Machine Learning II
Week #1

Jan Nagler

Deep Dynamics Group
Centre for Human and Machine Intelligence (HMI)
Frankfurt School of Finance & Management

Frankfurt, October 2019

Nonlinear Correlation

Connections

Causation

Clustering

Linear Prediction

Sample size realities

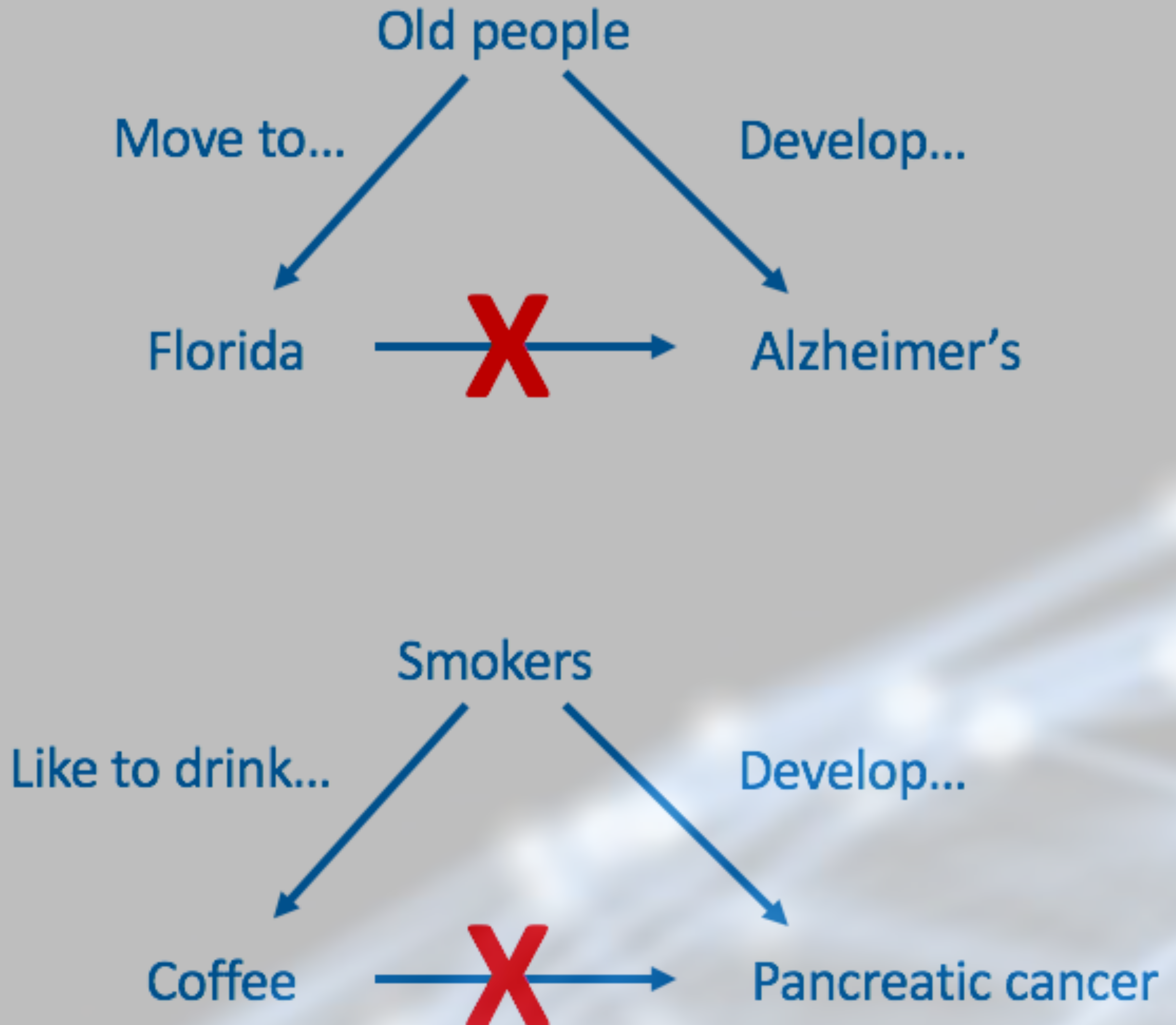
Methods

(Covariance, K-Means, PCA/OVA)

Data

(Surrogates creation, Wine-178)

Causality and Correlations



Anticorrelations (=neg. corr.) around year 2000

Pearson correlations = Covariance matrix (normalized) of Cryptos

Correlation between Top 15 Market Cap Currencies

@CryptoKit

Large-cap asset 3-month daily return correlation matrix (USD) 01/01/2017 - 10/04/2018

	BTC	ETH	XRP	BCH	LTC	EOS	ADA	XLM	NEO	MIOTA	XMR	DASH	TRX	XEM	ET
BTC															
ETH	0.45														
XRP	0.20	0.19													
BCH	0.21	0.25	0.14												
LTC	0.46	0.42	0.27	0.23											
EOS	0.31	0.31	0.17	0.23	0.27										
ADA	0.22	0.18	0.27	0.10	0.17	0.17									
XLM	0.28	0.27	0.50	0.12	0.31	0.20	0.31								
NEO	0.30	0.33	0.13	0.15	0.31	0.19	0.14	0.21							
MIOTA	0.44	0.39	0.18	0.23	0.35	0.32	0.32	0.34	0.26						
XMR	0.51	0.52	0.23	0.28	0.43	0.28	0.26	0.40	0.24	0.45					
DASH	0.42	0.45	0.10	0.31	0.37	0.23	0.16	0.20	0.29	0.34	0.56				
TRX	0.28	0.21	0.17	0.11	0.18	0.24	0.28	0.12	0.09	0.15	0.18	0.19			
XEM	0.28	0.35	0.22	0.21	0.34	0.23	0.24	0.32	0.22	0.37	0.32	0.29	0.13		
ETC	0.44	0.62	0.17	0.32	0.51	0.30	0.32	0.29	0.43	0.43	0.45	0.38	0.20	0.35	

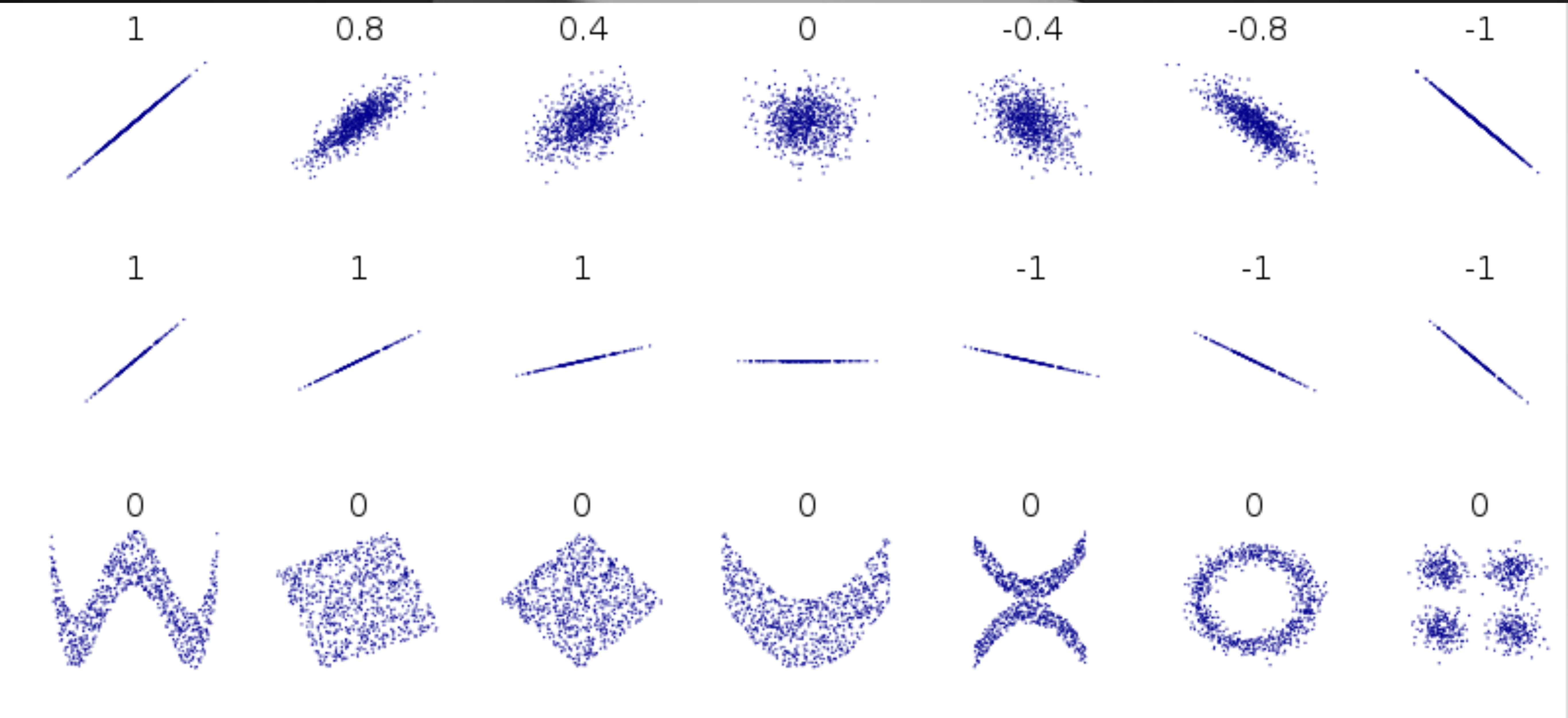
1.00	Strong Positive Relationship
0.50	Moderate Positive Relationship
0.30	Weak Positive Relationship
0.00	No Correlation
-0.30	Weak Negative
-0.50	Moderate Negative Relationship
-1.00	Strong Negative Relationship

