

Introduction to Unsupervised Learning

Ludwig Krippahl

Unsupervised Learning

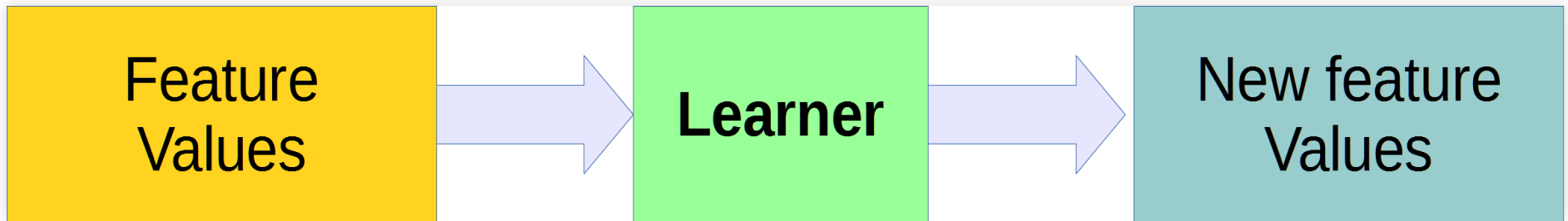
Summary

- Introduction to Unsupervised Learning
- Data visualization
- Feature Selection

Unsupervised Learning

Unsupervised Learning

- Data labels are not used in training
- No error measure for predictions
- The goal is to find structure in data



Unsupervised Learning

- Learning without labels does not mean data cannot be labeled
- Training does not adjust prediction to known values
- It's possible to use unsupervised learning in classification (transforming the feature space)
- Some tasks that can benefit from unsupervised learning:
 - Data visualization and understanding
 - Estimation of distribution parameters (e.g. KDE)
 - Feature selection and extraction
 - Clustering
- This lecture
 - Visualizing data
 - Selecting features

Visualizing Data

Visualizing Data

■ The Iris dataset



CC BY-SA Setosa: Szczecinkowaty; Versicolor: Gordon, Robertson; Virginica: Mayfield

Visualizing Data

- The Iris dataset: <https://archive.ics.uci.edu/ml/datasets/Iris>



- 3 classes
 - Setosa
 - Versicolor
 - Virginica
- 4 attributes:
 - Sepal length
 - Sepal width
 - Petal length
 - Petal width

Visualizing Data

- The Problem: features in 4 dimensions
- (We'll use the Pandas library for most examples)

Python Data Analysis Library

- pandas.pydata.org
- Utilities for:
 - Loading and handling data tables
 - Basic statistics and plotting
- Not required for this course but useful
- pandas.pydata.org/pandas-docs/stable/visualization.html

Visualizing Data

Examining individual features: histograms

- These examples can be done with Pyplot with some extra coding
- But Pandas makes it much more convenient
- Loads file into DataFrame, keeps column headers, automates plotting, ...

Iris csv data file

```
SepalLength,SepalWidth,PetalLength,PetalWidth,Name
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
...
5.1,2.5,3.0,1.1,Iris-versicolor
5.7,2.8,4.1,1.3,Iris-versicolor
...
6.2,3.4,5.4,2.3,Iris-virginica
5.9,3.0,5.1,1.8,Iris-virginica
```

