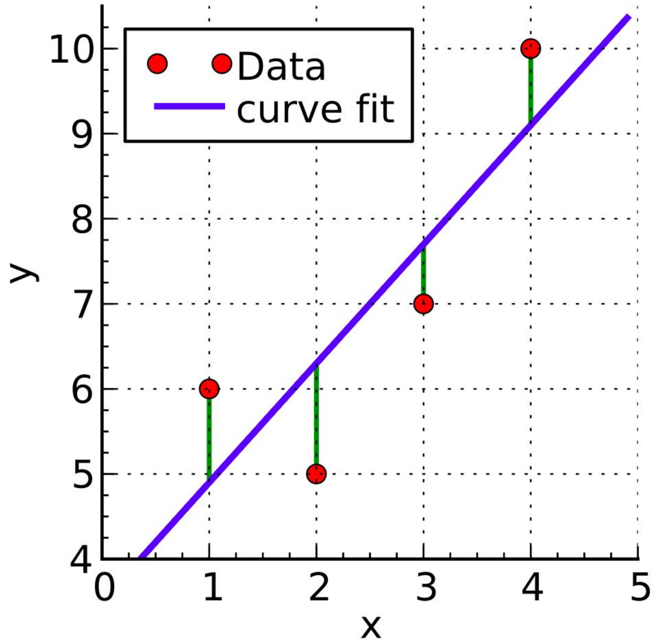
Regression methods
“*Data Science for Geosciences*”
Toulouse, January 28, 2020

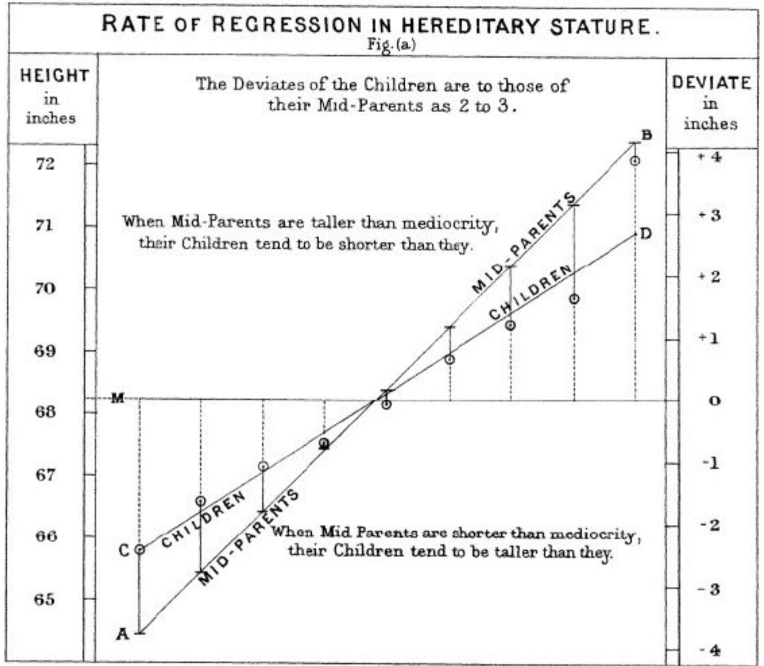
Pierre Tandeo
IMT Atlantique, Brest, France
pierre.tandeo@imt-atlantique.fr

Why is it called “regression”?

- Introduced by Legendre in 1805
- Named “**method of least squares**” by Gauss in 1809

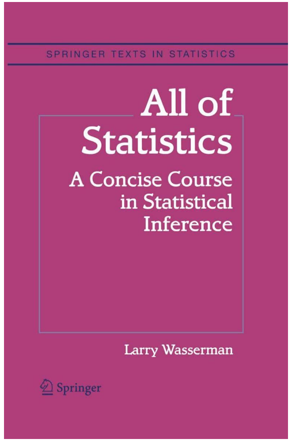


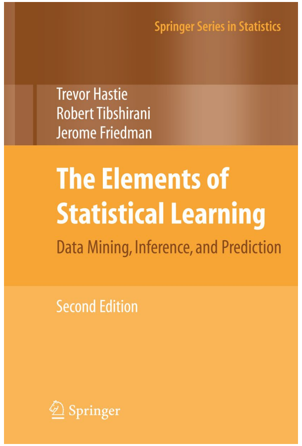
- Used by Galton in 1877 (*Nature*) as “**reversion**”
- Finally named “**regression toward the mean**” by Galton in 1885