Pearson correlations = Covariance matrix (normalized) of Cryptos

3-month daily return correlation matrix (USD) Dec 1 2018 - Mar 1 2019

	BTC	ADA	ETH	XLM	XMR	ZEC	NEO	LSK	XRP	ZRX	EOS	OMG	VET	DASH	QTU
BTC	1														
ADA	0.908	1													
ETH	0.889	0.900	1												
XLM	0.922	0.893	0.866	1											
XMR	0.929	0.886	0.881	0.878	1										
ZEC	0.909	0.843	0.865	0.861	0.915	1									
NEO	0.873	0.875	0.885	0.864	0.858	0.874	1								
LSK	0.901	0.830	0.837	0.836	0.838	0.847	0.818	1							
XRP	0.875	0.865	0.858	0.887	0.840	0.828	0.847	0.819	1						
ZRX	0.879	0.827	0.834	0.857	0.829	0.828	0.803	0.867	0.812	1					
EOS	0.884	0.889	0.851	0.860	0.824	0.813	0.836	0.770	0.818	0.769	1				
OMG	0.843	0.822	0.846	0.802	0.829	0.839	0.755	0.837	0.740	0.847	0.751	1			
VET	0.855	0.862	0.794	0.839	0.801	0.794	0.793	0.848	0.820	0.820	0.810	0.792	1		
DASH	0.885	0.800	0.799	0.819	0.875	0.892	0.769	0.802	0.752	0.807	0.770	0.840	0.745	1	
QTUM	0.794	0.831	0.826	0.801	0.727	0.746	0.805	0.784	0.793	0.773	0.820	0.754	0.800	0.697	
LTC	0.857	0.851	0.825	0.801	0.805	0.766	0.793	0.753	0.769	0.750	0.831	0.758	0.767	0.706	0.75
ETC	0.783	0.778	0.794	0.760	0.760	0.769	0.773	0.769	0.736	0.806	0.734	0.741	0.724	0.712	0.75
DCR	0.815	0.785	0.795	0.740	0.795	0.785	0.738	0.819	0.760	0.740	0.714	0.722	0.764	0.702	0.69
NEM	0.777	0.766	0.755	0.804	0.755	0.759	0.768	0.777	0.754	0.780	0.652	0.759	0.759	0.699	0.70
IOTA	0.768	0.756	0.768	0.759	0.760	0.752	0.675	0.694	0.738	0.756	0.729	0.777	0.697	0.752	0.78
BCH	0.790	0.752	0.738	0.738	0.778	0.752	0.622	0.745	0.679	0.727	0.675	0.784	0.664	0.815	0.62
ONT	0.697	0.760	0.678	0.743	0.710	0.676	0.747	0.624	0.701	0.618	0.751	0.657	0.709	0.599	0.67
BAT	0.719	0.703	0.728	0.704	0.707	0.702	0.703	0.711	0.674	0.668	0.626	0.688	0.650	0.656	0.62
XTZ	0.696	0.714	0.720	0.711	0.680	0.709	0.724	0.682	0.669	0.666	0.696	0.635	0.605	0.656	0.62
MKR	0.643	0.662	0.698	0.646	0.650	0.642	0.718	0.651	0.618	0.554	0.642	0.566	0.575	0.541	0.64
BNB	0.697	0.652	0.635	0.681	0.667	0.644	0.664	0.643	0.614	0.548	0.643	0.560	0.598	0.569	0.58
TRX	0.615	0.722	0.633	0.628	0.614	0.599	0.646	0.617	0.589	0.583	0.567	0.624	0.693	0.505	0.57

# Linear and nonlinear correlations
[number = linear (Pearson) correlation]



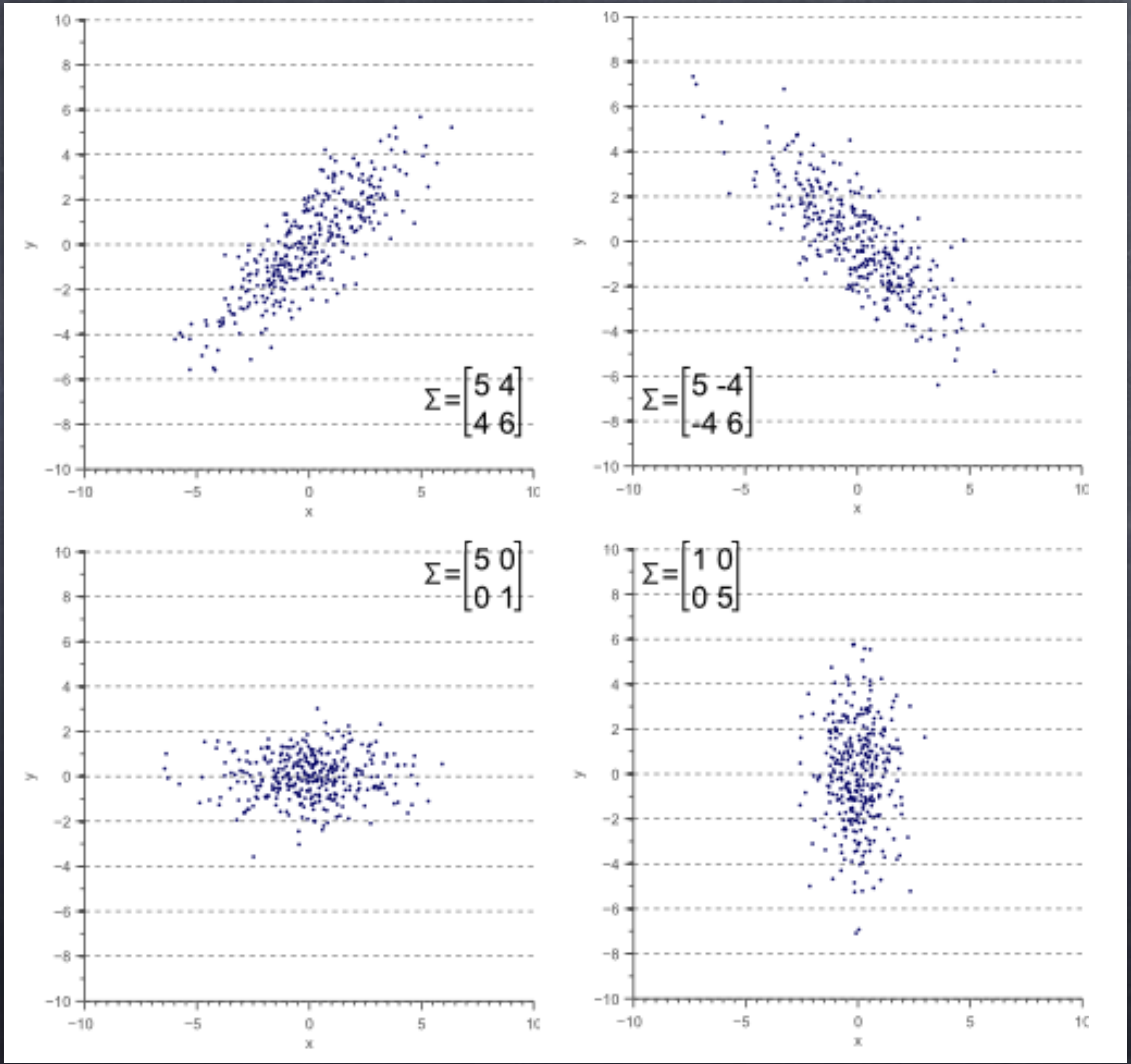
Recall

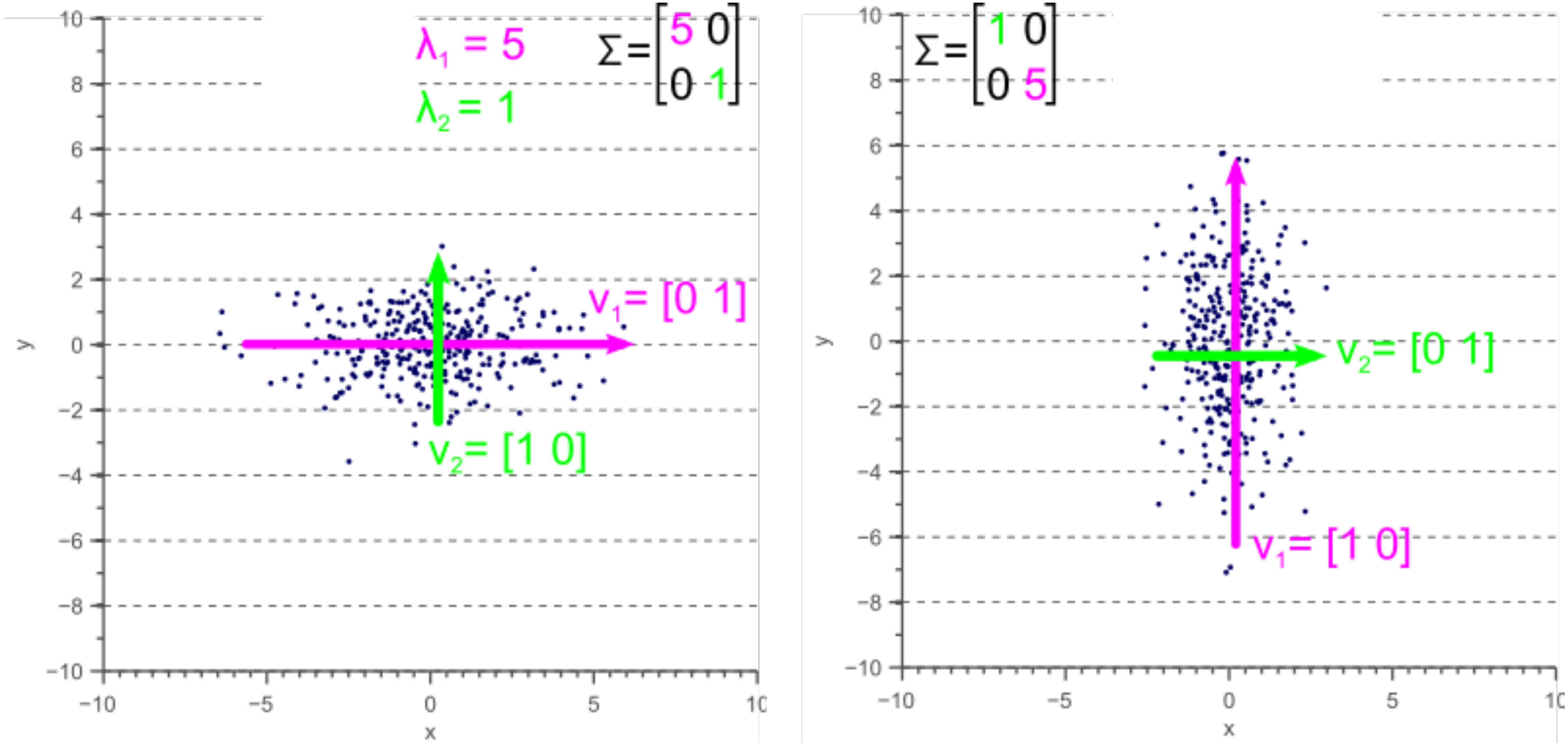
Covariance matrix

$$\Sigma = \left[\begin{array}{cc} \sigma(x,x) & \sigma(x,y) \\ \sigma(y,x) & \sigma(y,y) \end{array} \right]$$