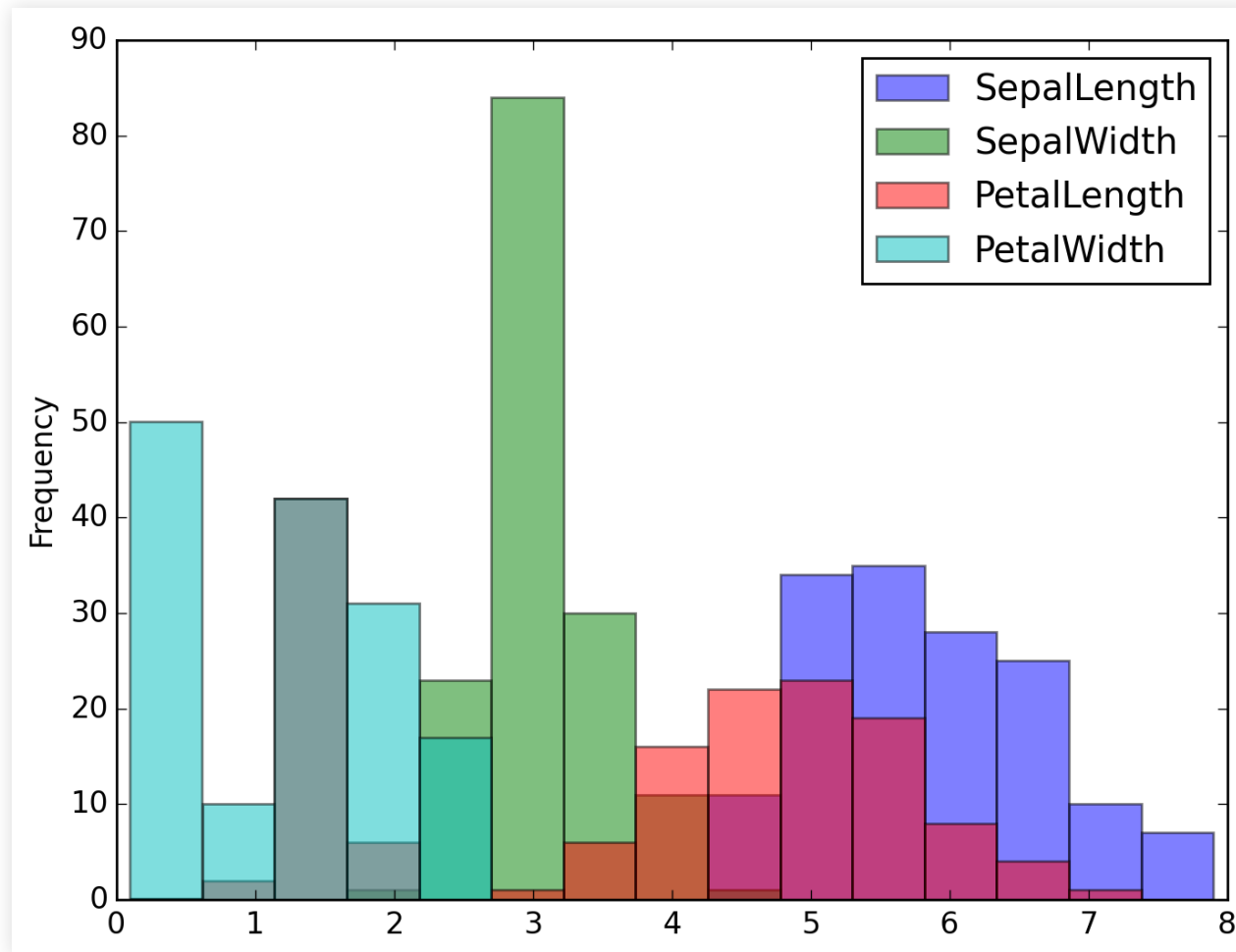
Visualizing Data

Examining individual features: histograms

```
from pandas import read_csv
import matplotlib.pyplot as plt

data = read_csv('iris.data')
data.plot(kind='hist', bins=15, alpha=0.5)
plt.savefig('L15-stackedhist.png', dpi=200, bbox_inches='tight')
plt.close()
```

Examining individual features: histograms



Visualizing Data

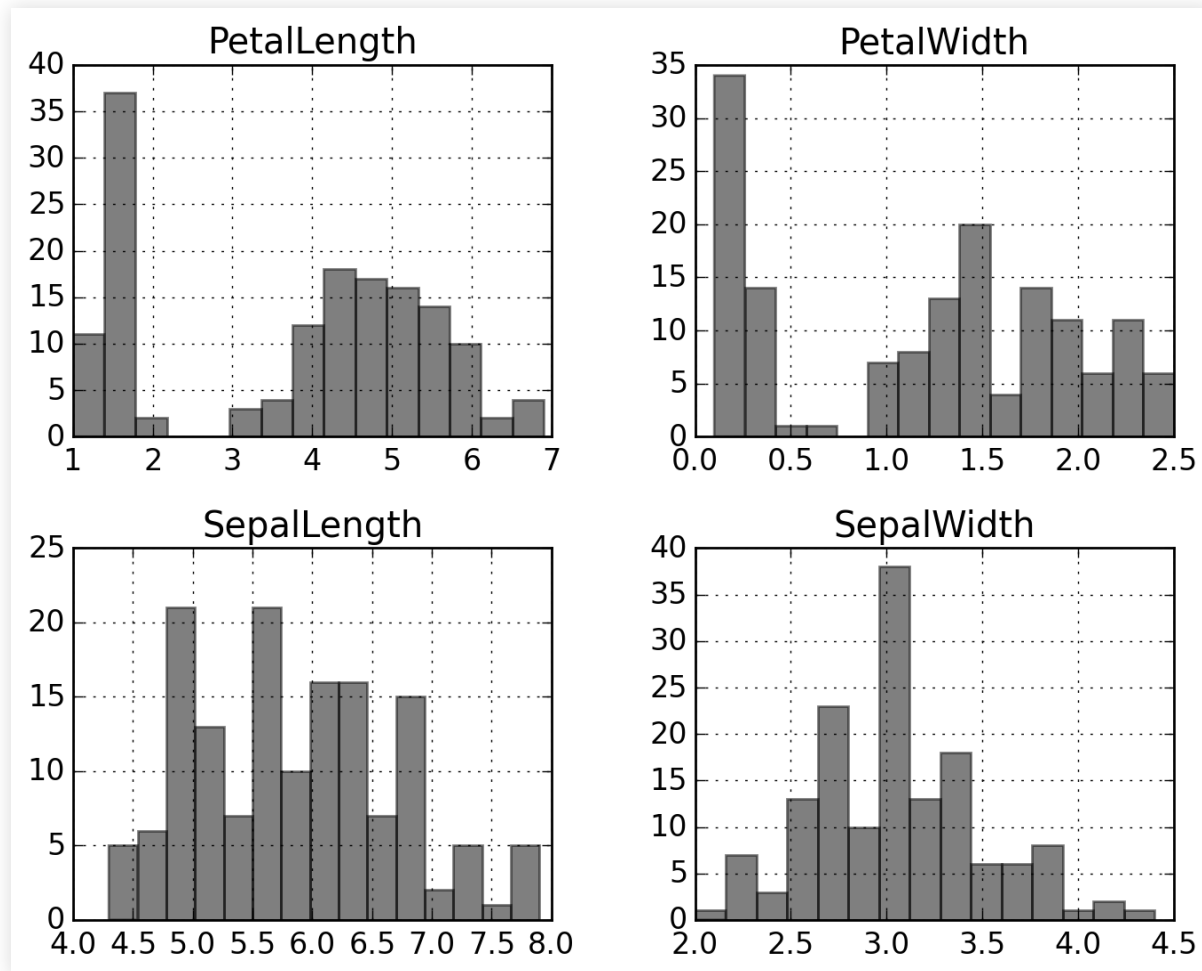
■ Separate histograms

```
from pandas import read_csv
import matplotlib.pyplot as plt

data = read_csv('iris.data')
data.hist(color='k', alpha=0.5, bins=15)
plt.savefig('L15-hists.png', dpi=200, bbox_inches='tight')
plt.close()
```

Visualizing Data

■ Separate histograms



Visualizing Data

Examining individual features: Box plot

- Boxes represent values at 25% (Q1) and 75% (Q3)
- Mid line for median
- Whiskers at:

$$\min_x \geq Q1 - w(Q3 - Q1)$$

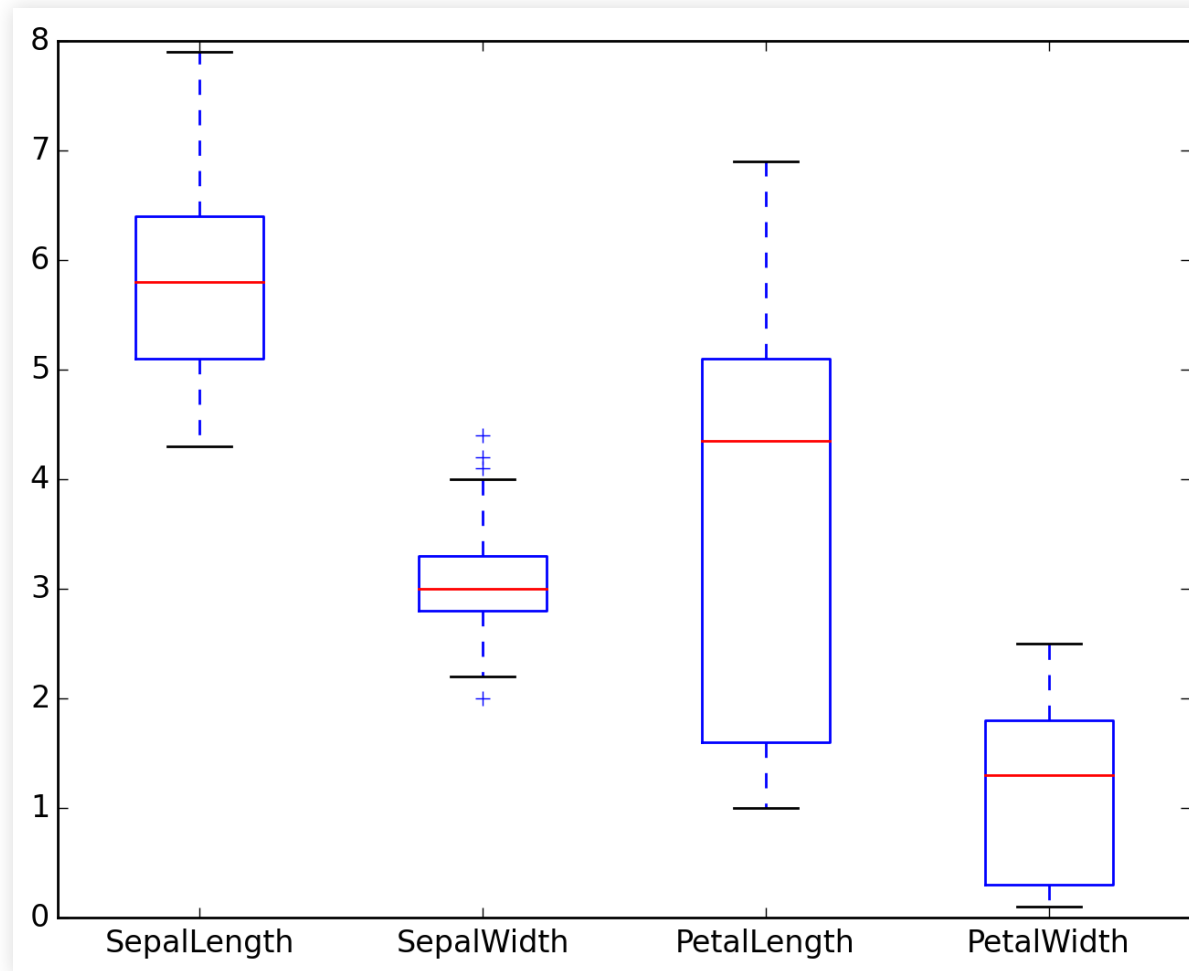
$$\max_x \leq Q3 + w(Q3 - Q1)$$

```
from pandas import read_csv
import matplotlib.pyplot as plt

data.plot(kind='box')
plt.savefig('L15-boxplot.png', dpi=200, bbox_inches='tight')
plt.close()
```

Visualizing Data

■ Examining individual features: box plot



Visualizing Data

Scatter Matrix plot

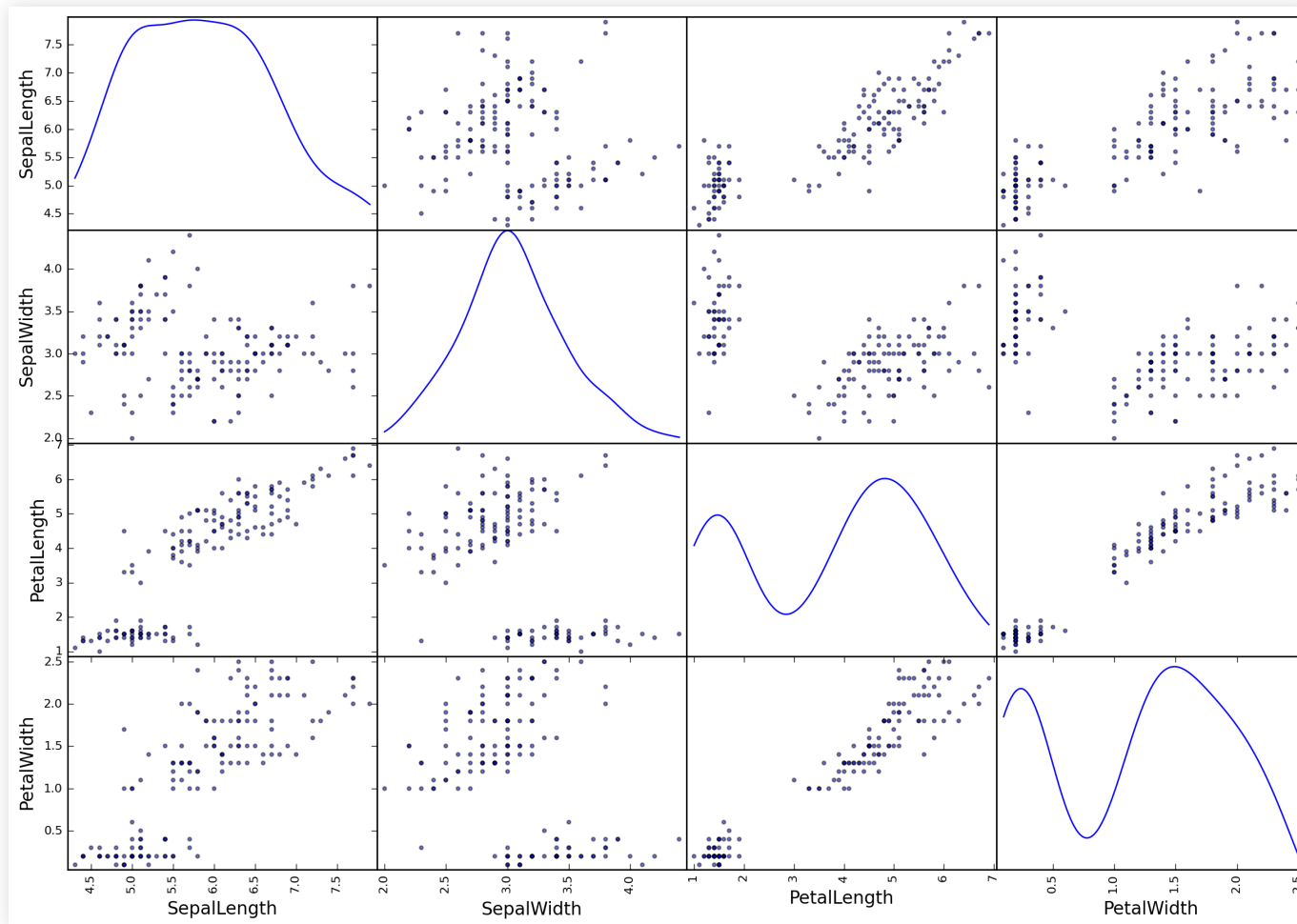
- Matrix of 2D projections into pairs of features
- Diagonal: histogram or KDE
- Works for moderate number of dimensions

```
from pandas import read_csv
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix

data = read_csv('iris.data')
scatter_matrix(data.ix[:,[0,1,2,3]], alpha=0.5, figsize=(15,10), diagonal='kde')
plt.savefig('L15-scatter.png', dpi=200, bbox_inches='tight')
plt.close()
```

