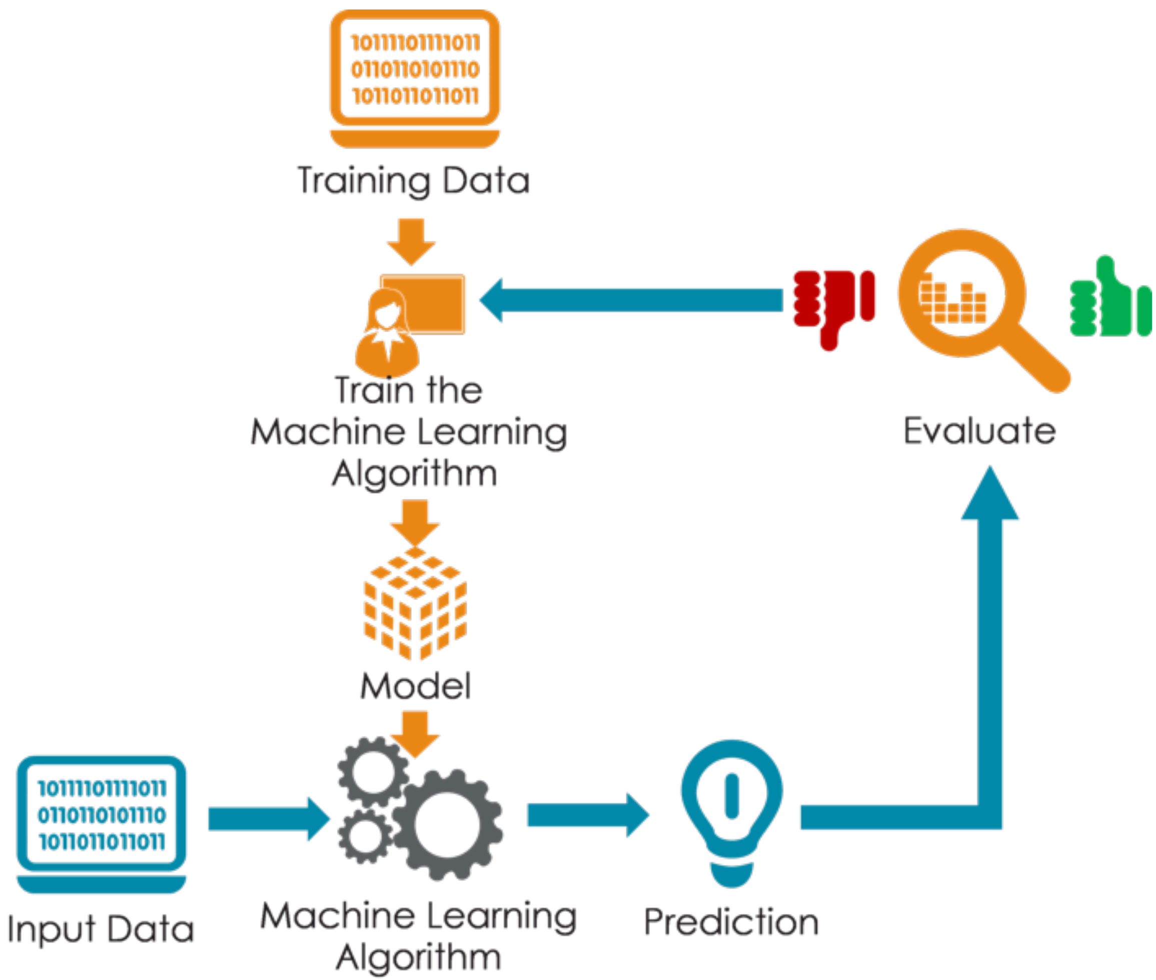
Machine Learning



SUPERVISED LEARNING

Predict outputs based on inputs

SUPERVISED LEARNING

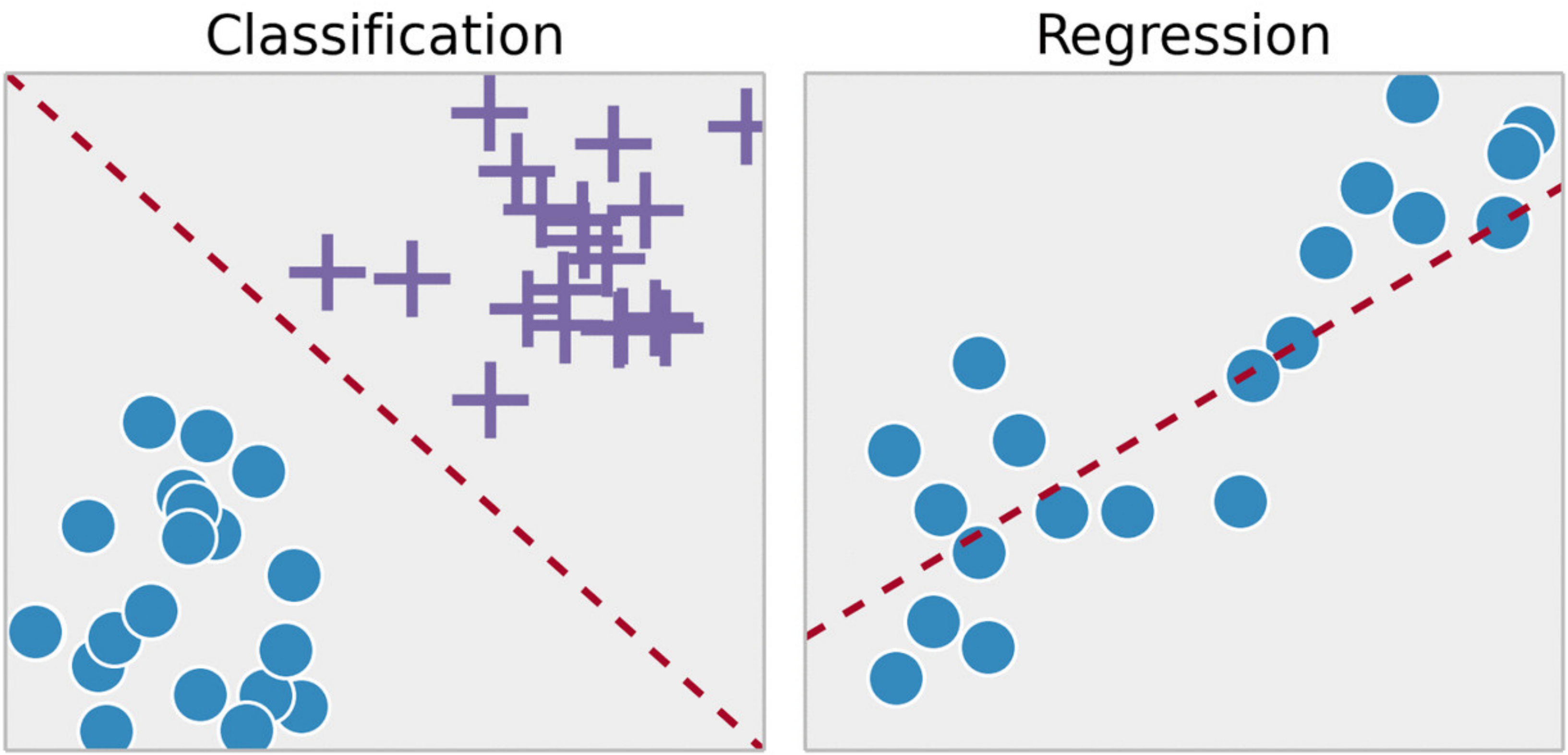
independent variables
predictors
attributes
features