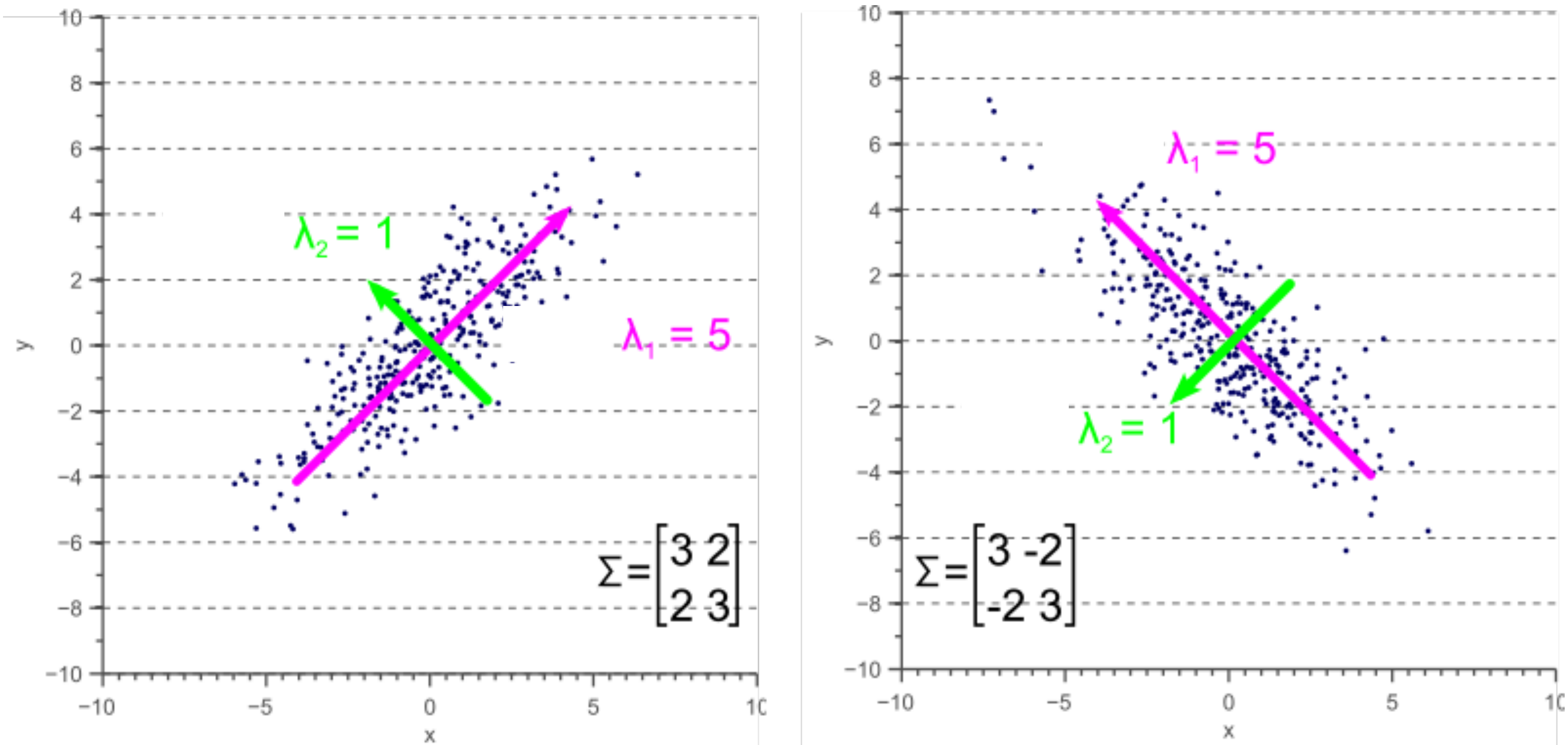
Examples: Linear correlation <-> Nonlinear correlations <-> Causation

Example: Understanding covariance

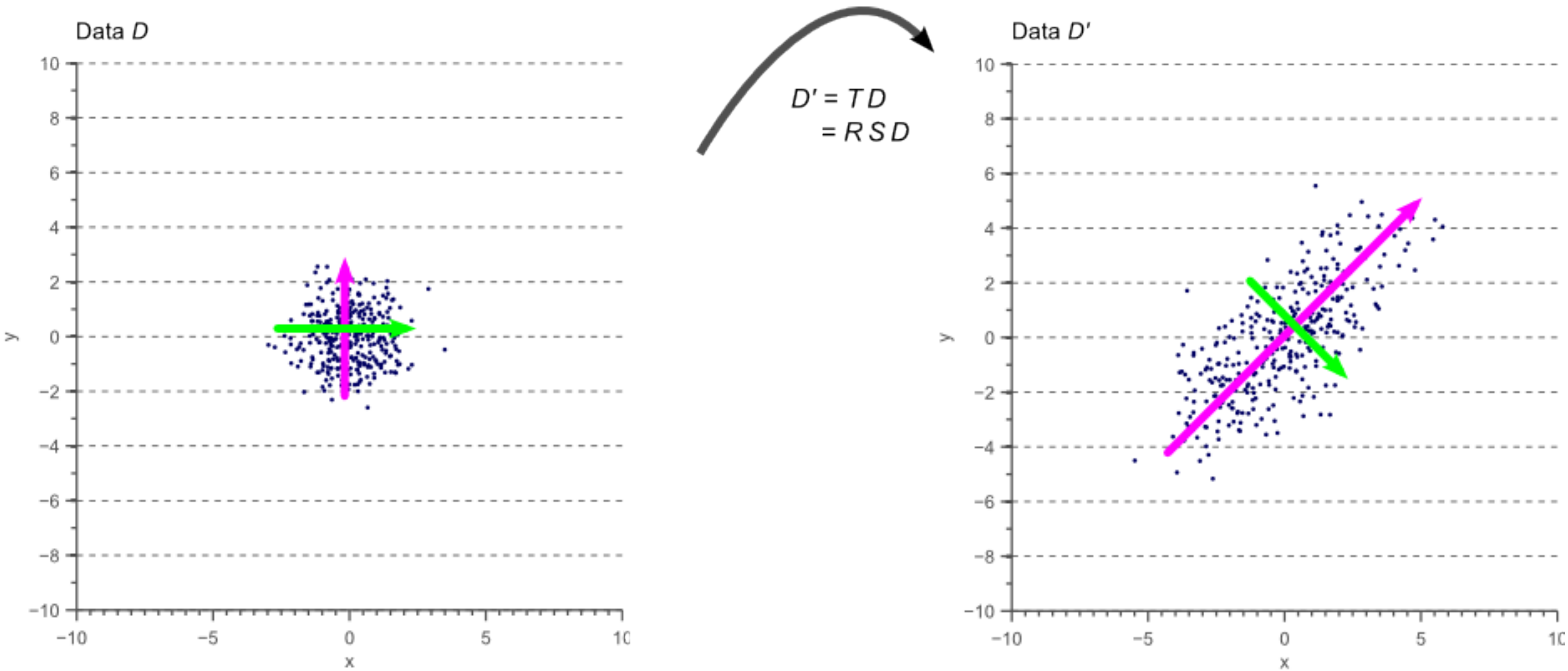




λ1 = 5
λ2 = 1
Σ = [5 0; 0 1]
v1= [0 1]
v2= [1 0]
Σ = [1 0; 0 5]
v2= [0 1]
v1= [1 0]
x
y


λ2 = 1
λ1 = 5
Σ=[3 2; 2 3]
λ1 = 5
λ2 = 1
Σ=[3 -2; -2 3]
x
y

Understanding covariance



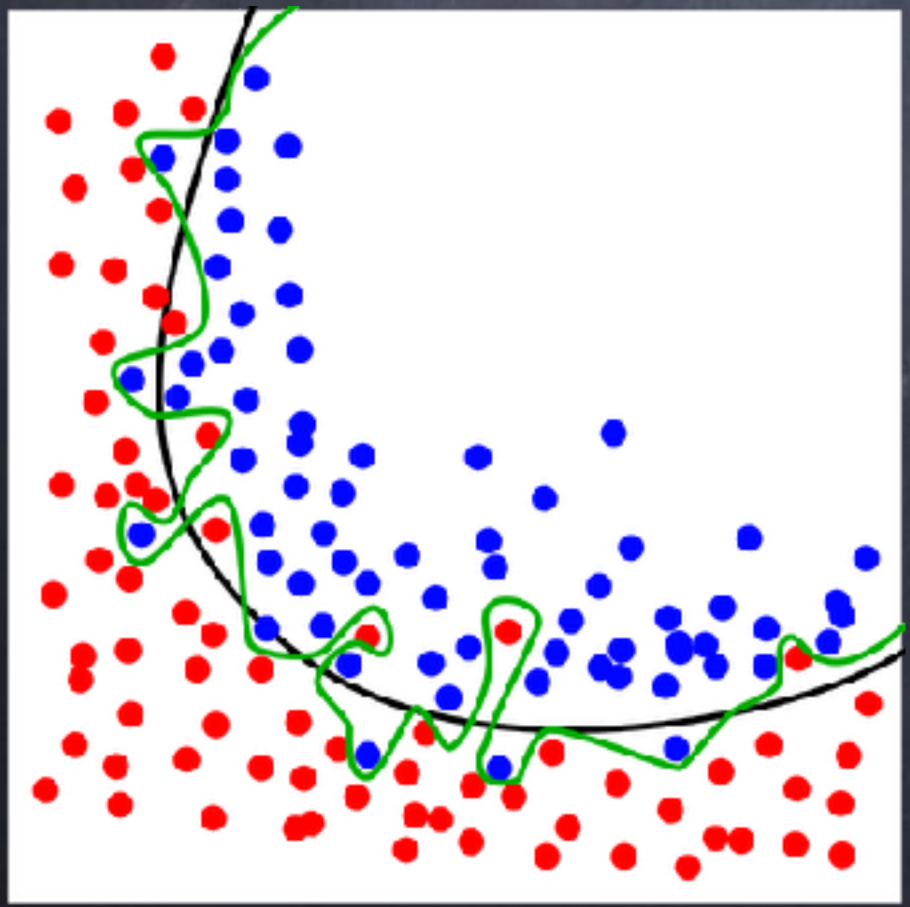
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\Sigma' = \begin{bmatrix} 4.25 & 3.10 \\ 3.10 & 4.29 \end{bmatrix} = R\,S\,S\,R^{-1}$$

PCA

Dimensionality reduction
(Curse of dimensionality)