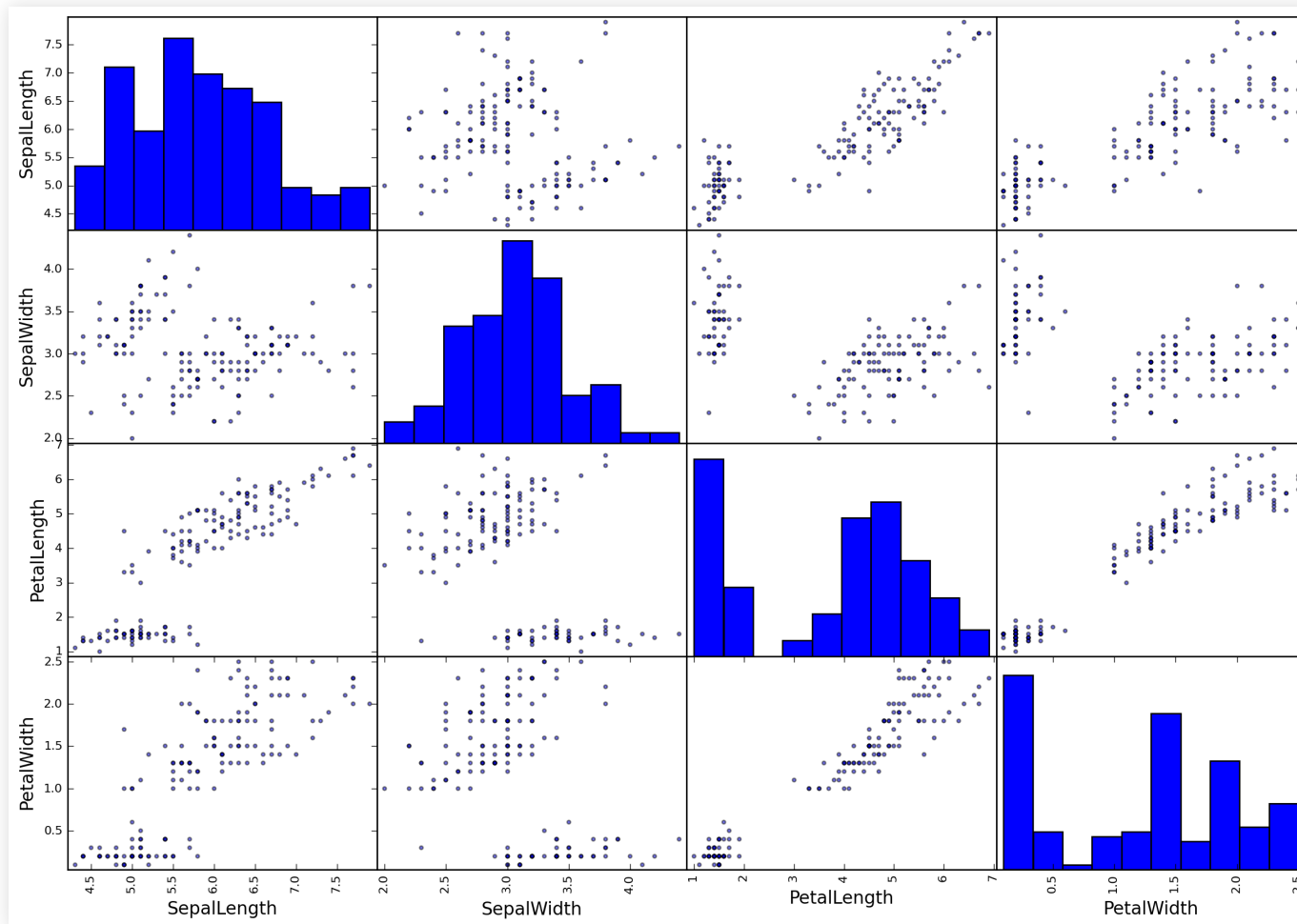
Visualizing Data

■ Scatter Matrix plot (KDE)



Visualizing Data

■ Scatter Matrix plot (histogram)



Visualizing Data

Parallel coordinates plot

- Plot each point as a set of line segments
- Y coordinates are the feature values