Predict outputs based on inputs

targets
responses
outcomes
dependent variables

Supervised Learning

Predict outputs based on inputs



Unsupervised Learning

There is no output measure

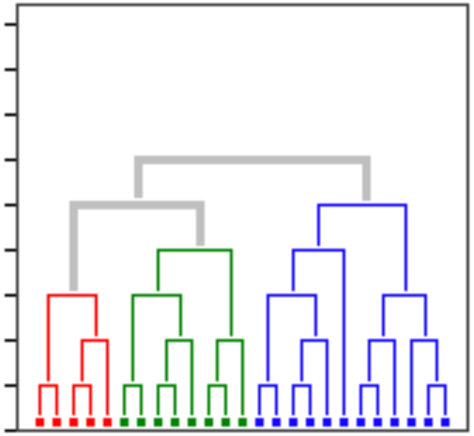
Describe associations/patterns among inputs

UNSUPERVISED LEARNING

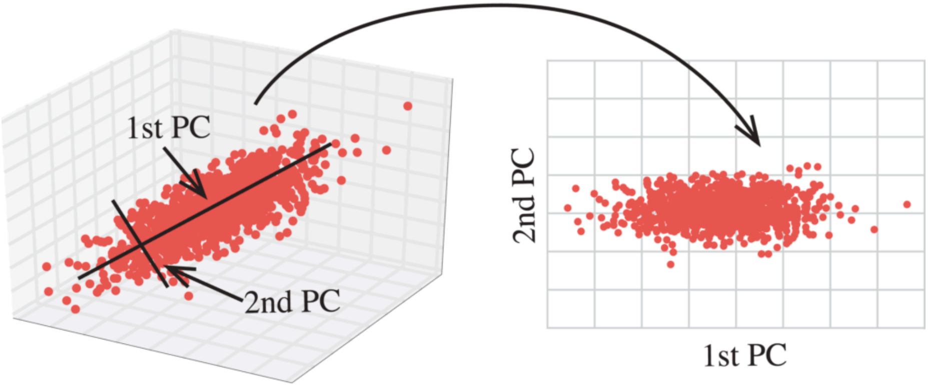
There is no output measure

Describe associations/patterns among inputs

Clustering



Dimensionality Reduction

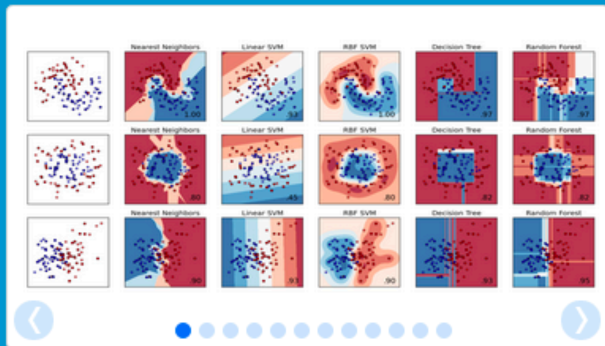


scikit learn

Home Installation Documentation Examples

Google™ Custom Search | Search ×

Fork me on GitHub



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: *SVM*, *nearest neighbors*, *random forest*, ... — *Examples*

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: *SVR*, *ridge regression*, *Lasso*, ... — *Examples*

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: *k-Means*, *spectral clustering*, *mean-shift*, ... — *Examples*

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: *PCA*, *Isomap*, *non-negative matrix factorization*. — *Examples*

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: *grid search*, *cross validation*, *metrics*. — *Examples*

Preprocessing

Feature extraction and normalization.

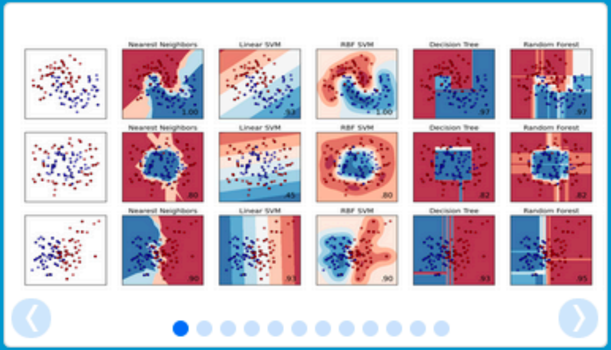
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: *preprocessing*, *feature extraction*. — *Examples*

scikit learn

Home | Installation | Documentation | Examples

Google™ Custom Search | Search ×

Fork me on GitHub



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: *SVM*, *nearest neighbors*, *random forest*, ... — *Examples*

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: *SVR*, *ridge regression*, *Lasso*, ... — *Examples*

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: *k-Means*, *spectral clustering*, *mean-shift*, ... — *Examples*

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: *PCA*, *Isomap*, *non-negative matrix factorization*. — *Examples*

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: *grid search*, *cross validation*, *metrics*. — *Examples*

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: *preprocessing*, *feature extraction*. — *Examples*

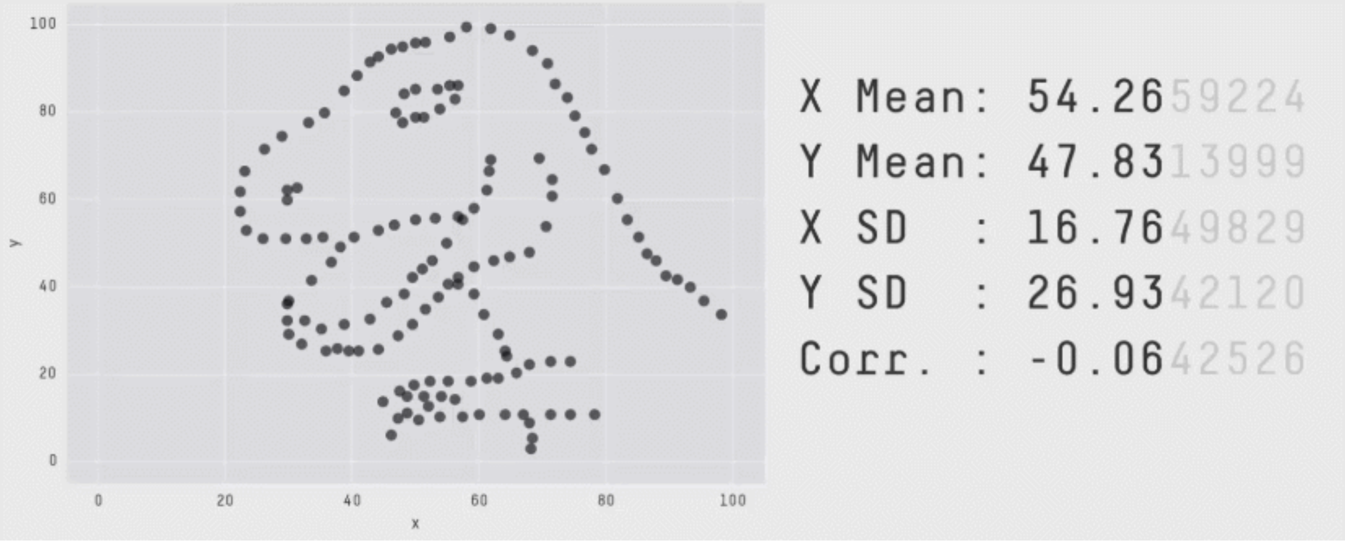
CORRELATION

- Measure the direction and strength of a linear relationship.

$$r = \frac{\sum_{i=1}^{n}\left(\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2 \sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}}$$

CORRELATION

- Measure the direction and strength of a **linear relationship**.



