**2.3**

**Visualization**

The conversion of data into visual or tabular format, so that the characteristics and relationships of the data can be analyzed by humans

Humans: well-developed ability to analyze and detect patterns/trends in data

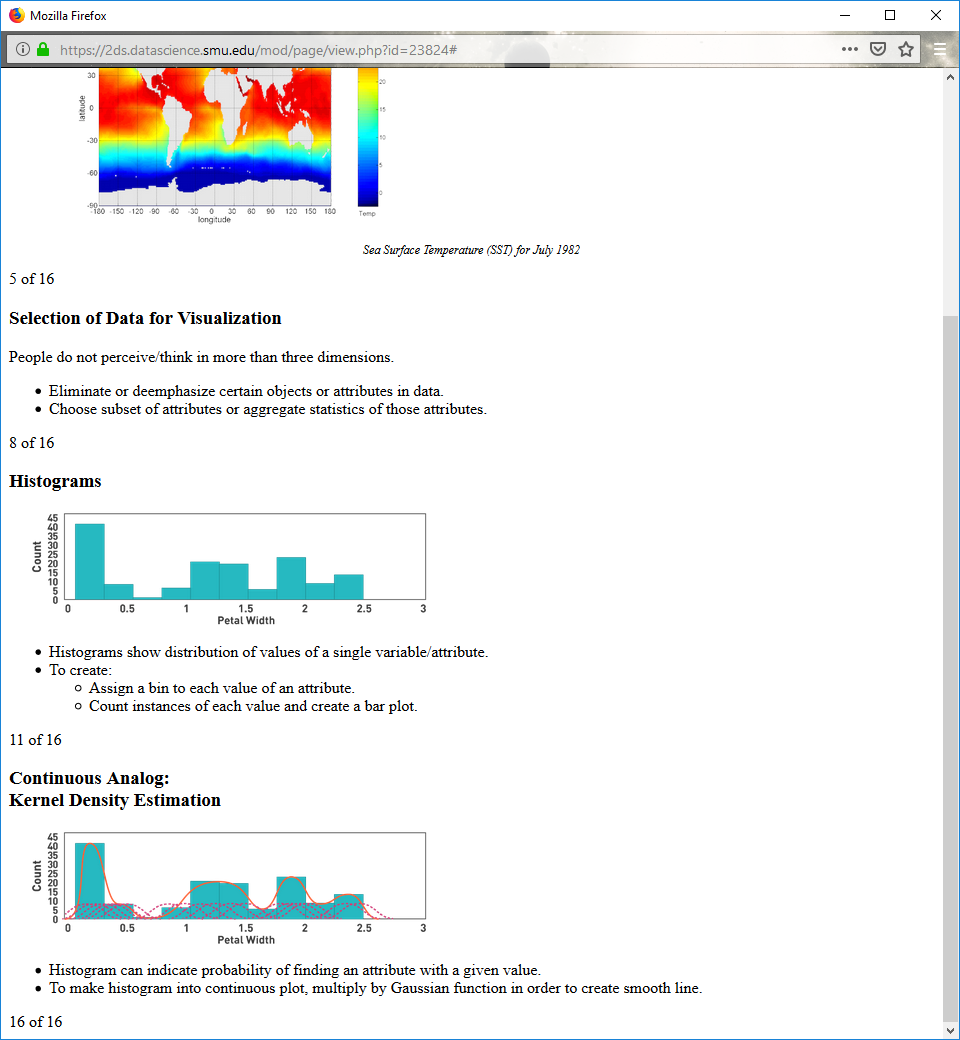
**Selection of Data for Visualization**

People do not perceive/think in more than three dimensions.

Eliminate or deemphasize certain objects or attributes in data.

Choose subset of attributes or aggregate statistics of those attributes.

**Histograms**



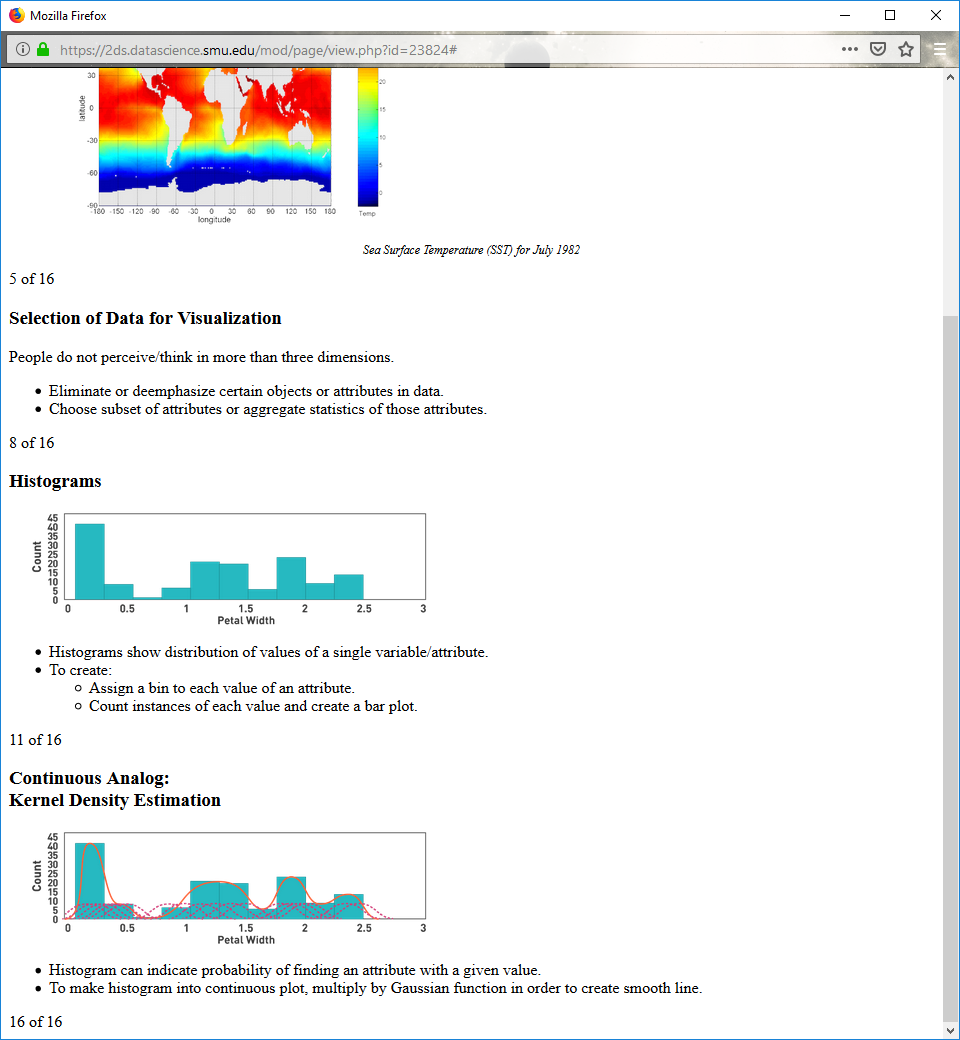
Histograms show distribution of values of a single variable/attribute.

To create:

Assign a bin to each value of an attribute.

Count instances of each value and create a bar plot.

**Continuous Analog: Kernel Density Estimation**



Histogram can indicate probability of finding an attribute with a given value.

To make histogram into continuous plot, multiply by Gaussian function in order to create smooth line.