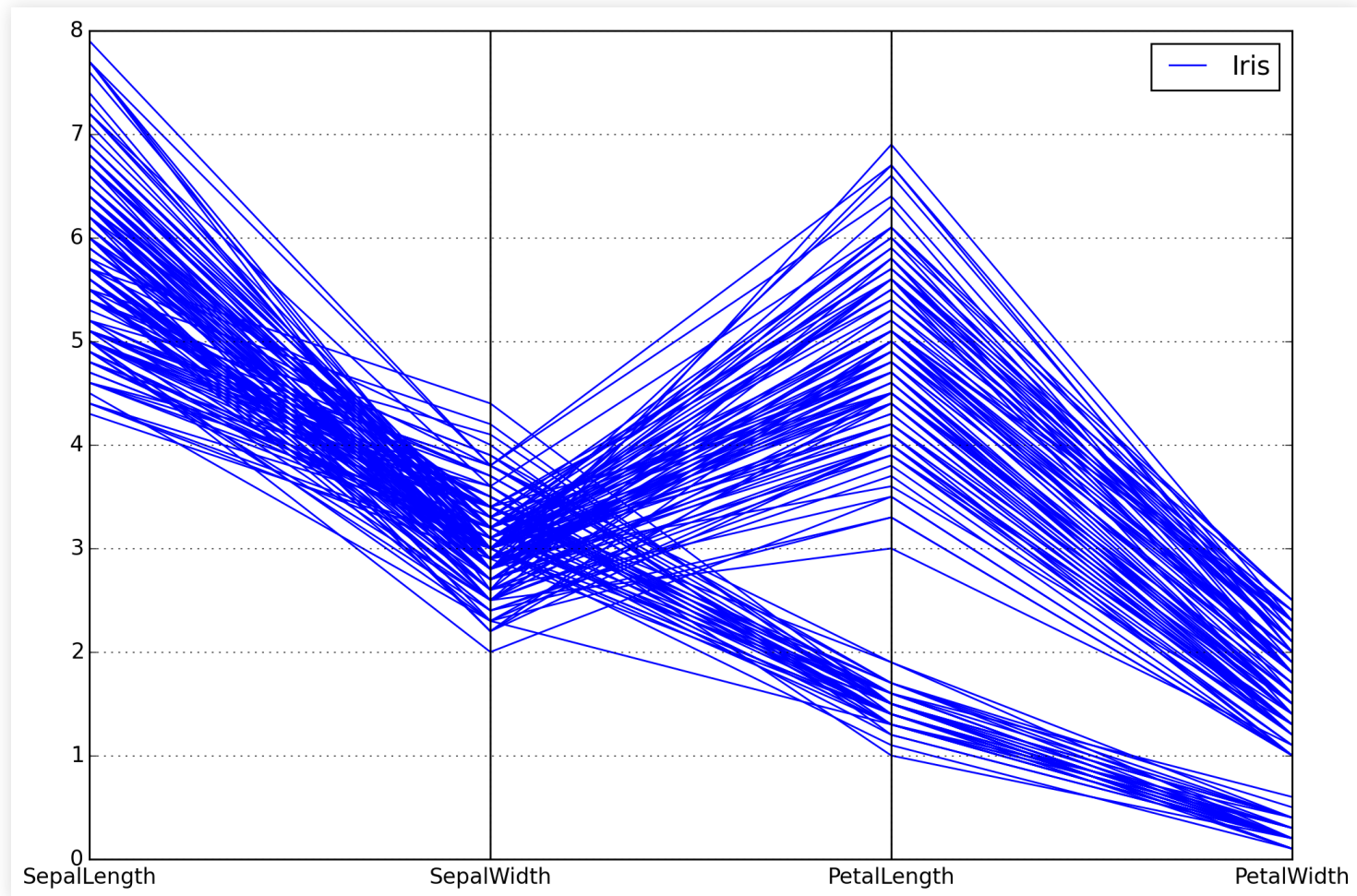
```
from pandas import read_csv
from pandas.plotting import parallel_coordinates
import matplotlib.pyplot as plt

data = read_csv('iris.data')
all_data=data.ix[:, [0,1,2,3]]
all_data['All']='Iris'

parallel_coordinates(all_data, 'All', color='b')
plt.savefig('L15-parallel-all.png', dpi=200, bbox_inches='tight')
plt.close()
```

Visualizing Data

■ Parallel coordinates plot



Visualizing Data

■ Parallel coordinates plot

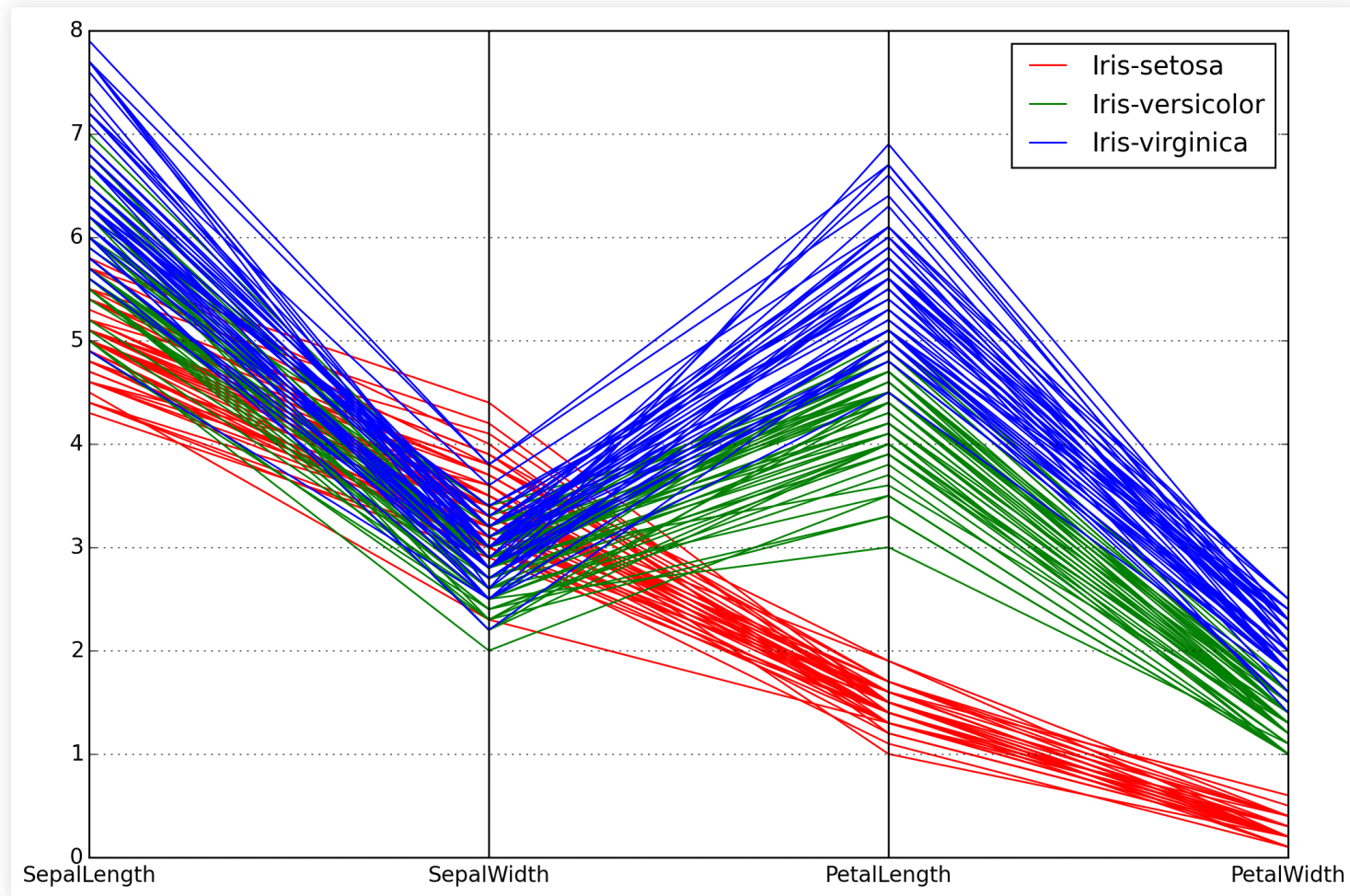
```
from pandas import read_csv
from pandas.plotting import parallel_coordinates
import matplotlib.pyplot as plt

data = read_csv('iris.data')

parallel_coordinates(data, 'Name', color=('r','g','b'))
plt.savefig('L15-parallel.png', dpi=200, bbox_inches='tight')
plt.close()
```