CORRELATION

- Measure the direction and strength of a linear relationship.

- Cannot be used for prediction purposes.

Regression Analysis

1. State the problem
2. Select potentially relevant variables
3. Specify the model
4. Fit the model
5. Critically evaluate the model
6. Address the original problem

LINEAR REGRESSION

- Express functional relationships among variables as an equation or “model”

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Linear Regression

- Express functional relationships among variables as an equation or “model”

parameters to be estimated
“coefficients”

random error
“noise”

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

intercept

slope

response variable

predictor variable

LINEAR REGRESSION

Simple linear regression:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Multiple linear regression:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$$

Linear Regression

Linear refers to the way that the parameters enter the model…

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$

Linear

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$$

Non-linear

$$Y = \beta_0 + e^{\beta_1 X_1} + \varepsilon$$

Data Notation

Observation Number	Response Y	Predictors X_1	X_2	$\cdots$	X_p
1	y_1	x_{11}	x_{12}	$\cdots$	x_{1p}
2	y_2	x_{21}	x_{22}	$\cdots$	x_{2p}
3	y_3	x_{31}	x_{32}	$\cdots$	x_{3p}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	y_n	x_{n1}	x_{n2}	$\cdots$	x_{np}

DATA NOTATION

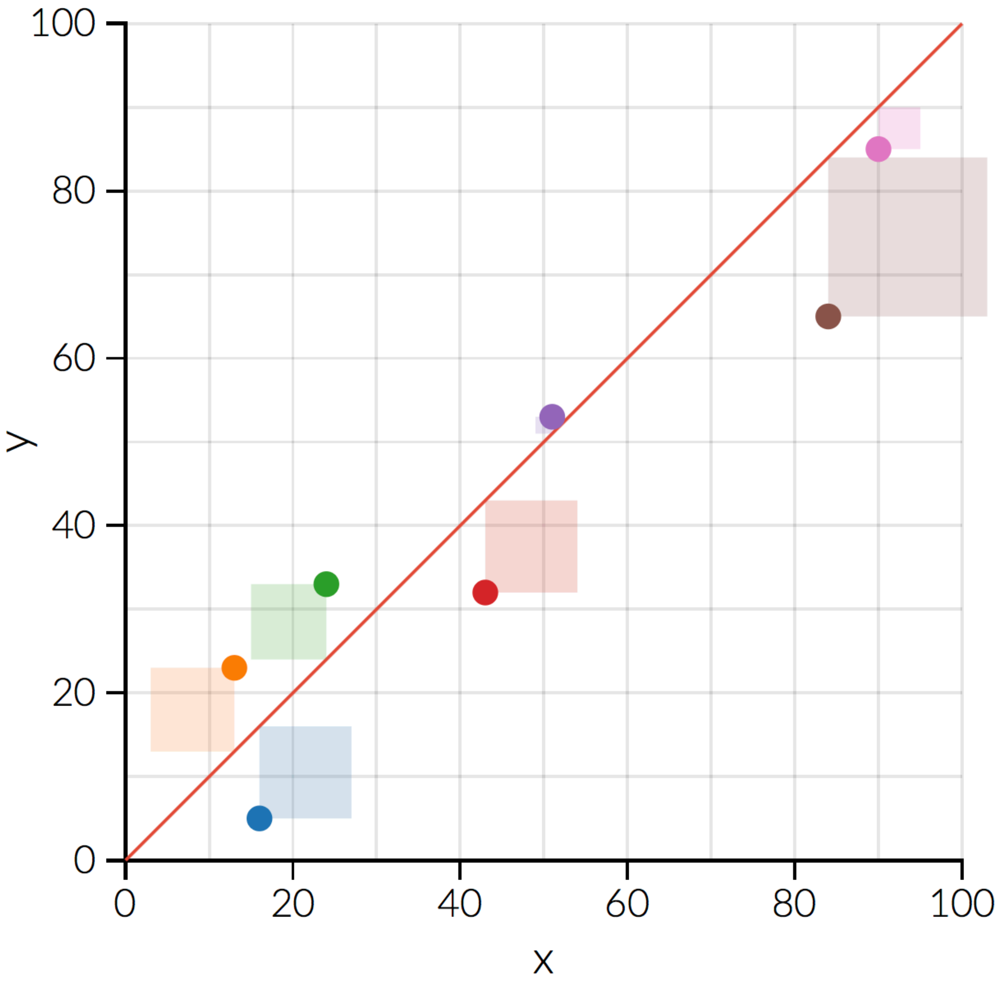
Observation Number	Response Y	Predictors X_1	X_2	$\cdots$	X_p
1	y_1	x_{11}	x_{12}	$\cdots$	x_{1p}
2	y_2	x_{21}	x_{22}	$\cdots$	x_{2p}
3	y_3	x_{31}	x_{32}	$\cdots$	x_{3p}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	y_n	x_{n1}	x_{n2}	$\cdots$	x_{np}

…what does this data structure resemble?

