
# Astroinformatics school - "Rise of the machines"

Curtin University
THE CURTIN INSTITUTE FOR COMPUTATION

4 to 6 February 2019

Presented by Rebecca Lange and Dan Marrable

# K-means

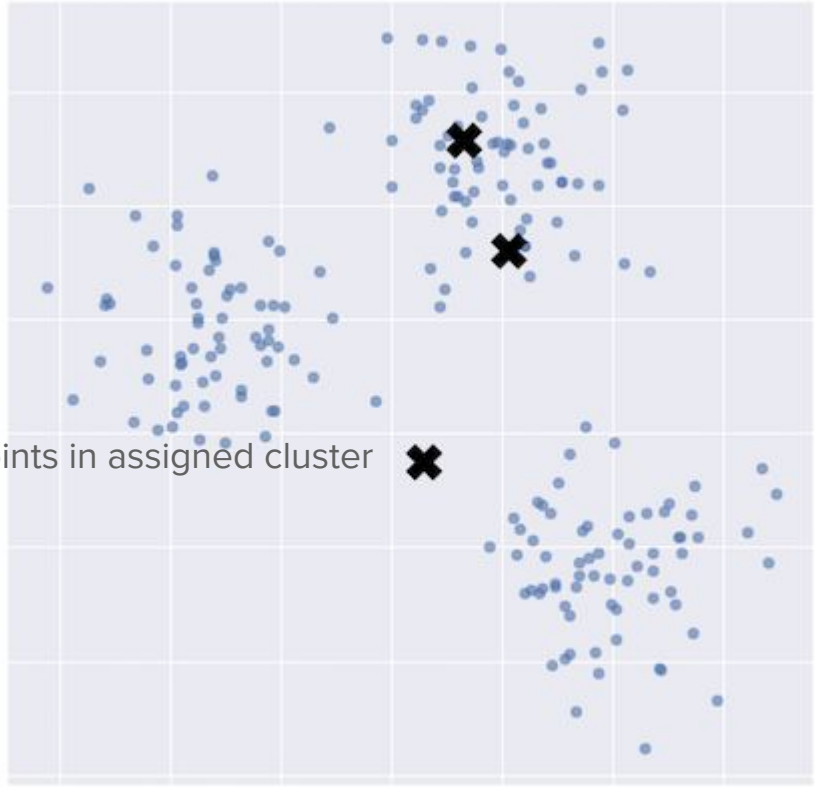
ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of pc
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid

# K-means

ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid



# K-means

ADACS
ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

ADACS
ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

ADACS
ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

ADACS
ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

ASTRONOMY DATA AND COMPUTING SERVICES

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# K-means

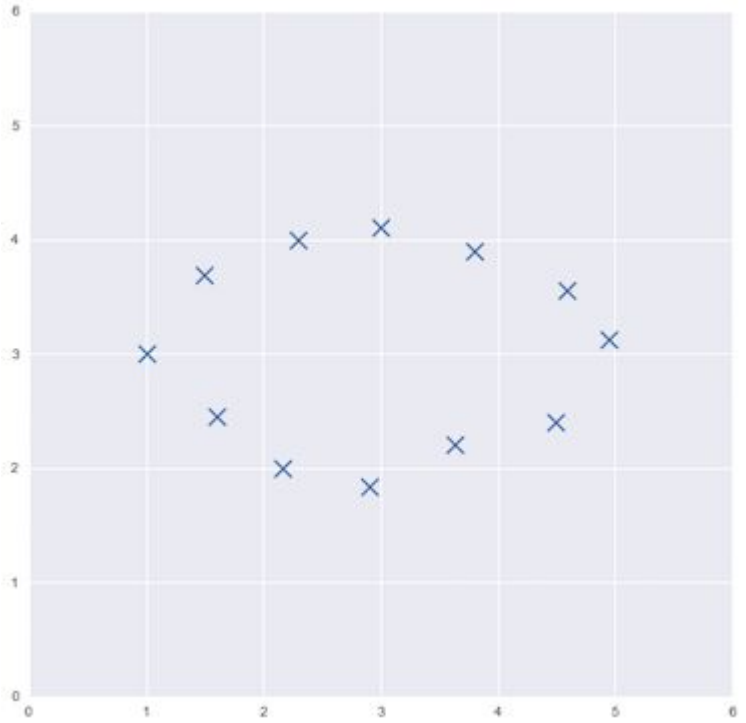
ASTRONOMY DATA AND COMPUTING SERVICES

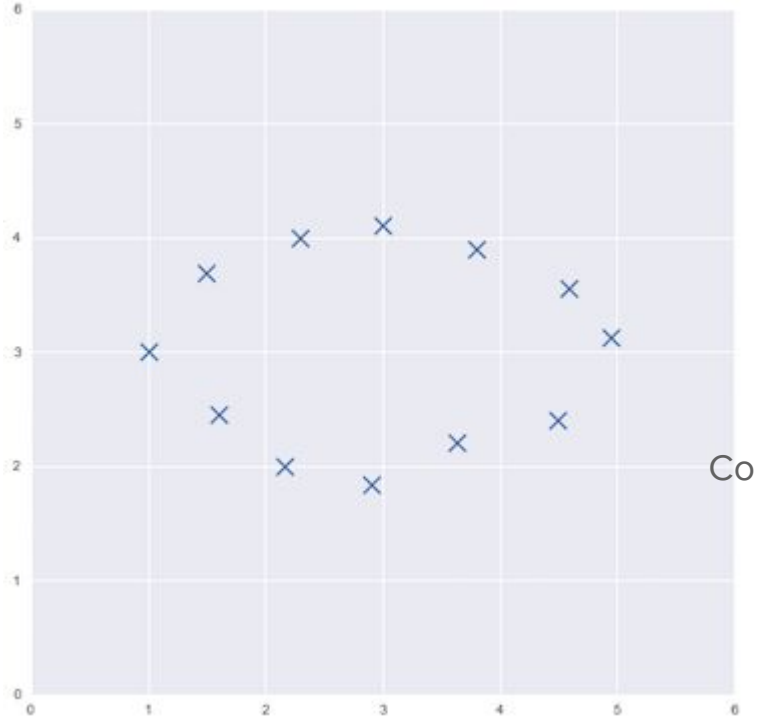
- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

# Principal Components Analysis (PCA)

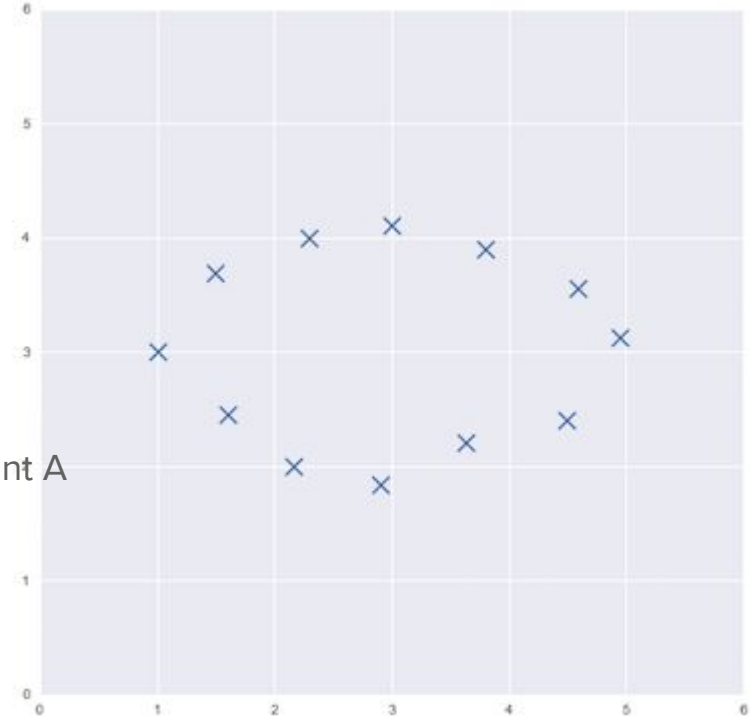
ADACS
ASTRONOMY DATA AND COMPUTING SERVICES

- Dimensionality reduction method
- Represent data in fewer dimensions
  - Preserve as much structure as possible
- New dimensions = principal components
  - Directions where data is most spread (variance)
  - Combination of original features
- Example
  - Reduce a dataset from 2D to 1D
  - Manually try two components