**Interpret Data with 2-D Plots**

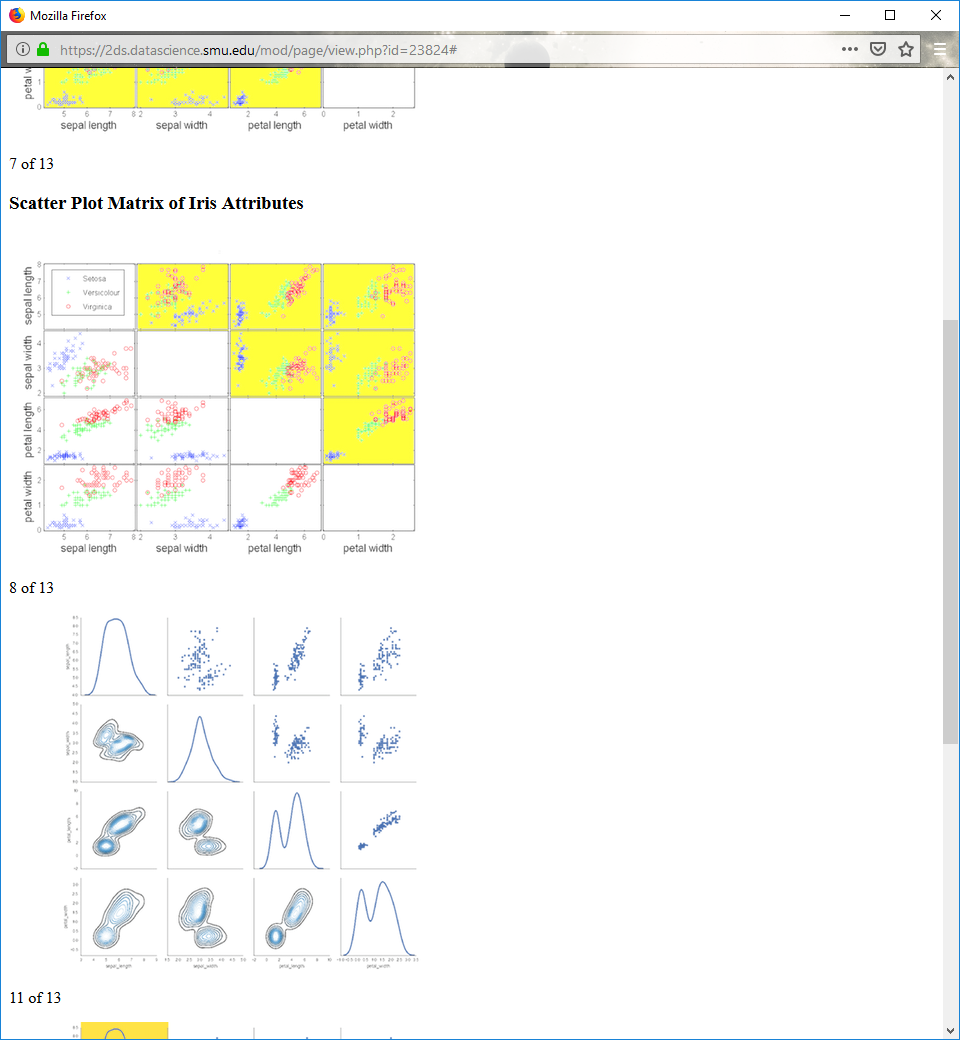
Break attribute down by dimension, then graph each dimension against itself and each of the others.

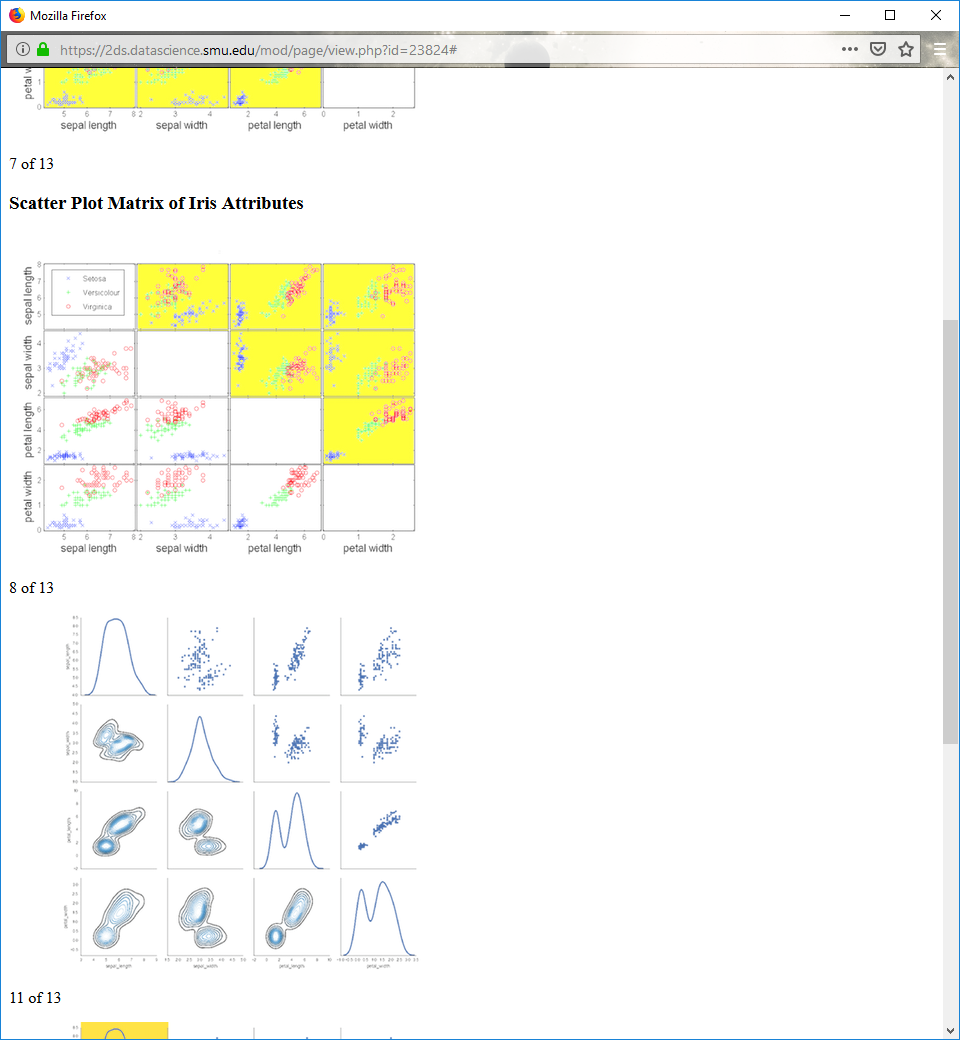
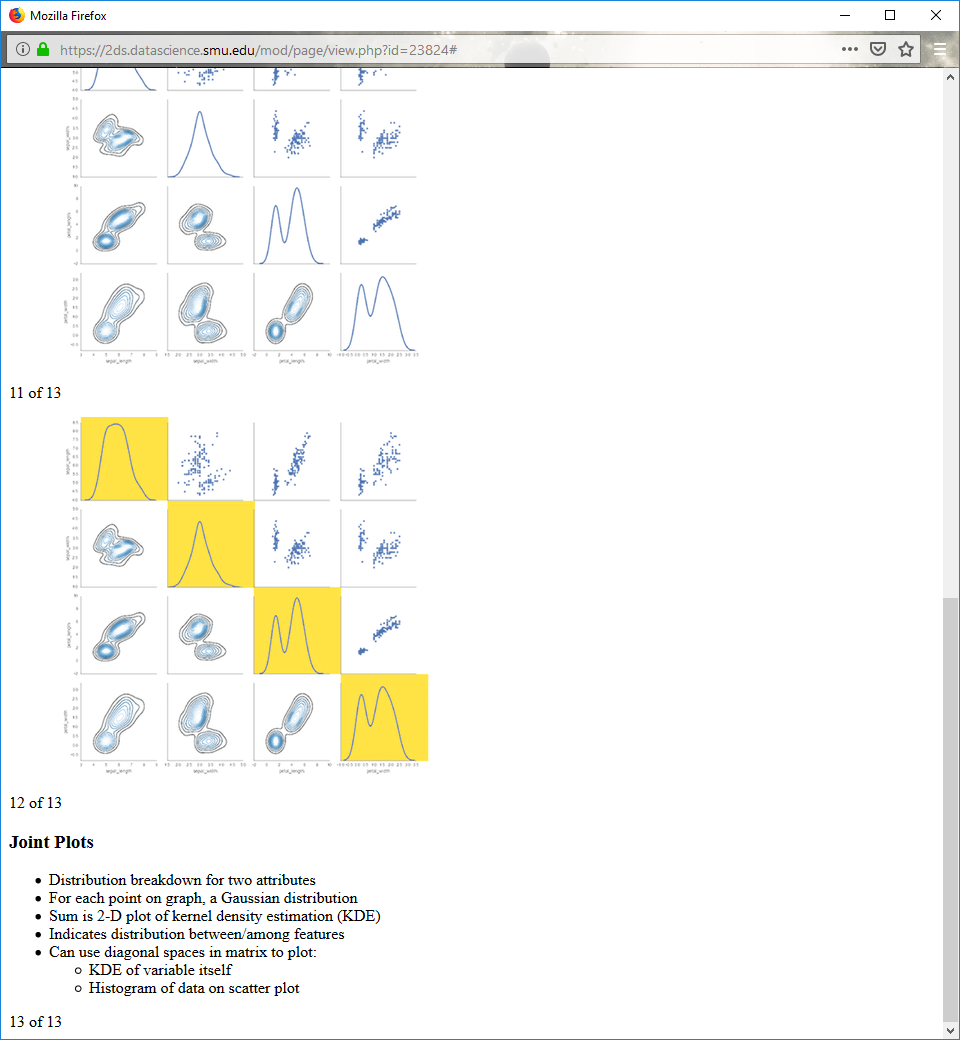
Scatter plots are an easy way to look at relationships between attributes.