Model Fitting

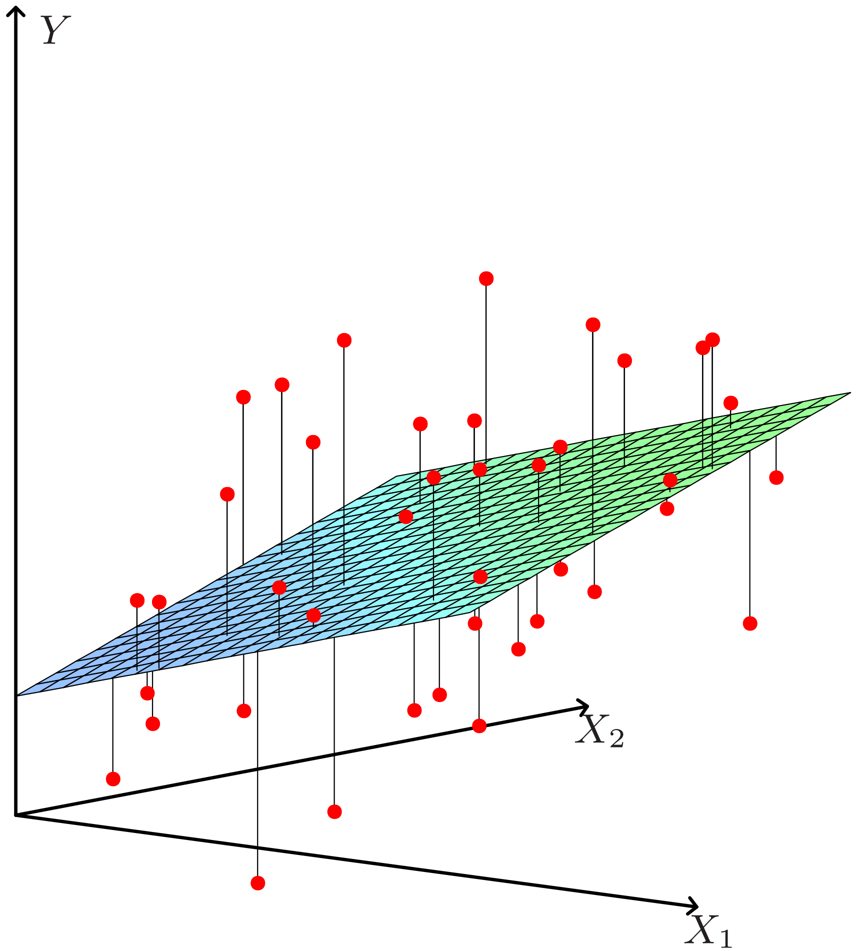
Minimize the sum of squared residuals…

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$



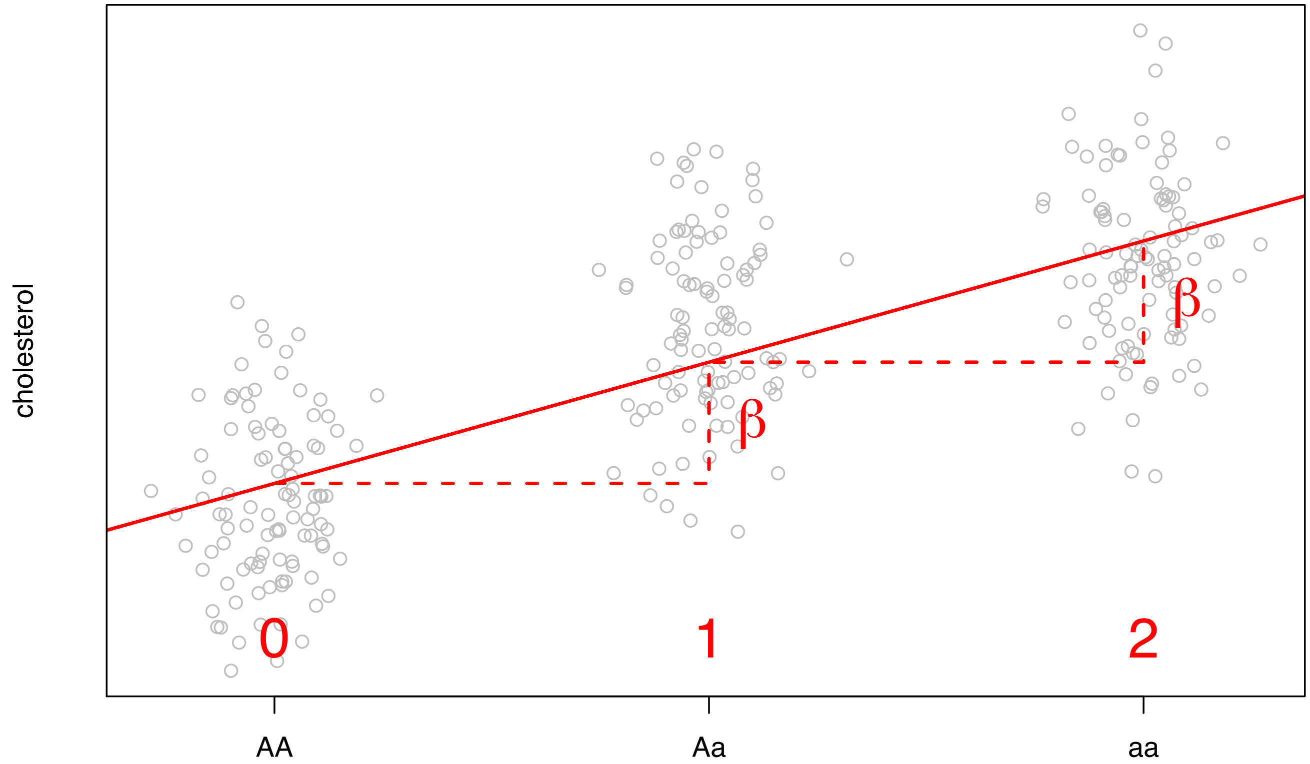
Model Fitting

Same concept applies in two or more dimensions…



Applications of Linear Regression

$$y = \beta_0 + \beta \times \#\text{minor alleles}$$



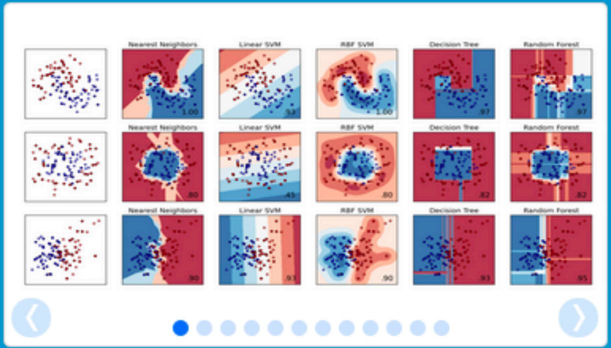
GENOME-WIDE ASSOCIATION STUDIES

Hundreds of thousands of linear regressions…

Home Installation Documentation Examples

Google™ Custom Search | Search

Fork me on GitHub



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: *SVM*, *nearest neighbors*, *random forest*, ... — *Examples*

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: *SVR*, *ridge regression*, *Lasso*, ... — *Examples*

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: *k-Means*, *spectral clustering*, *mean-shift*, ... — *Examples*

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: *PCA*, *Isomap*, *non-negative matrix factorization*. — *Examples*

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: *grid search*, *cross validation*, *metrics*. — *Examples*

Preprocessing

Feature extraction and normalization.

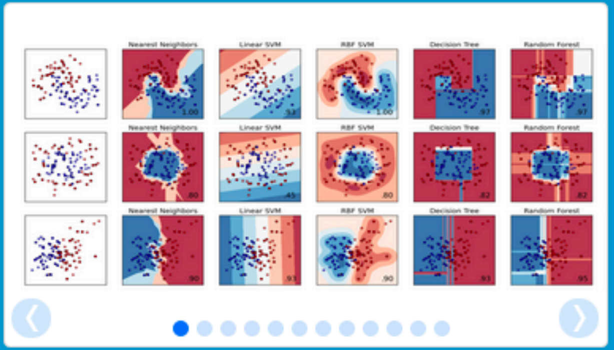
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: *preprocessing*, *feature extraction*. — *Examples*

scikit learn

Home Installation Documentation Examples

Google™ Custom Search Search

Fork me on GitHub



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: *SVM*, *nearest neighbors*, *random forest*, ... — *Examples*

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: *SVR*, *ridge regression*, *Lasso*, ... — *Examples*

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: *k-Means*, *spectral clustering*, *mean-shift*, ... — *Examples*

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: *PCA*, *Isomap*, *non-negative matrix factorization*. — *Examples*

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: *grid search*, *cross validation*, *metrics*. — *Examples*