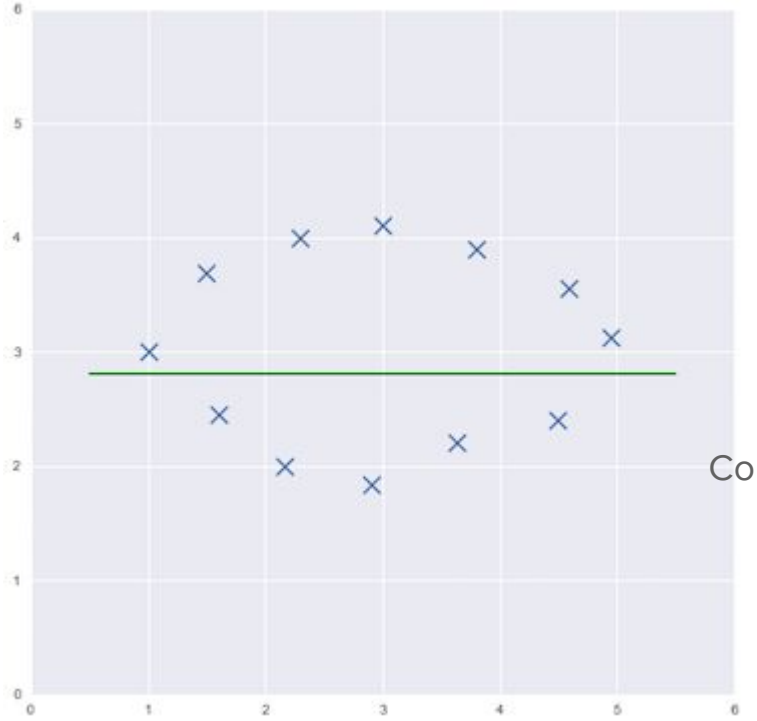




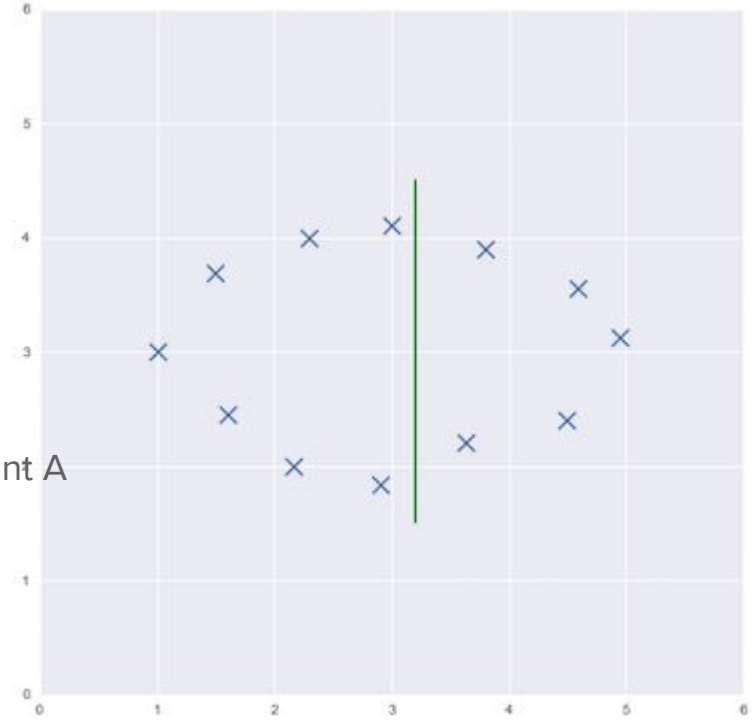

Component A



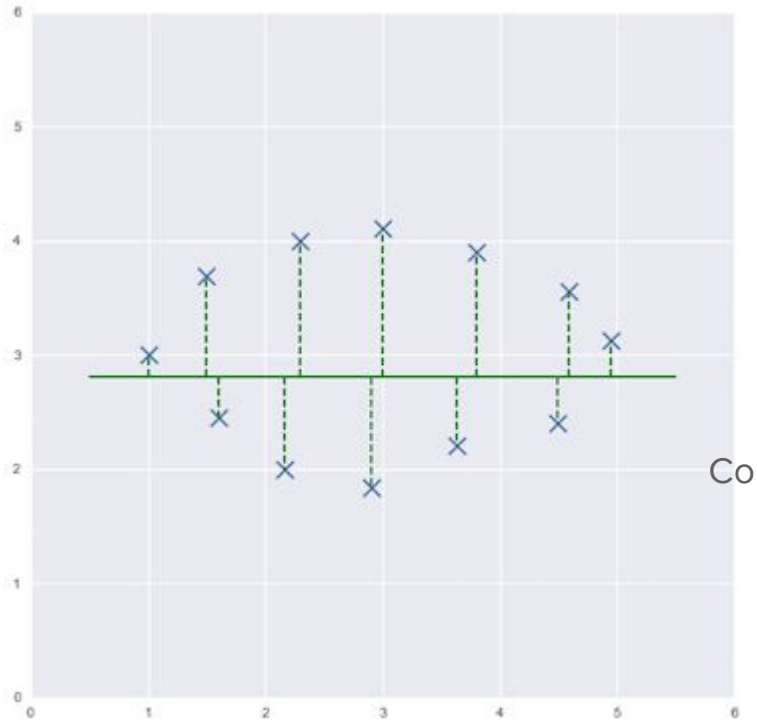

Component B



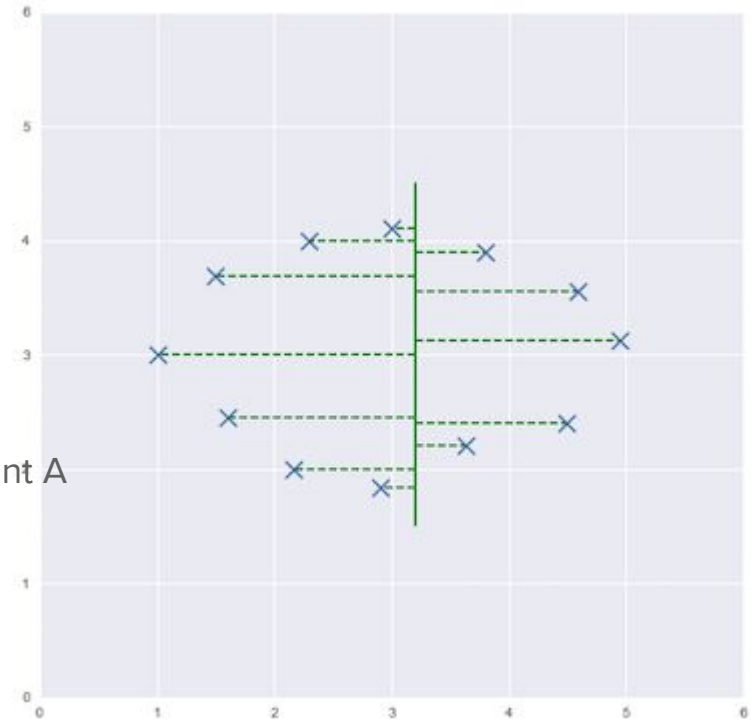

Component A




Component B




Component A