Avoidance of overfitting
Avoidance of variance
Feature extraction

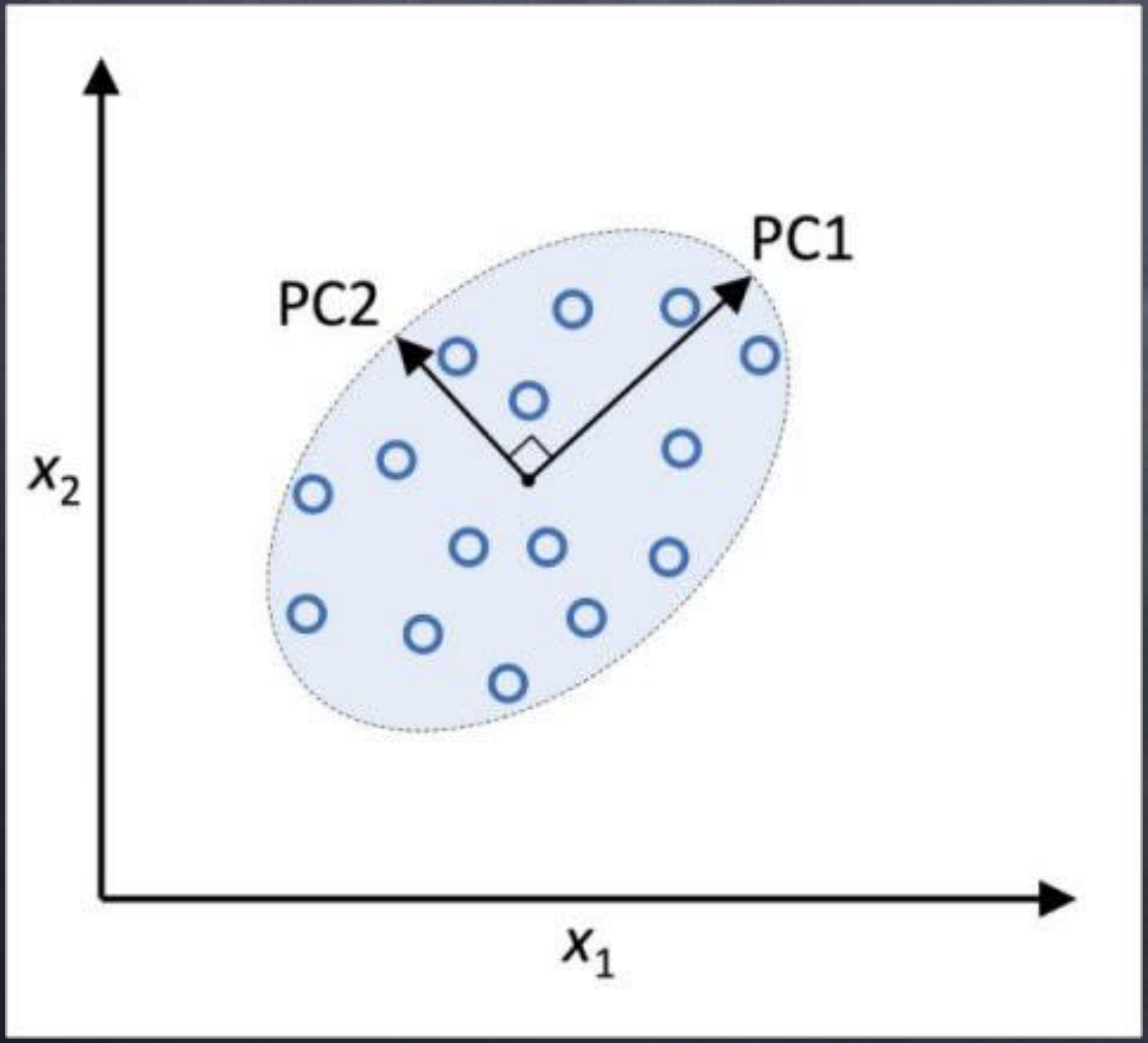


Methods:
PCA
(Principal Component Analysis)

Classifier with linear decision boundary, LDA
(Linear Discriminant Analysis)

P(rincipal) C(omponent) A(nalysis)

Main idea: Obtain principal components
from variance maximization,
given orthogonal feature axes



Main methodology:
Linear algebra

P(rincipal) C(omponent) A(nalysis)

Find linear transformation $W(d \times k)$

$$\mathbf{x} = (x_1, \ldots, x_d) \Rightarrow \mathbf{z} = (z_1, \ldots, z_k)$$

Objective

First principal component selected from largest possible variance, and all consequent principal components will have the largest variance given the constraint that these components are uncorrelated (orthogonal) to the other principal components

Even if the input features are correlated, resulting principal components will be orthogonal (uncorrelated).

P(rincipal) C(omponent) A(nalysis)

Standardize dataset of dim(d)

Compute covariance matrix

Decompose covariance matrix
using its eigenvectors and eigenvalues

Sort the eigenvalues by decreasing order
to rank the corresponding eigenvectors

Select k eigenvectors which correspond to the k largest eigenvalues,
where k is the dimensionality of the new feature subspace ($k \leq d$)

Construct a projection matrix W from the selected k eigenvectors.

Transform the d-dimensional input dataset X
using the projection matrix W
to obtain the new k-dimensional feature subspace

P(rincipal) C(omponent) A(nalysis)

Test: What is wrong?

$$\sigma_{jk} = \frac{1}{n} \sum_i (x_j^{(i)} - \mu_j) \cdot (y_k^{(i)} - \mu_k)$$

The eigenvalue problem

$$\mathbf{\Sigma v} = \lambda \mathbf{v}$$

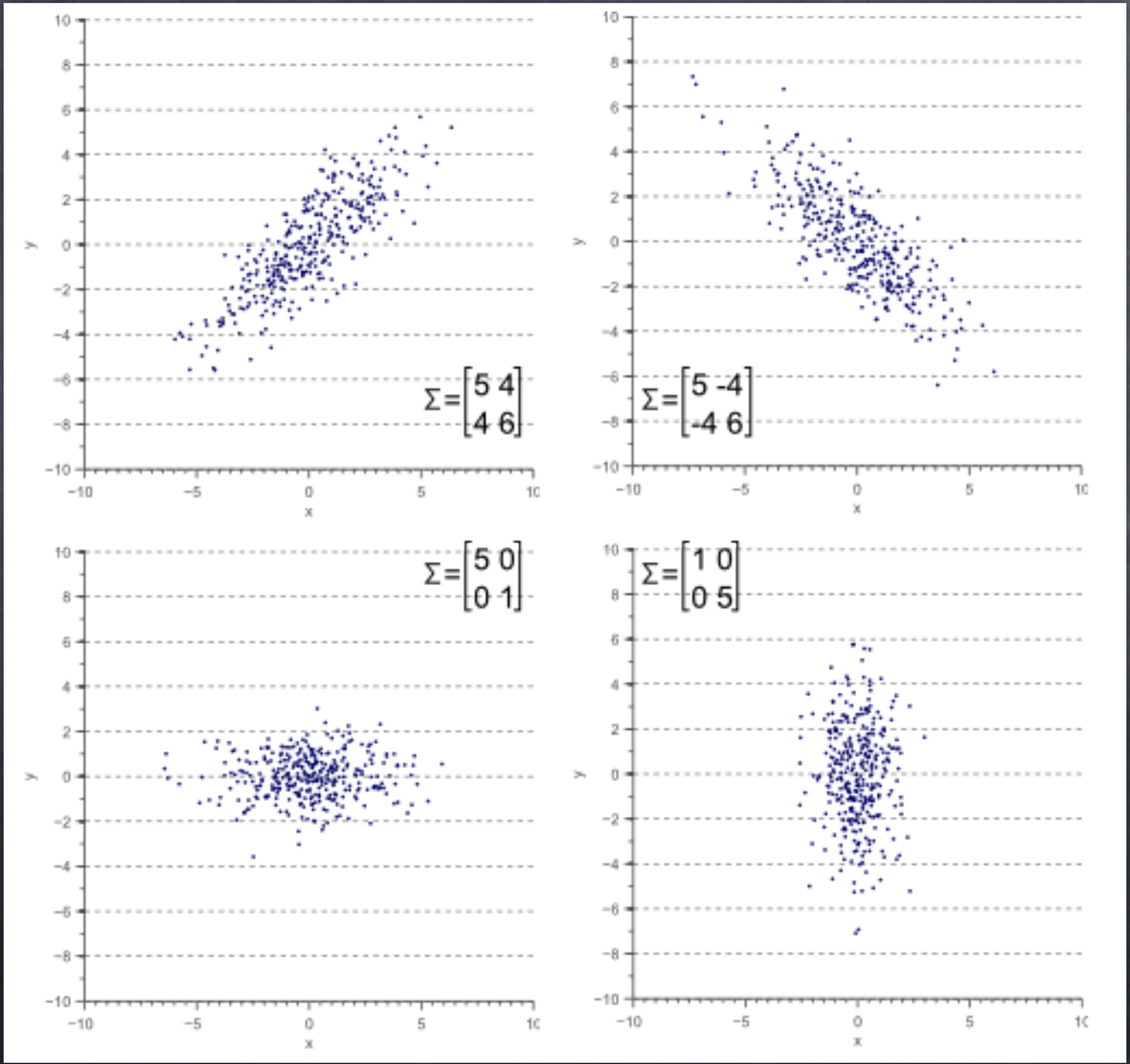
The eigenvectors of the covariance matrix represent
the directions of maximum variance,
hence they are the principal components

2-dim case:

$$\Sigma = \begin{bmatrix} \sigma(x,x) & \sigma(x,y) \\ \sigma(y,x) & \sigma(y,y) \end{bmatrix}$$

P(rincipal) C(omponent) A(nalysis)

Understanding covariance (and just be right but not wrong)

