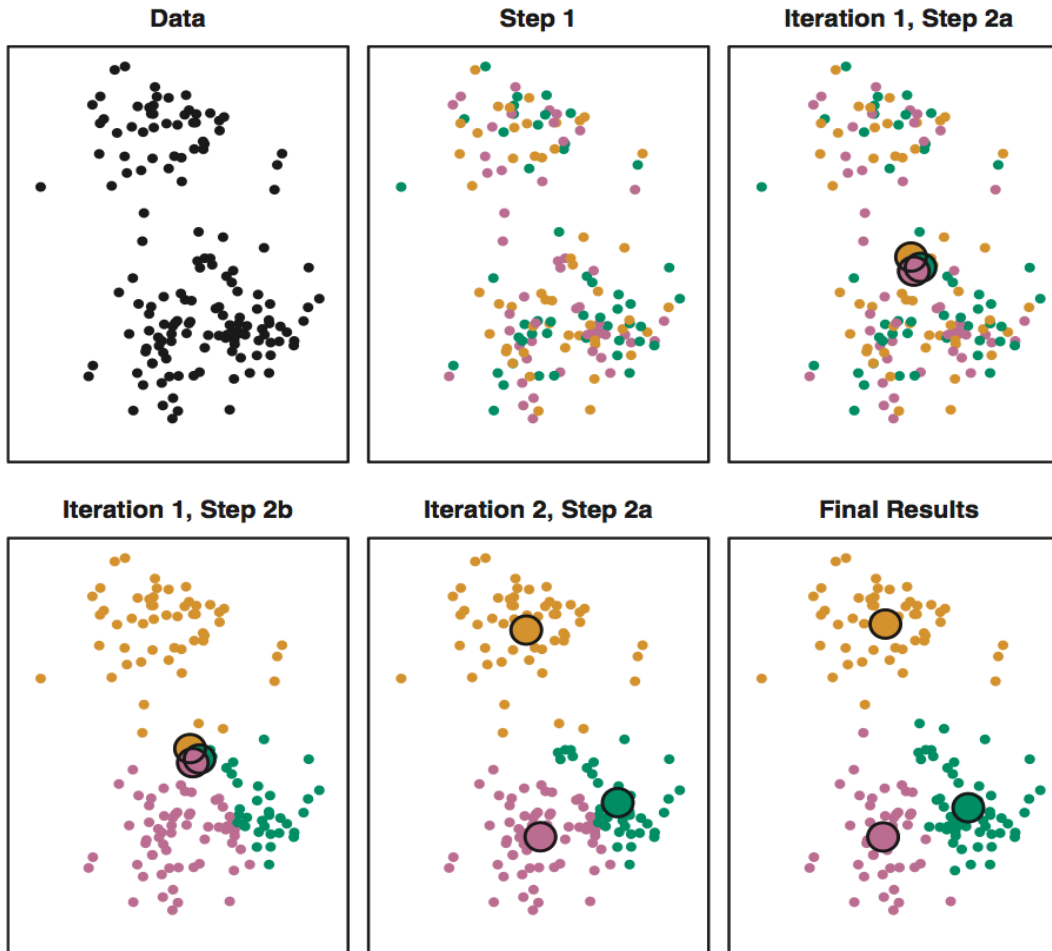
# K-means algorithm

- The center of a cluster is simply the point associated with the mean of each of the attributes for the observations assigned to that cluster – e.g. in the Age direction the center would be located at the mean Age for all subjects in the cluster.
- Note that for different initial random assignments of observations to clusters, K-means may lead to different final clustering (it is a *local* not *global* minimizer of dissimilarity)



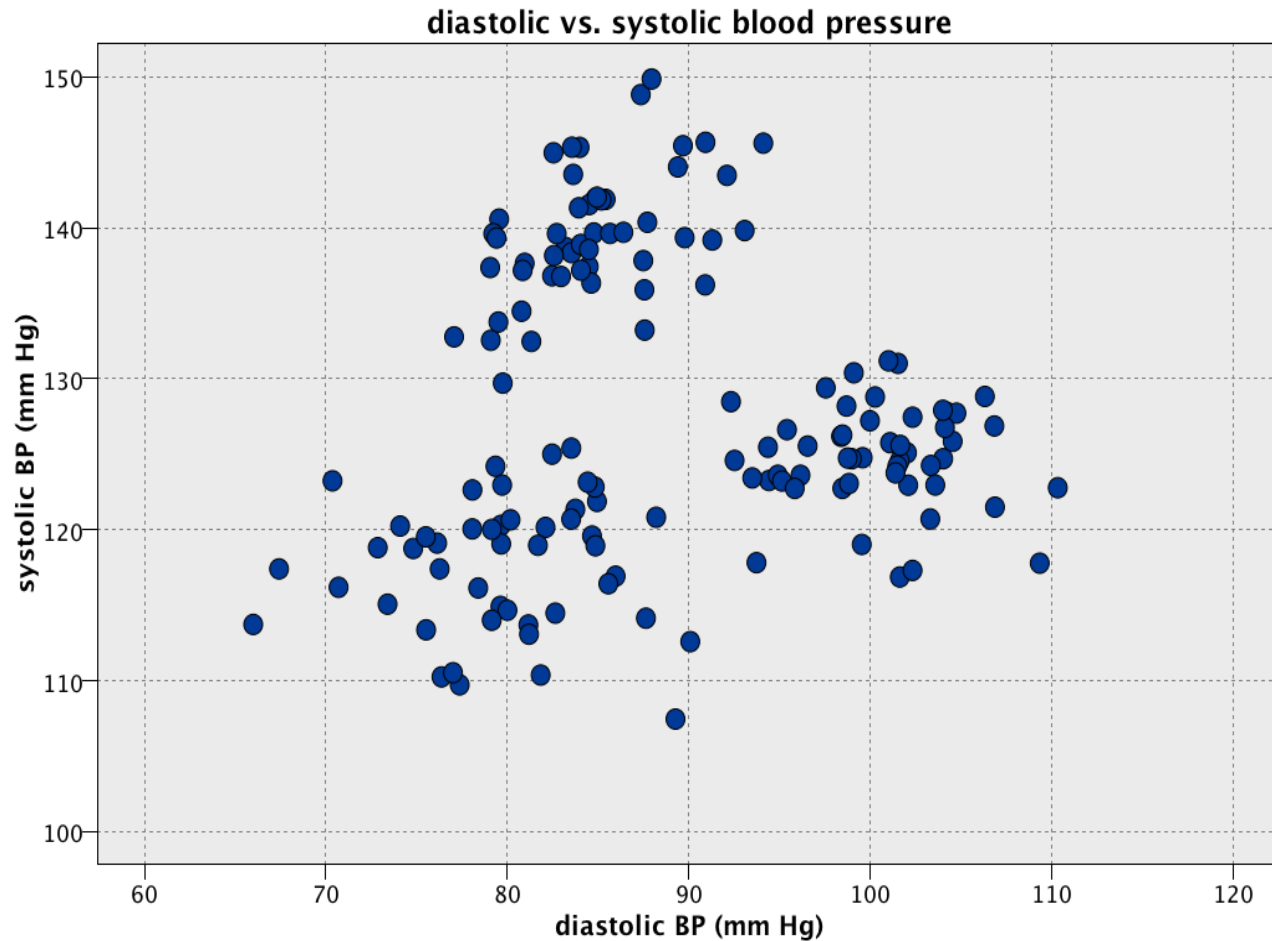
**Upshot - may not converge to best answer**

# K-means algorithm



Taken from ISLR,  
James et al., 2013  
p. 389

# K-means BP Modeler simulated example





# K-means BP Modeler simulated example

Type Node

| <div>  Preview </div>                                     |             |             |         |       |              |
|-----------------------------------------------------------|-------------|-------------|---------|-------|--------------|
| <div> Types Format Annotations </div>                     |             |             |         |       |              |
| <div>    Read Values Clear Values Clear All Values </div> |             |             |         |       |              |
| Field                                                     | Measurement | Values      | Missing | Check | Role         |
| # subjid                                                  | Nominal     | 1.0,2.0,... |         | None  | id Record... |
| # systolic                                                | Continuous  | [107.44...  |         | None  | Input        |
| # diastolic                                               | Continuous  | [65.993...  |         | None  | Input        |
| # group                                                   | Nominal     | 0.0,1.0,... |         | None  | None         |

Table Node

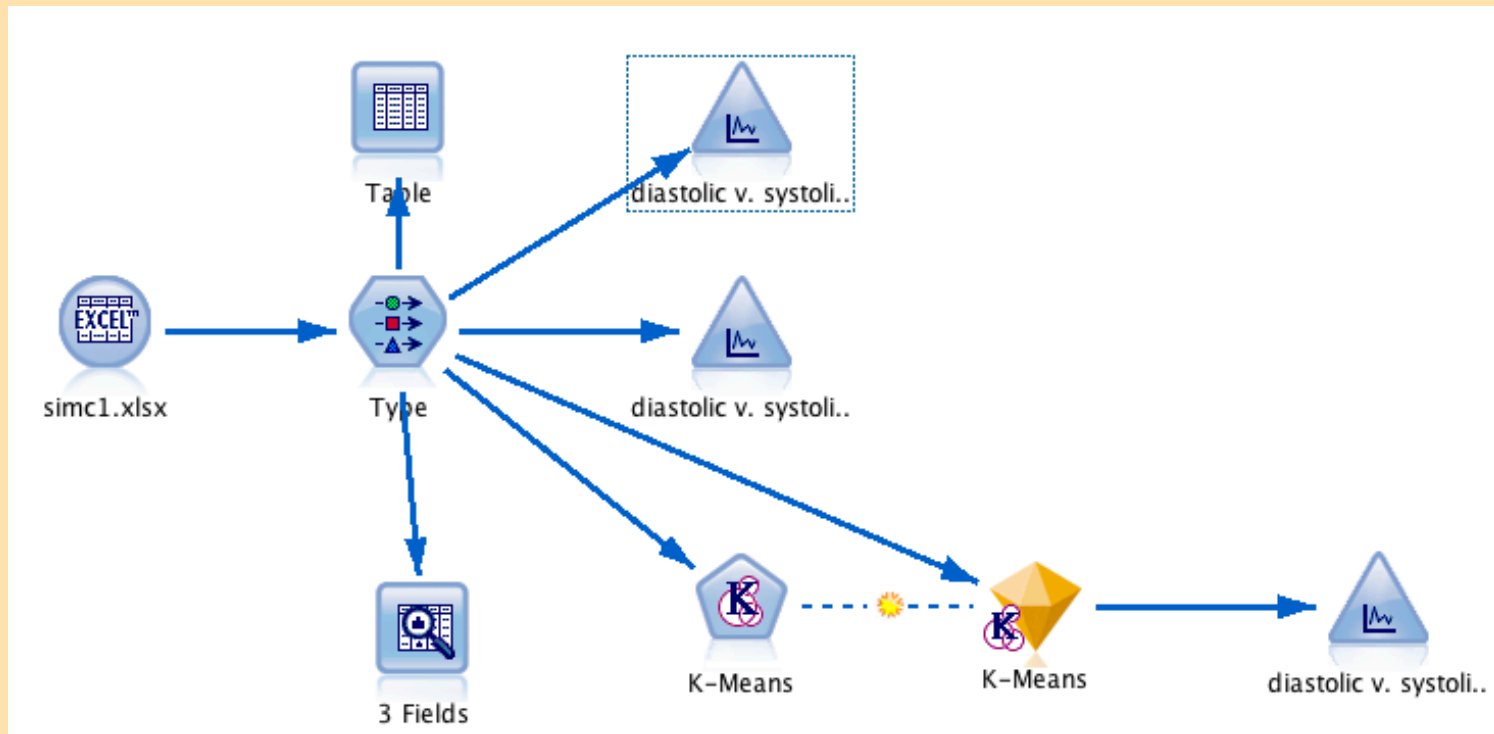
| Table | Annotations |          |           |       |  |
|-------|-------------|----------|-----------|-------|--|
|       | subjid      | systolic | diastolic | group |  |
| 1     | 1           | 113.69   | 81.16     | 0     |  |
| 2     | 2           | 112.58   | 90.09     | 0     |  |
| 3     | 3           | 118.74   | 74.81     | 0     |  |
| 4     | 4           | 119.58   | 84.67     | 0     |  |
| 5     | 5           | 120.05   | 78.07     | 0     |  |
| 6     | 6           | 125.00   | 82.47     | 0     |  |
| 7     | 7           | 113.71   | 65.99     | 0     |  |
| 8     | 8           | 122.95   | 79.72     | 0     |  |
| 9     | 9           | 116.91   | 85.97     | 0     |  |
| 10    | 10          | 121.35   | 83.76     | 0     |  |
| 11    | 11          | 121.86   | 84.95     | 0     |  |
| 12    | 12          | 117.39   | 67.42     | 0     |  |
| 13    | 13          | 120.28   | 79.65     | 0     |  |
| 14    | 14          | 116.18   | 70.70     | 0     |  |
| 15    | 15          | 114.47   | 82.64     | 0     |  |
| 16    | 16          | 124.19   | 79.35     | 0     |  |
| 17    | 17          | 119.05   | 79.66     | 0     |  |
| 18    | 18          | 109.71   | 77.38     | 0     |  |
| 19    | 19          | 114.91   | 79.62     | 0     |  |
| 20    | 20          | 122.82   | 84.83     | 0     |  |
| 21    | 21          | 116.13   | 78.40     | 0     |  |
| 22    | 22          | 120.69   | 83.52     | 0     |  |
| 23    | 23          | 113.36   | 75.53     | 0     |  |
| 24    | 24          | 118.93   | 84.86     | 0     |  |
| 25    | 25          | 116.41   | 85.57     | 0     |  |
| 26    | 26          | 119.11   | 76.13     | 0     |  |
| 27    | 27          | 118.81   | 72.86     | 0     |  |
| 28    | 28          | 124.58   | 92.53     | 0     |  |
| 29    | 29          | 115.06   | 73.40     | 0     |  |
| 30    | 30          | 120.01   | 79.15     | 0     |  |
| 31    | 31          | 129.71   | 79.75     | 0     |  |
| 32    | 32          | 120.65   | 80.18     | 0     |  |
| 33    | 33          | 113.07   | 81.19     | 0     |  |
| 34    | 34          | 107.45   | 89.27     | 0     |  |
| 35    | 35          | 117.81   | 93.74     | 0     |  |
| 36    | 36          | 118.53   | 75.53     | 0     |  |

Audit Node (include group as Target for visualization only)

| Audit       | Quality | Annotations |         |         |         |          |          |        |       |  |
|-------------|---------|-------------|---------|---------|---------|----------|----------|--------|-------|--|
| Field       | Graph   | Measurement | Min     | Max     | Mean    | Std. Dev | Skewness | Unique | Valid |  |
| # subjid    |         | Nominal     | 1.000   | 150.000 | --      | --       | --       | 150    | 150   |  |
| # systolic  |         | Continuous  | 107.449 | 149.852 | 127.557 | 9.944    | 0.286    | --     | 150   |  |
| # diasto... |         | Continuous  | 65.993  | 110.344 | 88.461  | 9.896    | 0.266    | --     | 150   |  |
| # group     |         | Nominal     | 0.000   | 2.000   | --      | --       | --       | 3      | 150   |  |

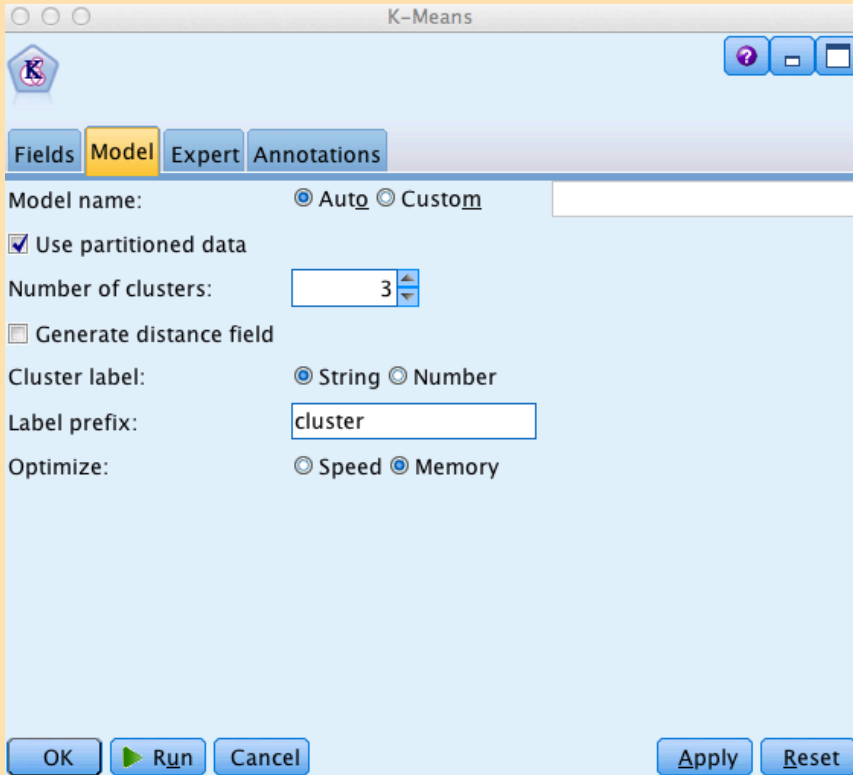
<sup>1</sup> Indicates a multimode result   <sup>2</sup> Indicates a sampled result

# K-means BP Modeler simulated example



# K-means Node

Pre-defined roles in Fields tab (all inputs) – systolic and diastolic are input variables



K-Means

Fields Model Expert Annotations

Model name: ☒ Auto ☐ Custom

☒ Use partitioned data

Number of clusters:

☐ Generate distance field

Cluster label: ☒ String ☐ Number

Label prefix:

Optimize: ☐ Speed ☒ Memory

OK Run Cancel Apply Reset

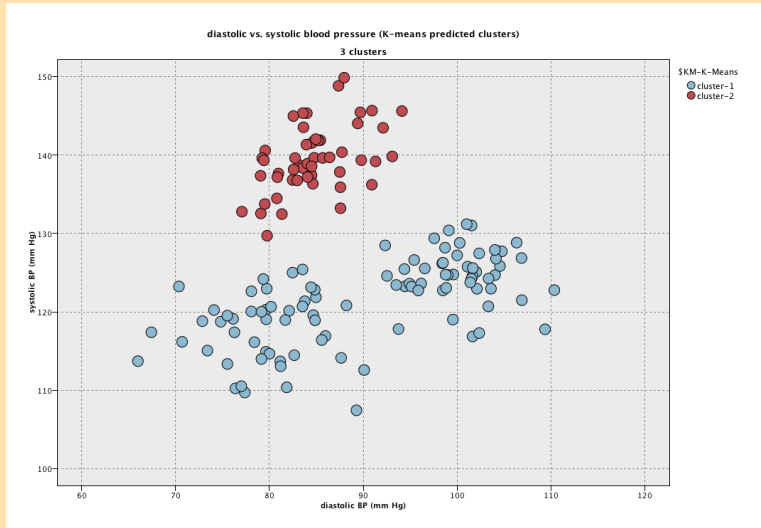
K-Means in IBM Modeler automatically normalized inputs using the following formula:

$$x_{i, \text{ new}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

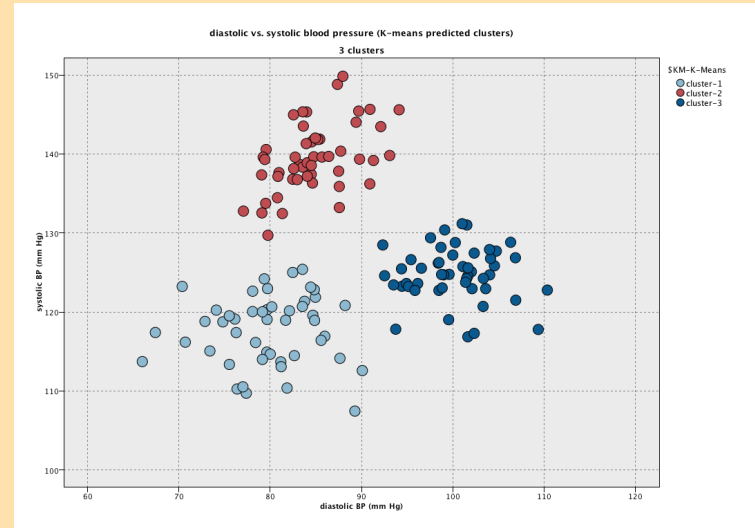
Use defaults in Expert tab

# K-means results

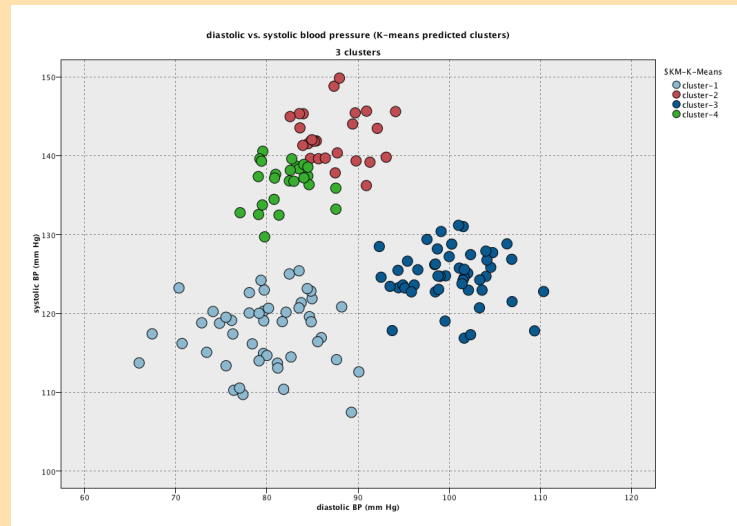
K=2



K=3

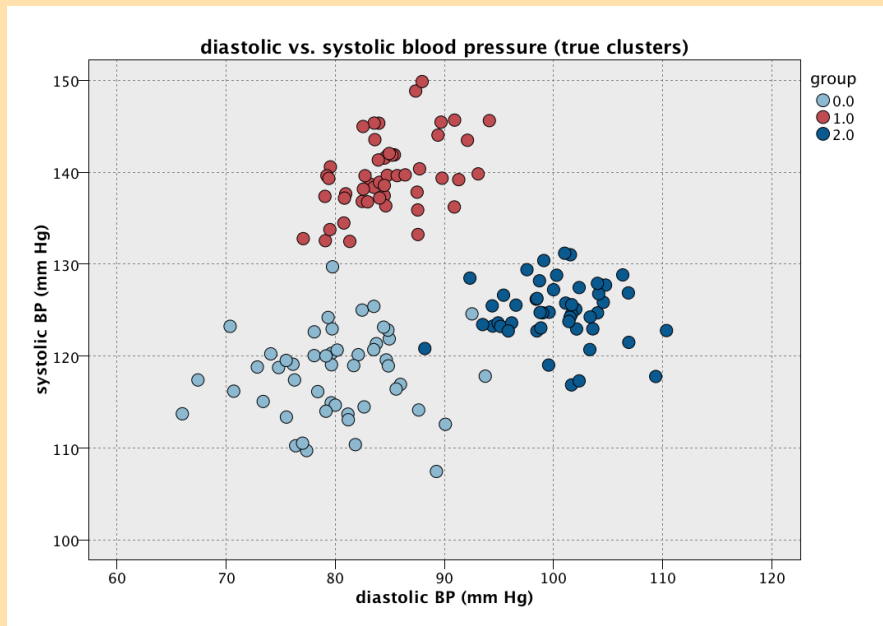


K=4

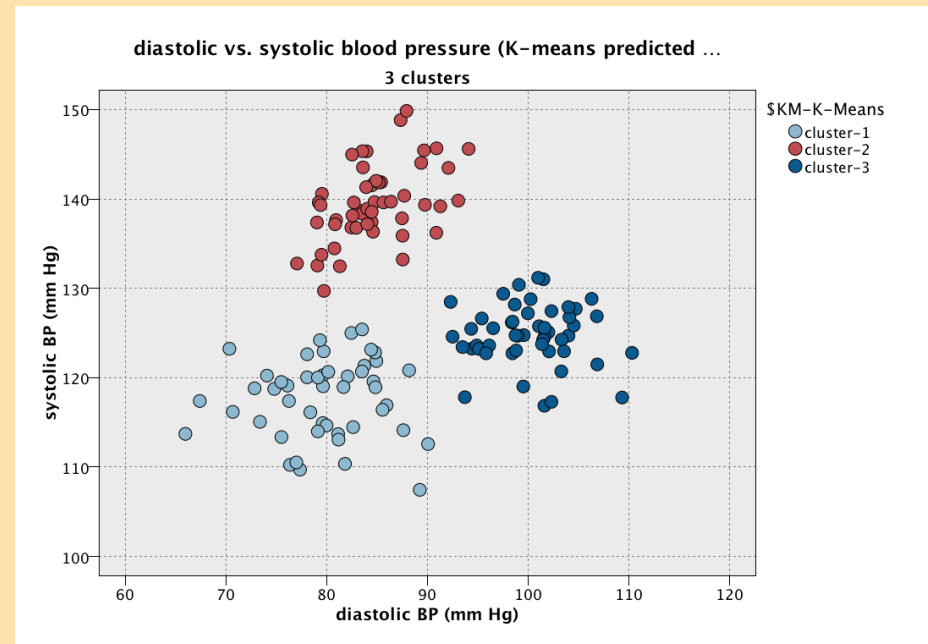


# K-means example

The truth

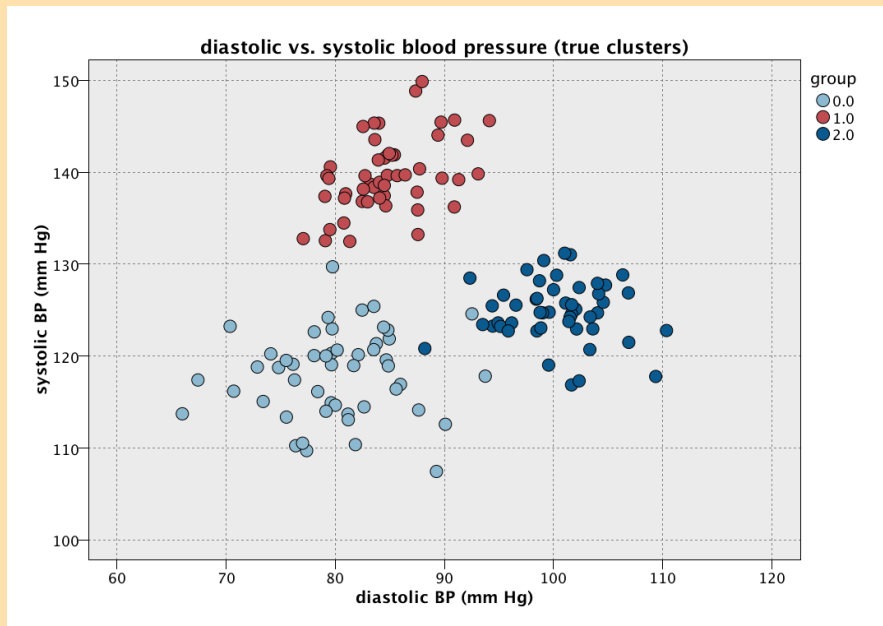


K-means ( $K = 3$ )

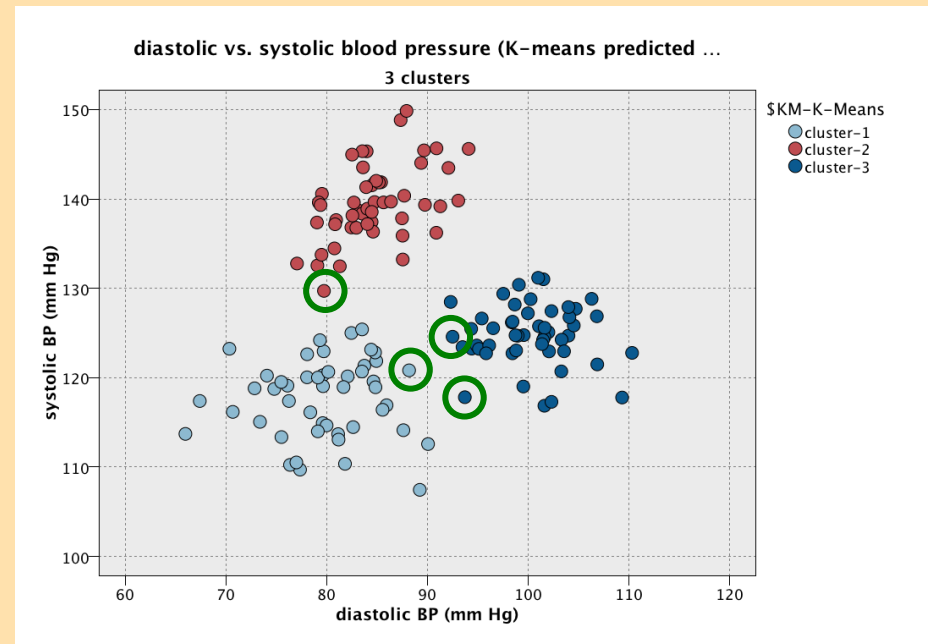


# K-means example

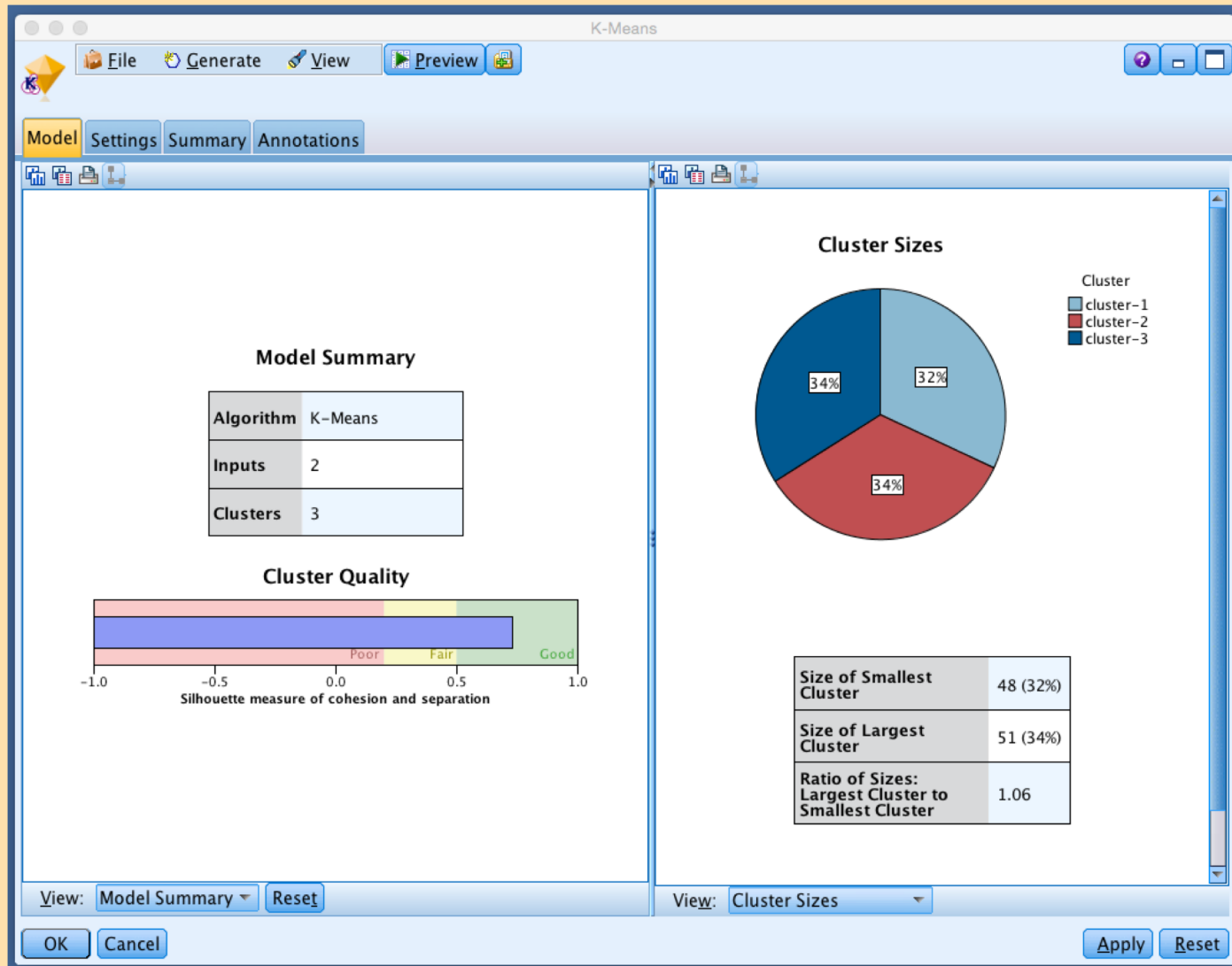
The truth



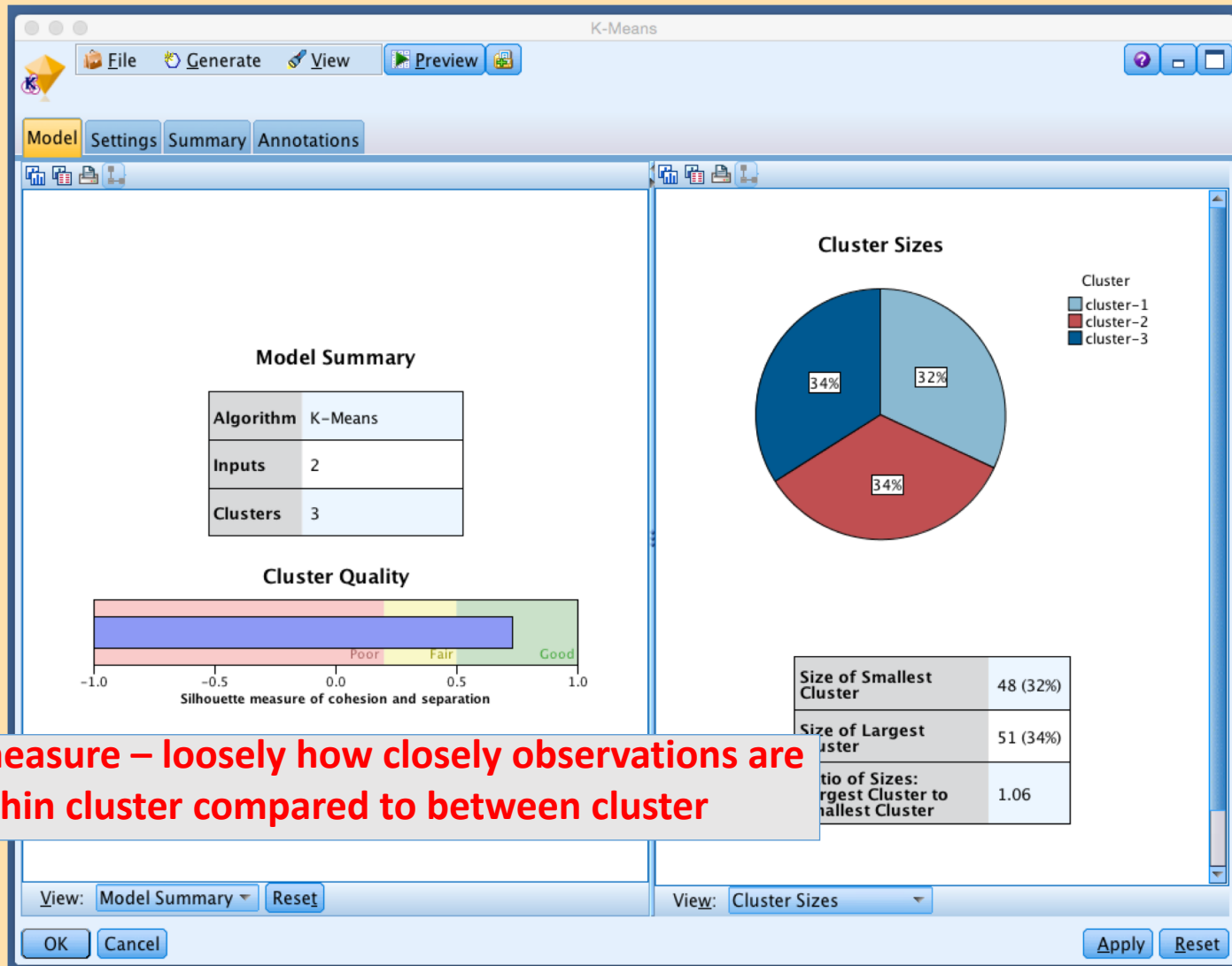
K-means ( $K = 3$ )