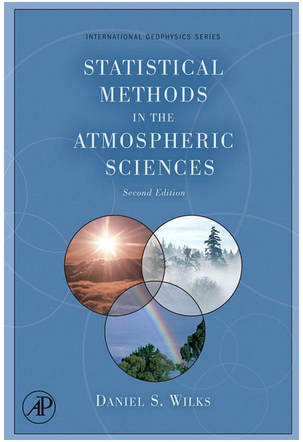


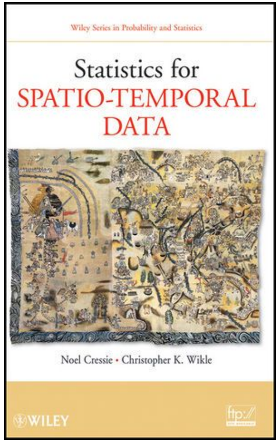
Good references

- Methodology:
  - Wasserman 2013:
    - statistics/probability point of view
    - rigorous
  - Hastie et al. 2009:
    - machine learning point of view
    - exhaustive (also clustering, classification)
- Methods/Applications:
  - Wilks 2011:
    - for climate data
    - physical point of view
  - Cressie and Wikle 2011:
    - some applications in climate
    - focus on spatio-temporal models









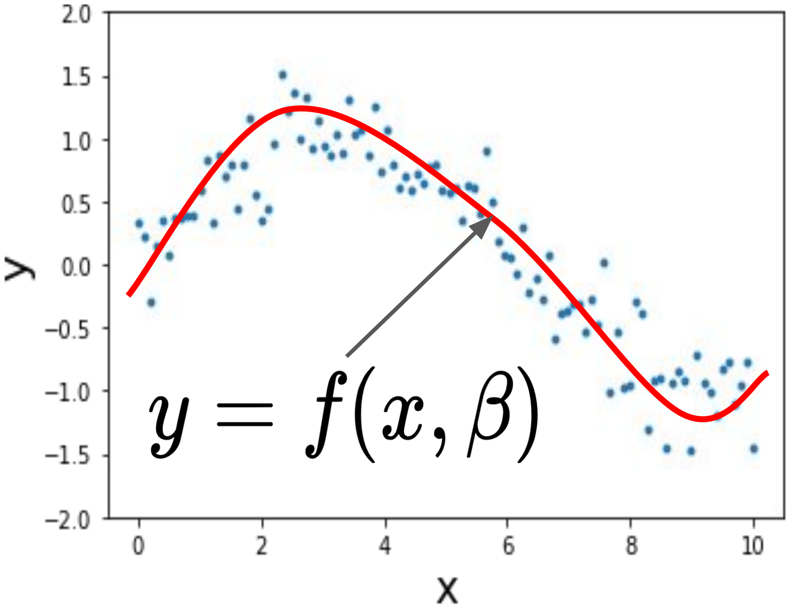
Notations

$$\mathbf{y} = f(\mathbf{X}, \beta) + \epsilon,$$
$$\text{with } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- y vector (here univariate):
  - **continuous variable**
  - “response variable” [stat]
  - “output variable” [ML]
- X matrix (here multivariate):
  - **continuous** or discrete variables
  - “covariates” or “predictors” [stat]
  - “features” or “input variables ”[ML]

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{bmatrix}$$

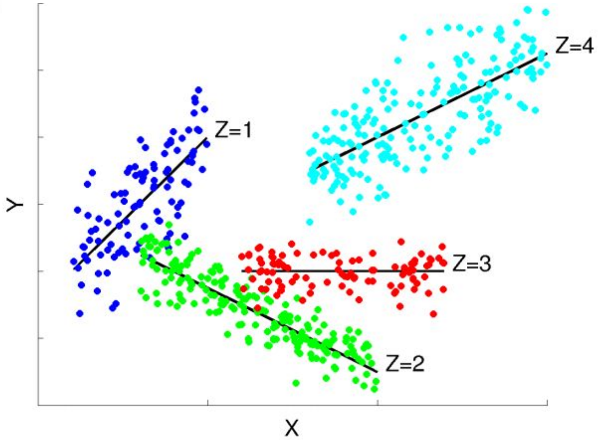
- Regression model:
  - “transfer function” f
  - beta “parameters” [stat] or “coefficients” [ML]
  - **independent additive Gaussian errors** epsilon