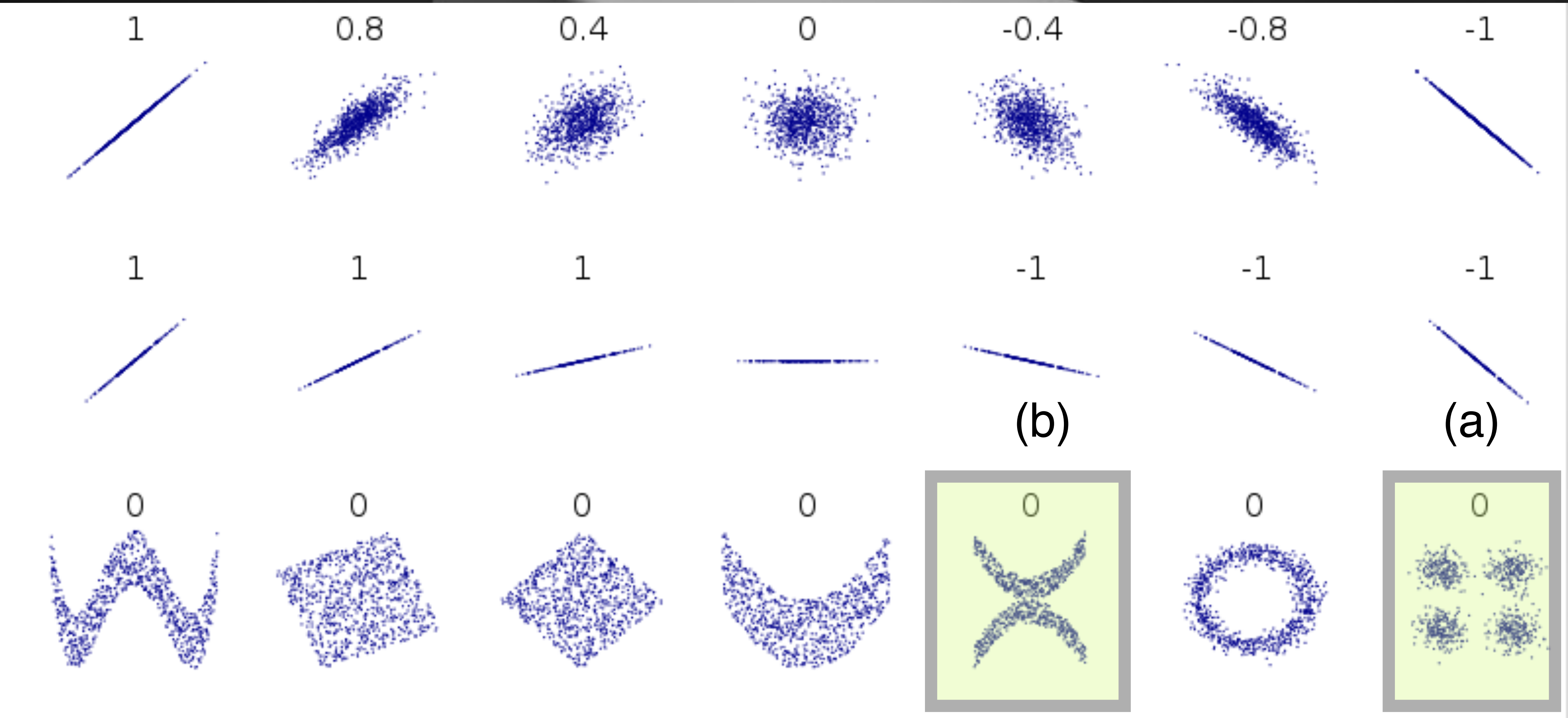


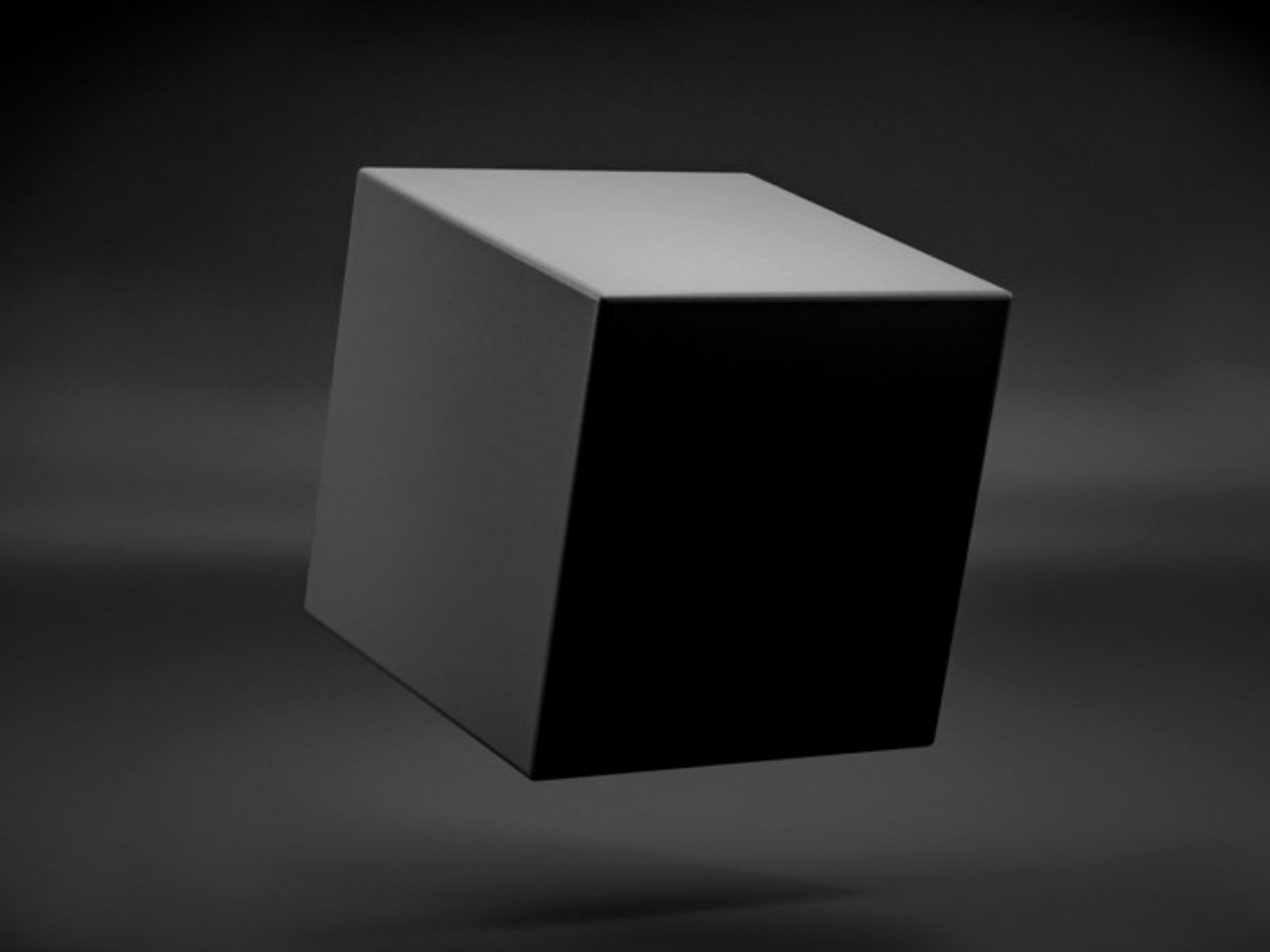
P(rincipal) C(omponent) A(nalysis)

Variance-explained-ratios

$$\frac{|\lambda_j|}{\sum_{i \leq d} |\lambda_i|}$$

Create surrogate data in more than 2 dimension!
0th: labeled data, 1st: PCA, 2nd: classification, for (a) & (b)





K-Means Clustering

centroid (*average*)

Number of clusters, *k*, a priori

K-Means Clustering: Preprocessing

Standardization =
Z-score normalization =
(zero mean, unit variance)

or

Min-Max Scaling

K-Means Clustering

Number of clusters, *k*, a priori
Somehow (e.g., randomly) pick *k* centroids from the sample points as initial cluster centers

Repeat
(1) Assign each point to the nearest centroid

(2) Move the centroids to the center of the samples that were assigned to it

Until
cluster assignments do not change or
within tolerance or
maximum number of iterations

K-Means Clustering

max_iter:
Maximum number of iterations

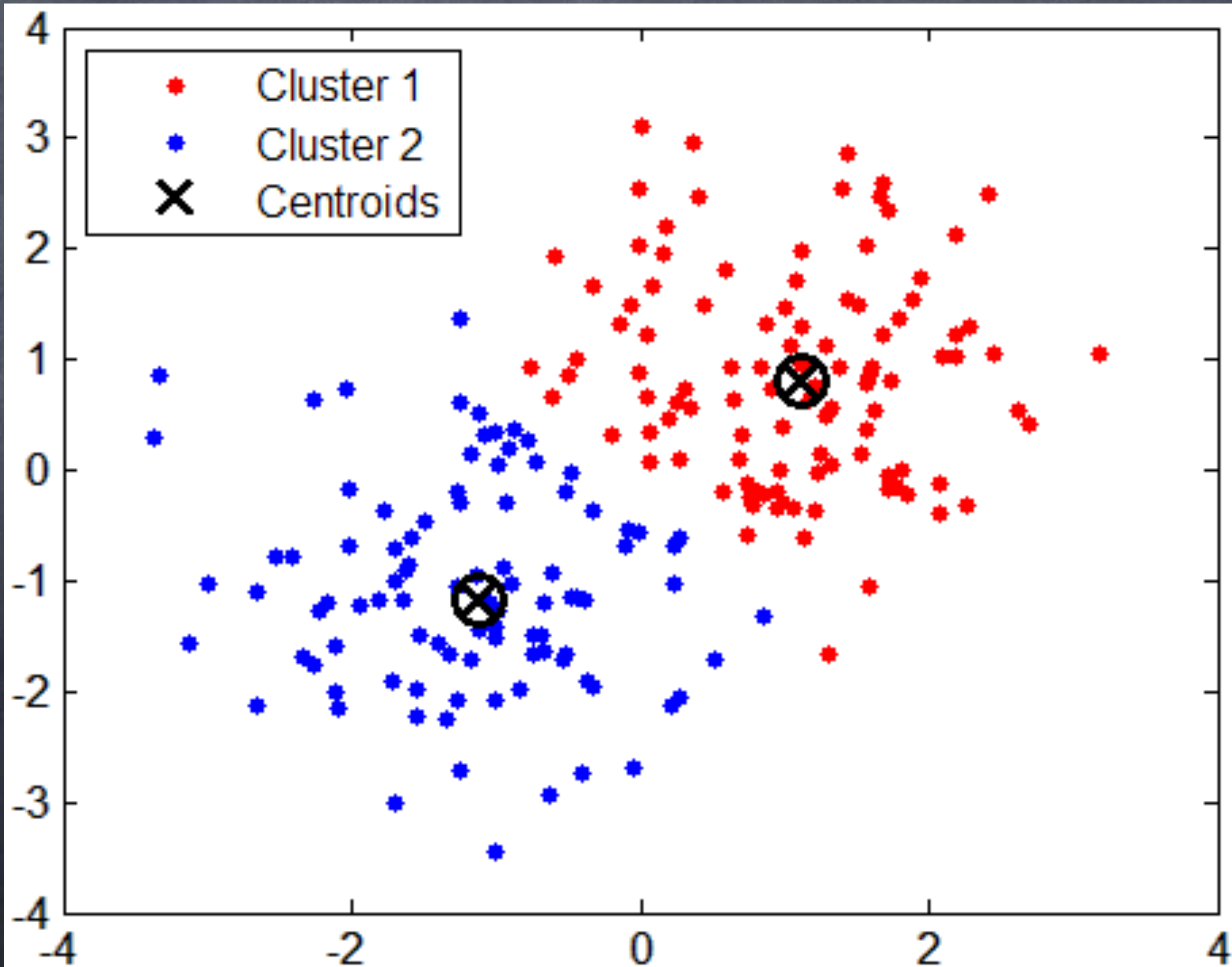
tol(erance):
Relative tolerance with regards to inertia/SSE
to declare convergence,
SSE(iter+1)/SEE(iter) < tol