# K-means example



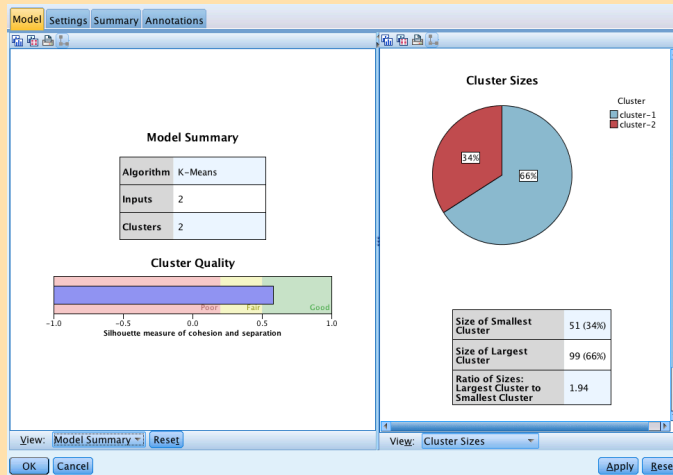
# K-means example



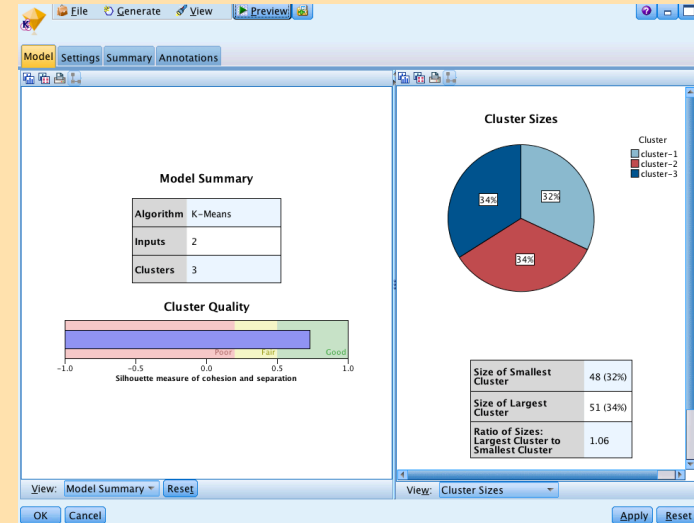


# K-means example

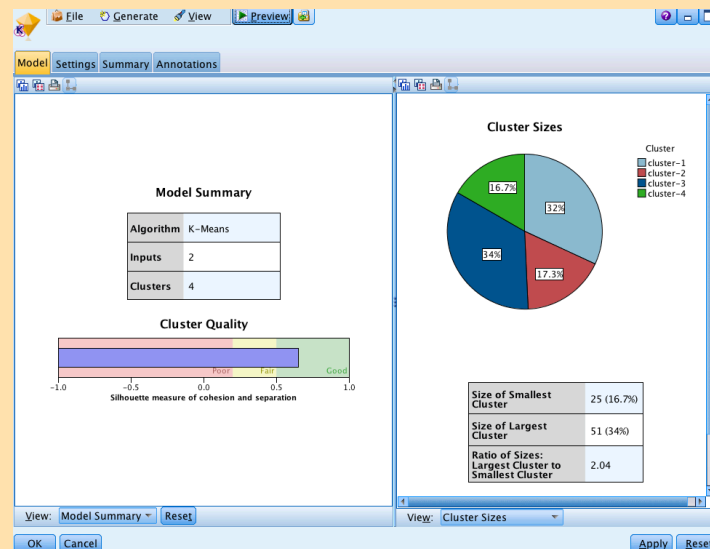
K=2



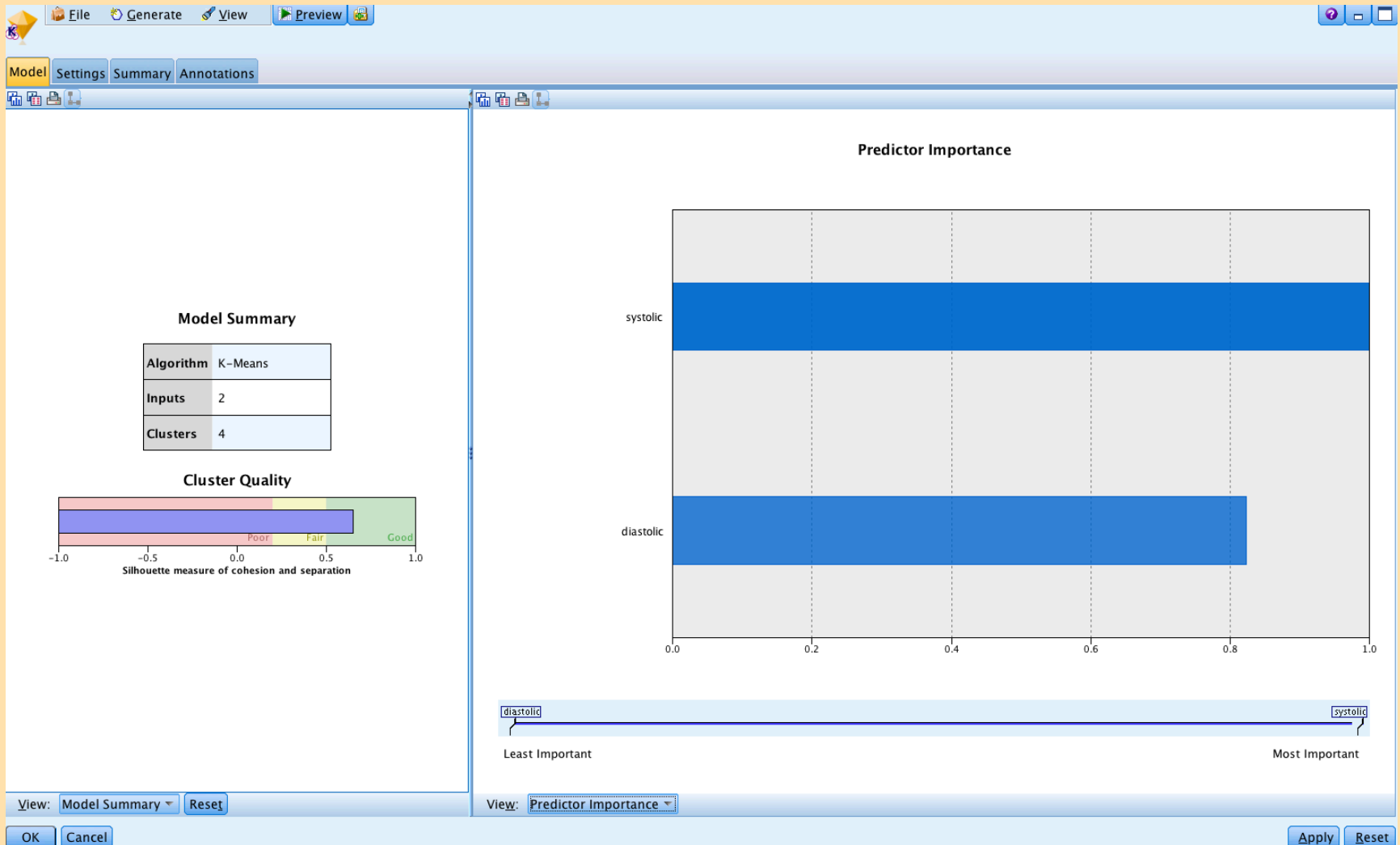
K=3



K=4



# K-means example

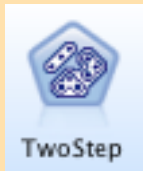


# K-means clustering

- K-means is fast – good for big data – in particular with large number of variables/attributes
- Number of clusters must be defined by user – only heuristic methods for choosing number of clusters (IBM Modeler gives “Silhouette measure”)
- Local minimizer, not global –different starts can lead to different results

# **Other Clustering Algorithms in IBM Modeler (in brief)**

# Other Clustering Algorithms in IBM Modeler (in brief)



## Two Step

- “Hierarchical clustering” method – successively splits data into increasing number of clusters (estimates optimal number of clusters – slower than K-means)



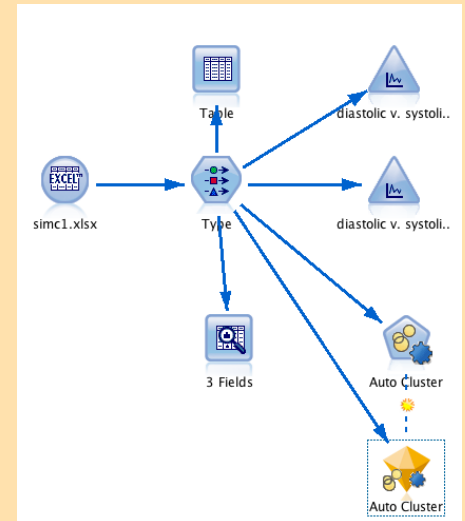
## Kohonen

- Based on Neural Networks (estimates optimal number of clusters – slower than K-means)



# Auto-Cluster

- Try all of Modelers methods and find “best”
- Uses K-means, TwoStep and Kohonen



| <div>  File Generate Preview </div>                                       |       |           |                   |            |                    |                      |                      |                 |                     |                  |          |
|---------------------------------------------------------------------------|-------|-----------|-------------------|------------|--------------------|----------------------|----------------------|-----------------|---------------------|------------------|----------|
| Model Summary Annotations                                                 |       |           |                   |            |                    |                      |                      |                 |                     |                  |          |
| Sort by: Use Ascending Descending Delete Unused Models View: Training set |       |           |                   |            |                    |                      |                      |                 |                     |                  |          |
| Use?                                                                      | Graph | Model     | Build Time (mins) | Silhouette | Number of Clusters | Smallest Cluster (N) | Smallest Cluster (%) | Largest Cluster | Largest Cluster (%) | Smallest Largest | Impor... |
| <input checked="" type="checkbox"/>                                       |       | TwoStep 1 | < 1               | 0.732      | 3                  | 48                   | 32                   | 51              | 34                  | 0.941            | 0.0      |
| <input type="checkbox"/>                                                  |       | K-means 1 | < 1               | 0.561      | 5                  | 25                   | 16                   | 48              | 32                  | 0.521            | 0.0      |
| <input type="checkbox"/>                                                  |       | Kohonen 1 | < 1               | 0.387      | 12                 | 1                    | 0                    | 31              | 20                  | 0.032            | 0.0      |

- Looks like TwoStep did “best” on the BP data

# **Data Reduction Methods**

# **Principle components analysis (PCA)**



# PCA

- Finds (linear) weighted sums of the variables/attributes that maximally explain variability in the data

$$a_1x_1 + a_2x_2 + \dots + a_px_p$$

- The first PC explains as much of the variability in the full data set as possible
- The second PC explains as much variability as possible after the variability from the first (PC) has been removed
- Constraint: each PC is a weighted sum (*linear combination*) of the original variables

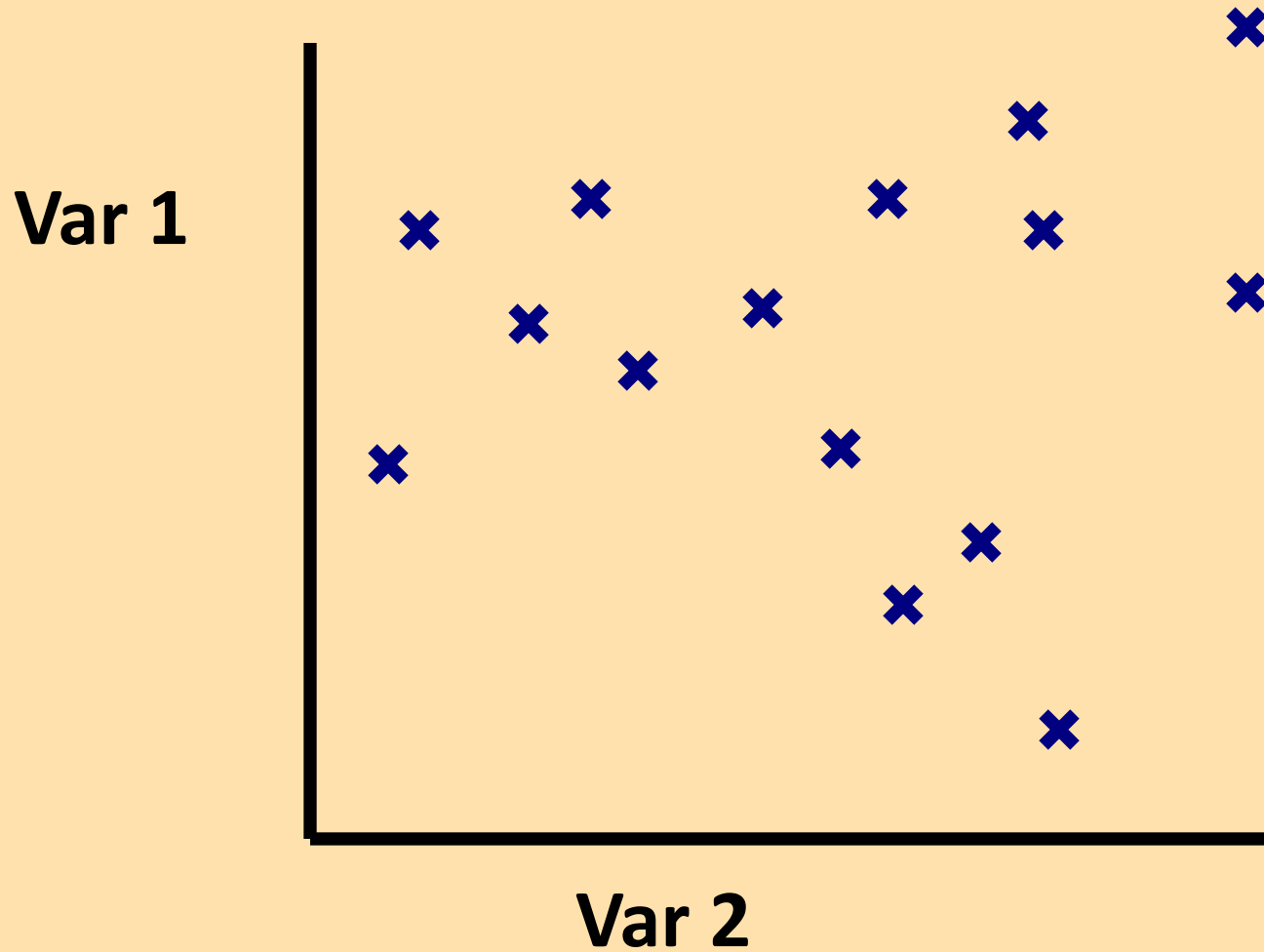
Each PC is a weighted average of the original variables (a linear combination).

$$a_1x_1 + a_2x_2 + \dots + a_px_p$$

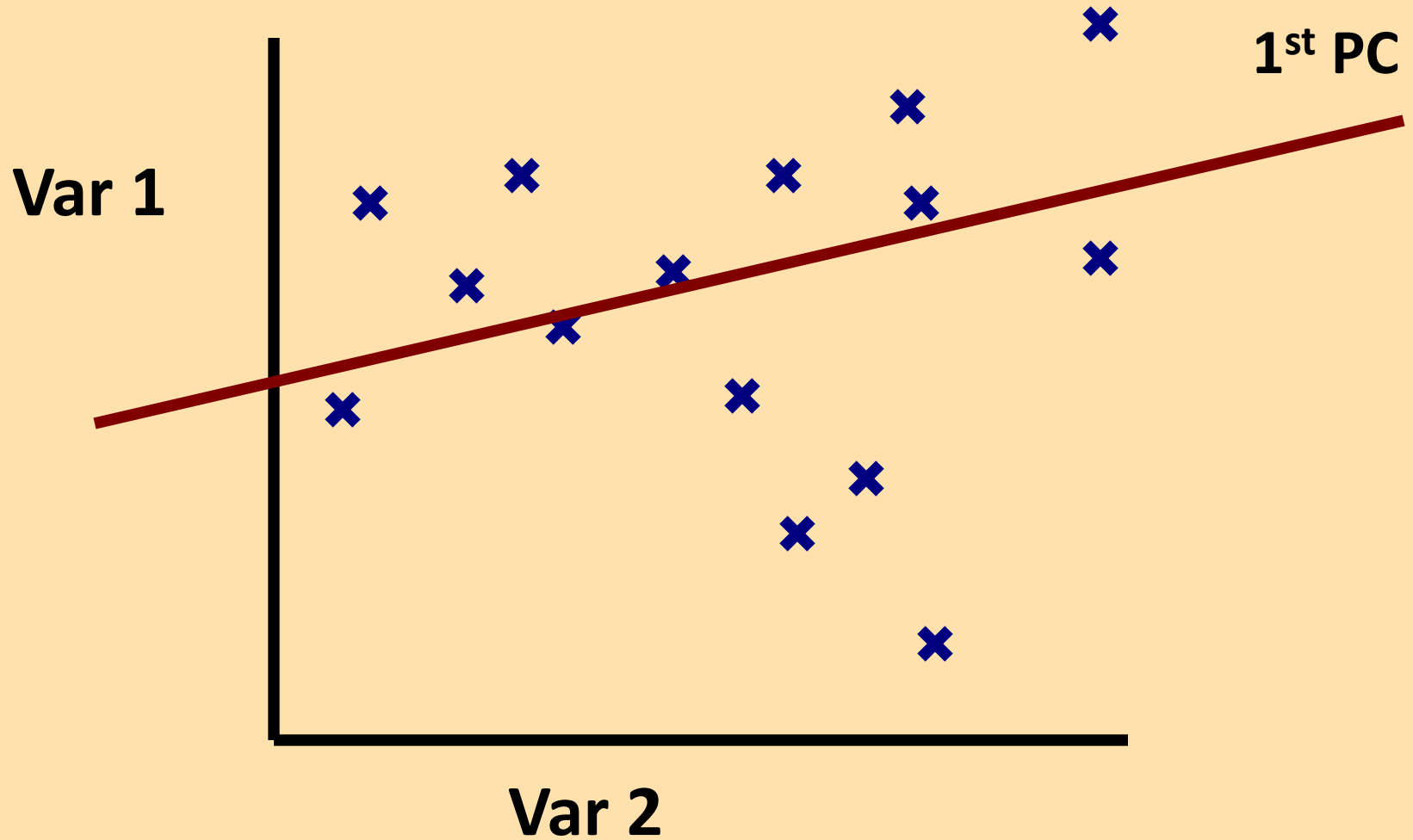
The data reduction goal is to *closely* represent the full data set in terms of just a few of these PCs