Preprocessing

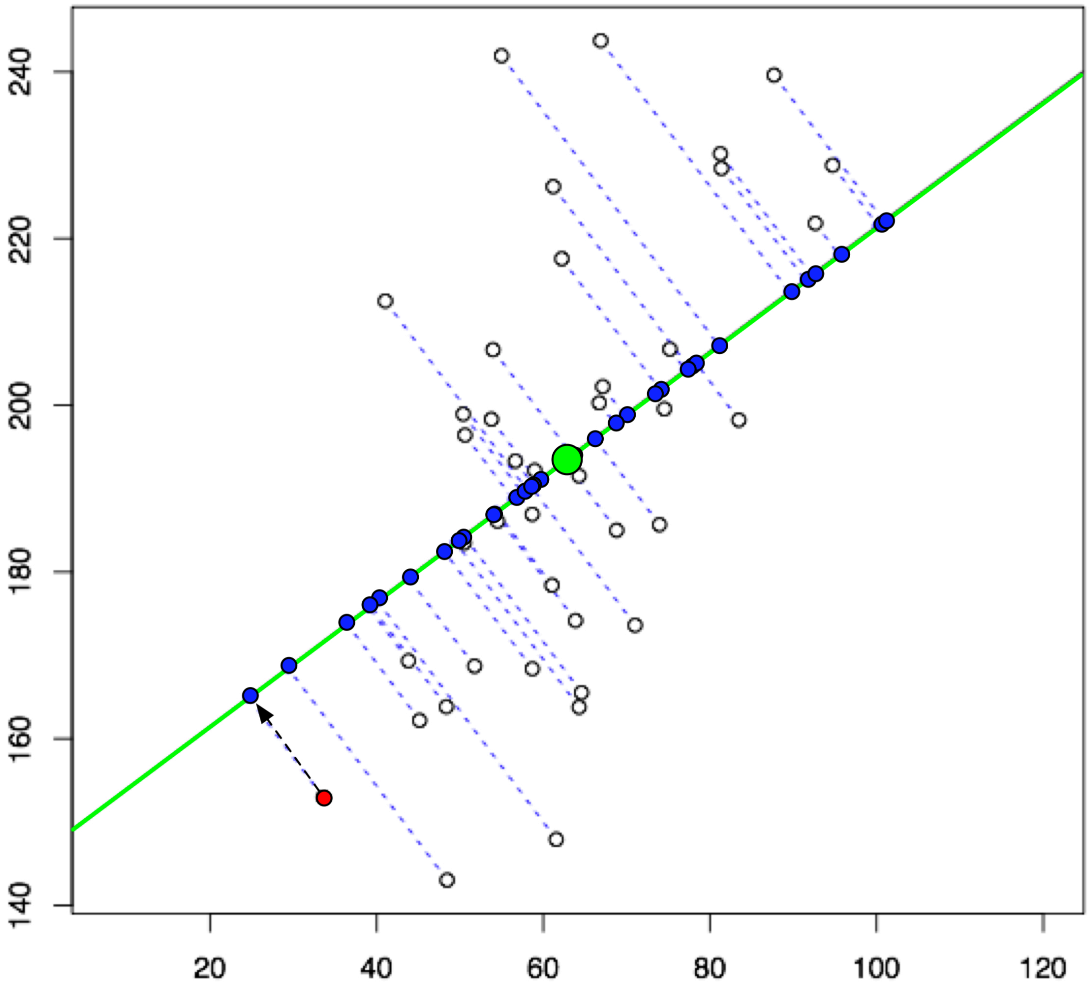
Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: *preprocessing*, *feature extraction*. — *Examples*

Principal Component Analysis

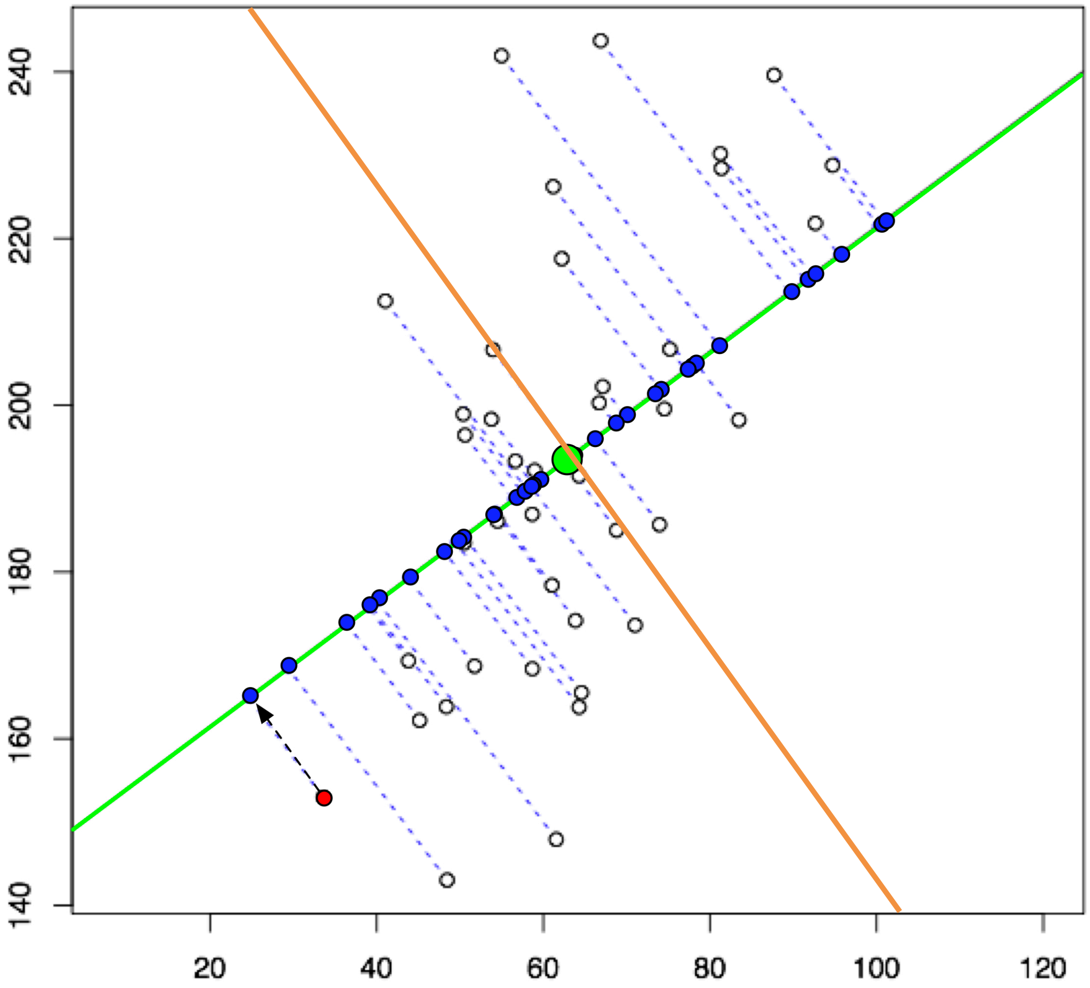
Transform a large number of possibly correlated variables into a smaller number of uncorrelated variables ("principal components") that capture as much information as possible from the original dataset.

Principal Component Analysis



The **first principal component** is a linear combination of original predictor variables which captures the maximum variance in the data set.