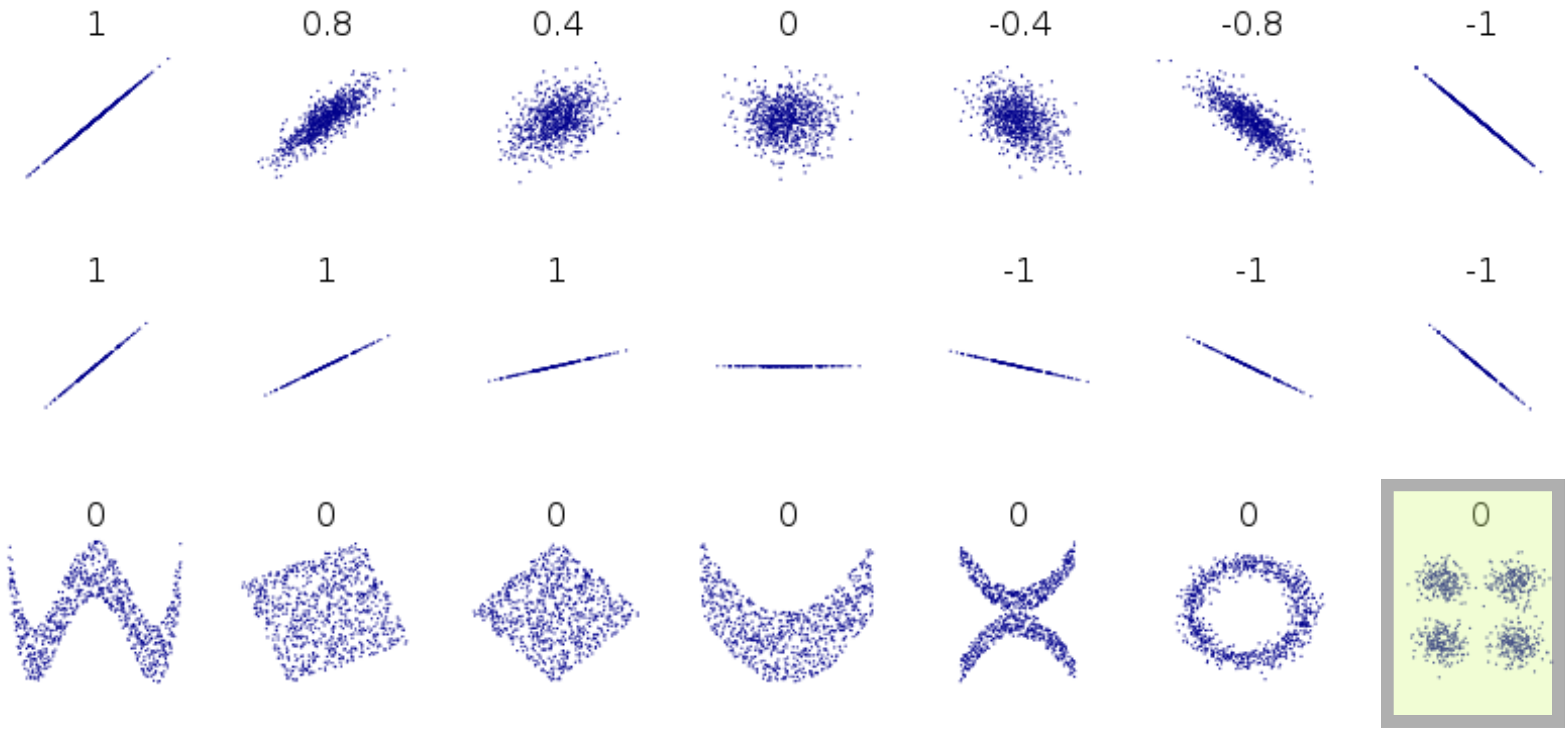
Sum of squared errors (SSE)

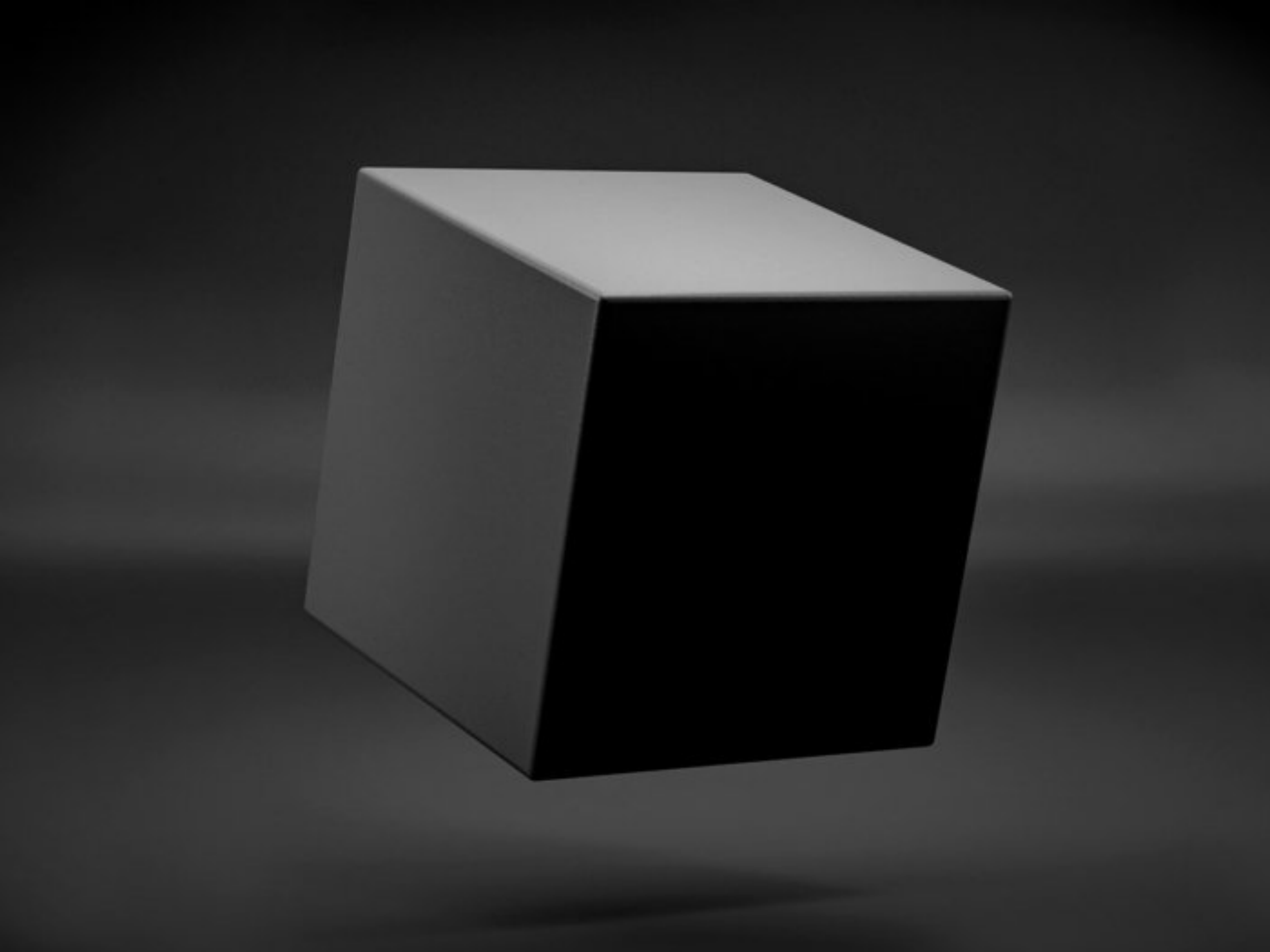
$$\mathrm{SSE} = \sum_{i}^{n} \sum_{j}^{k} w^{(i,j)} ||\mathbf{x}^{(i)} - \mathbf{y}^{(j)}||_2^2$$

$w^{(i,j)}$ is 1 if x(i) belongs to cluster j [sitting at y(j)], otherwise 0

$$\mathrm{SSE} = \sum_{i}^{n} \sum_{j}^{k} w^{(i,j)} \|\mathbf{x}^{(i)} - \mathbf{y}^{(j)}\|_2^2$$