For more than two variables, use scatter plot matrix.

Include all attributes in data set.

**Scatter Plot Matrix of Iris Attributes**



**Joint Plots**

Distribution breakdown for two attributes

For each point on graph, a Gaussian distribution

Sum is 2-D plot of kernel density estimation (KDE)

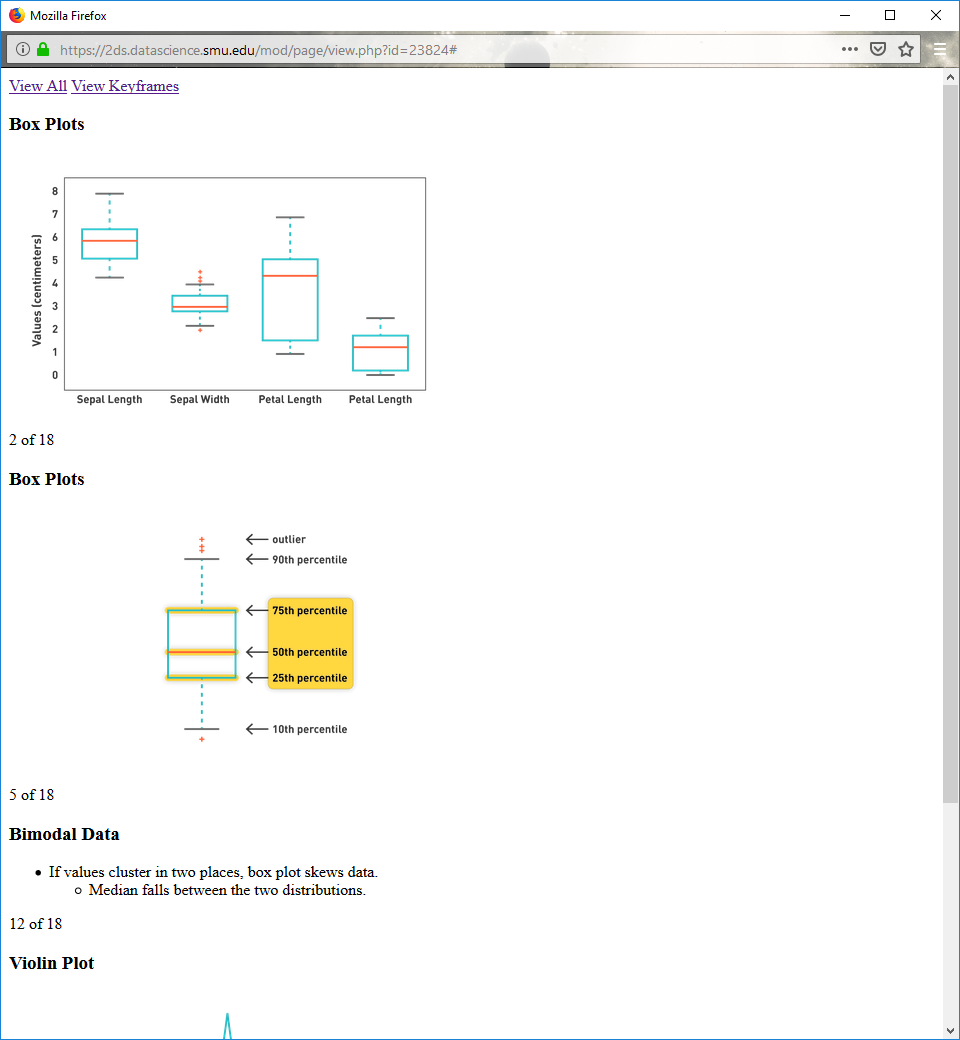
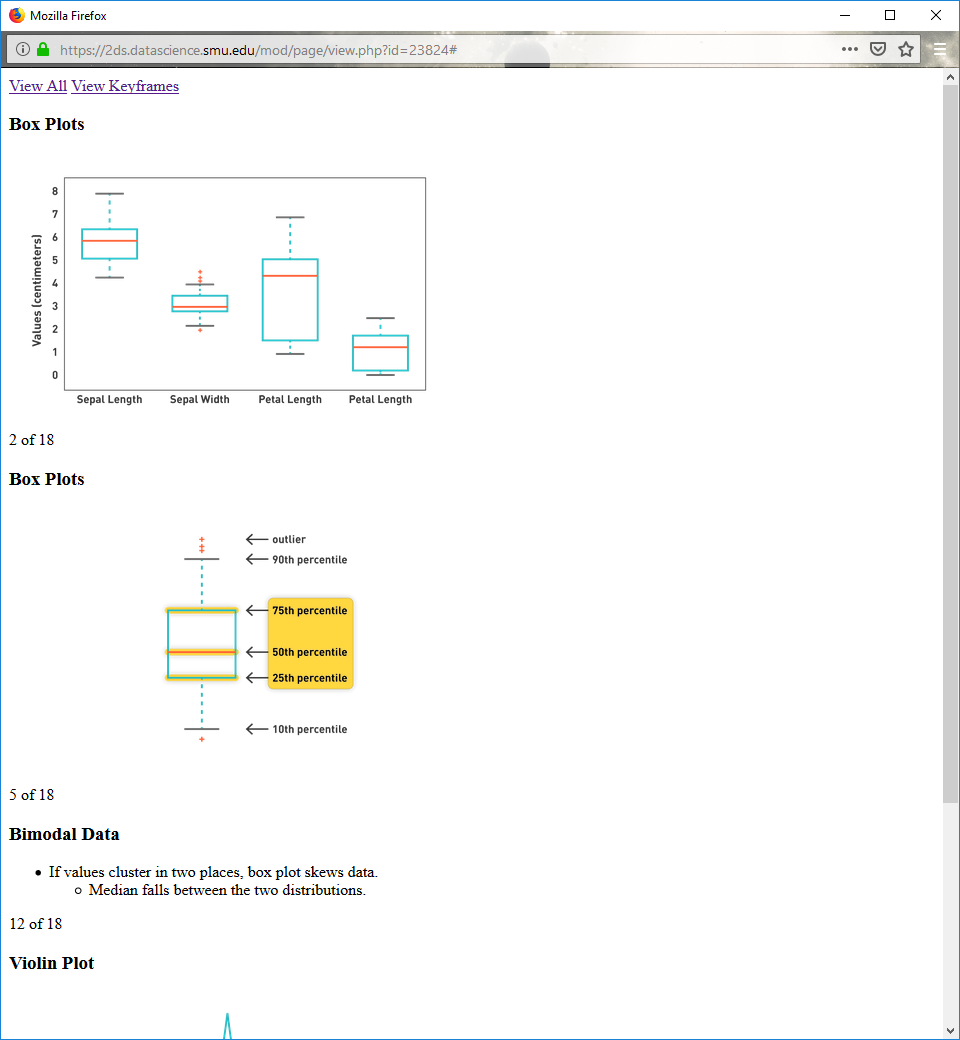
Indicates distribution between/among features