PRINCIPAL COMPONENT ANALYSIS

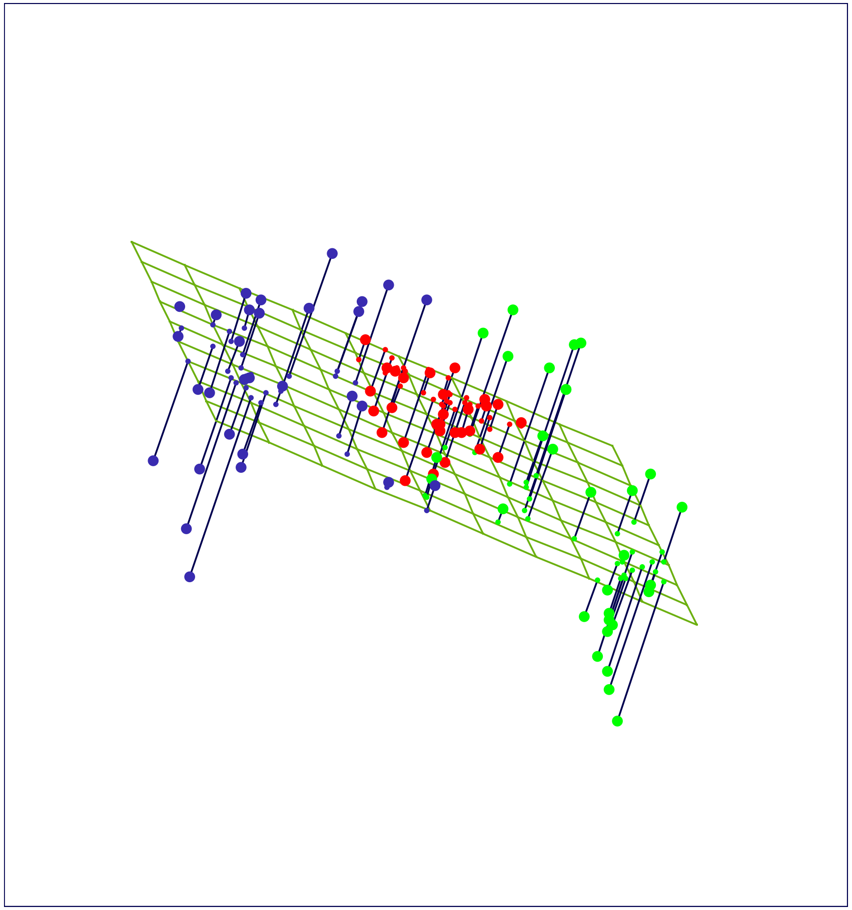


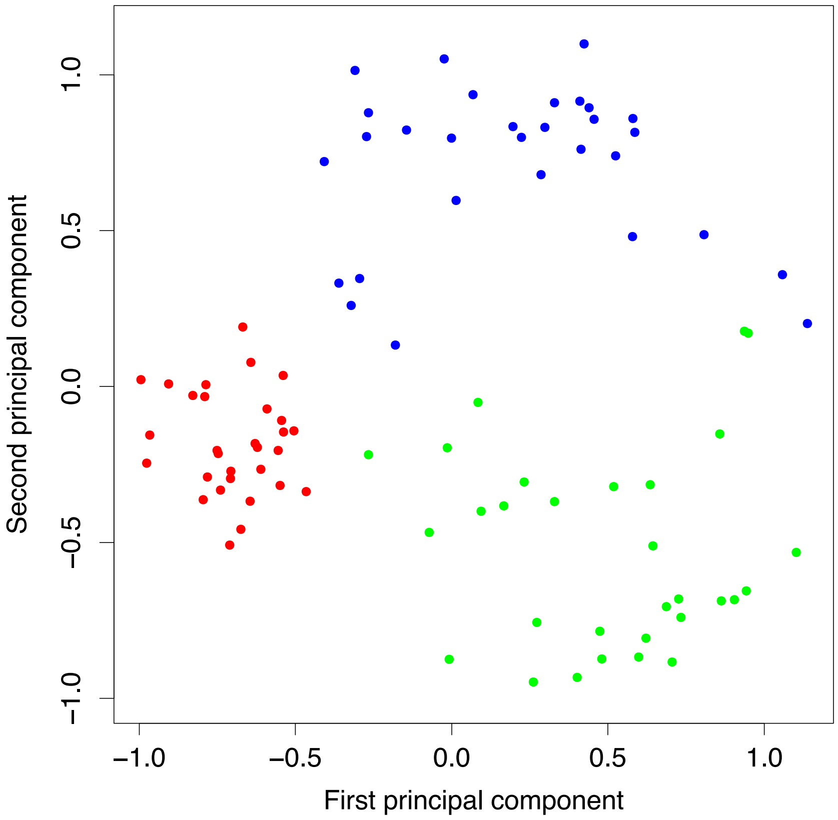
The **first principal component** is a linear combination of original predictor variables which captures the maximum variance in the data set.

The **second principal component** is a linear combination of original predictor variables which captures the second most variance under the constraint that it is orthogonal to PC1.