Sometimes it is also *hoped* that each of the PCs is meaningful

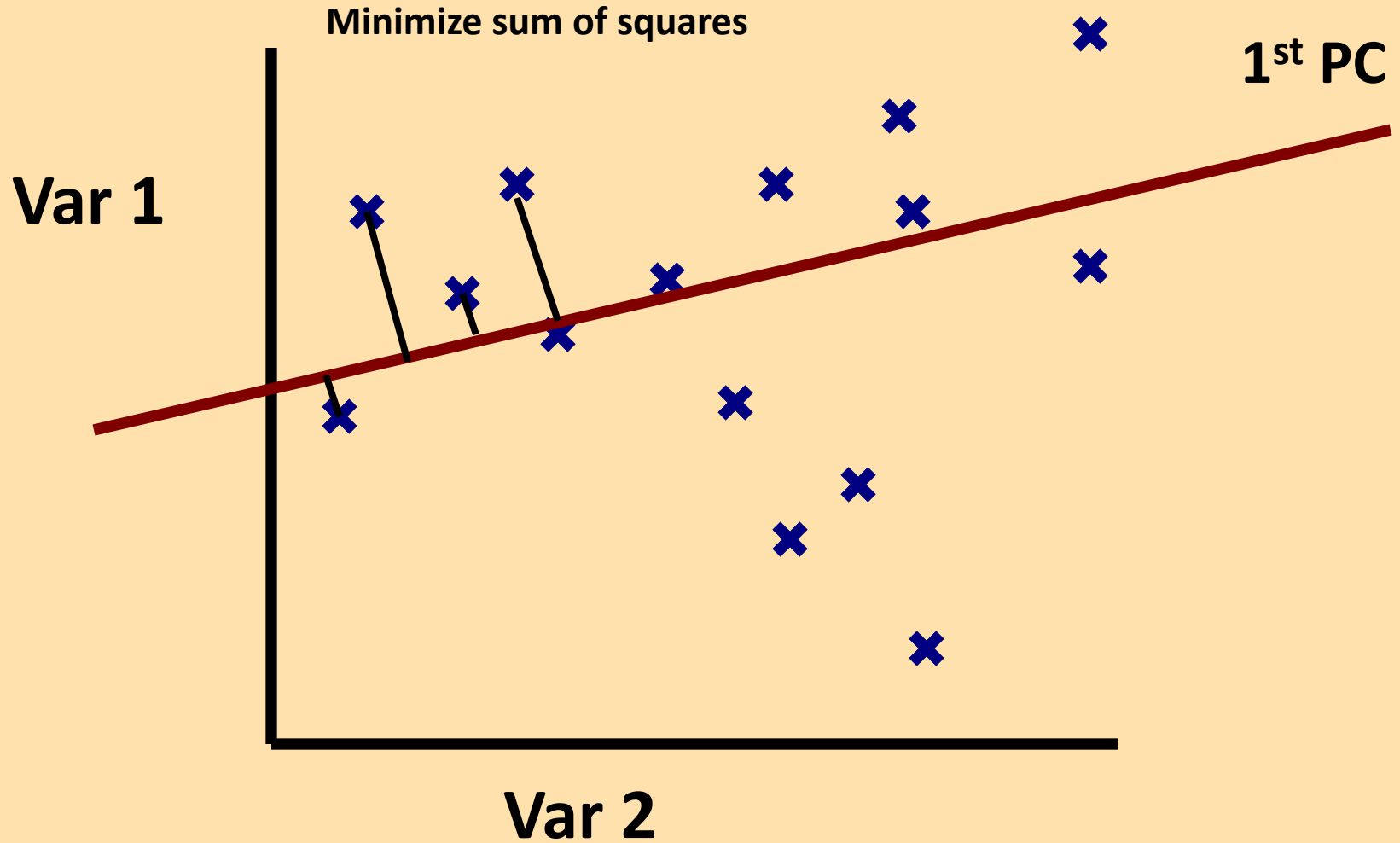
# 2D illustration



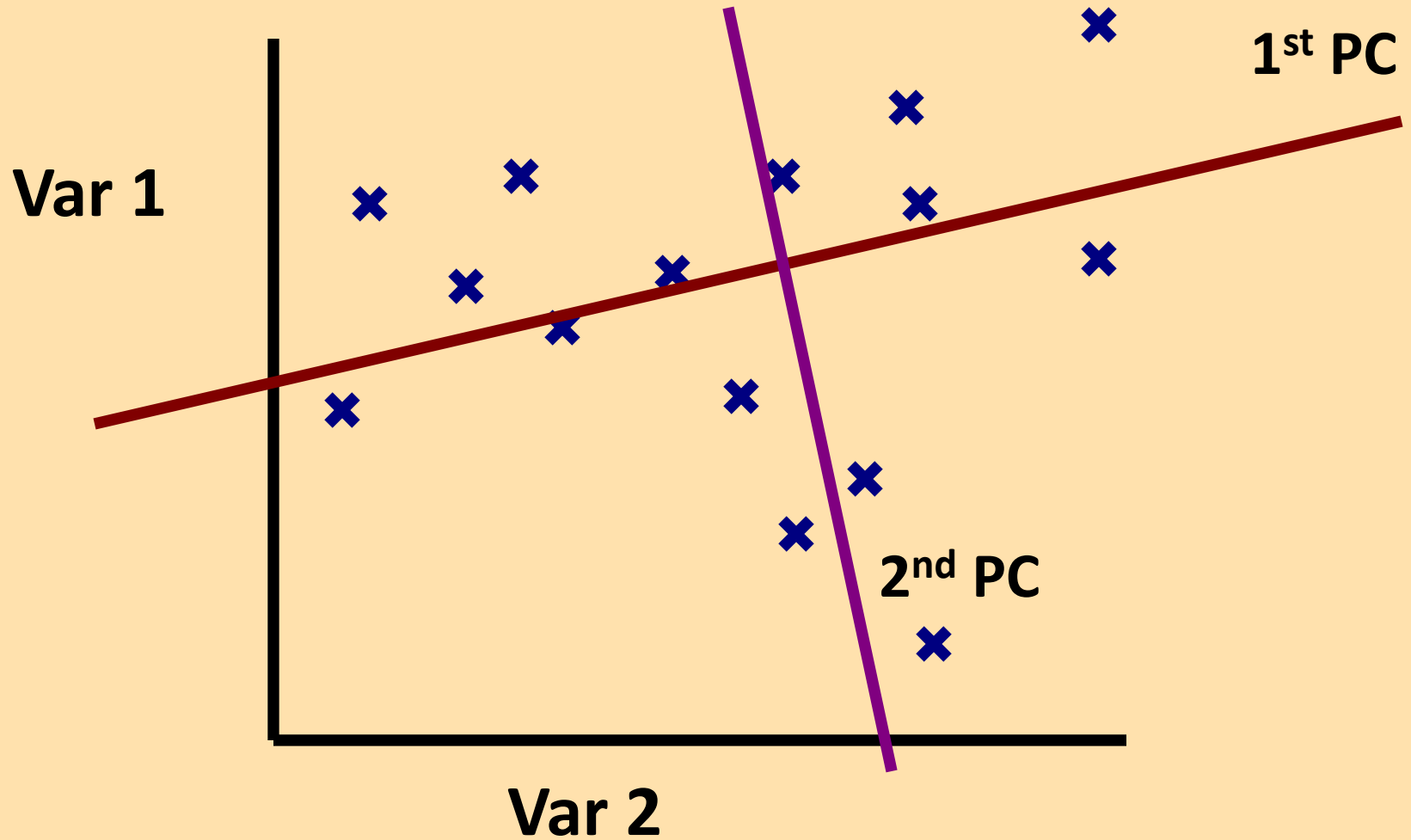
# 2D illustration



# 2D illustration

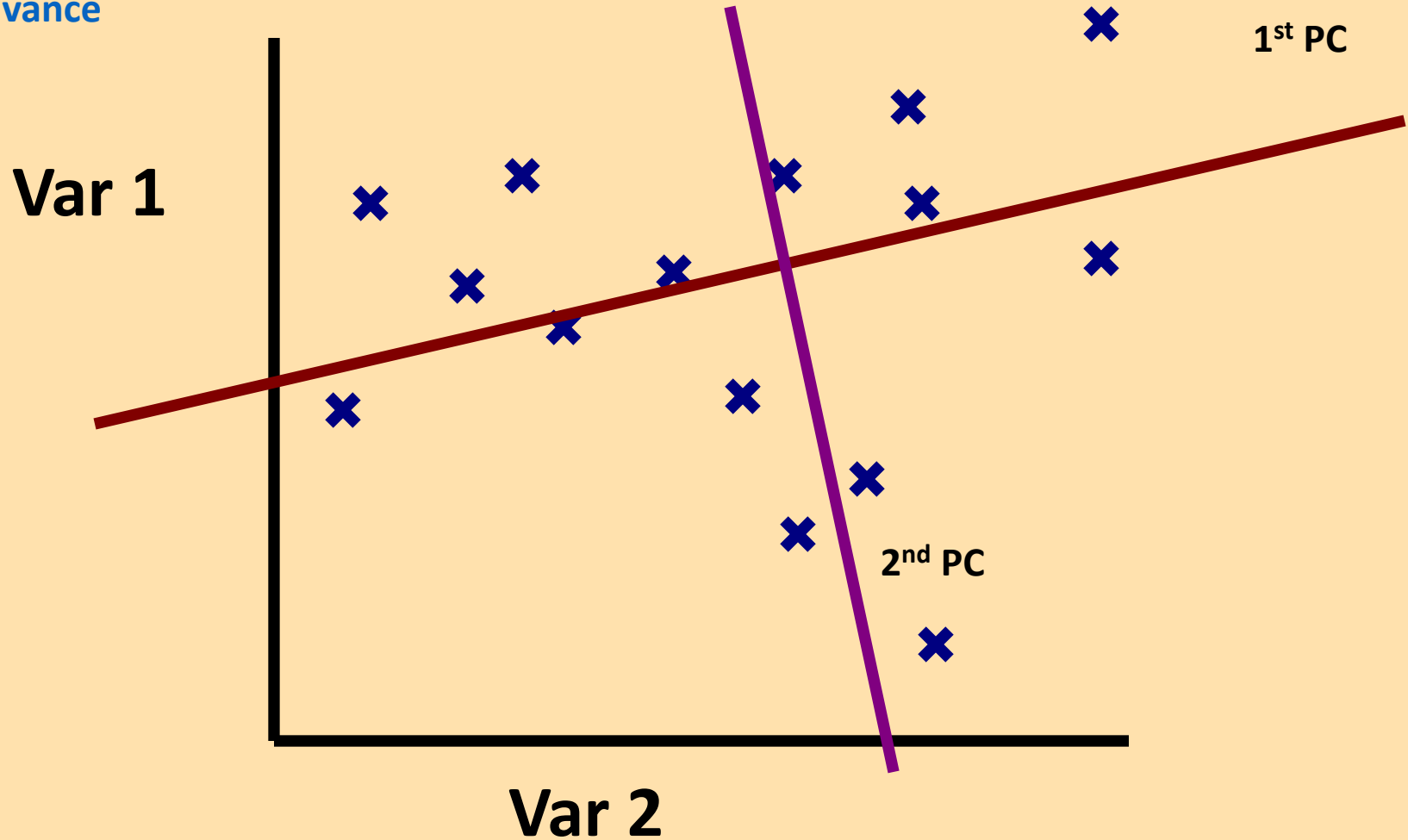


# 2D illustration



Note: that there is  
also a potential issue  
of scaling – variables  
often normalized in  
advance

# 2D illustration



# PCA example – knee shape

- Is the bone in ACL injured knees a different “shape” than a group of control knees?
- May suggest that particular bone shapes predispose to ACL injury.

## Osteoarthritis and Cartilage



Three-dimensional MRI-based statistical shape model and application to a cohort of knees with acute ACL injury



V. Pedroia <sup>†</sup>\*, D.A. Lansdown <sup>‡</sup>, M. Zaid <sup>†</sup>, C.E. McCulloch <sup>§</sup>, R. Souza <sup>‡</sup>, C.B. Ma <sup>‡</sup>, X. Li <sup>†</sup>

<sup>†</sup> Department of Radiology and Biomedical Imaging, University of California, San Francisco, USA

<sup>‡</sup> Department of Orthopaedic Surgery, University of California, San Francisco, USA

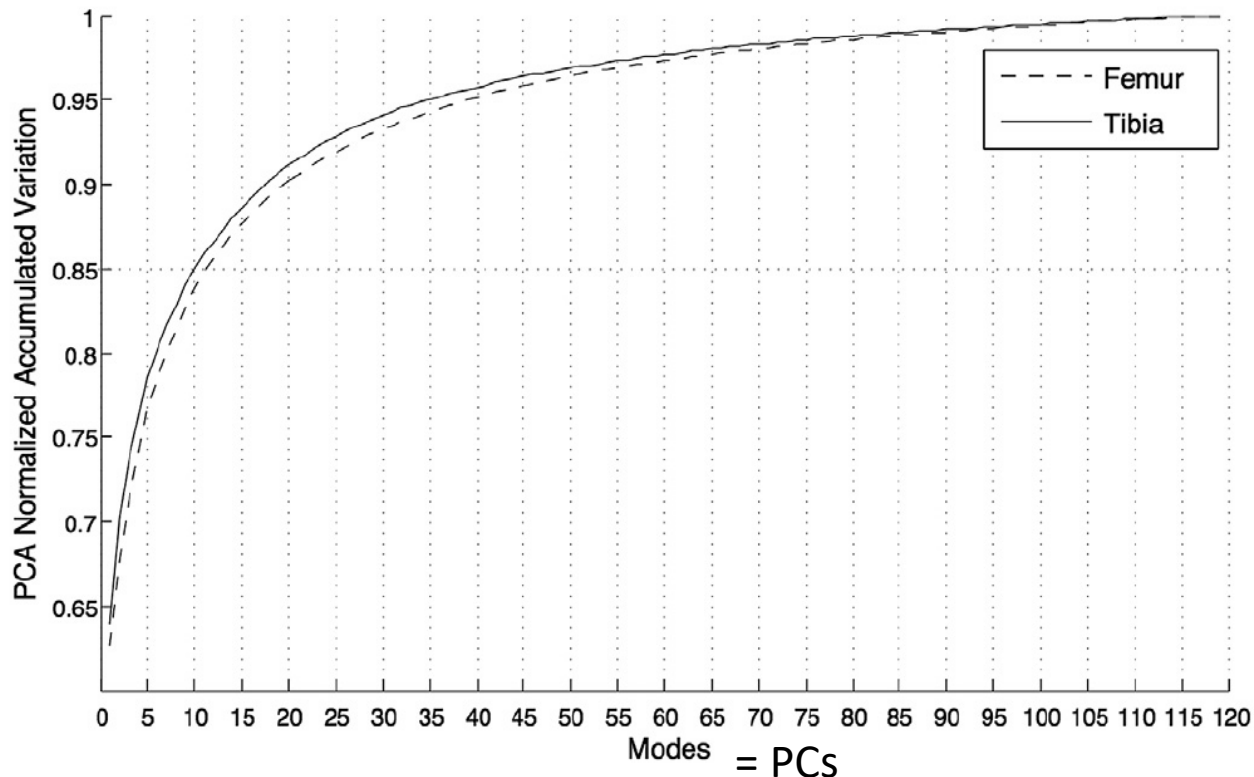
<sup>§</sup> Department of Epidemiology & Biostatistics, University of California, San Francisco, USA



# PCA example – knee shape

- Identify a large number of specific locations on the knee bones (“landmarks”).
- Measure the distance between all pairs of landmarks to quantify the shape.
- Used >8,000 landmarks in the tibia and >11,000 in the femur.
- All told there were >32,000,000 possible distances for the tibia and >62,000,000 for the femur to use as predictors to distinguish ACL vs non-ACL.
- Need to reduce!

# PCA illustration – knee shape



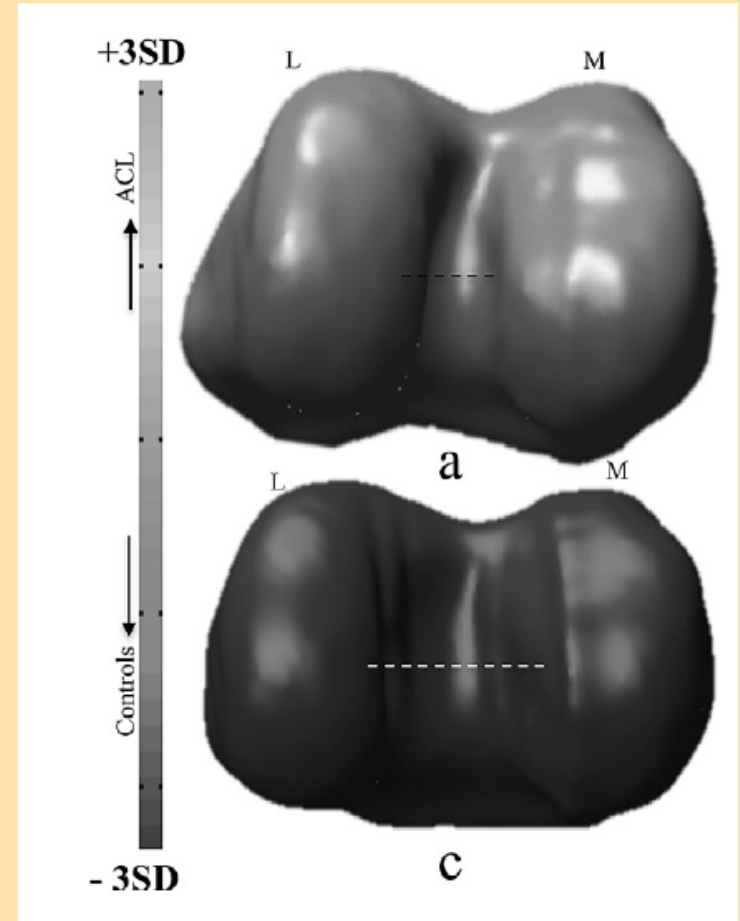
**Fig. 4.** Representation of the model's compactness: cumulative % of the variability expressed in function of the number of modes considered in the model.

For the femur, summarized 90% of the variation in 62 million variables using only about 20 principal components.

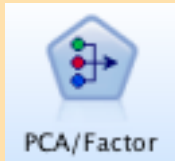
For the tibia, the first mode is related to size, the second to the medial posterior curvature of the tibial plateau, and the third to elevation of the anteromedial tibial plateau.

A number of the PCs (called modes here) had reasonable interpretations.

Though this level of interpretability is by no means a guarantee and is somewhat unusual.

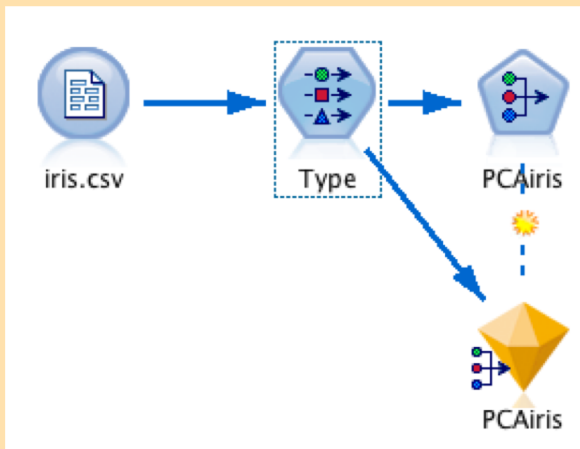


**Fig. 5.** Modeling of the principal component F2 (a, c) and T3 (b, d) of the scale-preserved model. In the first row Mean +3STD. in the second row Mean -3STD. Directions are labeled in the figure. M: medial, L: lateral, A: anterior P: posterior.



# PCA in Modeler

- Revisit iris data
- Just consider 4 input variables – Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width



Preview

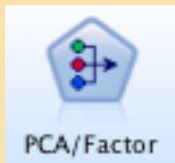
Types Format Annotations

Read Values Clear Values Clear All Values

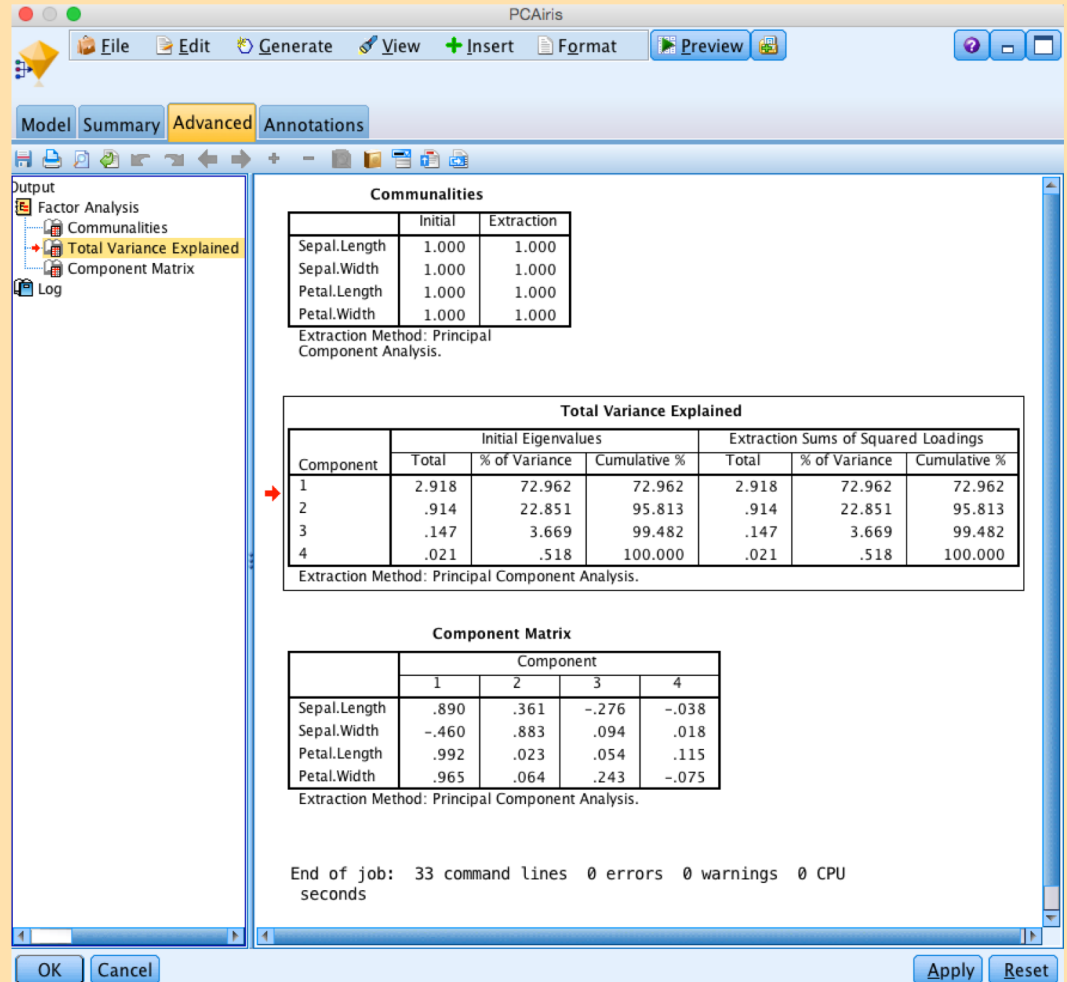
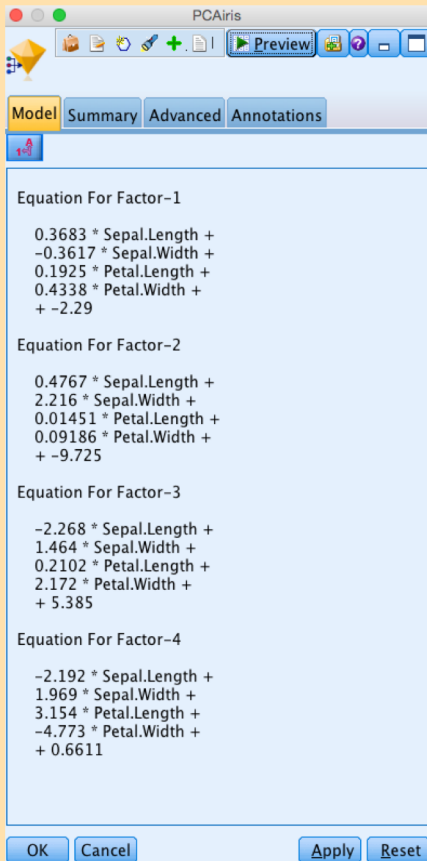
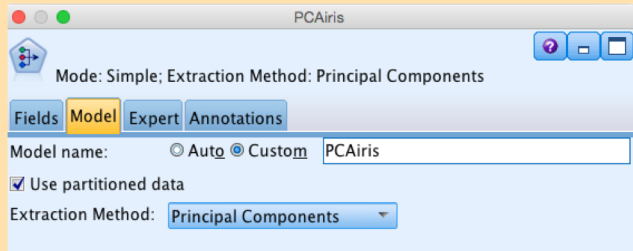
| Field        | Measurement | Values          | Missing | Check | Role      |
|--------------|-------------|-----------------|---------|-------|-----------|
| field1       | Continuous  | [1,150]         |         | None  | Record ID |
| Sepal.Length | Continuous  | [4.3,7.9]       |         | None  | Input     |
| Sepal.Width  | Continuous  | [2.0,4.4]       |         | None  | Input     |
| Petal.Length | Continuous  | [1.0,6.9]       |         | None  | Input     |
| Petal.Width  | Continuous  | [0.1,2.5]       |         | None  | Input     |
| Species      | Nominal     | setosa,versi... |         | None  | None      |

☒ View current fields ☐ View unused field settings

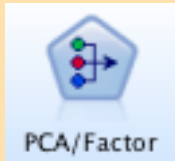
OK Cancel Apply Reset



# PCA in Modeler



Use first 2 or 3 PC's to represent full data



# PCA in Modeler

PCAiris

Mode: Simple; Extraction Method: Principal Components

Fields Model Expert Annotations

Model name: ☐ Auto ☒ Custom PCAiris

☒ Use partitioned data

Extraction Method:

PCAiris

Model Summary Advanced Annotations

Equation For Factor-1

$$0.3683 * \text{Sepal.Length} + -0.3617 * \text{Sepal.Width} + 0.1925 * \text{Petal.Length} + 0.4338 * \text{Petal.Width} + -2.29$$

Equation For Factor-2

$$0.4767 * \text{Sepal.Length} + 2.216 * \text{Sepal.Width} + 0.01451 * \text{Petal.Length} + 0.09186 * \text{Petal.Width} + -9.725$$

Equation For Factor-3

$$-2.268 * \text{Sepal.Length} + 1.464 * \text{Sepal.Width} + 0.2102 * \text{Petal.Length} + 2.172 * \text{Petal.Width} + 5.385$$

Equation For Factor-4

$$-2.192 * \text{Sepal.Length} + 1.969 * \text{Sepal.Width} + 3.154 * \text{Petal.Length} + -4.773 * \text{Petal.Width} + 0.6611$$

OK Cancel Apply Reset

PCAiris

File Edit Generate View Insert Format Preview

Model Summary Advanced Annotations

Output

- Factor Analysis
  - Communalities
  - Total Variance Explained
  - Component Matrix
  - Log

**Communalities**

|              | Initial | Extraction |
|--------------|---------|------------|
| Sepal.Length | 1.000   | 1.000      |
| Sepal.Width  | 1.000   | 1.000      |
| Petal.Length | 1.000   | 1.000      |
| Petal.Width  | 1.000   | 1.000      |

Extraction Method: Principal Component Analysis.