Can use diagonal spaces in matrix to plot:

KDE of variable itself

Histogram of data on scatter plot

**Box Plots**

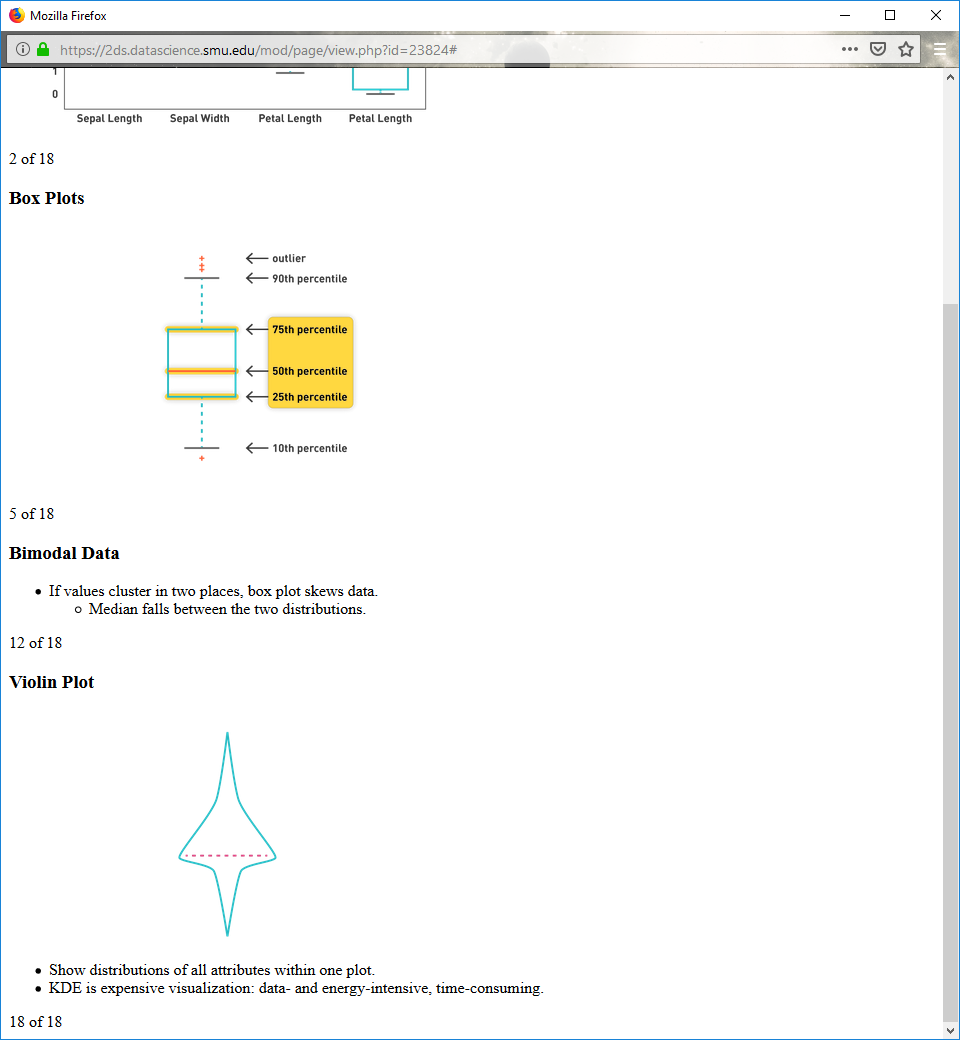
 

**Bimodal Data**

If values cluster in two places, box plot skews data.

Median falls between the two distributions.

**Violin Plot**



Show distributions of all attributes within one plot.

KDE is expensive visualization: data- and energy-intensive, time-consuming.

**Matrix Plots**

Similarity of each instance to every other instance

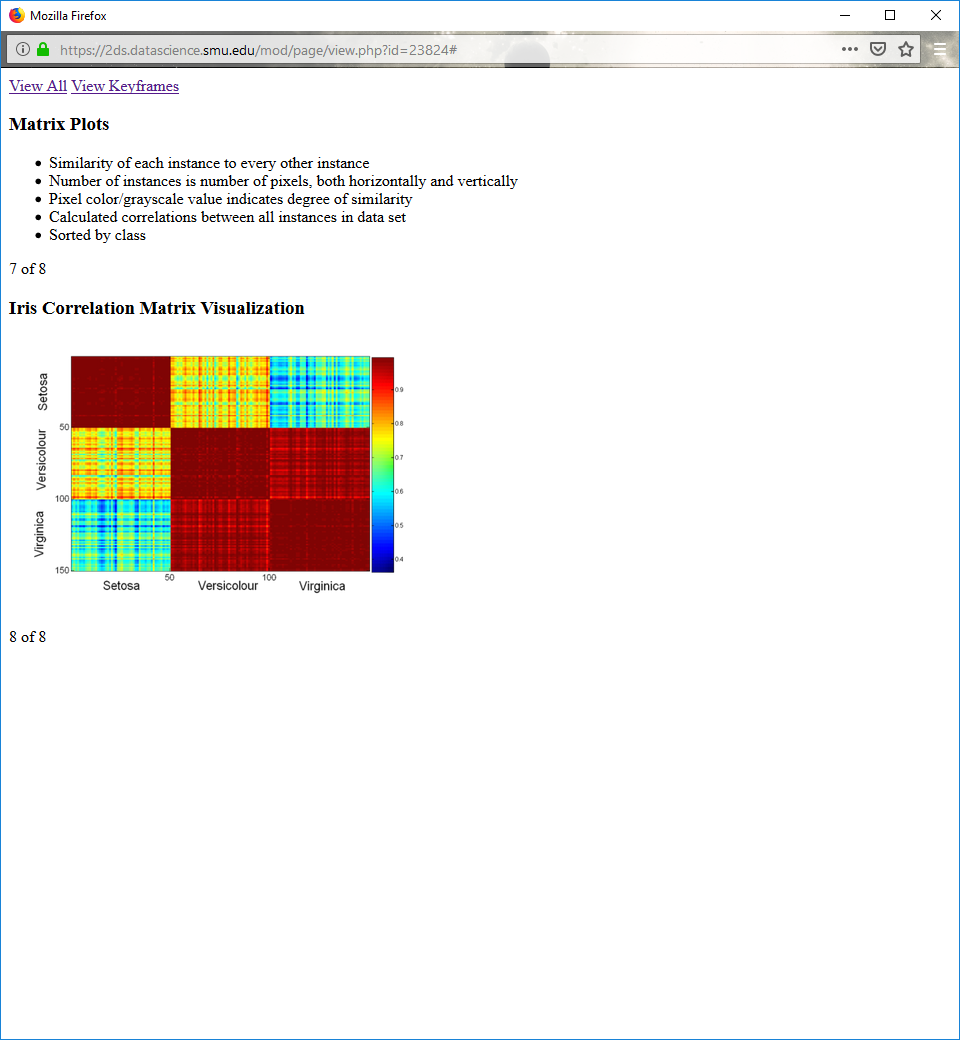
Number of instances is number of pixels, both horizontally and vertically