1
0.8
0.4
0
-0.4
-0.8
-1
1
1
1
-1
-1
-1
0
0
0
0
0
0
0

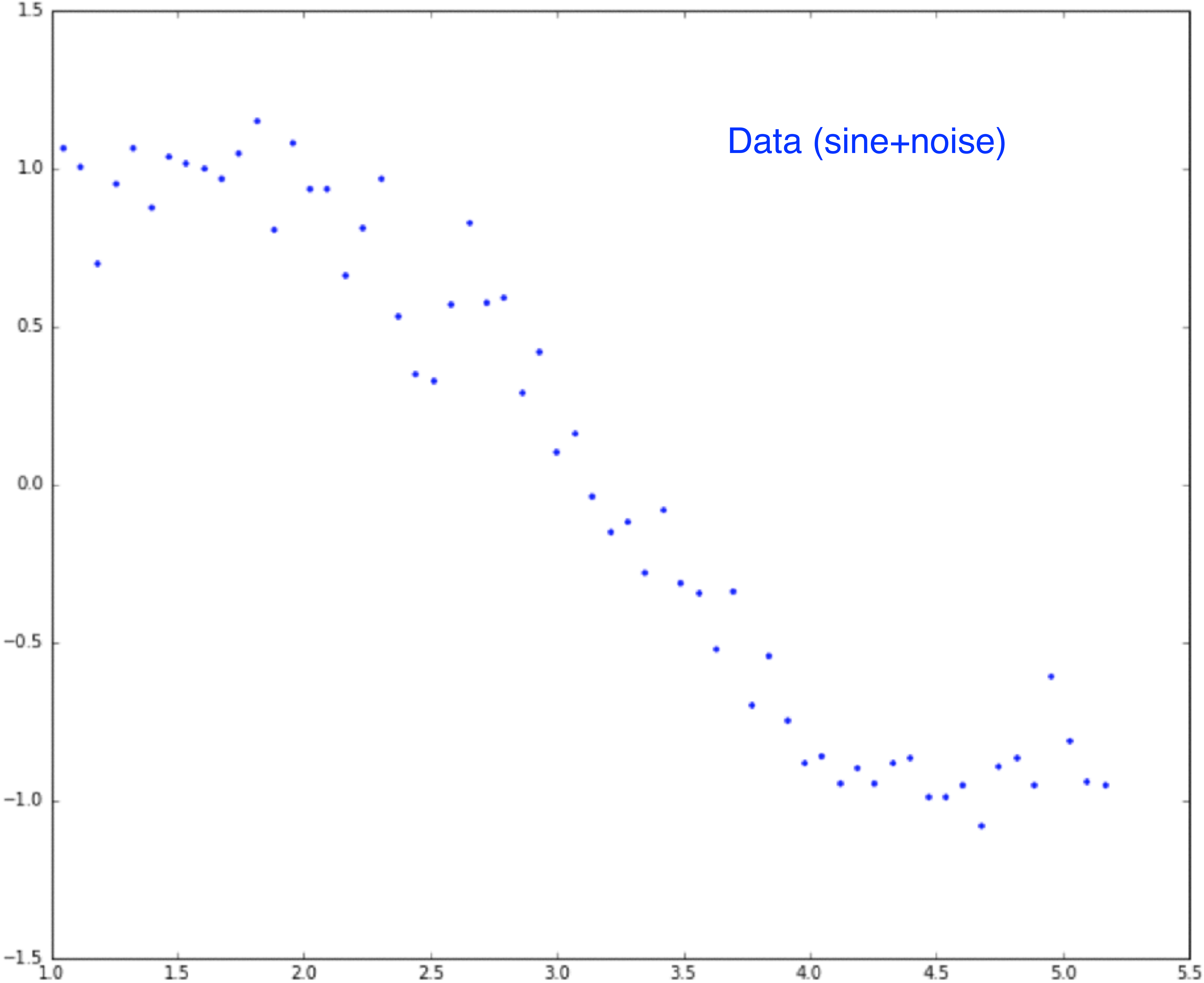


Regularization

Ridge regression

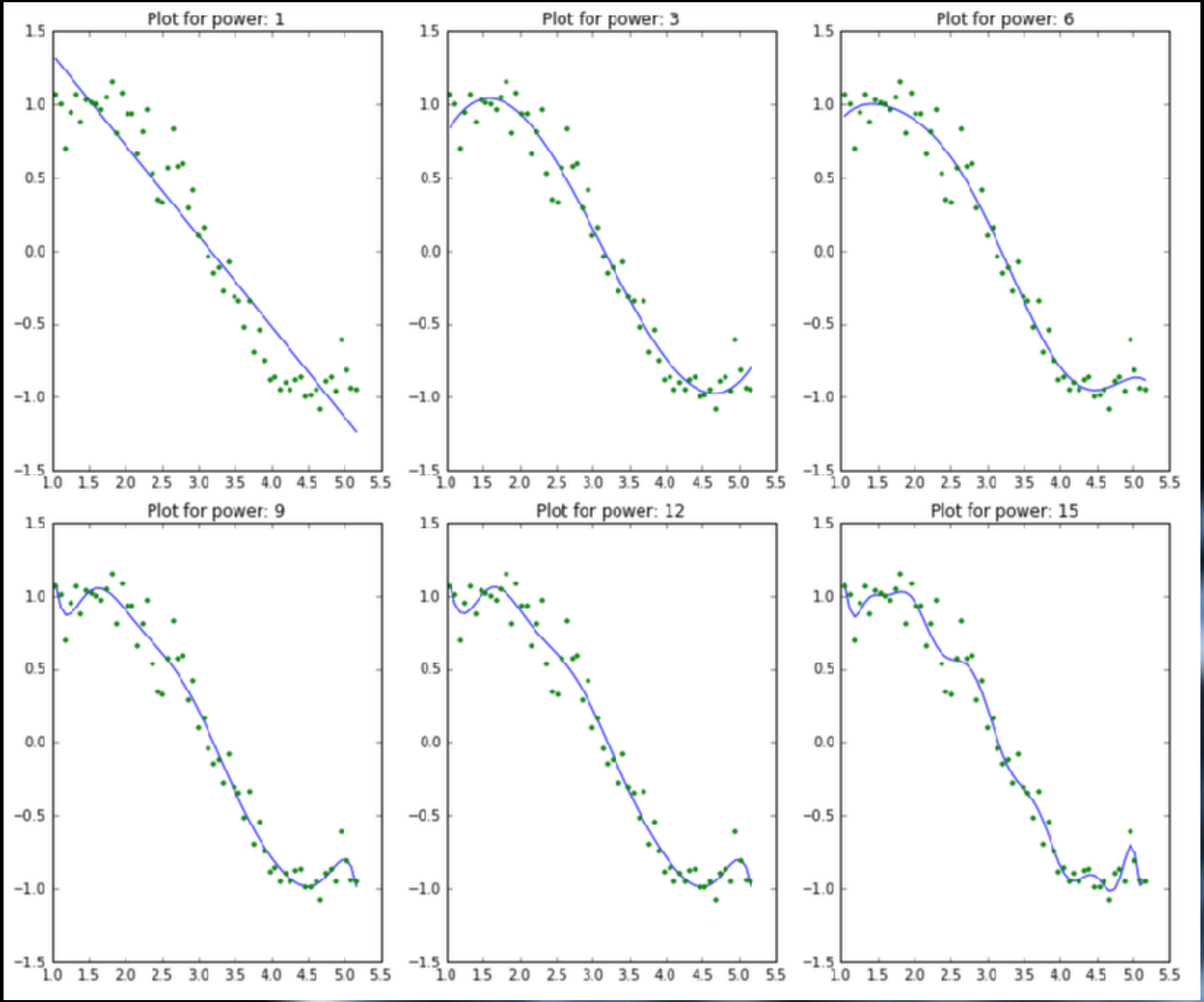
LASSO regression

Elastic Net regression



Courtesy: K. Jain

Fit: OLS Minimization for polynomials of order 1,3,6,9,12,15



Courtesy: K. Jain

Fit coefficients & Residuals
(OLS Minimization for polynomials of order 1-15)

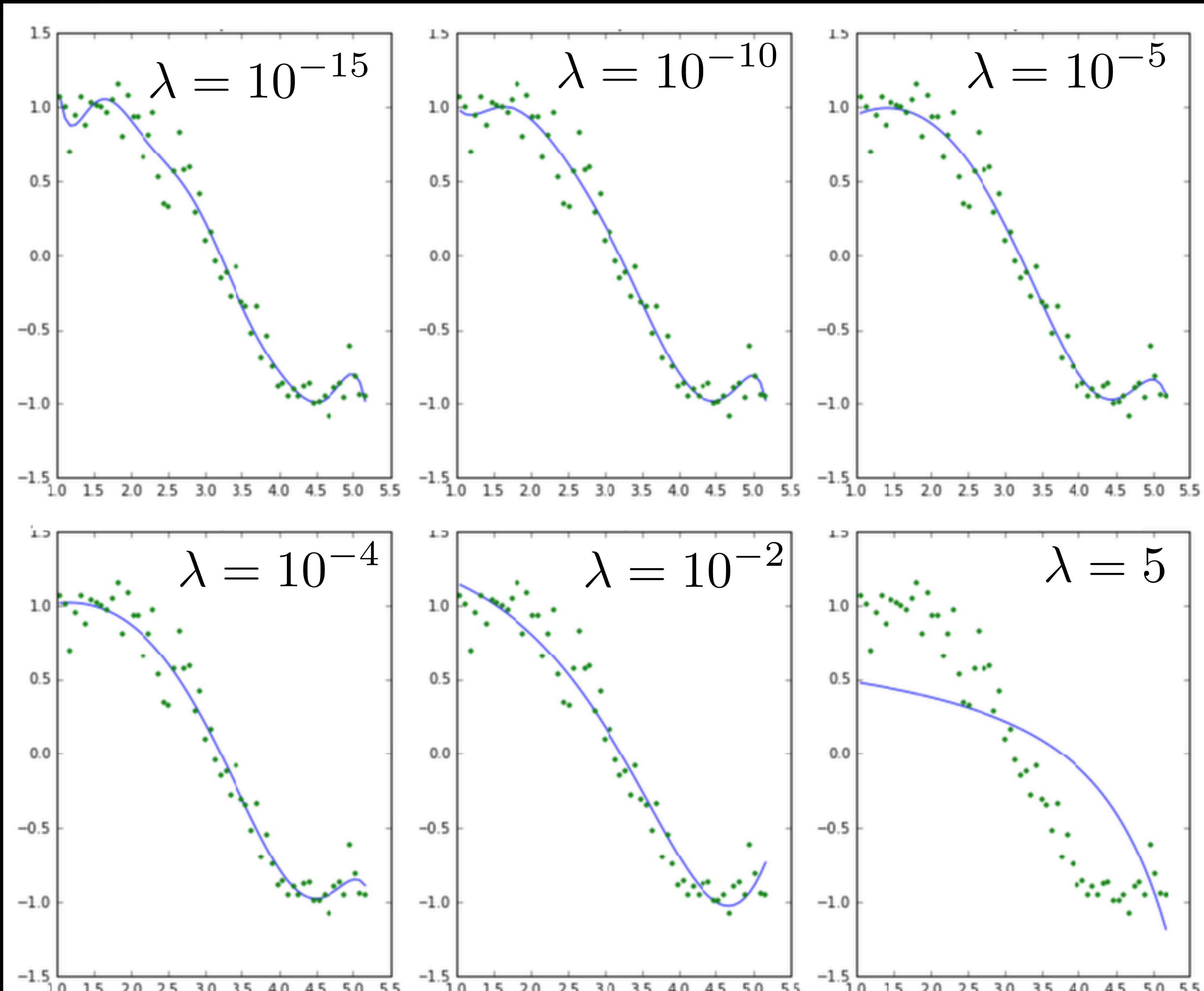
	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11
model_pow_1	3.3	2	-0.62	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_2	3.3	1.9	-0.58	-0.006	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_3	1.1	-1.1	3	-1.3	0.14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_4	1.1	-0.27	1.7	-0.53	-0.036	0.014	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_5	1	3	-5.1	4.7	-1.9	0.33	-0.021	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_6	0.99	-2.8	9.5	-9.7	5.2	-1.6	0.23	-0.014	NaN	NaN	NaN	NaN	NaN
model_pow_7	0.93	19	-56	69	-45	17	-3.5	0.4	-0.019	NaN	NaN	NaN	NaN
model_pow_8	0.92	43	-1.4e+02	1.8e+02	-1.3e+02	58	-15	2.4	-0.21	0.0077	NaN	NaN	NaN
model_pow_9	0.87	1.7e+02	-6.1e+02	9.6e+02	-8.5e+02	4.6e+02	-1.6e+02	37	-5.2	0.42	-0.015	NaN	NaN
model_pow_10	0.87	1.4e+02	-4.9e+02	7.3e+02	-6e+02	2.9e+02	-87	15	-0.81	-0.14	0.026	-0.0013	NaN
model_pow_11	0.87	-75	5.1e+02	-1.3e+03	1.9e+03	-1.6e+03	9.1e+02	-3.5e+02	91	-16	1.8	-0.12	0.0034
model_pow_12	0.87	-3.4e+02	1.9e+03	-4.4e+03	6e+03	-5.2e+03	3.1e+03	-1.3e+03	3.8e+02	-80	12	-1.1	0.062
model_pow_13	0.86	3.2e+03	-1.8e+04	4.5e+04	-6.7e+04	6.6e+04	-4.6e+04	2.3e+04	-8.5e+03	2.3e+03	-4.5e+02	62	-5.7
model_pow_14	0.79	2.4e+04	-1.4e+05	3.8e+05	-6.1e+05	6.6e+05	-5e+05	2.8e+05	-1.2e+05	3.7e+04	-8.5e+03	1.5e+03	-1.8e+02
model_pow_15	0.7	-3.6e+04	2.4e+05	-7.5e+05	1.4e+06	-1.7e+06	1.5e+06	-1e+06	5e+05	-1.9e+05	5.4e+04	-1.2e+04	1.9e+03

Coefficients increase exponentially with model complexity!

Regularization needed!