**Total Variance Explained**

| Component | Initial Eigenvalues |               |              | Extraction Sums of Squared Loadings |               |              |
|-----------|---------------------|---------------|--------------|-------------------------------------|---------------|--------------|
|           | Total               | % of Variance | Cumulative % | Total                               | % of Variance | Cumulative % |
| 1         | 2.918               | 72.962        | 72.962       | 2.918                               | 72.962        | 72.962       |
| 2         | .914                | 22.851        | 95.813       | .914                                | 22.851        | 95.813       |
| 3         | .147                | 3.669         | 99.482       | .147                                | 3.669         | 99.482       |
| 4         | .021                | .518          | 100.000      | .021                                | .518          | 100.000      |

Extraction Method: Principal Component Analysis.

**Component Matrix**

|              | Component |      |       |       |
|--------------|-----------|------|-------|-------|
|              | 1         | 2    | 3     | 4     |
| Sepal.Length | .890      | .361 | -.276 | -.038 |
| Sepal.Width  | -.460     | .883 | .094  | .018  |
| Petal.Length | .992      | .023 | .054  | .115  |
| Petal.Width  | .965      | .064 | .243  | -.075 |

Extraction Method: Principal Component Analysis.

End of job: 33 command lines 0 errors 0 warnings 0 CPU seconds

OK Cancel Apply Reset

Correlations of each predictor with each PC

Use first 2 or 3 PC's to represent full data

# Choosing a Machine Learning Method

- Common question: “Which machine learning algorithm should I use?”
- Answer: It depends,
  1. Do you have supervised or unsupervised data?
  2. If supervised, do you have continuous, binary class, or multi-class outcome(s)?
  3. If unsupervised, are you interested in looking for latent groupings or data reduction?
  4. Do you have predictors that are continuous, categorical, ordinal, text-based, or some combination of the above?
  5. How large is your training set?
  6. Do you need a fast algorithm?
  7. Do you have time to explore multiple approaches?
  8. Do you need an interpretable model?
  9. For classification, do you need class probabilities, or will “hard” classes suffice?

1. Do you have supervised or unsupervised data?
  - Supervised data → use classification or regression methods
  - Unsupervised data → use clustering or data reduction algorithms
2. If supervised, do you have continuous, binary class, or multi-class outcomes?
  - Continuous data → use regression methods (e.g. multiple linear regression)
  - Class data → use classification methods (e.g. recursive partitioning regression trees); possible choices of method will be more limited for problems with more than 2 classes for the outcome.
3. If unsupervised, are you interested in looking for natural groupings, or data reduction/simplification?
  - To group data into similar sets → clustering algorithms, e.g. K-nearest neighbors (kNN)
  - To reduce data → use data reduction algorithms (e.g. principal components analysis)



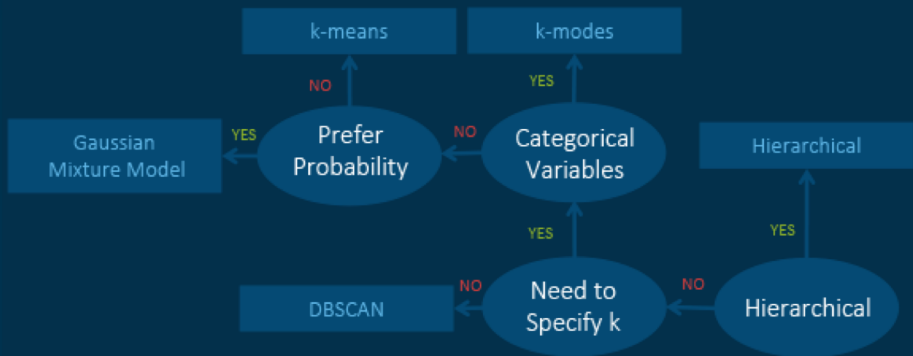
4. Do you have predictors that are continuous, categorical, ordinal, text-based, or some combination of the above?
  - Some algorithms will not work with certain types of predictors
5. How large is your training set?
  - Small data → “high bias/low variance” methods (e.g. Naïve Bayes); higher level of assumptions – converges faster if assumptions hold
  - Large data → “low bias/high variance” methods (e.g. kNN); lower level of assumptions – over-fits with small datasets, but lower asymptotic error (i.e. less error for super-large datasets)
6. Do you need a fast algorithm? (e.g. for real-time work)
  - More complex algorithms can be relatively slow, especially if there are algorithm parameters to be optimized (e.g. Support Vector Machines) – not likely to be an issue in most medicine datasets
7. Do you have time to explore multiple approaches?
  - If so, you may as well try many and see which does “best” for your data (making sure you are dutifully evaluating on independent data) or use ensemble methods

8. Do you need an interpretable model?
9. For classification, do you need class probabilities, or will “hard” classes suffice?
  - Some classification methods don’t provide class probabilities, so if you want probabilities your options are less

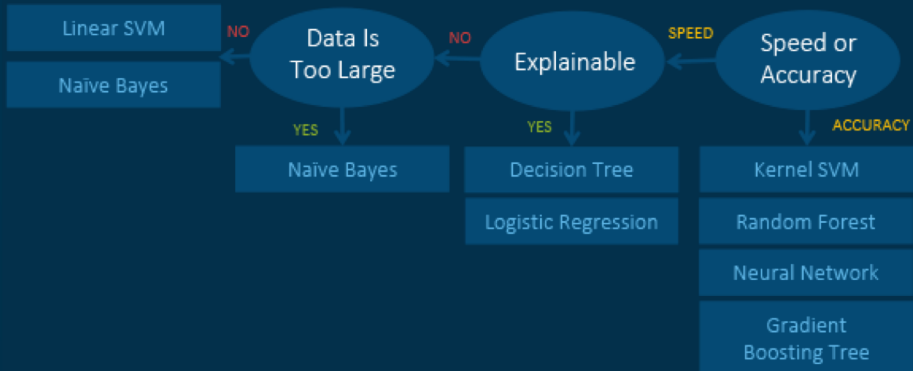
There are also many “cheat sheets” for choosing machine learning algorithms on the internet, but you can see there is a lot of variability...

# Machine Learning Algorithms Cheat Sheet

## Unsupervised Learning: Clustering

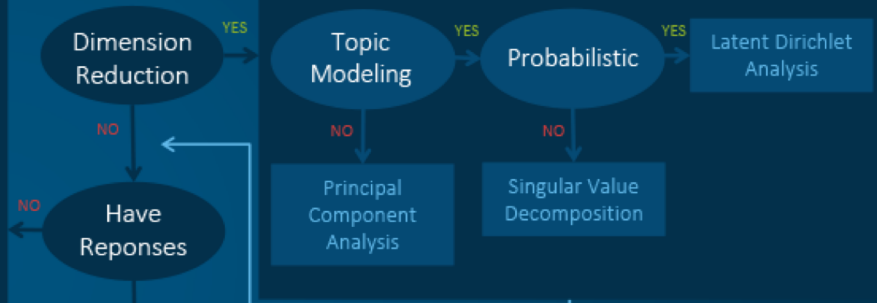


## Supervised Learning: Classification

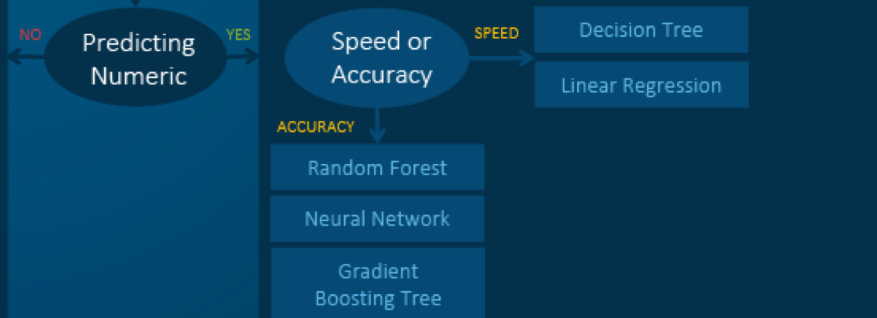


# START

## Unsupervised Learning: Dimension Reduction



## Supervised Learning: Regression



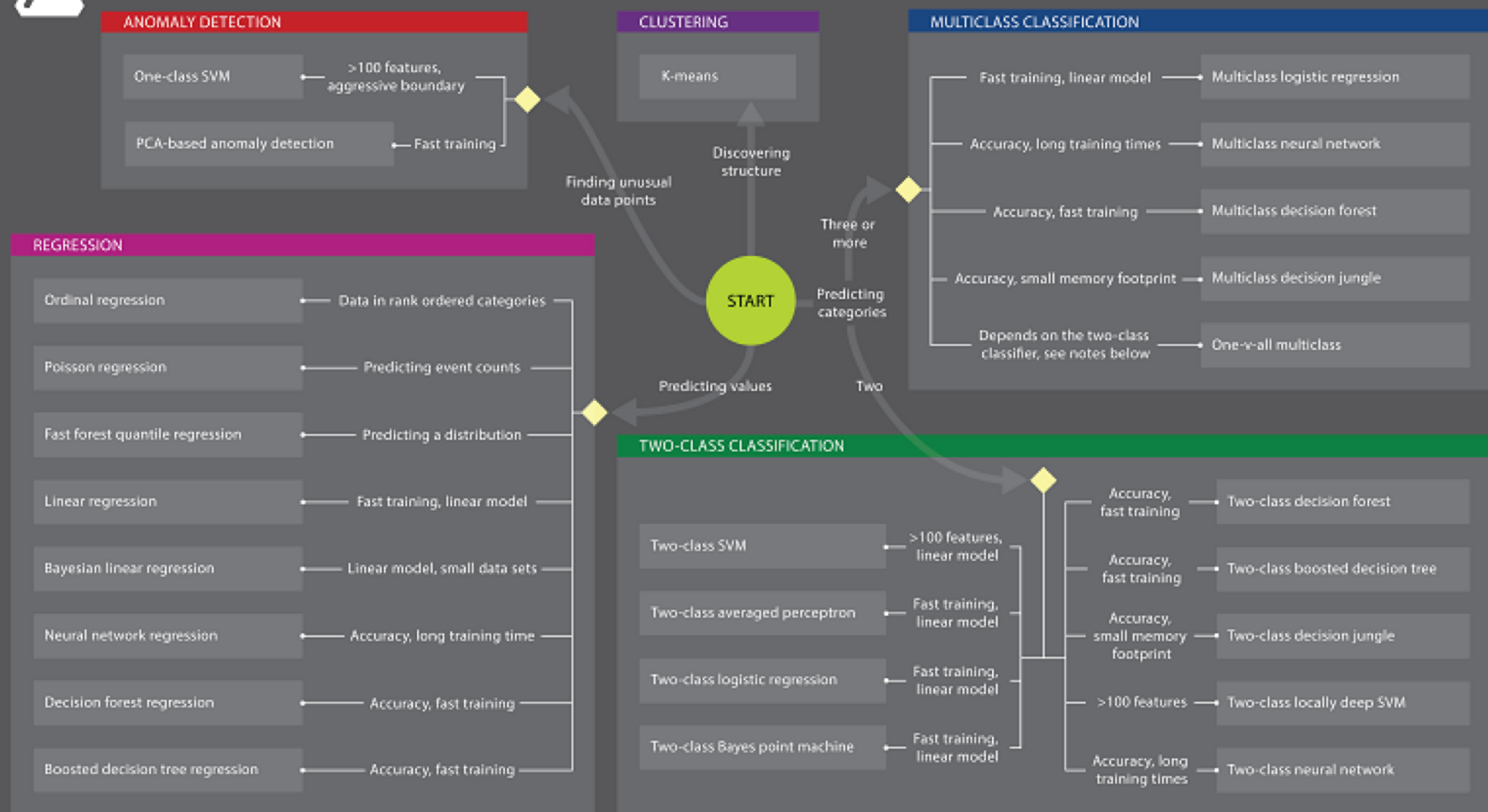
START





# Microsoft Azure Machine Learning: Algorithm Cheat Sheet

This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.



# Our own quick guide

| Supervised learning methods                                                                       |            |                                                                                |                                                                                        |                                                                                        |
|---------------------------------------------------------------------------------------------------|------------|--------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Interpretability                                                                                  | Model type | Outcome type                                                                   |                                                                                        |                                                                                        |
|                                                                                                   |            | Numeric                                                                        | Binary                                                                                 | Category                                                                               |
| More<br><br>Less | Regression | Stepwise linear regression<br>LASSO linear regression                          | Stepwise logistic regression<br>LASSO logistic regression                              | Stepwise multinomial regression<br>LASSO multinomial regression                        |
|                                                                                                   | Bayes      | --                                                                             | Naïve Bayes<br>Tree augmented naïve Bayes<br>Bayesian Networks                         | Naïve Bayes<br>Tree augmented naïve Bayes<br>Bayesian Networks                         |
|                                                                                                   | Tree       | Regression tree<br>Averaged regression tree (bagging, boosting, random forest) | Classification tree<br>Averaged classification tree (bagging, boosting, random forest) | Classification tree<br>Averaged classification tree (bagging, boosting, random forest) |
|                                                                                                   | Other      | --                                                                             | Linear support vector machine*                                                         | --                                                                                     |
|                                                                                                   |            | --                                                                             | Support vector machine*                                                                | --                                                                                     |
|                                                                                                   |            | --                                                                             | Neural Network*                                                                        | --                                                                                     |
|                                                                                                   |            |                                                                                |                                                                                        |                                                                                        |
|                                                                                                   |            |                                                                                |                                                                                        |                                                                                        |
|                                                                                                   |            |                                                                                |                                                                                        |                                                                                        |
|                                                                                                   |            |                                                                                |                                                                                        |                                                                                        |

\*Only gives predicted categories, not predicted outcome probabilities

# Ensemble Methods

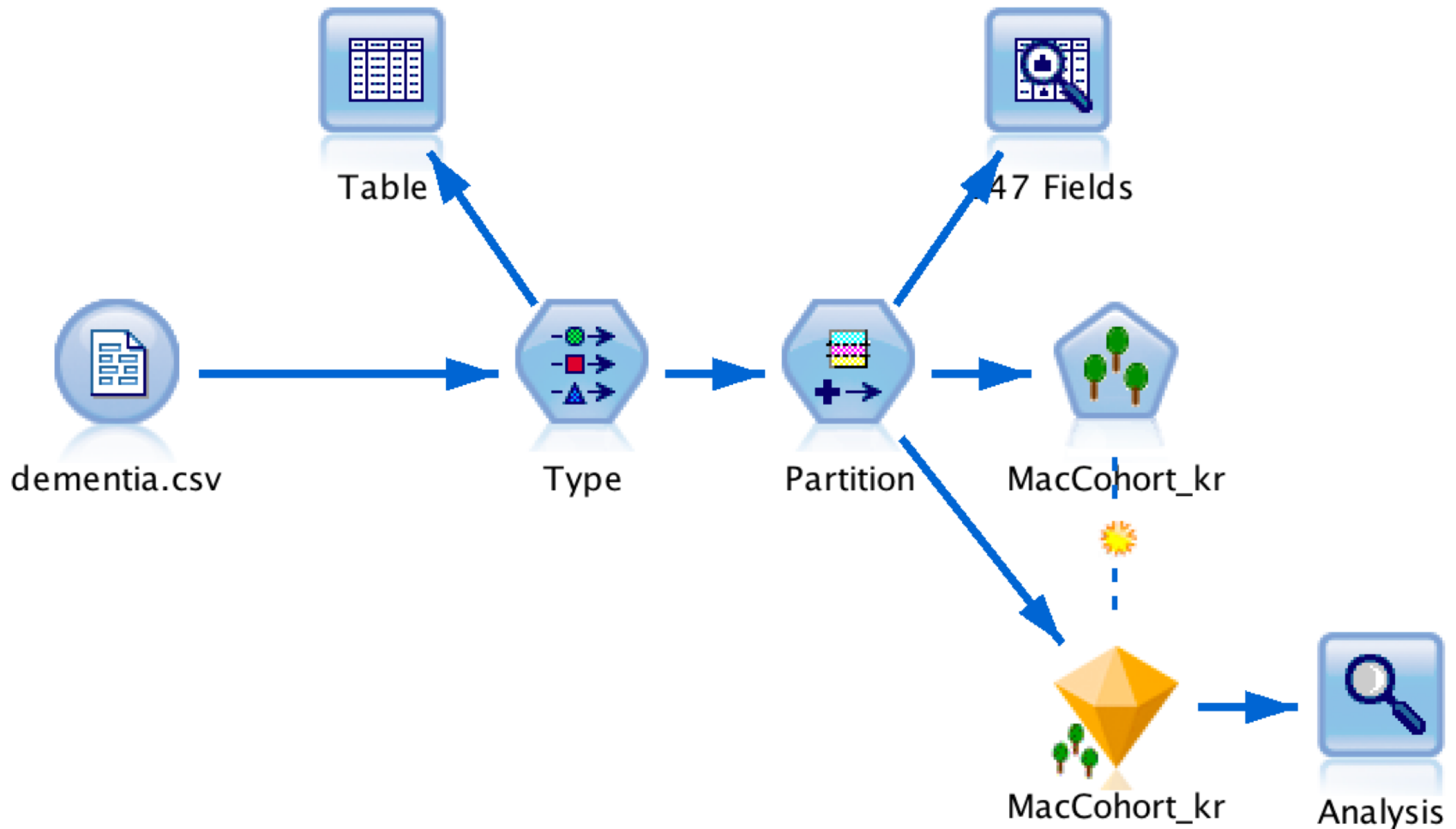
- Run a set of classifiers and allow them to “vote” on predictions.
- **Advantage:**
  - improvement in predictive accuracy.
- **Disadvantages:**
  - more computation
  - it is difficult to understand an ensemble of classifiers
- Examples: bagging, boosting, and stacking