Pixel color/grayscale value indicates degree of similarity

Calculated correlations between all instances in data set

Sorted by class

**Iris Correlation Matrix Visualization**



**Parallel Coordinate Plots**

Each instance treated separately, represented as a line

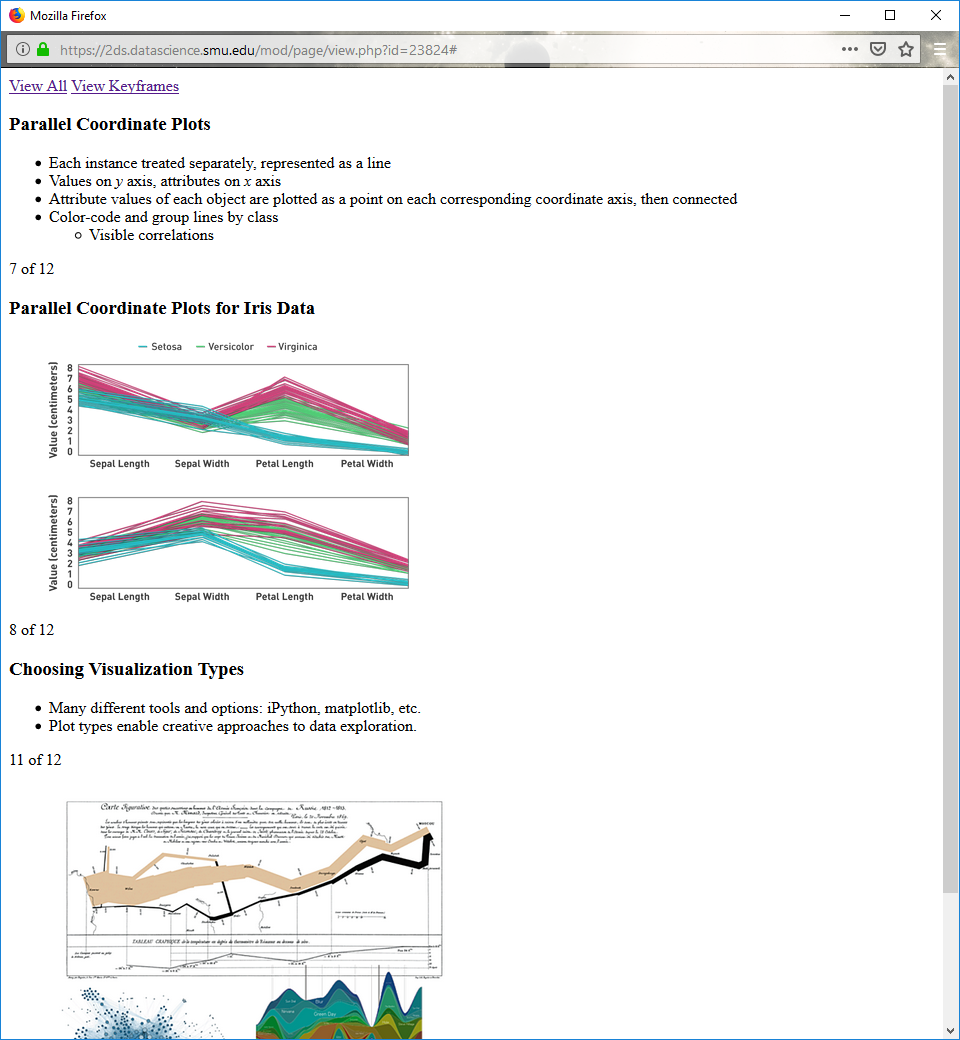
Values on y axis, attributes on x axis

Attribute values of each object are plotted as a point on each corresponding coordinate axis, then connected

Color-code and group lines by class

Visible correlations

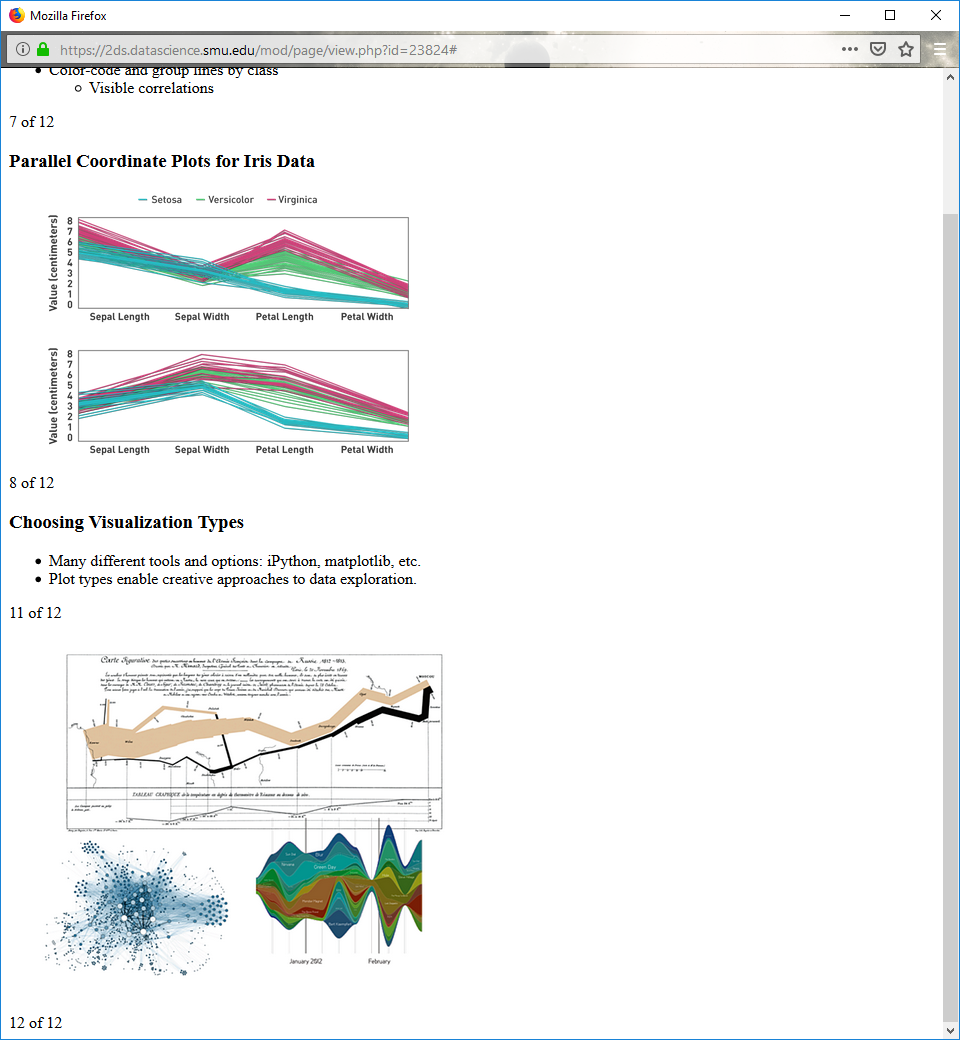
**Parallel Coordinate Plots for Iris Data**



**Choosing Visualization Types**

Many different tools and options: iPython, matplotlib, etc.

Plot types enable creative approaches to data exploration.