Courtesy: K. Jain

Ridge regression (regularization)



Courtesy: K. Jain

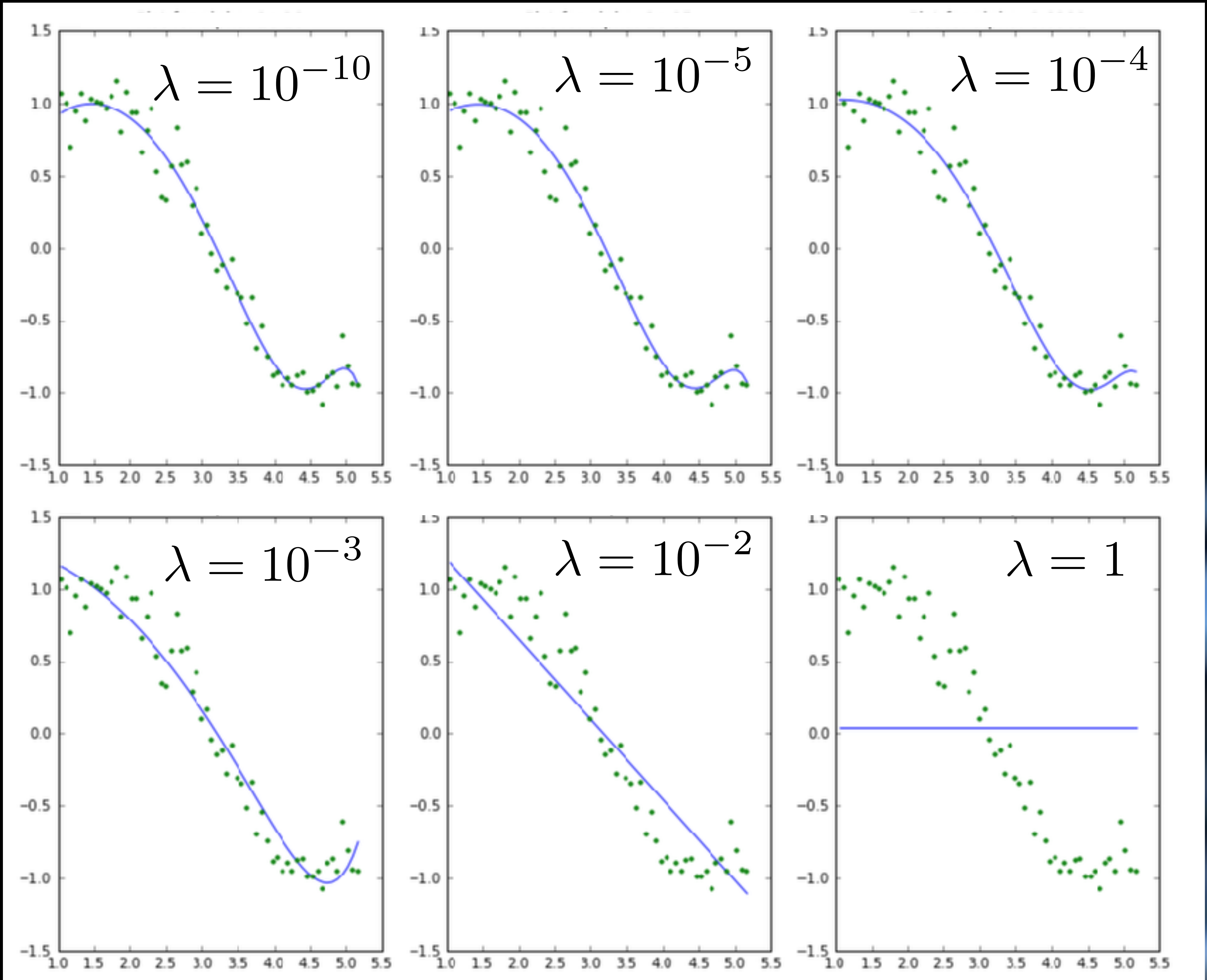
Fit coefficients & Residuals
(OLS Minimization with L2 penalty = Ridge regularization)

λ

	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11	ce
alpha_1e-15	0.87	95	-3e+02	3.8e+02	-2.4e+02	66	0.96	-4.8	0.64	0.15	-0.026	-0.0054	0.00086	0.
alpha_1e-10	0.92	11	-29	31	-15	2.9	0.17	-0.091	-0.011	0.002	0.00064	2.4e-05	-2e-05	-4
alpha_1e-08	0.95	1.3	-1.5	1.7	-0.68	0.039	0.016	0.00016	-0.00036	-5.4e-05	-2.9e-07	1.1e-06	1.9e-07	2e
alpha_0.0001	0.96	0.56	0.55	-0.13	-0.026	-0.0028	-0.00011	4.1e-05	1.5e-05	3.7e-06	7.4e-07	1.3e-07	1.9e-08	1.
alpha_0.001	1	0.82	0.31	-0.087	-0.02	-0.0028	-0.00022	1.8e-05	1.2e-05	3.4e-06	7.3e-07	1.3e-07	1.9e-08	1.
alpha_0.01	1.4	1.3	-0.088	-0.052	-0.01	-0.0014	-0.00013	7.2e-07	4.1e-06	1.3e-06	3e-07	5.6e-08	9e-09	1.
alpha_1	5.6	0.97	-0.14	-0.019	-0.003	-0.00047	-7e-05	-9.9e-06	-1.3e-06	-1.4e-07	-9.3e-09	1.3e-09	7.8e-10	2.
alpha_5	14	0.55	-0.059	-0.0085	-0.0014	-0.00024	-4.1e-05	-6.9e-06	-1.1e-06	-1.9e-07	-3.1e-08	-5.1e-09	-8.2e-10	-1
alpha_10	18	0.4	-0.037	-0.0055	-0.00095	-0.00017	-3e-05	-5.2e-06	-9.2e-07	-1.6e-07	-2.9e-08	-5.1e-09	-9.1e-10	-1
alpha_20	23	0.28	-0.022	-0.0034	-0.0006	-0.00011	-2e-05	-3.6e-06	-6.6e-07	-1.2e-07	-2.2e-08	-4e-09	-7.5e-10	-1

Courtesy: K. Jain

LASSO regression (regularization)



Courtesy: K. Jain

Fit coefficients & Residuals

(OLS Minimization with L1 penalty = LASSO regularization)

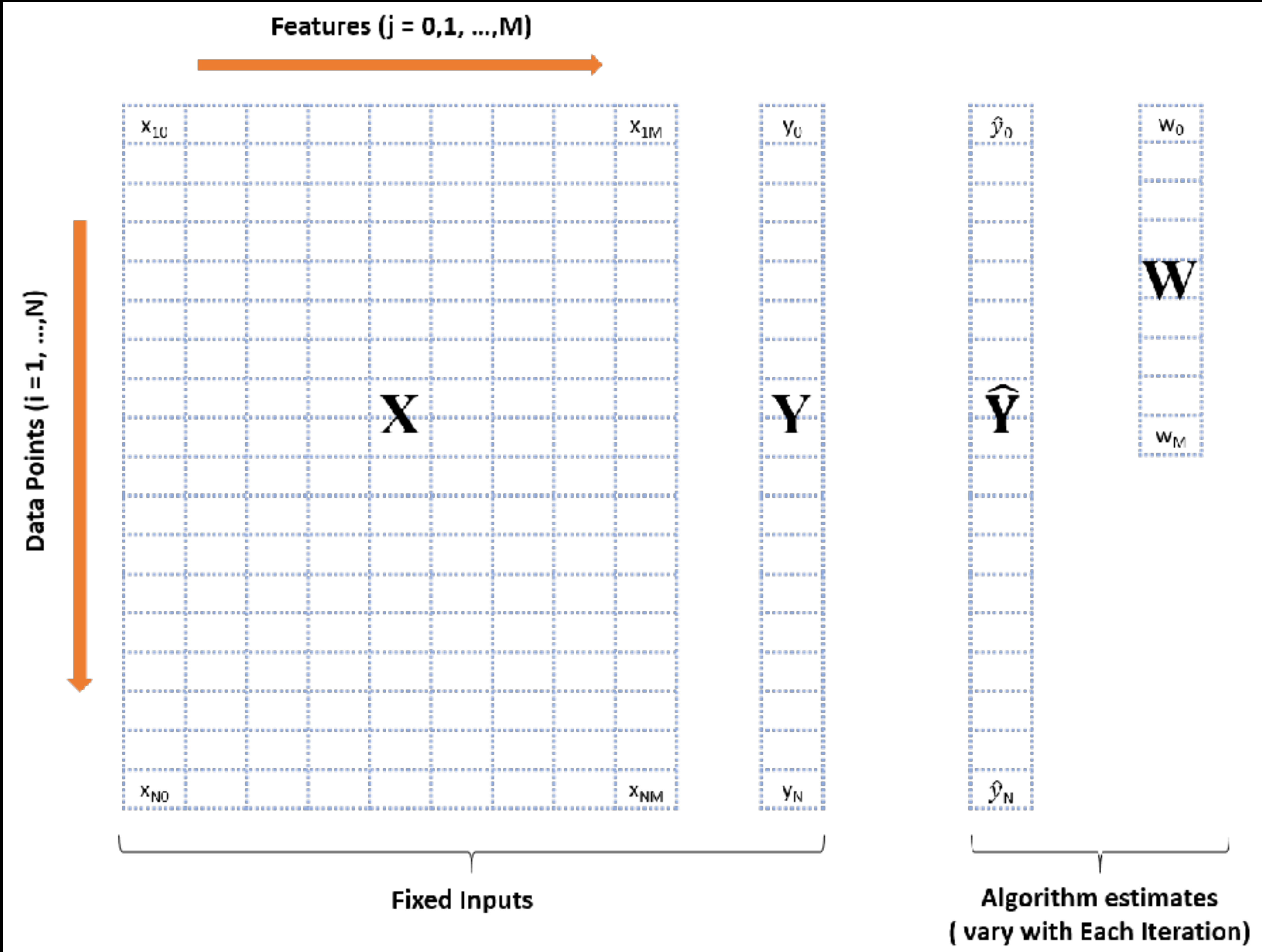
λ

	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11	coe
alpha_1e-15	0.96	0.22	1.1	-0.37	0.00089	0.0016	-0.00012	-6.4e-05	-6.3e-06	1.4e-06	7.8e-07	2.1e-07	4e-08	5.4
alpha_1e-10	0.96	0.22	1.1	-0.37	0.00088	0.0016	-0.00012	-6.4e-05	-6.3e-06	1.4e-06	7.8e-07	2.1e-07	4e-08	5.4
alpha_1e-08	0.96	0.22	1.1	-0.37	0.00077	0.0016	-0.00011	-6.4e-05	-6.3e-06	1.4e-06	7.8e-07	2.1e-07	4e-08	5.3
alpha_1e-05	0.96	0.5	0.6	-0.13	-0.038	-0	0	0	0	7.7e-06	1e-06	7.7e-08	0	0
alpha_0.0001	1	0.9	0.17	-0	-0.048	-0	-0	0	0	9.5e-06	5.1e-07	0	0	0
alpha_0.001	1.7	1.3	-0	-0.13	-0	-0	-0	0	0	0	0	0	1.5e-08	7.5
alpha_0.01	3.6	1.8	-0.55	-0.00056	-0	-0	-0	-0	-0	-0	-0	0	0	0
alpha_1	37	0.038	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
alpha_5	37	0.038	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
alpha_10	37	0.038	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0

HIGH SPARSITY

Courtesy: K. Jain

Regression: (more) general case



Courtesy: K. Jain