# Bagging

- Simplest way of combining predictions that belong to the same type.
- Combine via voting or averaging
- Each model receives equal weight
- Basic idea of bagging:
  - Re-sample several training sets of size  $n$  from the dataset (instead of just using the one training set of size  $n$ )
  - Build a classifier for each training set
  - Combine the classifier's predictions (majority vote)
- This improves performance in almost all cases if learning scheme is *unstable* (e.g. decision trees depend heavily on where the first cut occurs – random forests)



# Random Forests in Modeler



# Partition Node



Partition

**Generate** **Preview**

**Settings** Annotations

Partition field:

Partitions: ☒ Train and test ☐ Train, test and validation

Training partition size:  Label:  Value =

Testing partition size:  Label:  Value =

Validation partition size:  Label:  Value =

Total size: 100%

Values: ☐ Use system-defined values ("1", "2" and "3")  
☒ Append labels to system-defined values  
☐ Use labels as values

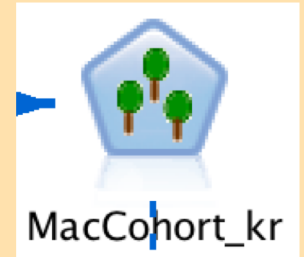
☒ Repeatable partition assignment

Seed:  **Generate**

☐ Use unique field to assign partitions:

**OK** **Cancel** **Apply** **Reset**

# Random Trees Node



MacCohort\_kr

Fields Build Options Model Options Annotations

Basics  
Costs  
Advanced

Model building

Number of models to build: 100

Sample size: 1.0

☐ Handle imbalanced data

☐ Use weighted sampling for variable selection

Tree Growth

Maximum number of nodes: 10000

Maximum tree depth: 10

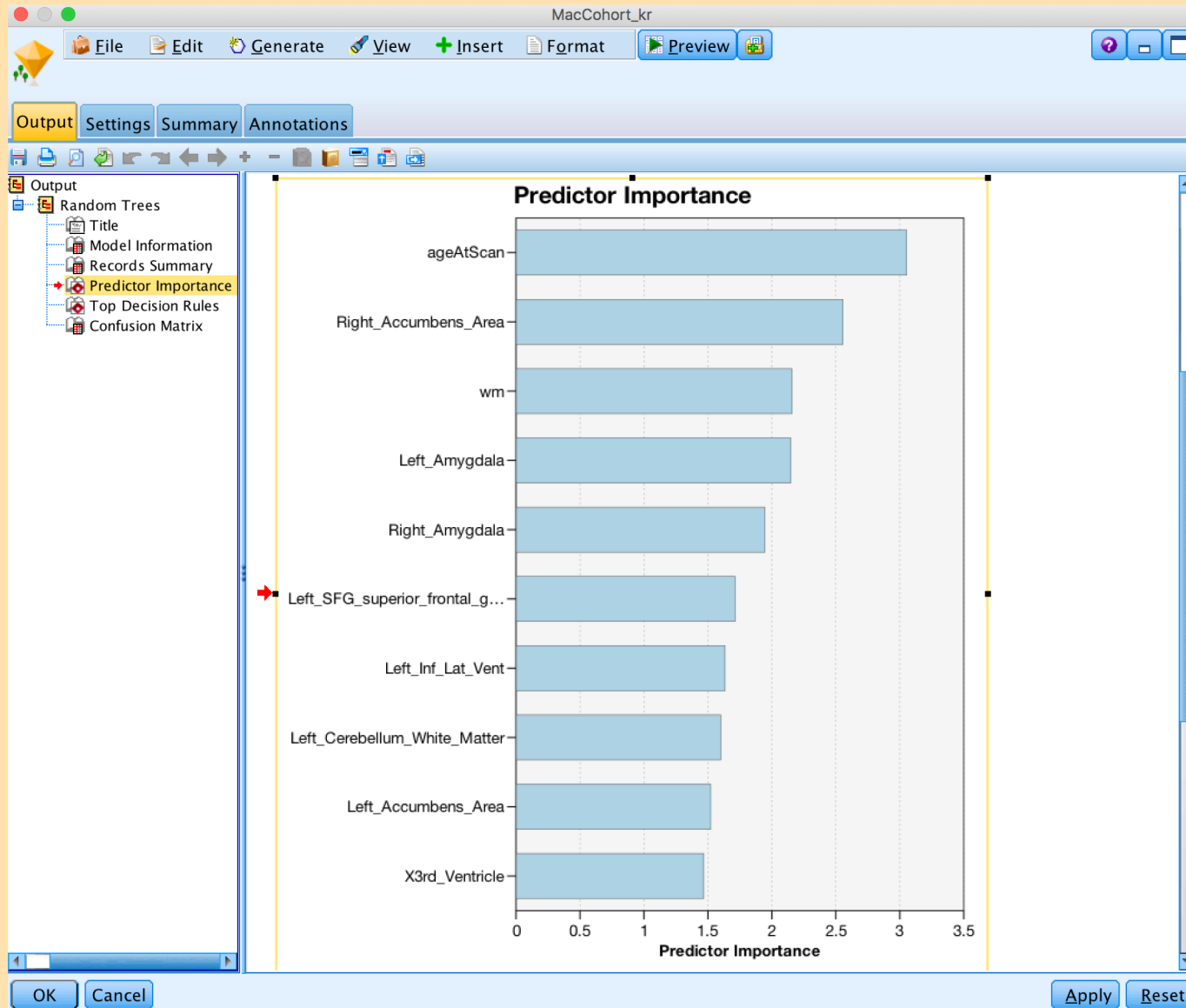
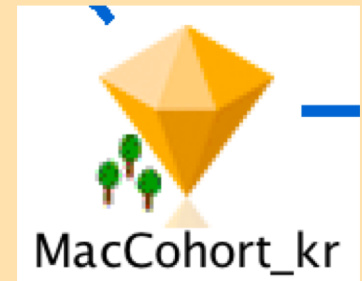
Minimum child node size: 5

☐ Specify number of predictors to use for splitting 1

☒ Stop building when accuracy can no longer be improved

OK Run Cancel Apply Reset

# Golden Nugget



# Analysis Node



Analysis

Analysis of [MacCohort\_kr]

File

Edit

?

X

Analysis

Annotations

Collapse All

Expand All

Results for output field MacCohort\_kr

Comparing \$R-MacCohort\_kr with MacCohort\_kr

| 'Partition' | 1_Training |        | 2_Testing |        |
|-------------|------------|--------|-----------|--------|
| Correct     | 465        | 97.69% | 117       | 63.59% |
| Wrong       | 11         | 2.31%  | 67        | 36.41% |
| Total       | 476        |        | 184       |        |

Coincidence Matrix for \$R-MacCohort\_kr (rows show actuals)

'Partition' = 1\_Training

|            | AD  | CONTROL | PSP | bvFTD | nfPPAunspc | svPPA |
|------------|-----|---------|-----|-------|------------|-------|
| AD         | 106 | 1       | 0   | 1     | 1          | 1     |
| CONTROL    | 3   | 229     | 0   | 0     | 0          | 0     |
| PSP        | 0   | 1       | 27  | 0     | 0          | 0     |
| bvFTD      | 0   | 1       | 0   | 49    | 0          | 0     |
| nfPPAunspc | 0   | 1       | 0   | 0     | 24         | 0     |
| svPPA      | 0   | 0       | 0   | 1     | 0          | 30    |

'Partition' = 2\_Testing

|            | AD | CONTROL | PSP | bvFTD | nfPPAunspc | svPPA |
|------------|----|---------|-----|-------|------------|-------|
| AD         | 25 | 11      | 0   | 4     | 1          | 0     |
| CONTROL    | 9  | 73      | 0   | 1     | 0          | 0     |
| PSP        | 4  | 8       | 2   | 2     | 0          | 0     |
| bvFTD      | 2  | 3       | 0   | 12    | 1          | 0     |
| nfPPAunspc | 2  | 7       | 1   | 3     | 0          | 0     |
| svPPA      | 7  | 0       | 0   | 1     | 0          | 5     |

OK

# Boosting

- Also uses voting/averaging but models are weighted based on their performance
- Iterative procedure: new models are influenced by performance of previously built ones
  - New model is encouraged to become expert for instances classified incorrectly by earlier models
  - Intuitive justification: models should be experts that complement each other
- There are many variants of boosting, one of the most celebrated is *AdaBoost*

# Stacking

- Uses a “*meta-learner*” instead of voting to combine predictions across models
  - Predictions of initial models (*level-0 models*) are used as input for meta-learner (*level-1 model*)
- Individual initial models can be of many different types
- Complex methodology to build meta-learner by learning again from the initial set of models (machine learning on top of machine learning)

**A few more classification  
methods in brief...**

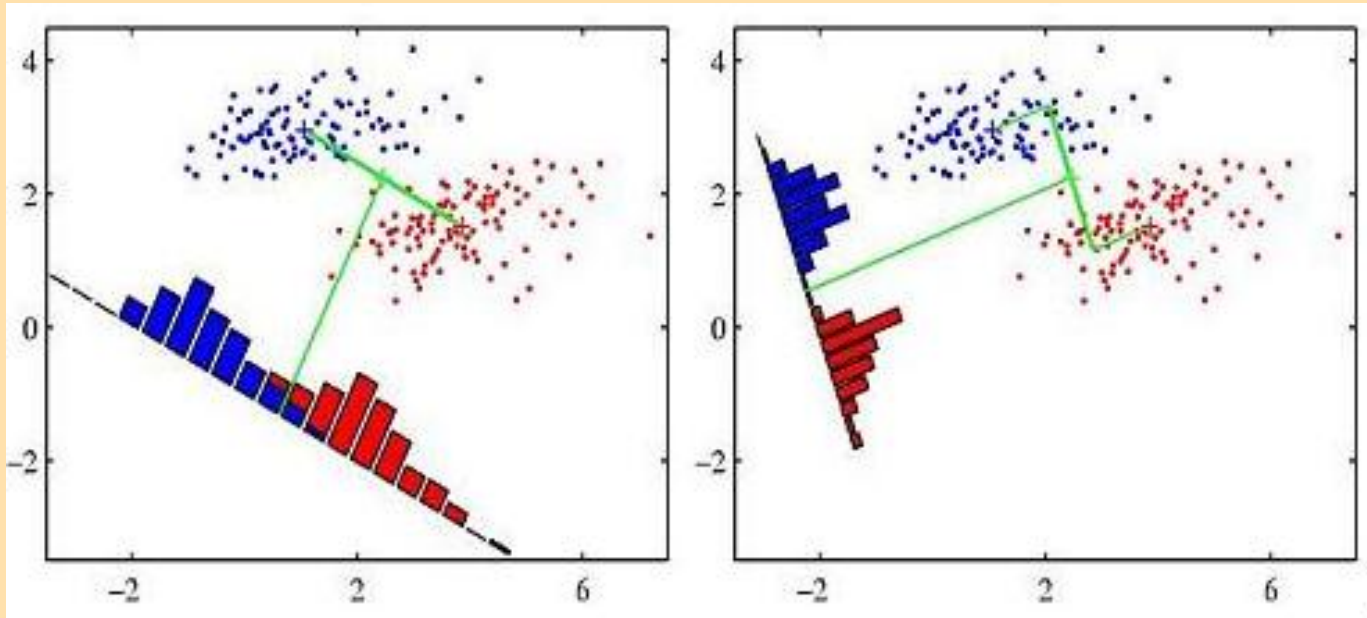
**First, Discriminant Analysis**



# Linear discriminant analysis (LDA)

- Consider 2 or more classes with correct class known for each subject.
- Objective: find a linear combination  $a_1x_1 + a_2x_2 + \dots + a_px_p$  of the variables that maximizes the ratio of the between-class to within-class variance – this is the first linear discriminant function.
- The 2<sup>nd</sup> linear discriminant finds the best linear combination *orthogonal (perpendicular)* to the first to maximize the same ratio etc.

# Discriminant analysis

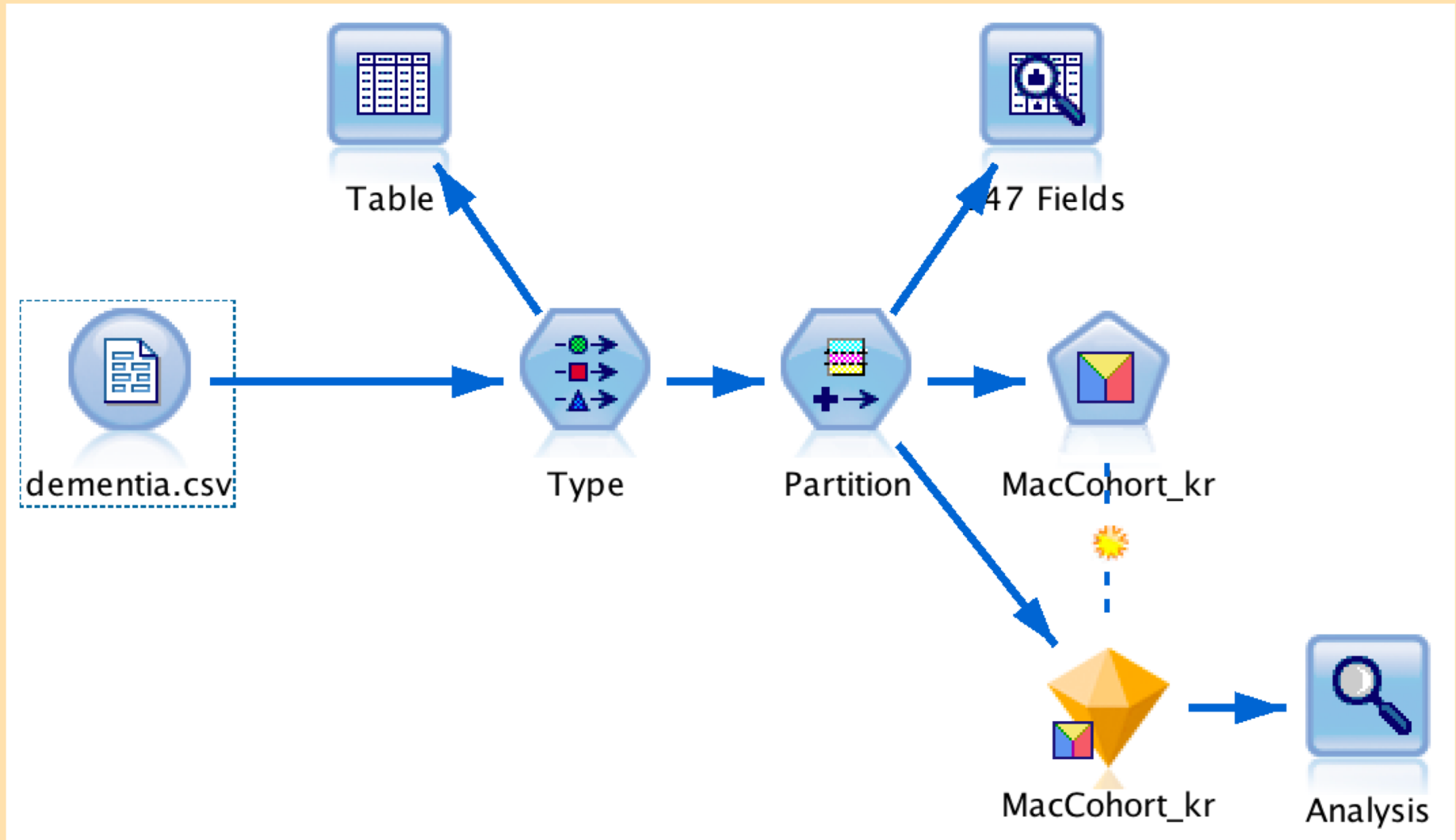


- Gaussian (normal) probability densities: linear discriminant analysis (LDA) based on assuming within-class covariances are equal
- Provides the probability of belonging to each class, not just a classification.

# Discriminant analysis

- Long standing method (goes back to Ronald Fisher – see lab)
- We are more interested in summary patterns of what is important in determining classification
- Can extend to quadratic discriminant analysis (QDA) and more complex non-linear forms

# Linear Discriminant Analysis in Modeler



# Partition Node



Partition

Generate Preview

**Settings** Annotations

Partition field:

Partitions: ☒ Train and test ☐ Train, test and validation

Training partition size:  Label:  Value =

Testing partition size:  Label:  Value =

Validation partition size:  Label:  Value =

Total size: 100%

Values: ☐ Use system-defined values ("1", "2" and "3")  
☒ Append labels to system-defined values  
☐ Use labels as values

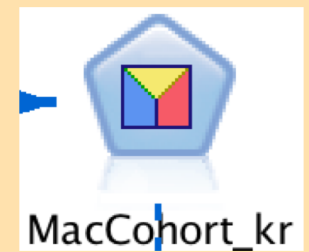
☒ Repeatable partition assignment

Seed:  Generate

☐ Use unique field to assign partitions:

OK Cancel Apply Reset

# LDA Node



MacCohort\_kr

Fields **Model** Expert Analyze Annotations

Model name: ☒ Auto ☐ Custom

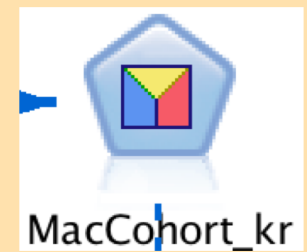
☒ Use partitioned data

☒ Build model for each split

Method:

OK  Cancel

# LDA Node



MacCohort\_kr

Fields Model **Expert** Analyze Annotations

Mode: ☐ Simple ☒ Expert

Prior probabilities:

☒ All groups equal ☐ Compute from group sizes

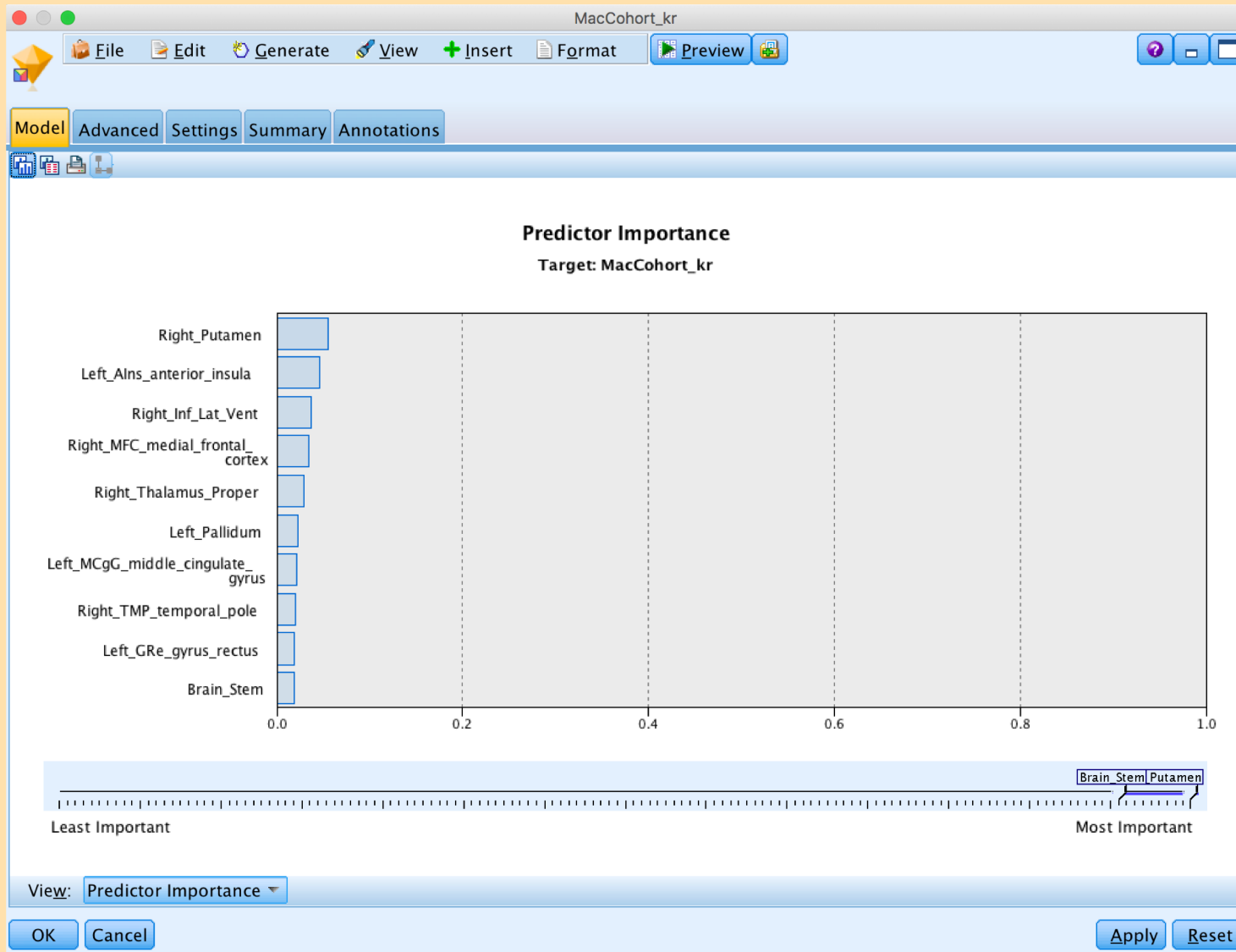
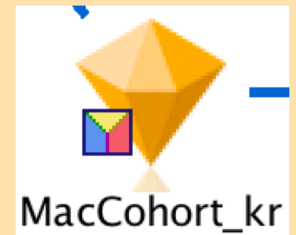
Use covariance matrix:

☒ Within-groups ☐ Separate-groups

Output... Stepping...