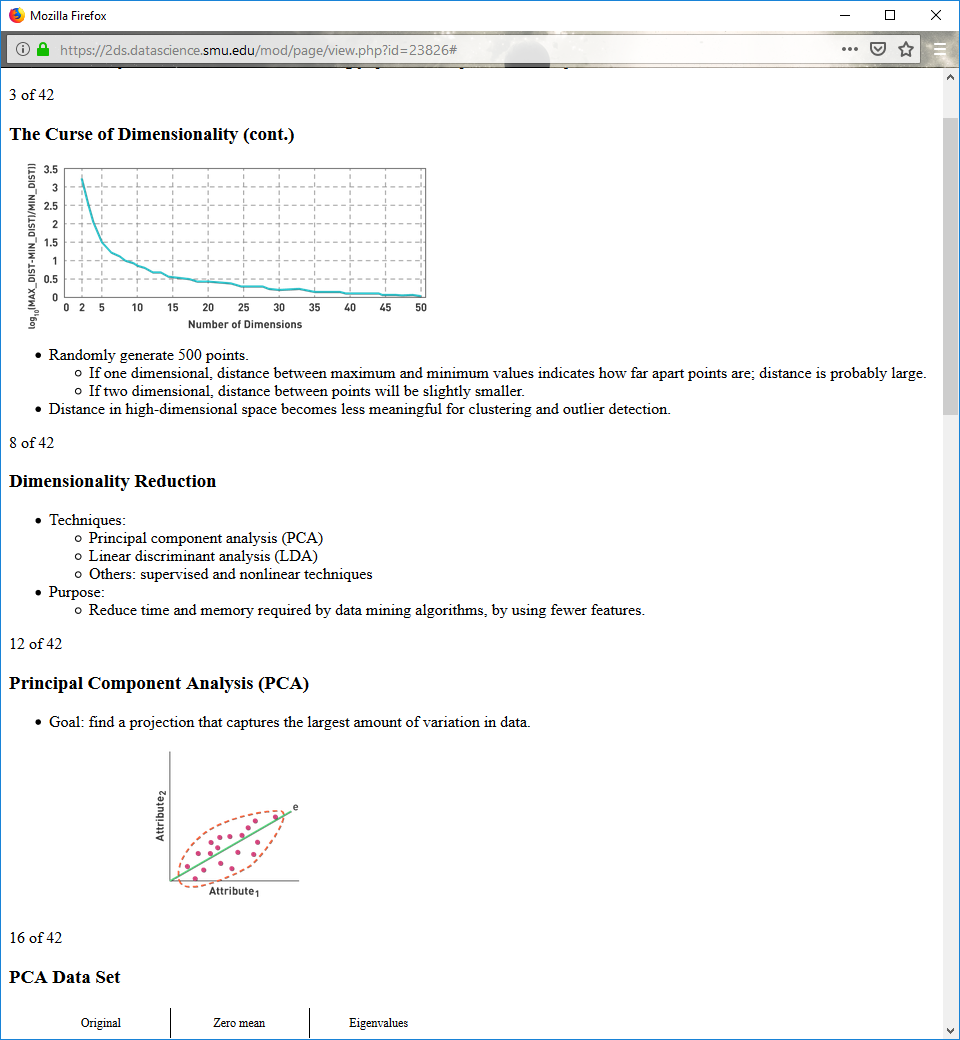


**2.5**

**The Curse of Dimensionality**

As dimensionality increases, data becomes increasingly sparse in the space that it occupies.

**The Curse of Dimensionality (cont.)**



Randomly generate 500 points.

If one dimensional, distance between maximum and minimum values indicates how far apart points are; distance is probably large.

If two dimensional, distance between points will be slightly smaller.

Distance in high-dimensional space becomes less meaningful for clustering and outlier detection.

**Dimensionality Reduction**

Techniques:

Principal component analysis (PCA)

Linear discriminant analysis (LDA)

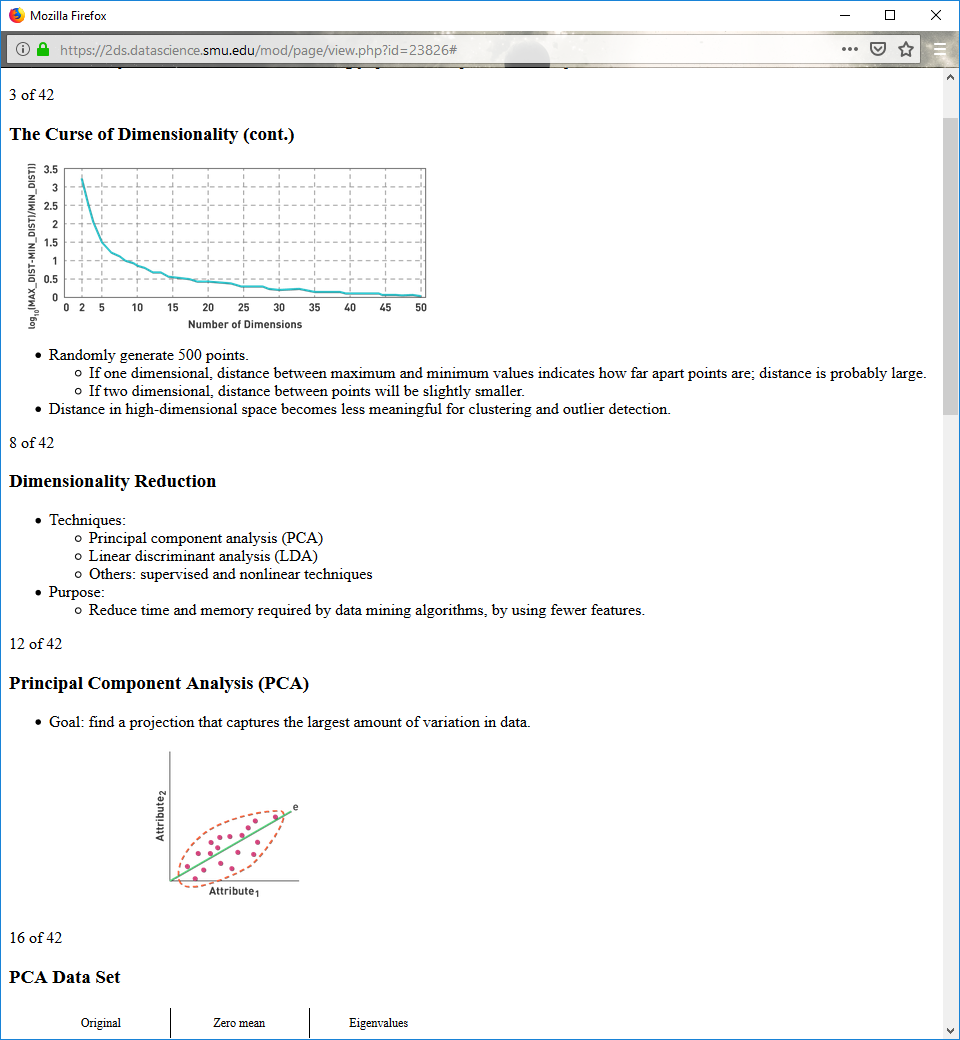
Others: supervised and nonlinear techniques

Purpose:

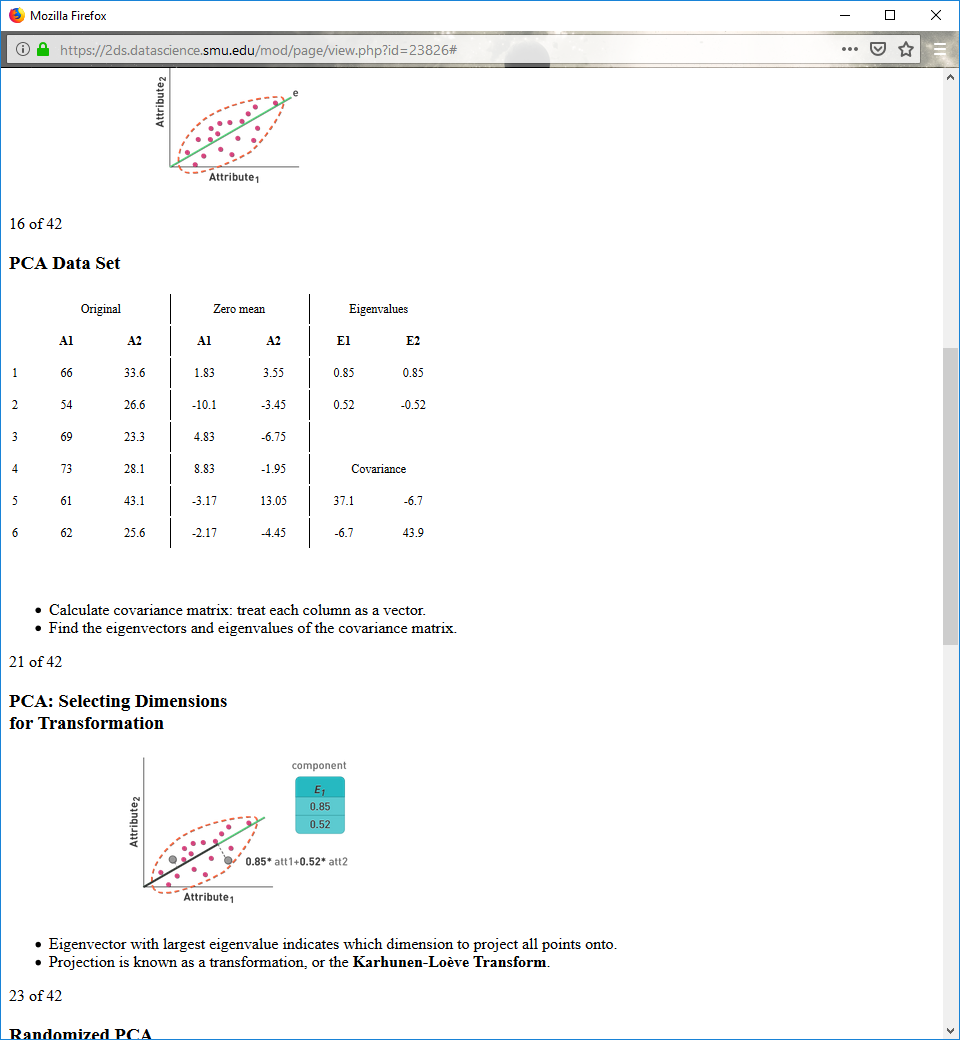
Reduce time and memory required by data mining algorithms, by using fewer features.