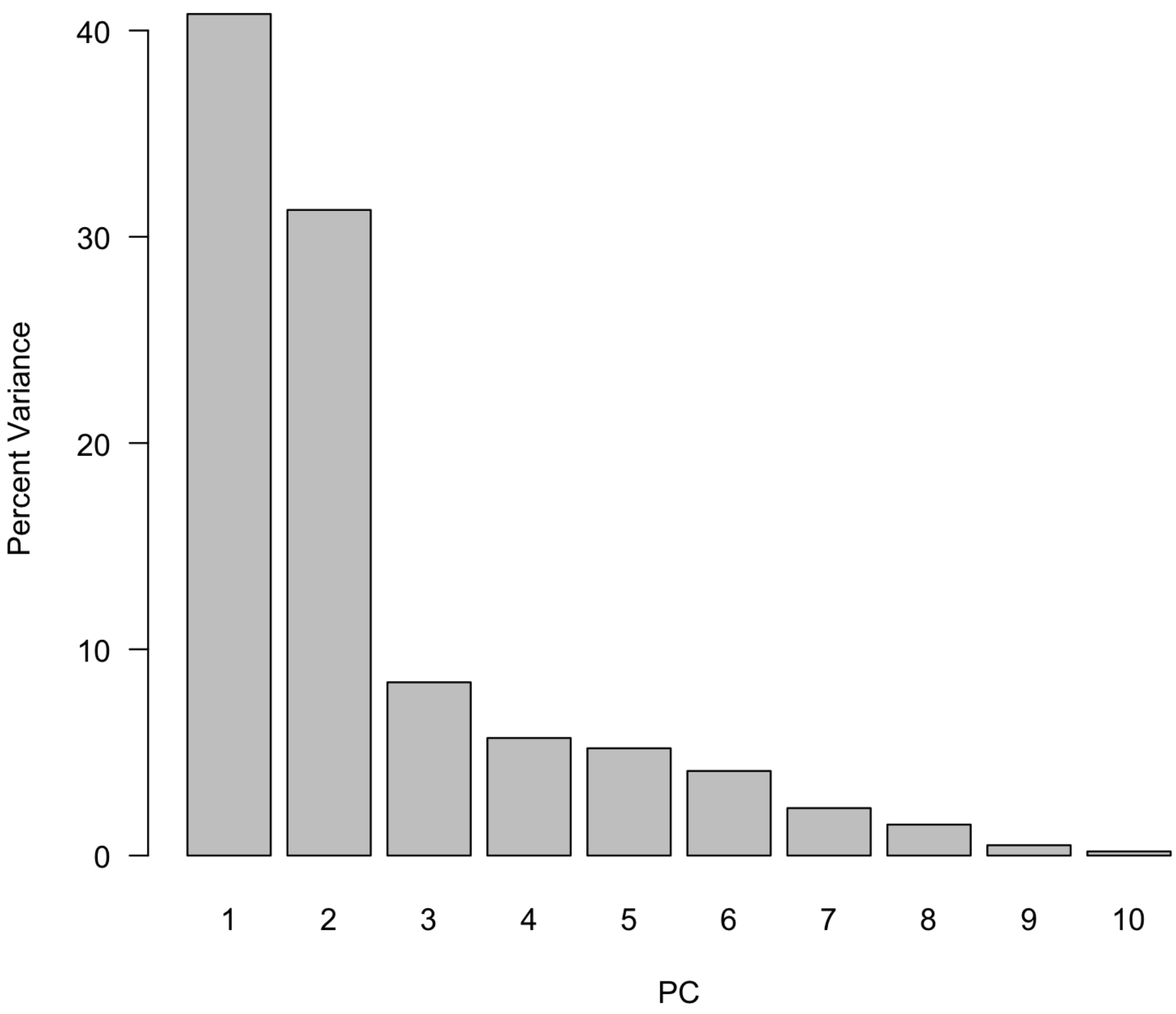
Principal Component Analysis

Reduce dimensionality and visualize data structure…

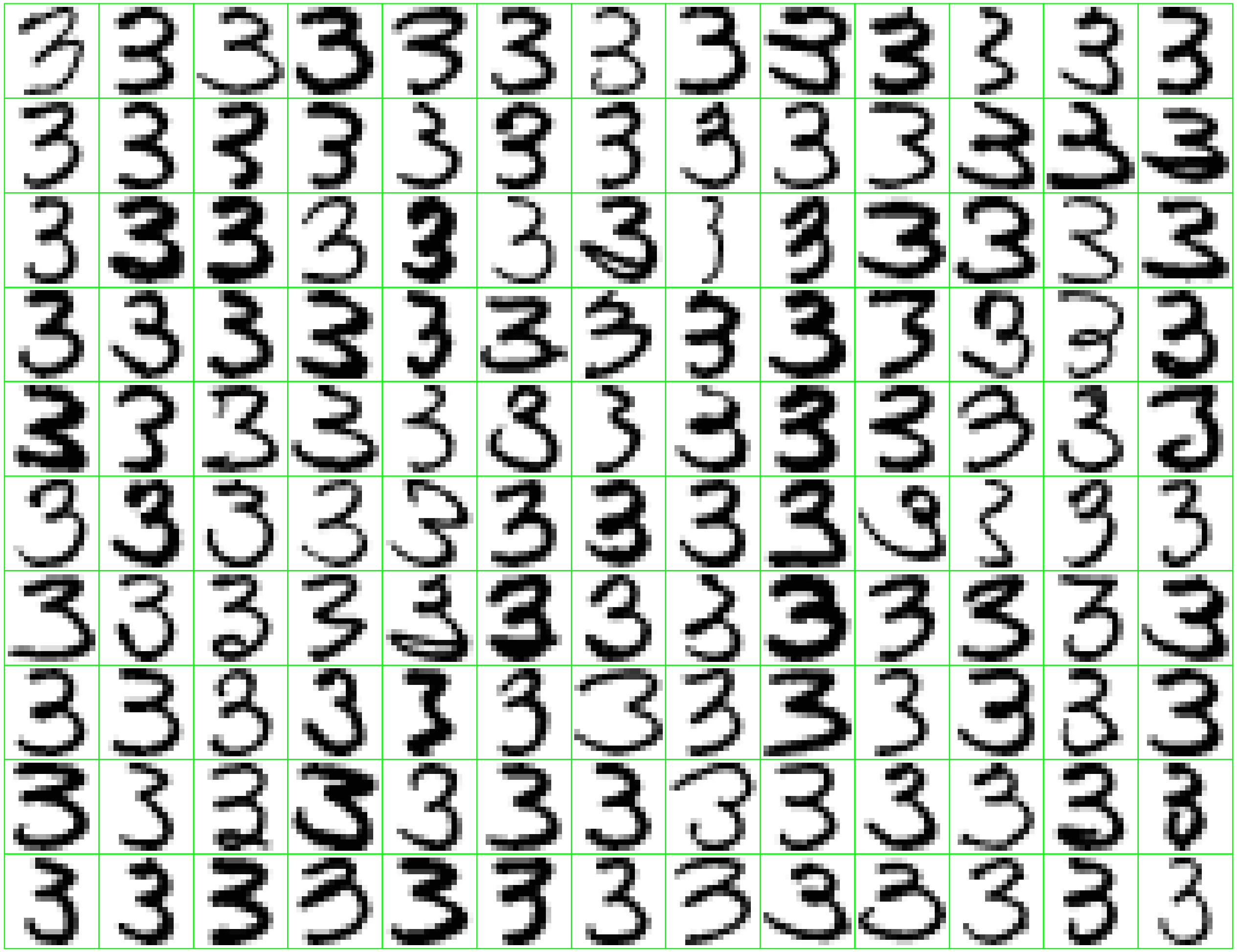




Principal Component Analysis

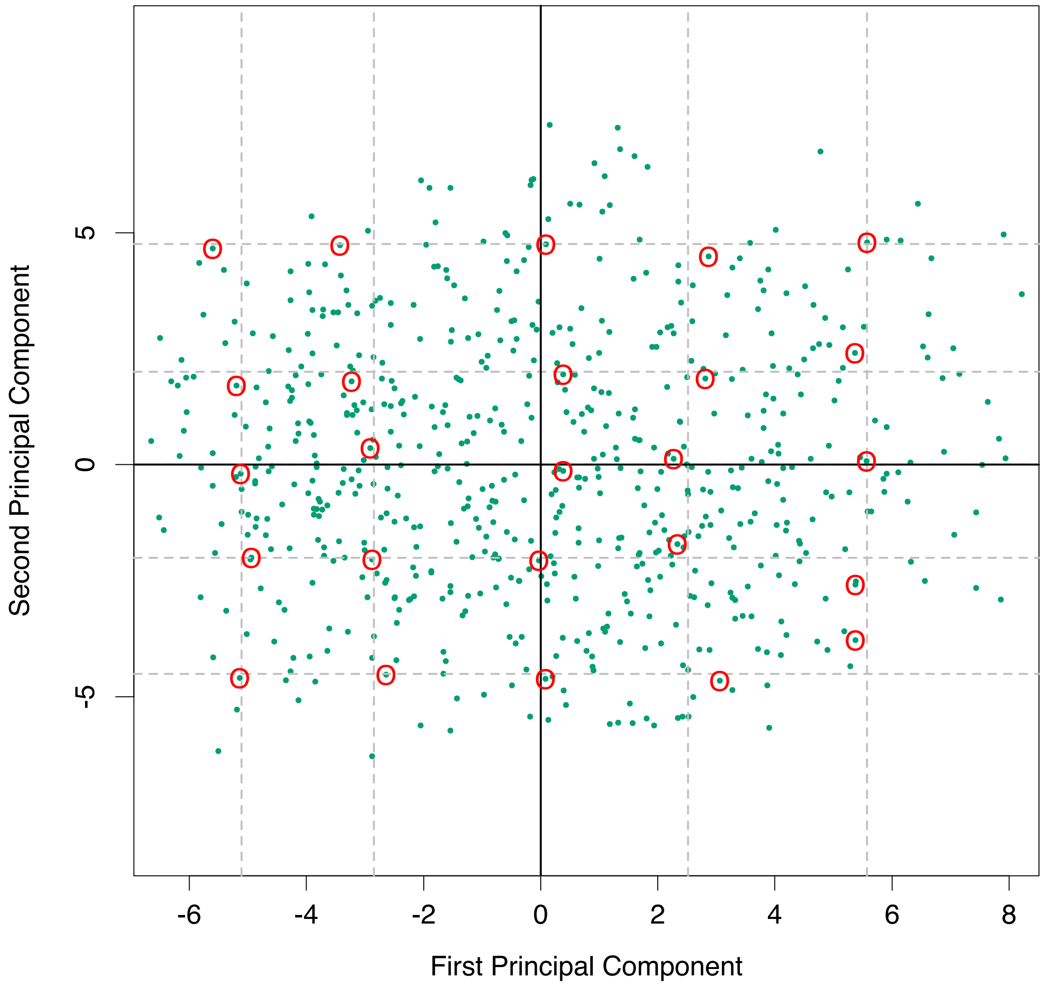


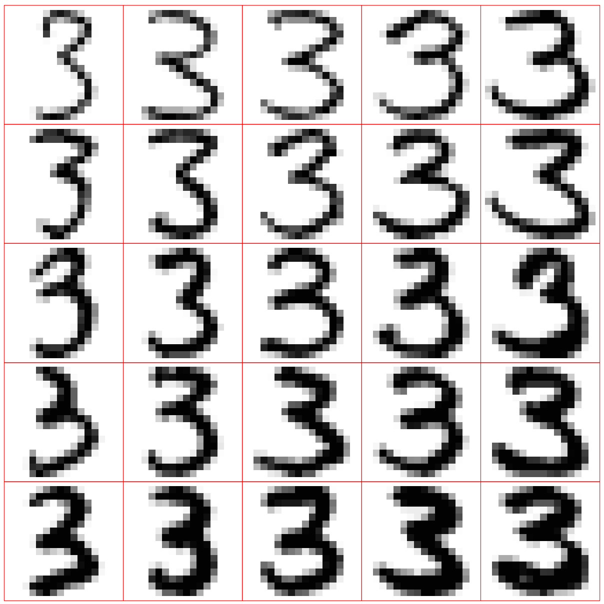
Applications of PCA



Hastie *et al*. (2009)

Applications of PCA





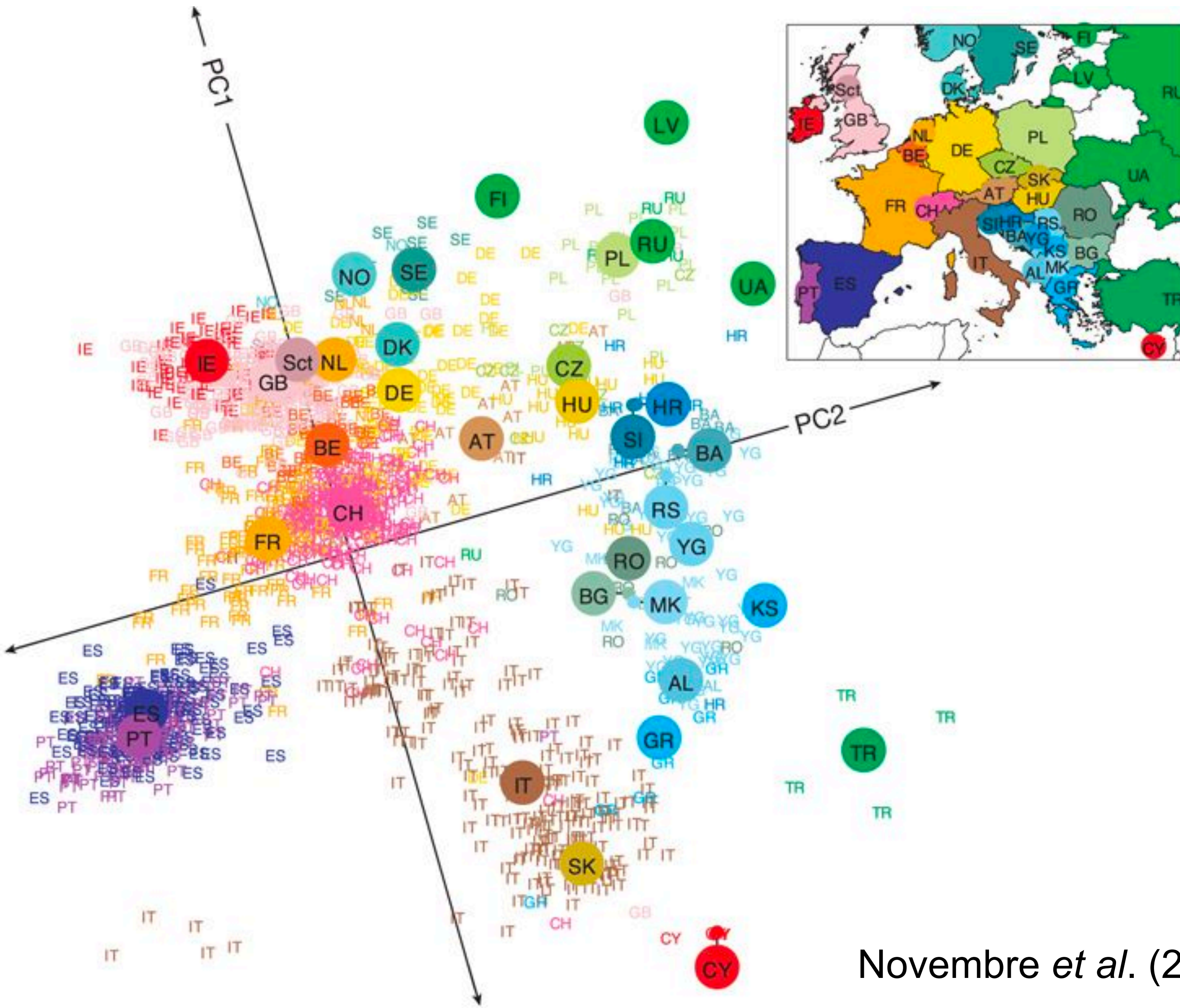
Applications of PCA

	rs10001	rs10002	rs10003	rs10004	rs10005	rs10006	...
Sample1	AA	GG	CC	AC	GG	CC	
Sample2	AA	GG	CT	AC	CG	CC	
Sample3	AT	GG	TT	AA	CG	TT	
Sample4	TT	GG	TT	AA	GG	CC	
Sample5	AA	GG	TT	AA	GG	TT	
Sample6	AA	GA	CC	AC	GG	CC	
Sample7	AT	GG	CC	CC	GG	CC	
Sample8	AA	GG	CT	AA	GG	CT	
...							

genotypes can
also be represented
counts of non-reference
alleles (i.e., 0, 1, 2)

$$\begin{bmatrix} 0 & 2 & 1 & 2 & 0 & 2 & 2 & 2 \\ 2 & 0 & 0 & 2 & 1 & 0 & 2 & 2 \\ 0 & 1 & 0 & 2 & 0 & 0 & 2 & 0 \\ 1 & 2 & 2 & 0 & 2 & 1 & 0 & 0 \end{bmatrix}$$

APPLICATIONS OF PCA



Novembre *et al*. (2008)