Visualizing Data

■ Parallel coordinates plot



Visualizing Data

Andrew's curves

- (Andrews, D. F. 1972. Plots of High Dimensional Data. Biometrics, 28:125-136)
- Convert each data point $\vec{x} = \{x_1, x_2, x_3, \dots\}$ into:

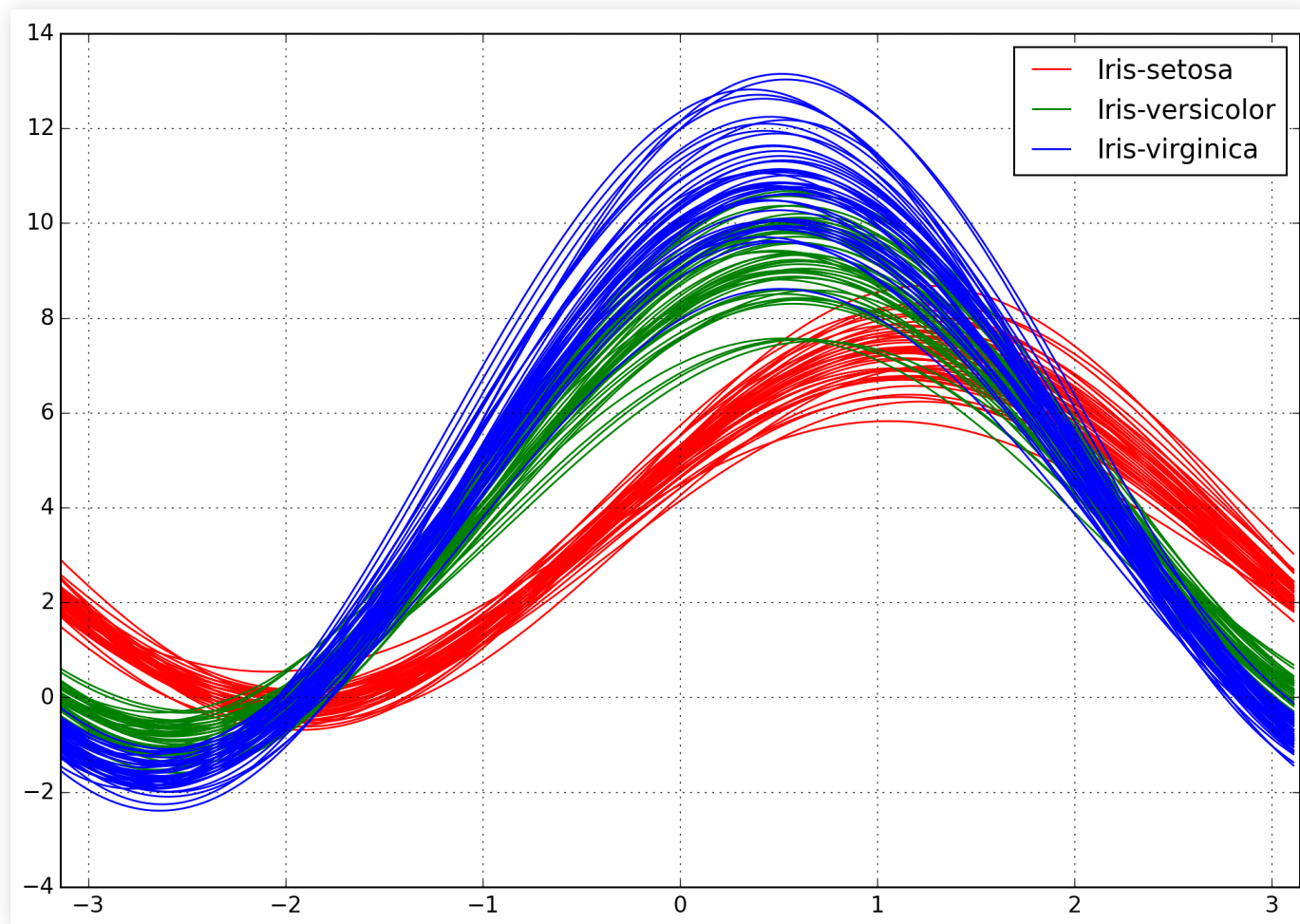
$$f_{\vec{x}}(t) = \frac{x_1}{\sqrt{2}} + x_2 \sin(t) + x_3 \cos(t) + x_4 \sin(2t) + x_5 \cos(2t) \dots$$

```
from pandas import read_csv
import matplotlib.pyplot as plt
from pandas.plotting import andrews_curves

data = read_csv('iris.data')
andrews_curves(data, 'Name', color=('r', 'g', 'b'))
plt.savefig('L15-andrews.png', dpi=200, bbox_inches='tight')
plt.close()
```