**Principal Component Analysis (PCA)**

Goal: find a projection that captures the largest amount of variation in data.



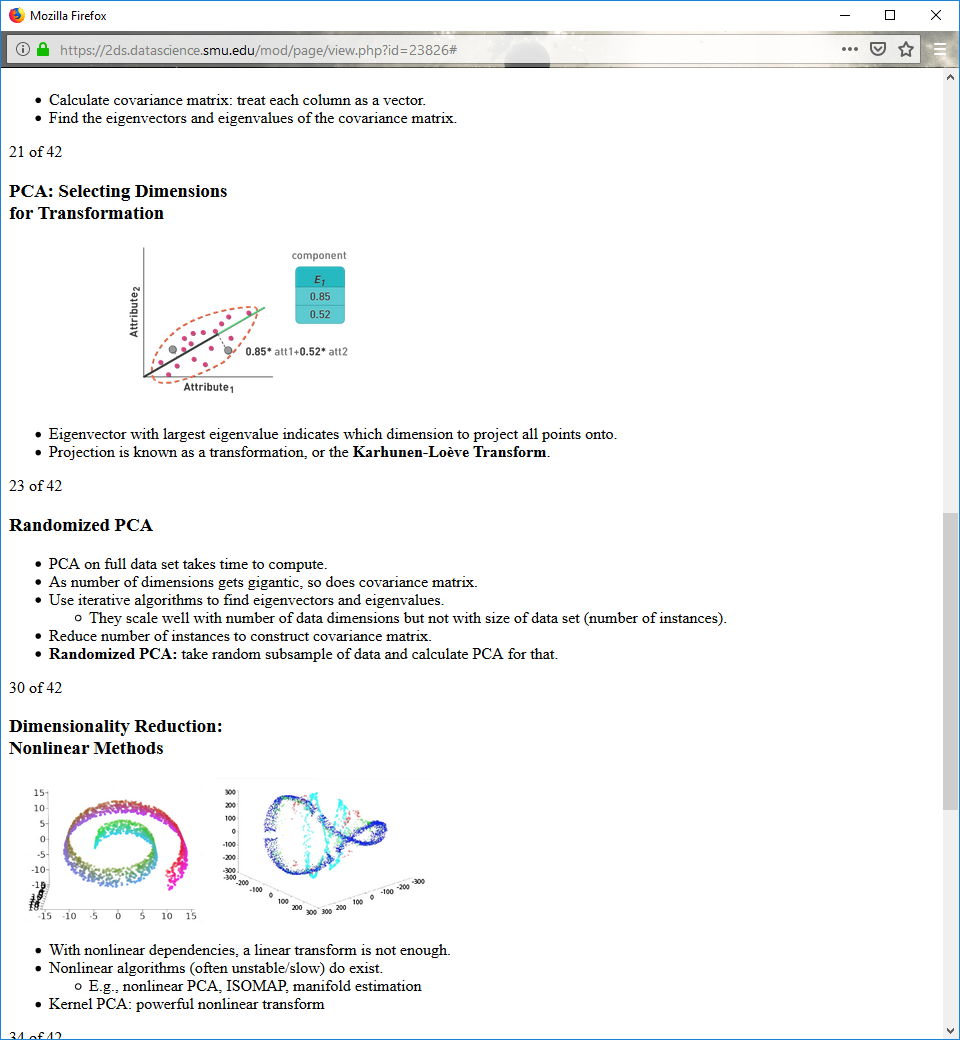
**PCA Data Set**



Calculate covariance matrix: treat each column as a vector.

Find the eigenvectors and eigenvalues of the covariance matrix.

**PCA: Selecting Dimensions** **for Transformation**



Eigenvector with largest eigenvalue indicates which dimension to project all points onto.

Projection is known as a transformation, or the **Karhunen-Loève Transform**.

**Randomized PCA**

PCA on full data set takes time to compute.

As number of dimensions gets gigantic, so does covariance matrix.

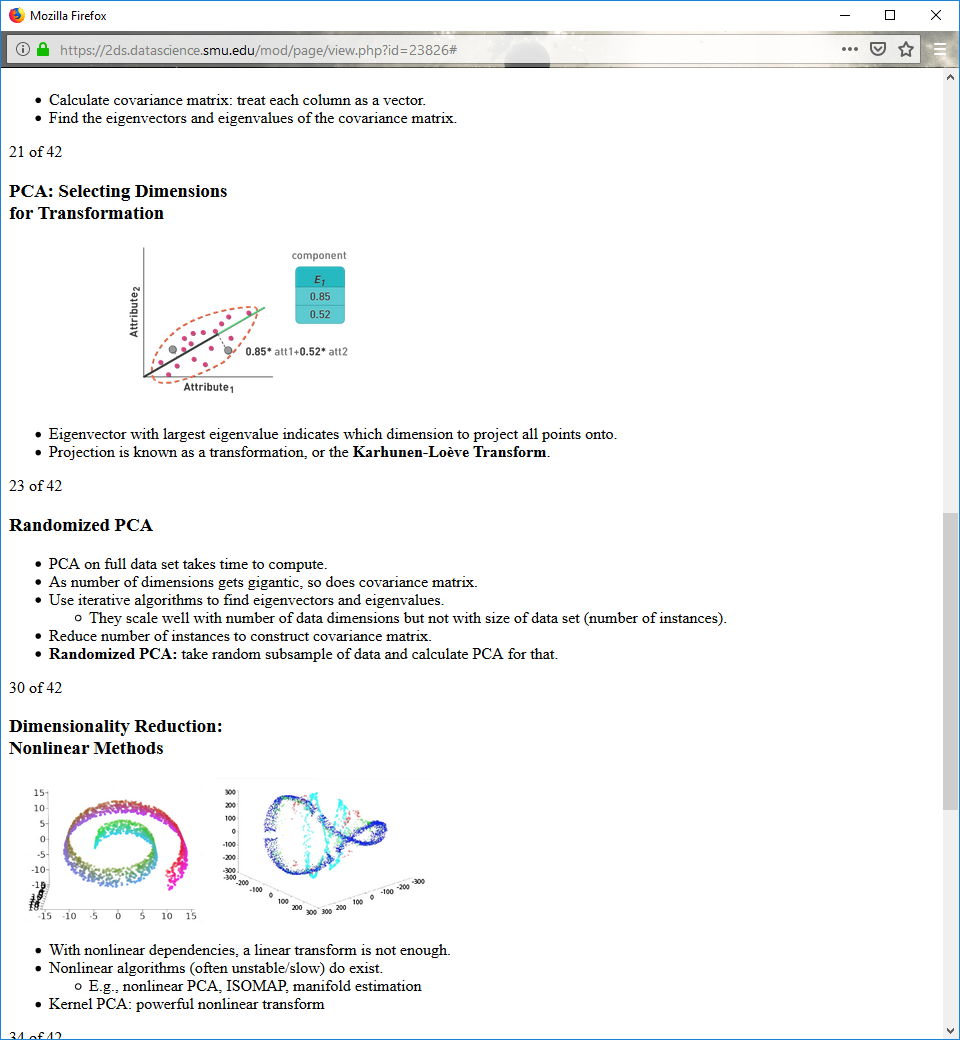
Use iterative algorithms to find eigenvectors and eigenvalues.