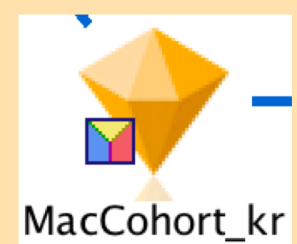
OK **Run** Cancel Apply Reset

# Golden Nugget





# Golden Nugget



Model Advanced Settings Summary Annotations

Output Log Discriminant Analysis Case Pro Group Statistics Analysis 1 Variables Fa Summary of Eigenval Wilks' La Standardized Structure Ma Functions at Log

|   |                   |     |       |      |
|---|-------------------|-----|-------|------|
| 5 | .653 <sup>a</sup> | 6.0 | 100.0 | .629 |
|---|-------------------|-----|-------|------|

a. First 5 canonical discriminant functions were used in the analysis.

**Wilks' Lambda**

| Test of Function(s) | Wilks' Lambda | Chi-square | df  | Sig. |
|---------------------|---------------|------------|-----|------|
| 1 through 5         | .005          | 2005.849   | 705 | .000 |
| 2 through 5         | .033          | 1314.801   | 560 | .000 |
| 3 through 5         | .117          | 826.448    | 417 | .000 |
| 4 through 5         | .325          | 432.248    | 276 | .000 |
| 5                   | .605          | 193.356    | 137 | .001 |

**Standardized Canonical Discriminant Function Coefficients**

|                               | Function |       |        |       |       |
|-------------------------------|----------|-------|--------|-------|-------|
|                               | 1        | 2     | 3      | 4     | 5     |
| gm                            | -.592    | -.253 | -1.161 | -.117 | .252  |
| wm                            | .832     | -.249 | 1.230  | -.277 | -.684 |
| csf                           | -.003    | .472  | .185   | .697  | .422  |
| ageAtScan                     | .739     | -.018 | .386   | .054  | -.419 |
| Educ                          | .124     | -.160 | .035   | .046  | -.024 |
| X3rd_Ventricle                | -.321    | .229  | .130   | .289  | .015  |
| X4th_Ventricle                | -.509    | .310  | .153   | -.786 | -.041 |
| Right_Accumbens_Area          | -.627    | -.605 | -.524  | -.021 | .191  |
| Left_Accumbens_Area           | .409     | .625  | .637   | -.312 | -.238 |
| Right_Amygdala                | -.082    | .161  | -.163  | -.017 | -.304 |
| Left_Amygdala                 | .580     | .333  | .224   | .370  | .057  |
| Brain_Stem                    | .232     | -.206 | -.214  | .836  | -.143 |
| Right_Caudate                 | .233     | .604  | .628   | .177  | .119  |
| Left_Caudate                  | -.457    | -.660 | .010   | -.105 | -.055 |
| Right_Cerebellum_Exterior     | .003     | .279  | .158   | -.250 | .030  |
| Left_Cerebellum_Exterior      | .414     | -.544 | -.155  | .317  | -.373 |
| Right_Cerebellum_White_Matter | -.094    | -.267 | -.180  | .925  | -.208 |

# Analysis Node

Analysis of [MacCohort\_kr]

File

Edit

?

X

Analysis

Annotations

Collapse All

Expand All

Results for output field MacCohort\_kr

Comparing \$D-MacCohort\_kr with MacCohort\_kr

| 'Partition' | 1_Training |        | 2_Testing |       |
|-------------|------------|--------|-----------|-------|
| Correct     | 449        | 94.33% | 131       | 71.2% |
| Wrong       | 27         | 5.67%  | 53        | 28.8% |
| Total       | 476        |        | 184       |       |

Coincidence Matrix for \$D-MacCohort\_kr (rows show actuals)

| 'Partition' = 1_Training | AD  | CONTROL | PSP | bvFTD | nfPPAunspc | svPPA |
|--------------------------|-----|---------|-----|-------|------------|-------|
| AD                       | 102 | 5       | 1   | 0     | 2          | 0     |
| CONTROL                  | 3   | 225     | 2   | 1     | 1          | 0     |
| PSP                      | 0   | 0       | 27  | 1     | 0          | 0     |
| bvFTD                    | 1   | 2       | 1   | 44    | 1          | 1     |
| nfPPAunspc               | 3   | 1       | 0   | 0     | 21         | 0     |
| svPPA                    | 0   | 0       | 0   | 1     | 0          | 30    |

| 'Partition' = 2_Testing | AD | CONTROL | PSP | bvFTD | nfPPAunspc | svPPA |
|-------------------------|----|---------|-----|-------|------------|-------|
| AD                      | 30 | 8       | 1   | 1     | 1          | 0     |
| CONTROL                 | 1  | 74      | 6   | 0     | 2          | 0     |
| PSP                     | 3  | 5       | 5   | 1     | 2          | 0     |
| bvFTD                   | 4  | 3       | 0   | 10    | 0          | 1     |
| nfPPAunspc              | 1  | 4       | 4   | 0     | 3          | 1     |
| svPPA                   | 2  | 0       | 0   | 1     | 1          | 9     |

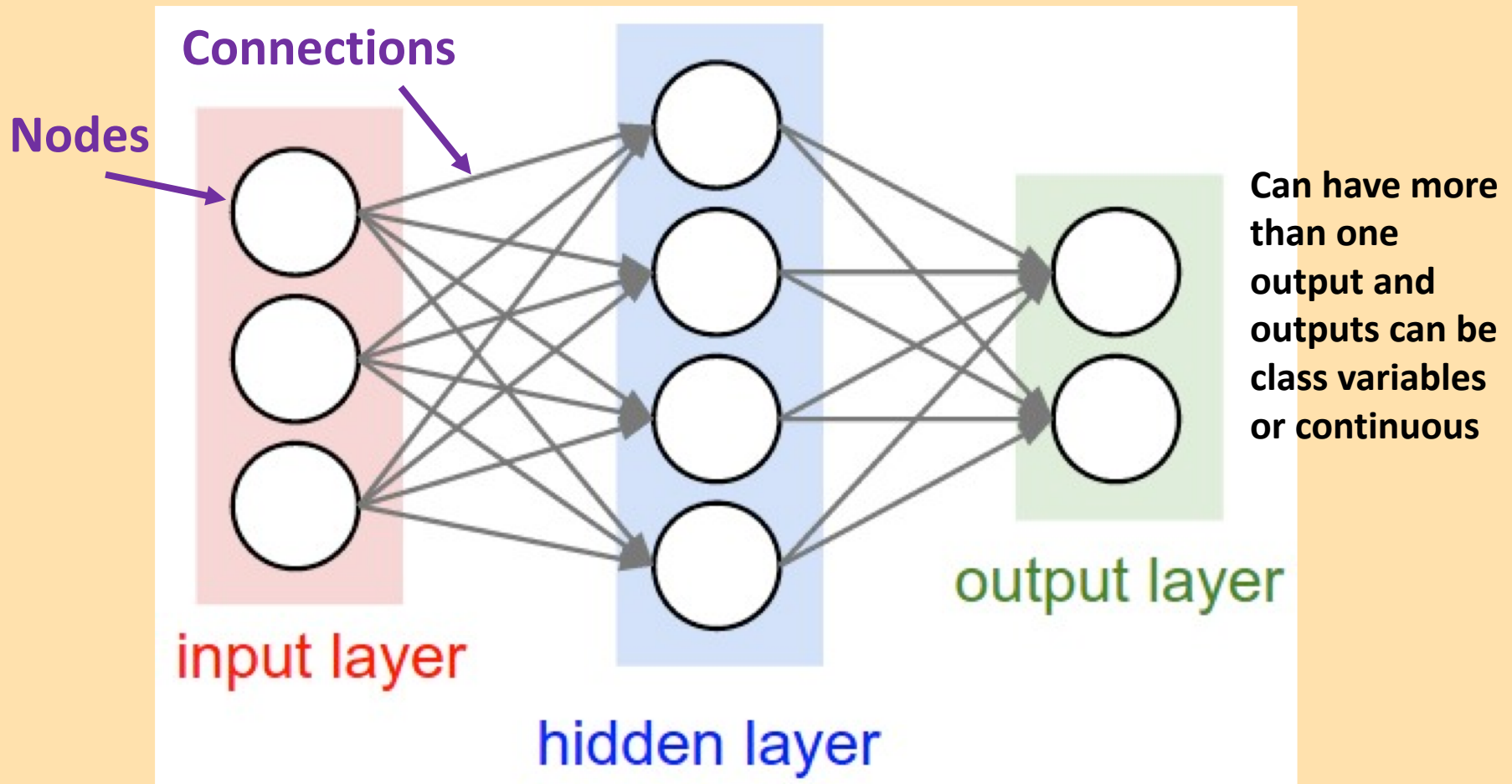
OK

The image part with relationship ID rtd2 was not found in the file.

# Neural networks (neural nets)

- More properly called an ‘artificial’ neural network
- “A computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.”
- A *learning rule* is developed to “learn by example” so that the weights of connections can be modified based on the observed data
- At one time considered as a model for the brain...

# Neural networks (neural nets)



Underlying math is complex, but every node is a complex function of it's inputs, and the parameters/weights of the function are learned from the training data

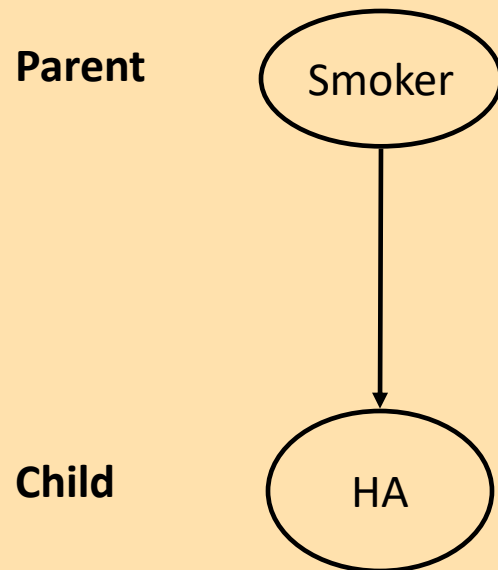
# **Bayesian Networks (Bayes Nets)**

# Bayesian Networks

- Looks for pathways or complex relationships between variables
- Unlike most classification methods, Bayes Nets actually has a “causal” interpretation
- Bayesian networks do not have pre-defined outcomes/predictors, but determine order/structure among a set of variables

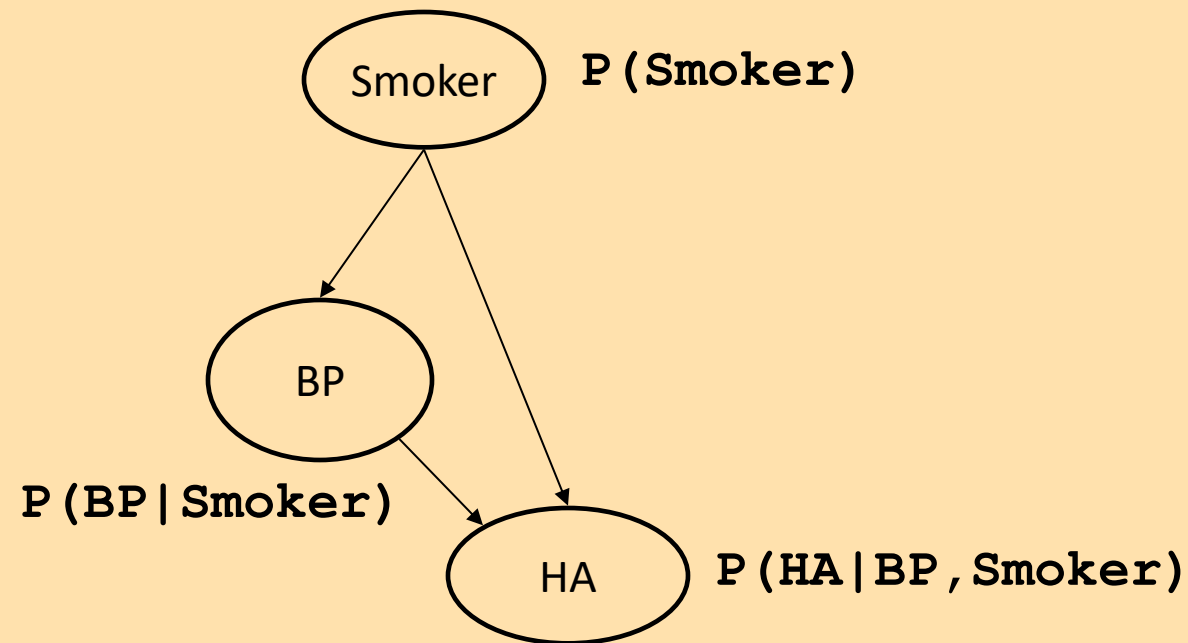
# Components of Bayes Nets

- Each node is a variable with an associated probability distribution
- Each arc (arrow) denotes conditional dependence



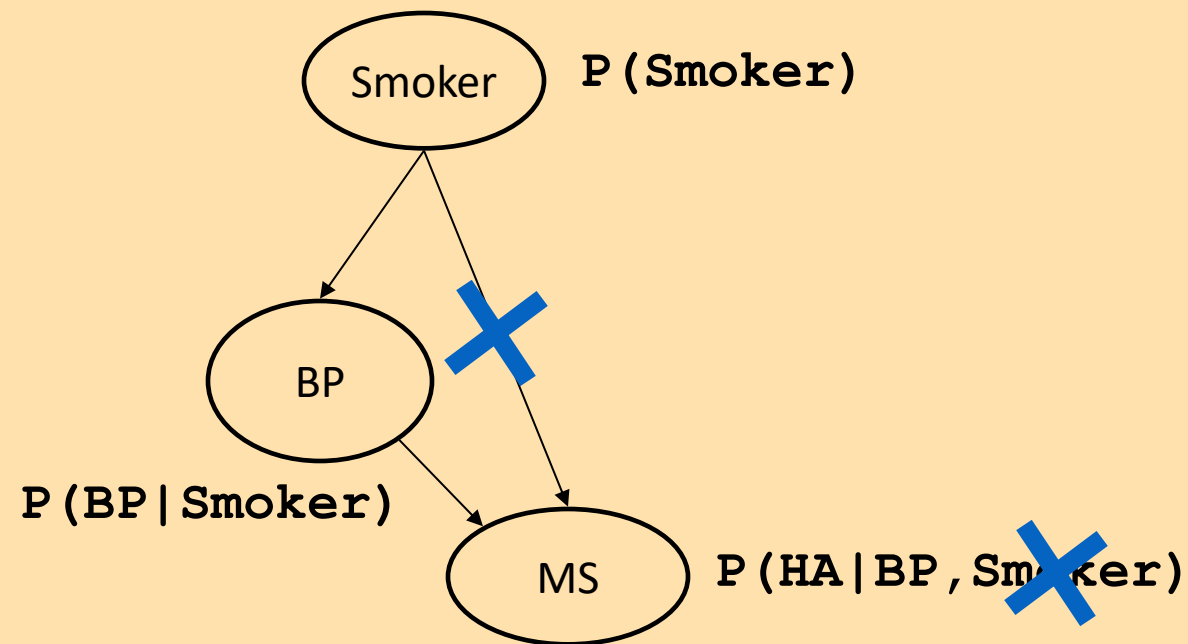
HA = event of Heart Attack

# Small illustration





# Small illustration

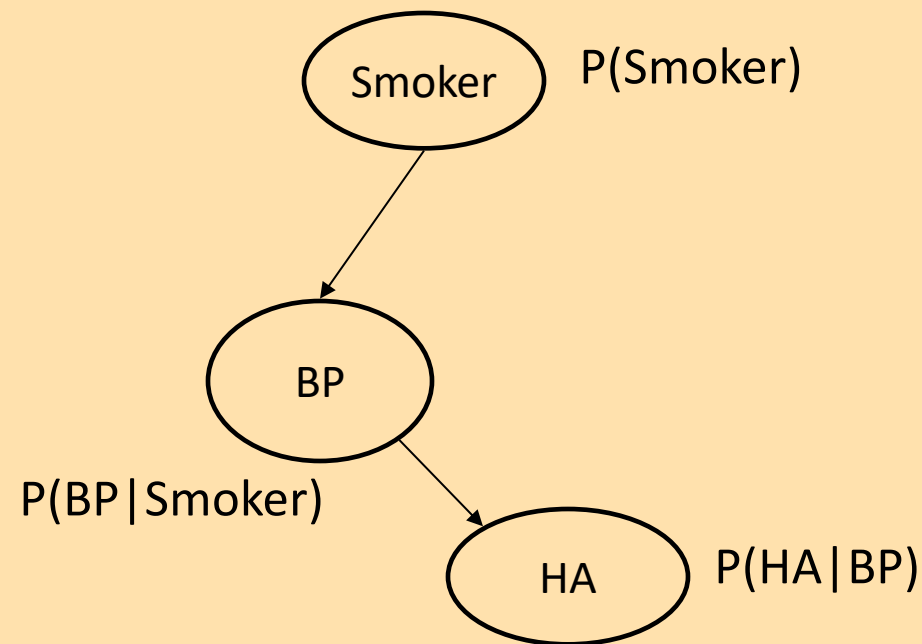


# Conditional independence

Have determined risk of heart attack (HA) is *conditionally independent* of Smoker given BP

Smoker, BP, and HA are all related to each other, but once the value of BP is settled, Smoker and HA become independent of each other

Reduced model is simpler to estimate and interpret



# Learning Bayes Nets

- Compute optimal network among a set of nodes
- Requires a “score” of what is “good”
- Full evaluation = “NP-hard” problem (i.e. computationally impossible for large problems)
- Employ greedy type algorithms
- Perhaps have multiple starts