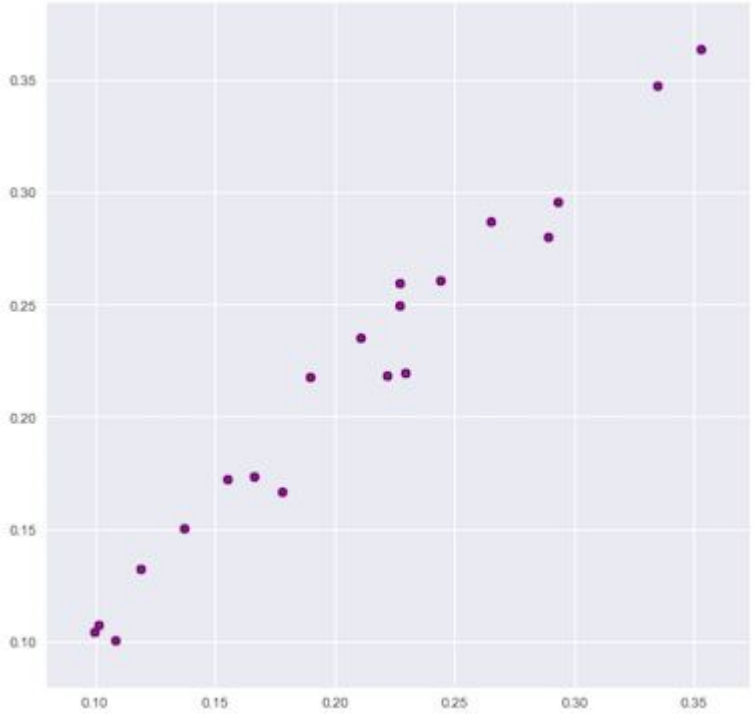

Component B

Which component represents a larger spread of the data (variance)? Component A

# PCA

ADACS
ASTRONOMY DATA AND COMPUTING SERVICES

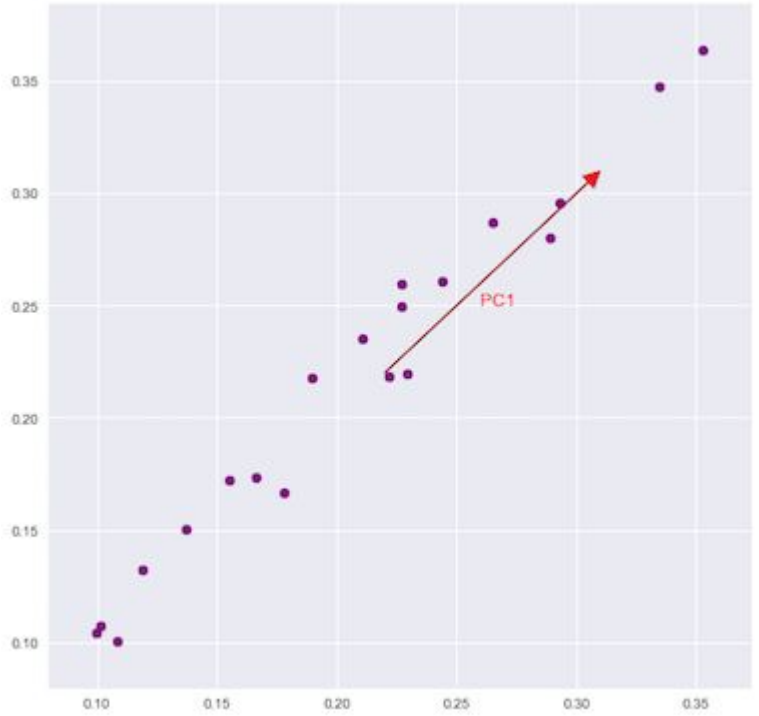
- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - ... until last dimension of dataset



# PCA

ASTRONOMY DATA AND COMPUTING SERVICES

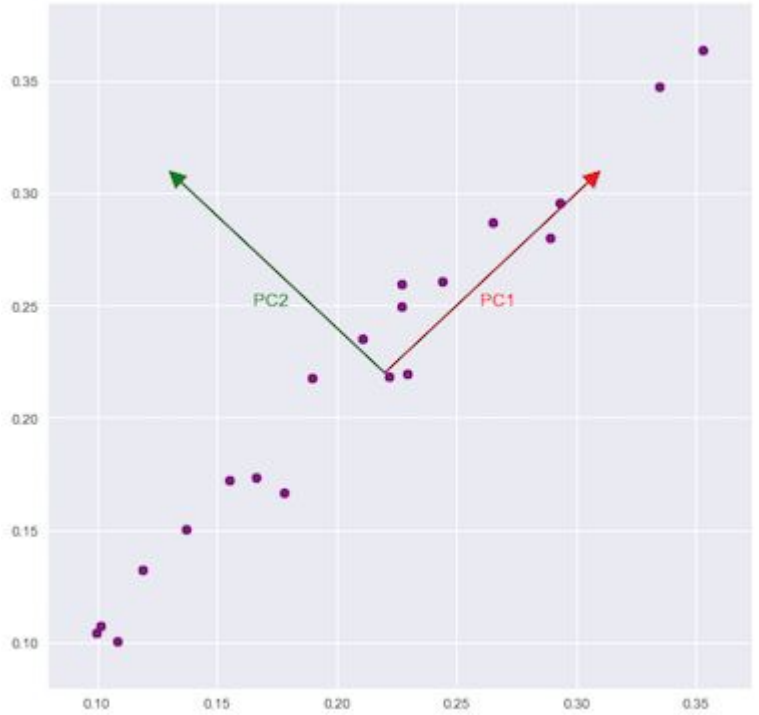
- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - ... until last dimension of dataset