They scale well with number of data dimensions but not with size of data set (number of instances).

Reduce number of instances to construct covariance matrix.

**Randomized PCA:** take random subsample of data and calculate PCA for that.

**Dimensionality Reduction: Nonlinear Methods**

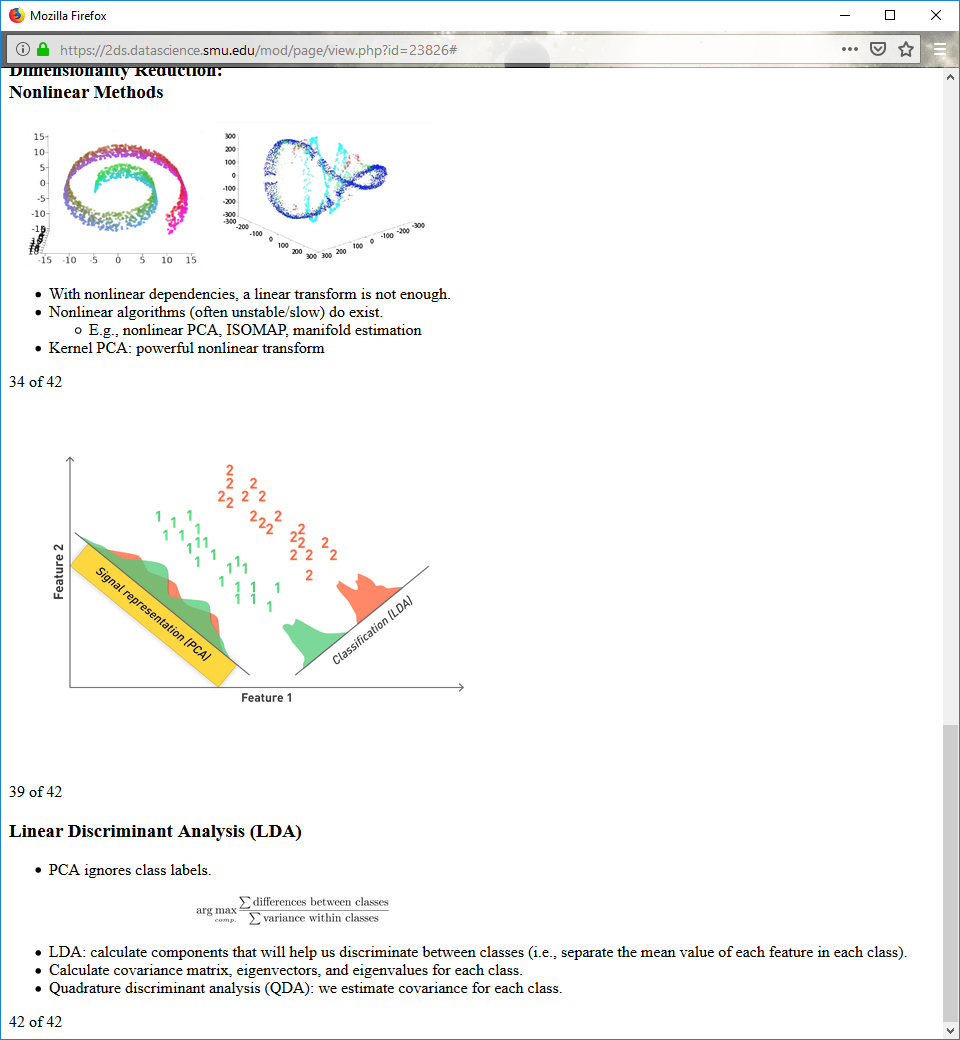


With nonlinear dependencies, a linear transform is not enough.

Nonlinear algorithms (often unstable/slow) do exist.

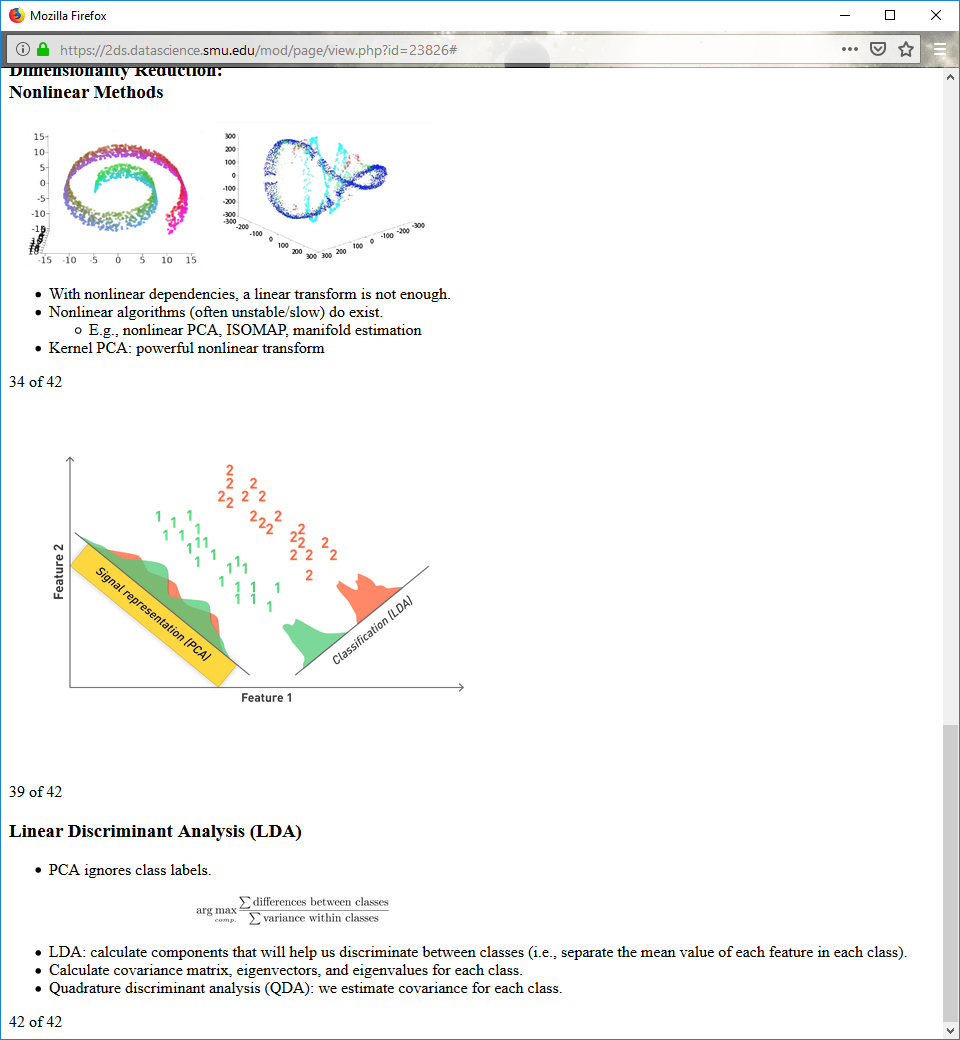
E.g., nonlinear PCA, ISOMAP, manifold estimation

Kernel PCA: powerful nonlinear transform



**Linear Discriminant Analysis (LDA)**

PCA ignores class labels.



LDA: calculate components that will help us discriminate between classes (i.e., separate the mean value of each feature in each class).

Calculate covariance matrix, eigenvectors, and eigenvalues for each class.

Quadrature discriminant analysis (QDA): we estimate covariance for each class.