# Example Multiple Sclerosis – variable set

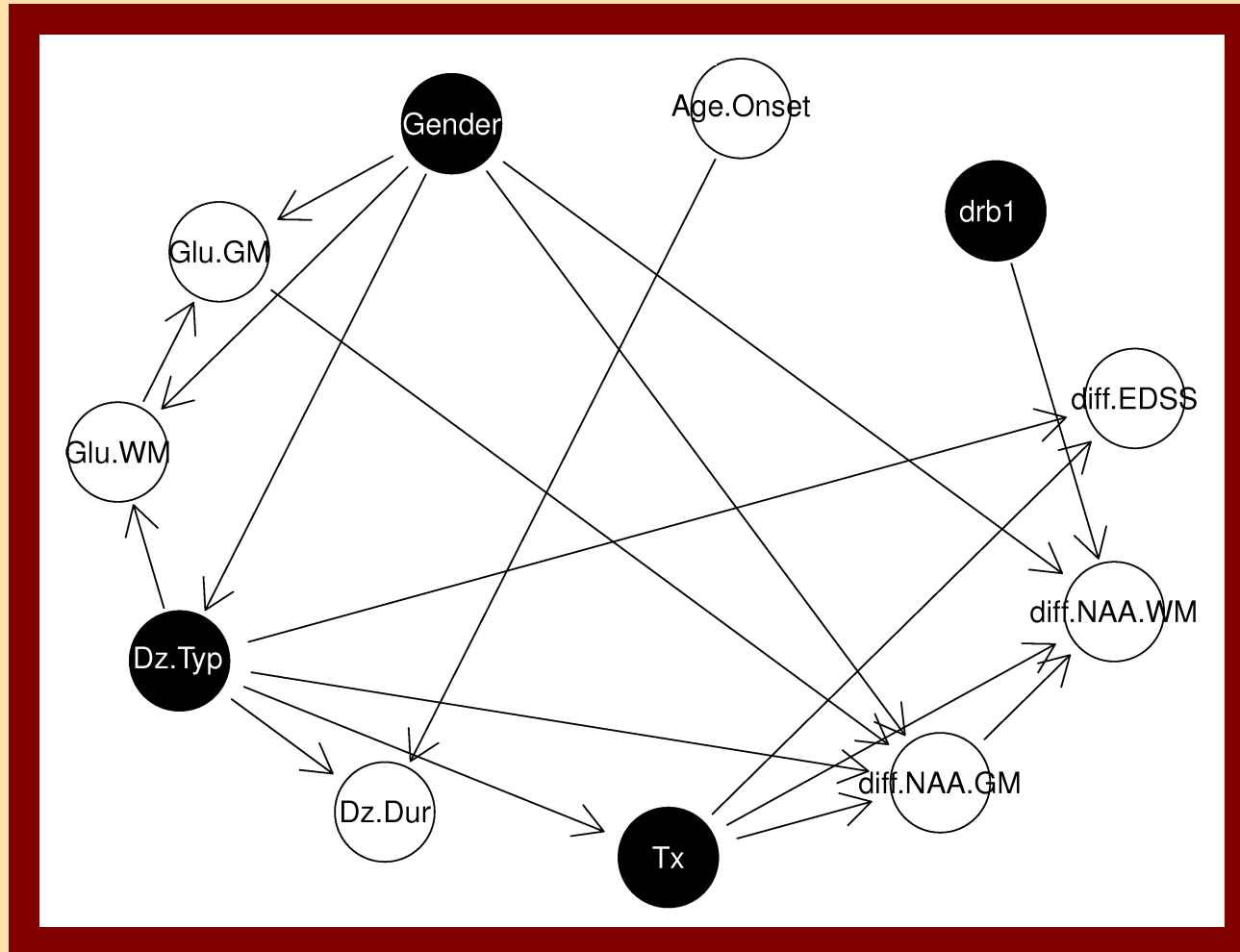
## Categorical

- Treatment Y/N (Tx)
- Gender (Gender)
- Disease Type (Dz.Type)
- HLA<sub>DRB1</sub>\*1501 (drb1)

## Continuous

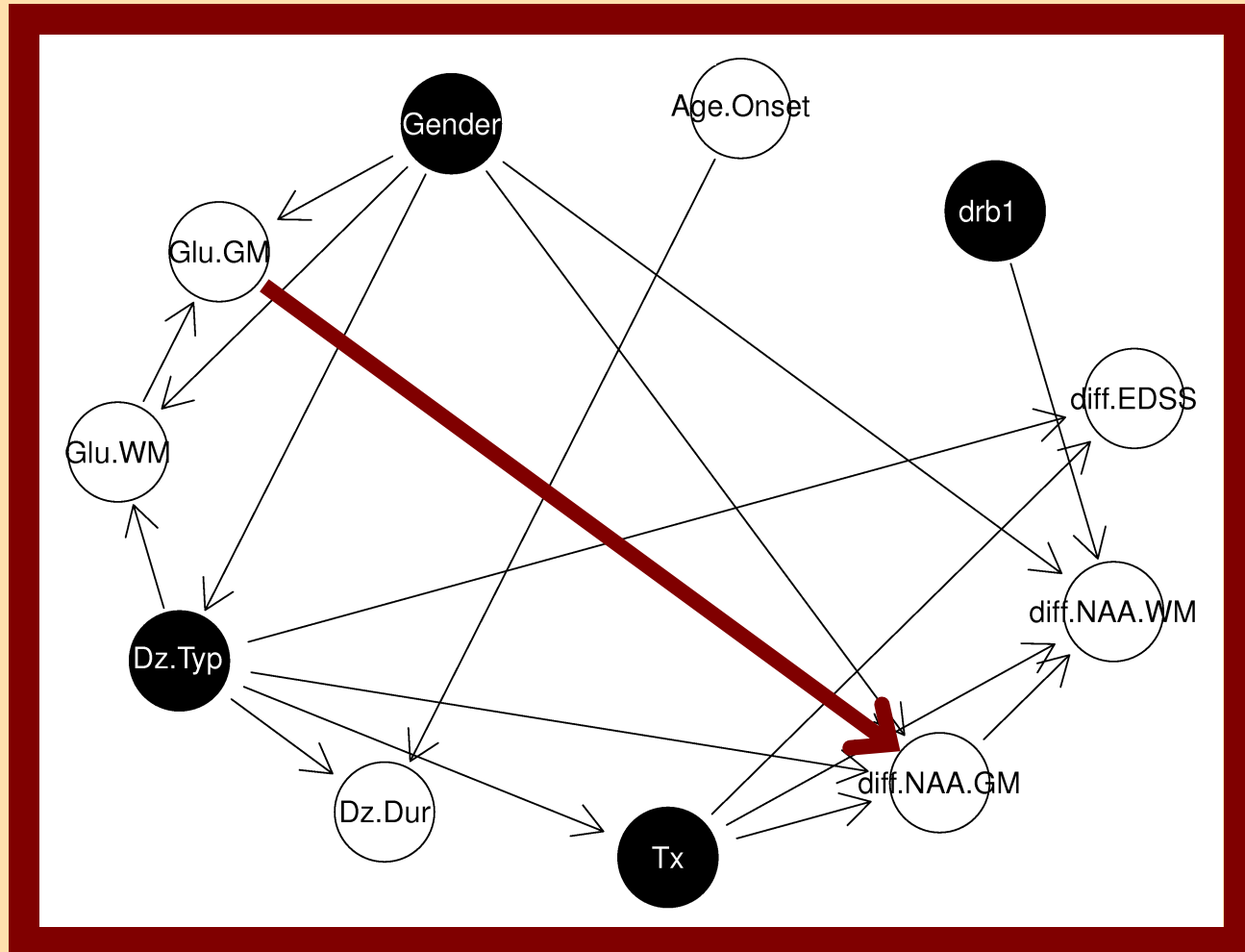
- Disease Duration (Dz.Dur)
- Age at Disease Onset (Age.Onset)
- Baseline Glu GM (Glu.GM)
- Baseline Glu WM (Glu.WM)
- 1 year change in NAA GM (diff.NAA.GM)
- 1 year change in NAA WM (diff.NAA.WM)
- 1 year change in EDSS (diff.EDSS)

# Fitted Bayes' Net

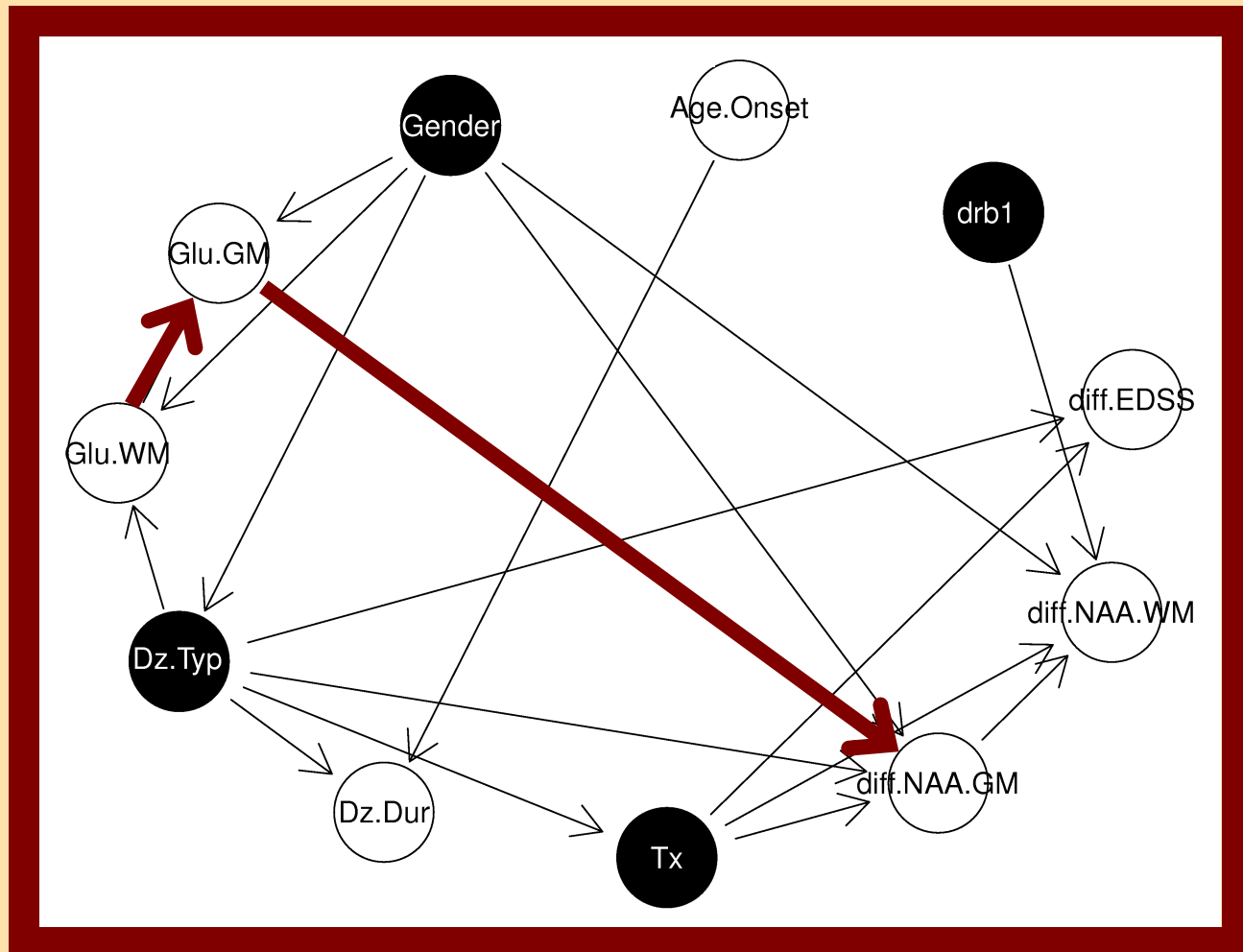


Arrows can be interpreted as defining causal effects – see all the caveats in Dr. McCulloch's lecture on Thursday when it comes to causal modeling

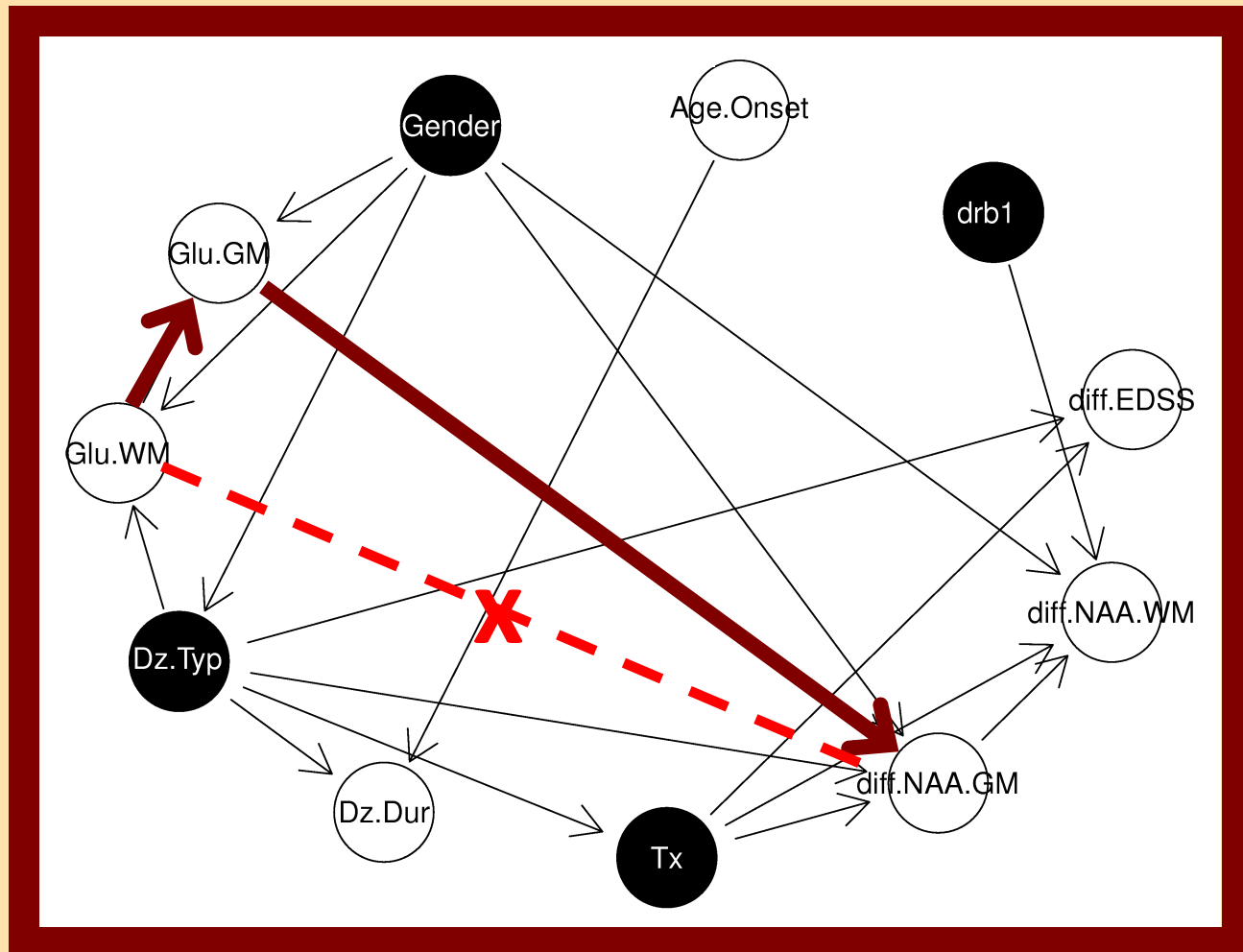
# Glu Effect on NAA GM



# Glu Effect on NAA GM

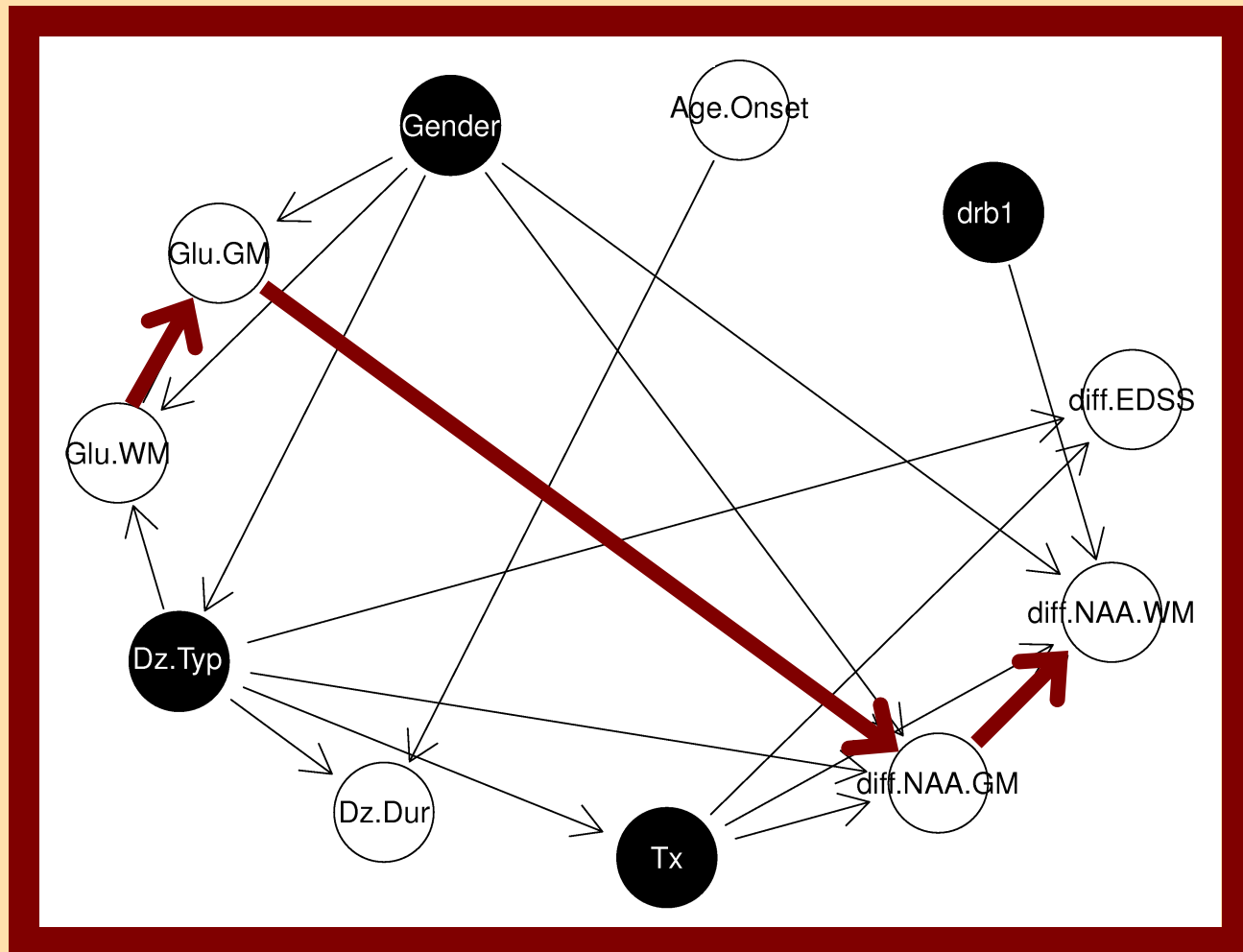


# Glu Effect on NAA GM





# Glu Effect on NAA GM



# Bayesian networks

- Can lead to complex insights into the relationships between variables
- Have to be careful to not over-interpret estimated “causal” relationships – see Dr. McCulloch’s lecture Monday with regard to general interpretation of causality
- Computation can be expensive
- We will explore Bayes Nets in Modeler during lab:



# Machine Learning Computational Tools

- We have focused on IBM Modeler but there are other tools/software/programming languages if you want to dig further. Here are some free ones
- Weka -- <http://www.cs.waikato.ac.nz/ml/weka/> - there is an associated book - Data Mining: Practical Machine Learning Tools and Techniques 3<sup>rd</sup> edition, Witten, Frank and Hall, 2011
- R -- <https://www.r-project.org/> -- R has many machine learning libraries and tools. They have a machine learning and statistical learning “Task View” page which much of what is available in R (including an R interface to the Weka package) – packages mlr and caret are wrappers for calling multiple machine learning algorithms
- Python -- <https://www.python.org/> -- if you want to go even further down the programming line – many numerical and scientific libraries – see e.g. <http://www.southampton.ac.uk/~fangohr/training/python/pdfs/Python-for-Computational-Science-and-Engineering.pdf> to get started

**Next Lecture - Monday**  
**Dr. Pletcher**  
**“Digital Health”**