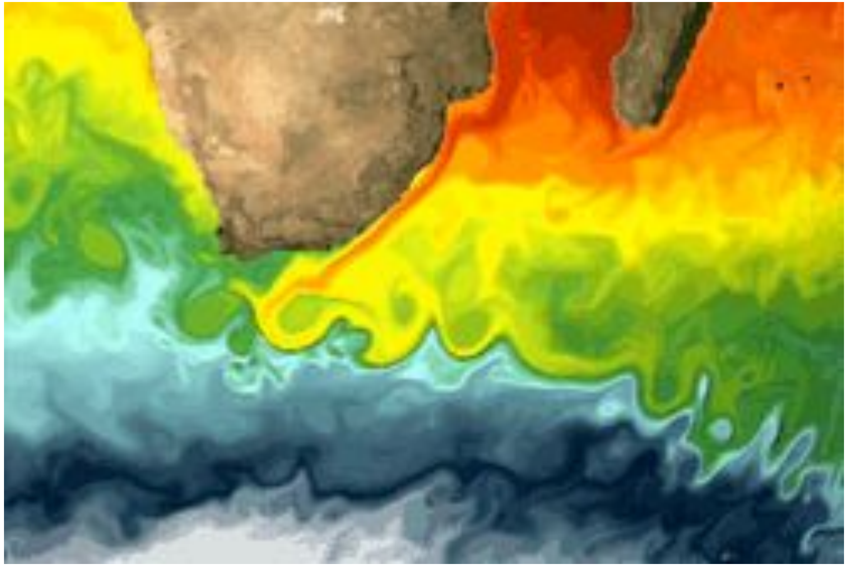
Specificity of geophysical data

- Nonlinear, chaotic, underlying hidden processes, etc...
- Organized in space and time, governed by physical/biological laws, high dimensional, etc...

Nonlinear and choatic Lorenz system



Mixture of regressions

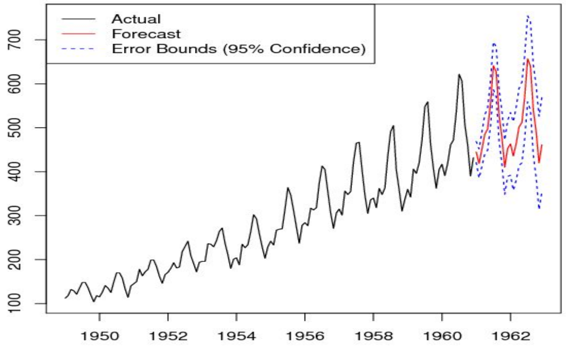


Sea surface temperature evolution

Regressions for spatio-temporal processes

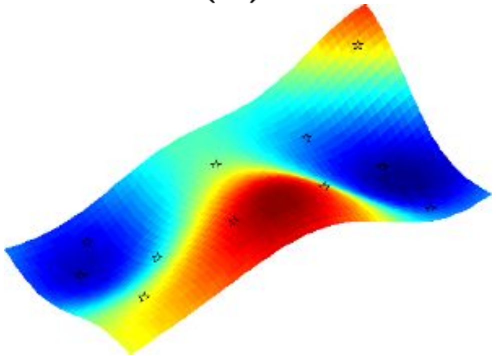
- Temporal process:
  - process evolving in time
  - the input (X) and output (y) are the same variable at different times
  - example is AR(p) process

$$y_t = \underbrace{\sum_{i=1}^{p} \theta_i y_{t-i}}_{\mathbf{X}\beta} + \epsilon$$



- Spatial process:
  - process evolving in space
  - the input (X) and output (y) are the same variables at different locations
  - example of kriging

$$y_s = \underbrace{\sum_{l \in v(s)} \theta_l y_l}_{\mathbf{X}\beta} + \epsilon$$



Linear methods (linear regression)

- Find the beta that minimizes the cost function:

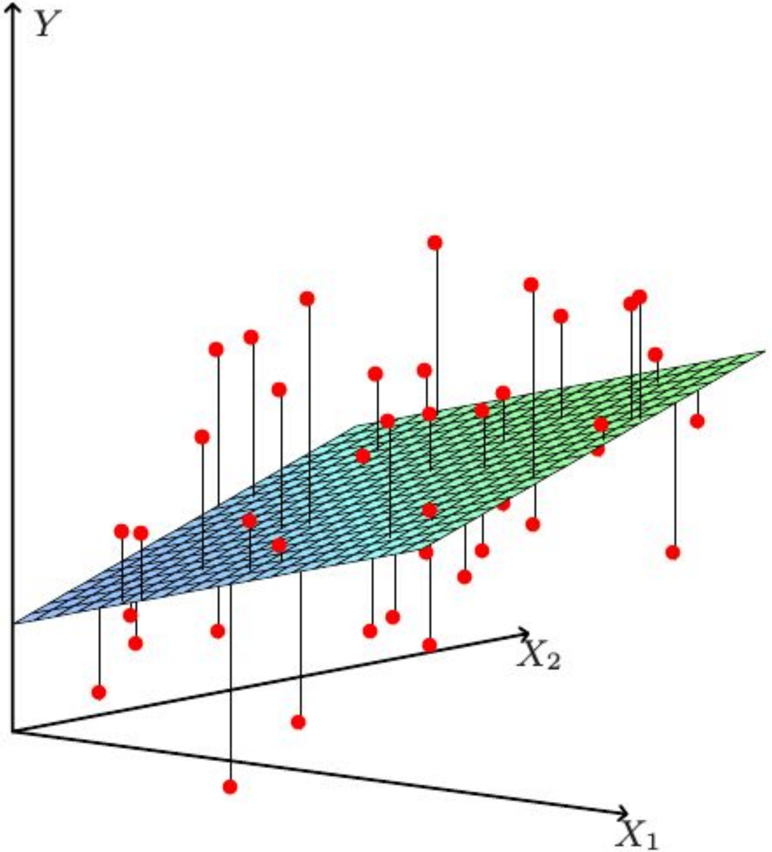
$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2$$

- Analytic solution:

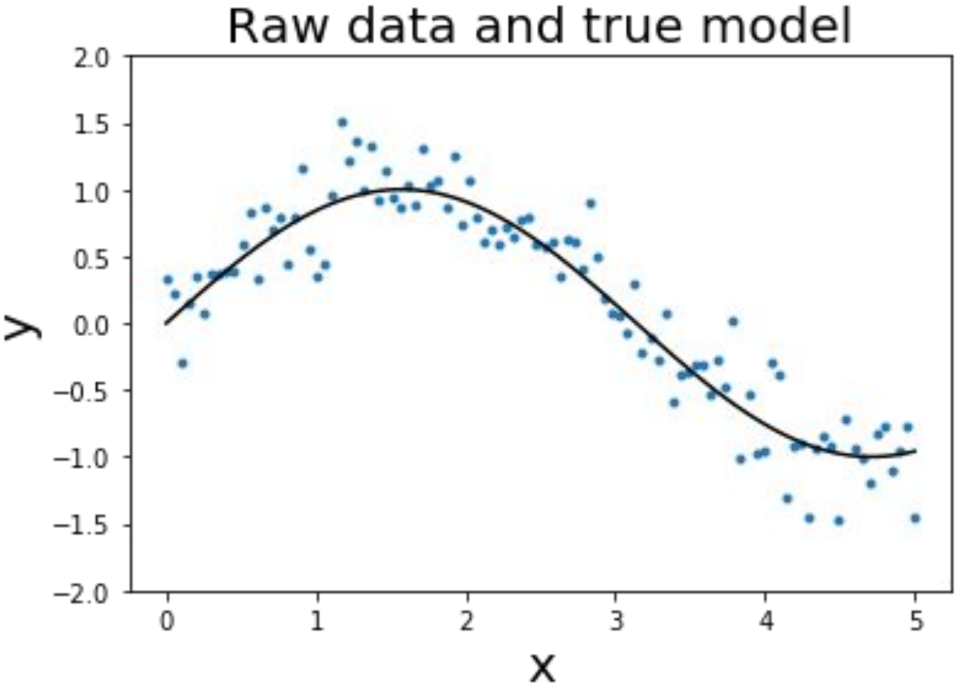
$$\widehat{\beta} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$$

- Predictions given by:

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\beta}$$



Linear methods (data and models)



- Simulated data:

$$y = sin(x) + \epsilon,$$

$$\text{with } \epsilon \sim \mathcal{N}\left(0, (1/4)^2\right)$$

- Simple linear regression:

$$y = \beta_0 + \beta_1 x$$

- Multiple linear regression (polynomial):

$$y = \beta_0 + \sum_{j=1}^{15} \beta_j x^j$$

⇨ play this Jupyter Notebook

https://github.com/DataScience4Geoscience/Toulouse2020/tree/master/Notebooks/Regression/DSG_2020_regression_course.ipynb