# PCA

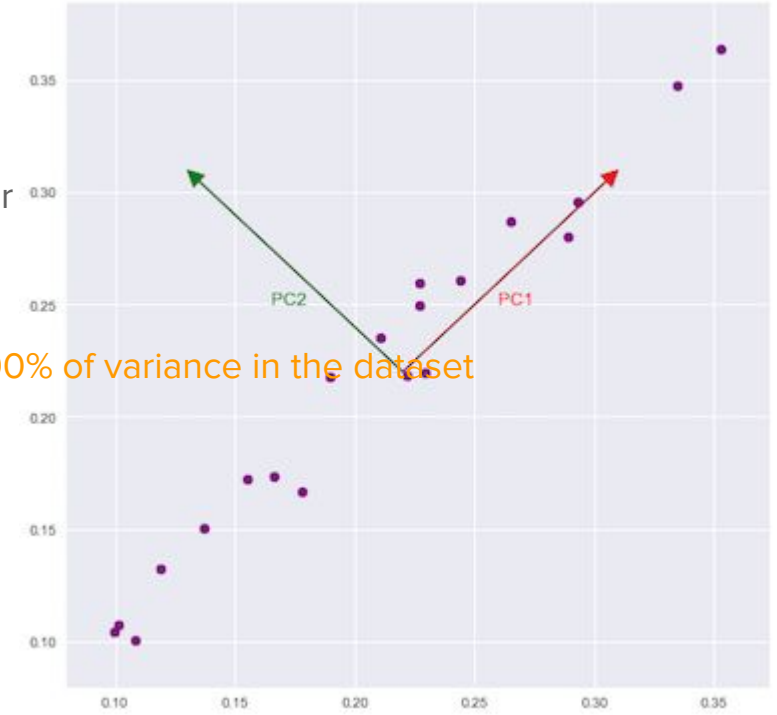
ASTRONOMY DATA AND COMPUTING SERVICES

- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - ... until last dimension of dataset



# PCA

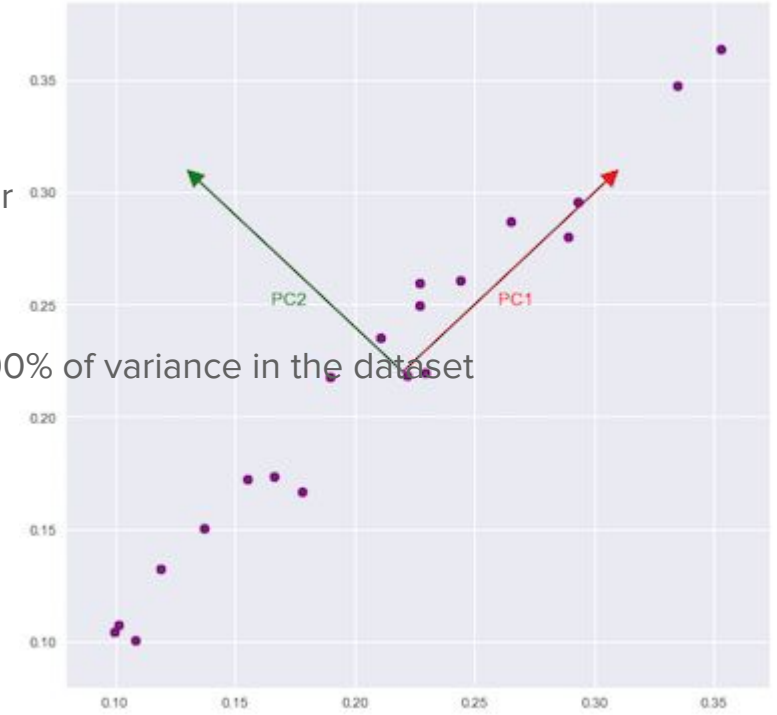
- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - … until last dimension of dataset
- Reduce dataset to *k* components
  - E.g. keep top *k* components that represent at least 90% of variance in the dataset
  - Transform original data to the new dimensions




**PC1** represents 99.6% of variance

# PCA

- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - … until last dimension of dataset
- Reduce dataset to *k* components
  - E.g. keep top *k* components that represent at least 90% of variance in the dataset
  - Transform original data to the new dimensions
- Algorithm
  - Normalise data with zero mean
  - Calculate covariance matrix
  - Calculate:
    - Eigenvectors = principal component
    - Eigenvalues = amount of variance represented




**PC1** represents 99.6% of variance