Visualizing Data

■ Andrew's curves



Visualizing Data

Radial visualization

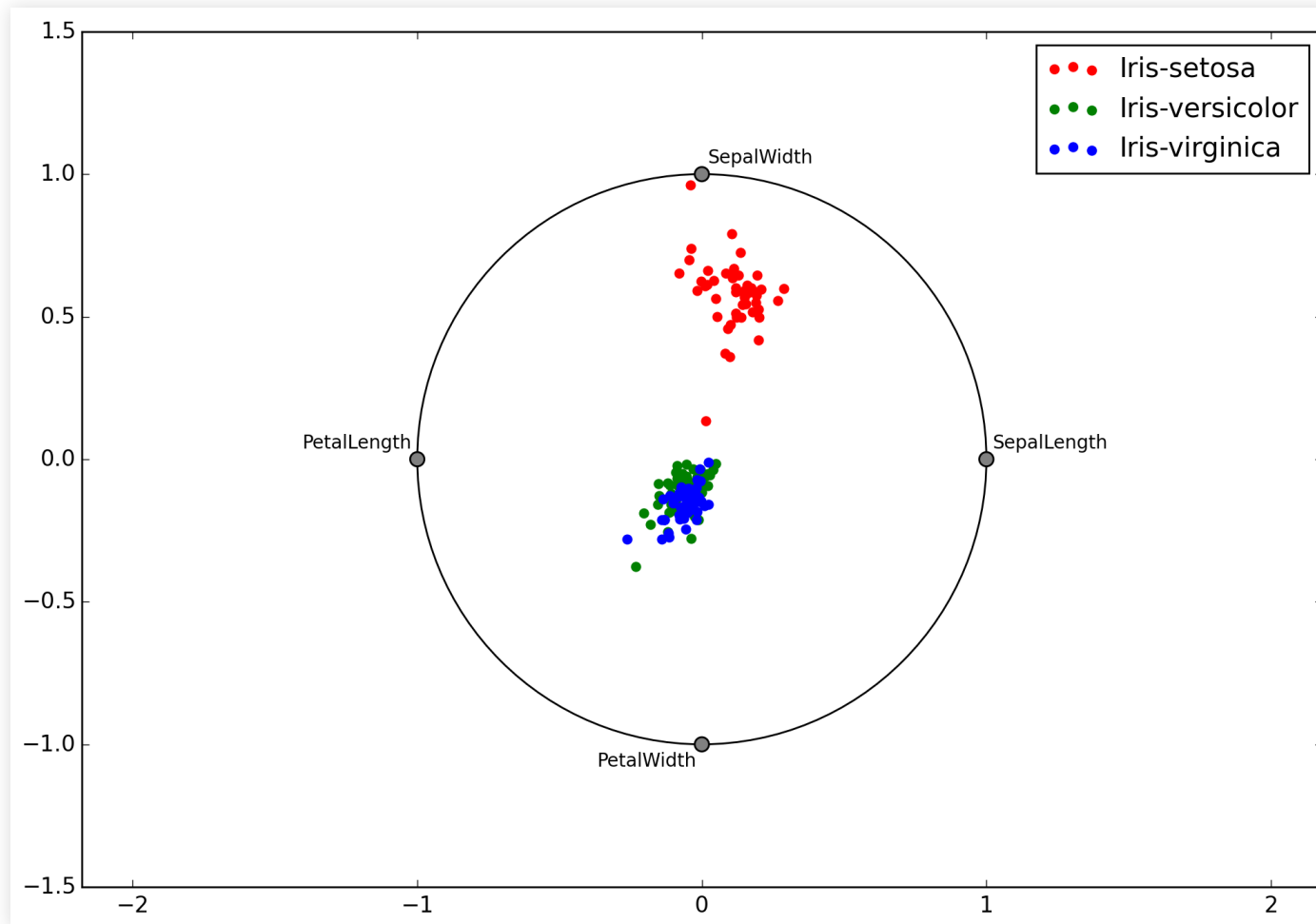
- (RADVIZ, Hoffman et al. Visualization'97)
- Feature axes spread radially
- Each data point is "pulled" along each axis according to value

```
from pandas import read_csv
import matplotlib.pyplot as plt
from pandas.plotting import radviz

data = read_csv('iris.data')
radviz(data, 'Name', color=('r','g','b'))
plt.savefig('L15-radviz.png', dpi=200, bbox_inches='tight')
plt.close()
```

Visualizing Data

■ Radial visualization



Visualizing Data

Problem: visualize data in more than 2 dimensions

- Need some way to represent relations
 - Pairwise plots
 - Coordinate transformations (parallel, radial)
- Need to understand distributions
 - Histograms, KDE, box plots

Feature Selection

Feature Selection

Why select features?

- Not all features are equally useful (e.g noisy data)
- Too many features for available data
- Need to simplify model
- Decide if features are worth measuring

Feature selection

- Reduce dimensionality by picking the best features
- The problem is in deciding which are the best...

Feature Selection

Feature selection: Univariate filter

- Compare features with variable to predict (supervised learning)
- E.g. test each filter against class

- Chi-square (χ^2) test

$$\chi^2 = \sum_{i=1}^N \frac{(O_i - E_i)^2}{E_i}$$

- Expected (assuming independency) vs Observed
- (features must be boolean or frequencies)

Feature selection: Univariate filter

- Analysis of Variance (ANOVA) F-test
- If feature values have the same distribution over all classes:

$$F = \frac{\text{variance between classes}}{\text{variance within classes}}$$

$$F = \frac{\sum_{i=1}^K n_i (\bar{x}_i - \bar{x})^2 / (K - 1)}{\sum_{i=1}^K \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2 / (N - K)}$$

Feature Selection

■ F-test with Scikit-Learn:

```
from sklearn.feature_selection import f_classif
from sklearn import datasets

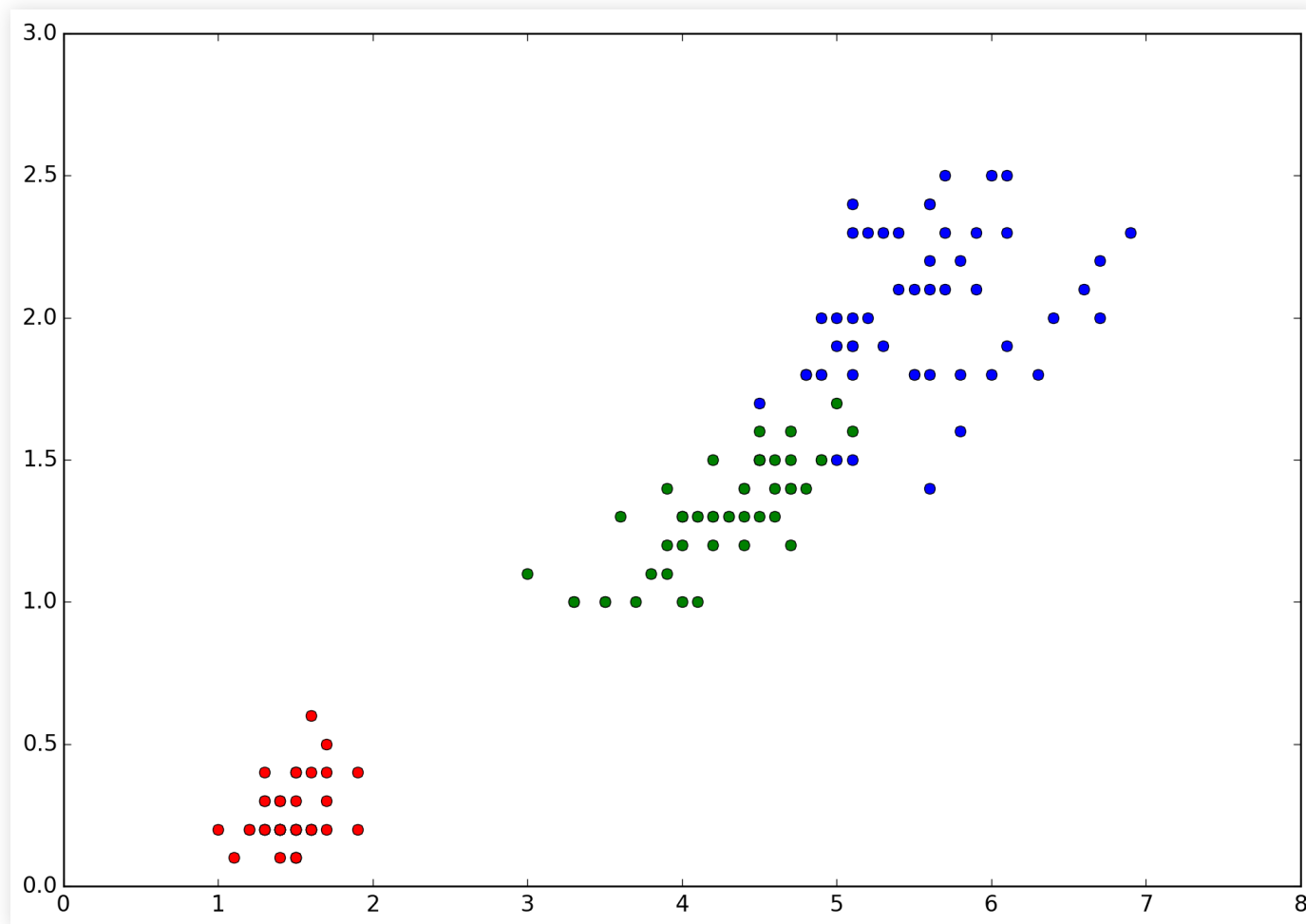
iris = datasets.load_iris()
X = iris.data
y = iris.target
f, prob = f_classif(X, y)
print(f)
print(prob)
```

■ Most correlated with class are 3 and 4

```
[ 119.26450218  47.3644614  1179.0343277  959.32440573]
[ 1.66966919e-31  1.32791652e-16  3.05197580e-91  4.37695696e-85]
```

Feature Selection

- F-test, best 2 features for Iris:



Feature Selection

Feature selection: Univariate filter

■ Feature selection with Scikit-Learn:

```
from sklearn.feature_selection import f_classif
from sklearn import datasets
from sklearn.feature_selection import SelectKBest

iris = datasets.load_iris()
X = iris.data
y = iris.target
X_new = SelectKBest(f_classif, k=2).fit_transform(X, y)
```

Feature Selection

Feature selection: Univariate filter

- Compare each attribute to predicted variable
- Requires target values Y
- (But then can be used for supervised or unsupervised learning)
- Other possible measures
 - Pearson Correlation
 - Mutual information
 - Distance correlation

Feature selection with model-based ranking:

- Train model with one feature; compare performance
- (Wrapper methods)

Feature selection: Multivariate filter

- Correlation-based feature selection
- Ideally, features should correlate with the class but not with each other
- Relevant feature : correlates with class (supervised learning only)
- Redundant feature : correlates with another feature (unsupervised learning too)

Wrapper Methods

- Wrapper methods use a score and a search algorithm
 - Score to evaluate performance of a feature set (classification, cluster quality)
 - The search algorithm explores different combinations of features to find the best
- Wrapper methods generally use the same machine learning algorithm for which the features will be used

Deterministic Wrapper

■ Sequential forward selection:

- Add one feature at a time, choosing best improvement
- Stop when reaching desired number

■ Sequential backward elimination:

- Start with all features and exclude feature leading to best improvement
- Stop when reaching desired number

Non-deterministic Wrapper

■ Use non-deterministic search algorithms

- Simulated annealing
- Genetic algorithms

Embedded Feature selection

- Feature selection is part of the learning algorithm
 - Decision trees: more relevant features used first; others may not be used
 - Feature-weighted Naive Bayes: features are weighted differently based on, for example, mutual information between class and conditional feature distribution (how dissimilar a posteriori is to a priori)
- Feature selection can also be embedded by regularization
 - L1 regularization penalizes the absolute value of parameters, forcing some to 0.
 - (Logistic Regression in Scikit-Learn can be done with L1 regularization)

Feature selection methods

- Filter: applied first, independent of learning algorithm
- Wrapper: searches combinations using some scoring function related to the learning algorithm
- Embedded: part of the learning algorithm

Summary

Unsupervised Learning

Summary

- Unsupervised learning: no error on predictions
- Visualizing data
- Feature selection
- Mostly for supervised learning
- But wrapper methods can be used in unsupervised learning

Further reading

- Pandas, visualization tutorial
- Scikit Learn, Feature Selection tutorial
- Alpaydin, Section 6.2 (6.9: more references)
- Saeys et. al., A review of feature selection techniques in bioinformatics (Bioinformatics 2007)