Linear methods (issues: least squares parameters)

	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	c
model_pow_1	3.3	2	-0.62	NaN	NaN	NaN	NaN	NaN	[illegible]	[illegible]	N	NaN	NaN	N
model_pow_2	3.3	1.9	-0.58	-0.006	NaN	NaN	NaN	NaN	[illegible]	[illegible]	N	NaN	NaN	N
model_pow_3	1.1	-1.1	3	-1.3	0.14	NaN	NaN	NaN	[illegible]	[illegible]	N	NaN	NaN	N
model_pow_4	1.1	-0.27	1.7	-0.53	-0.036	0.014	NaN	NaN	[illegible]	[illegible]	N	NaN	NaN	N
model_pow_5	1	3	-5.1	4.7	-1.9	0.33	.021	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_6	0.99	-2.8	9.5	-9.7	5.2	-1.6	0.23	-0.014	NaN	NaN	NaN	NaN	NaN	N
model_pow_7	0.93	19	-56	69	-45	17	-3.5	0.4	-0.019	NaN	NaN	NaN	NaN	N
model_pow_8	0.92	43	[illegible]	[illegible]	[illegible]	[illegible]	-15	2.4	[illegible]	0.0077	NaN	NaN	NaN	N
model_pow_9	0.87	1.7e+02	[illegible]	[illegible]	[illegible]	[illegible]	-1.6e+02	37	-5.2	0.42	[illegible]	[illegible]	N	N
model_pow_10	0.87	1.4e+02	[illegible]	[illegible]	[illegible]	[illegible]	-87	15	-0.81	-0.14	[illegible]	[illegible]	N	N
model_pow_11	0.87	-75	[illegible]	[illegible]	[illegible]	[illegible]	9.1e+02	-3.5e+02	91	-16	[illegible]	[illegible]	034	N
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	0
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	1
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	-

Alternance of positive/negative parameters

Without cross-validation, the error always decreases with the number of parameters

Large increase of parameter estimates

Linear methods (solution: ridge and lasso regressions)

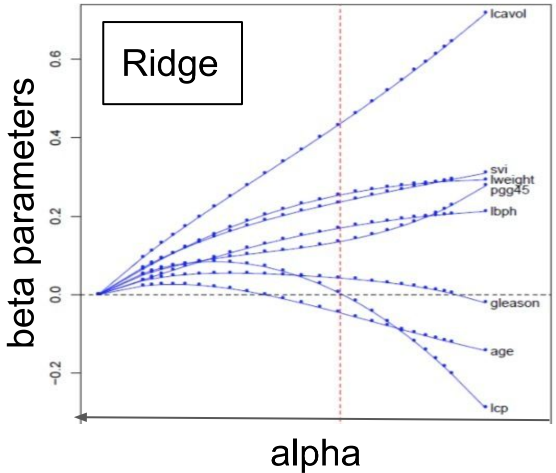
- Deal with:
  - Numerical problems of least squares
  - Highly correlated X predictors (ridge)
  - Large number of predictors X (lasso)
- Interests:
  - Robust to new independent data
  - Avoid overfitting (ridge)
  - Used for model selection (lasso)

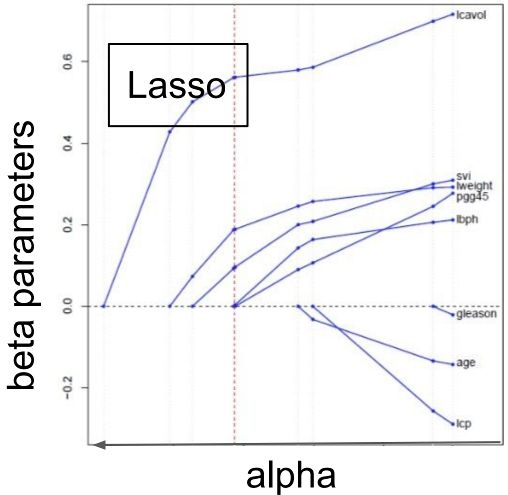
- Ridge cost function:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2 + \alpha\sum_{j=1}^{p}\beta_j^2$$

- Lasso cost function:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2 + \alpha\sum_{j=1}^{p}|\beta_j|$$





Linear methods (solution: ridge and lasso parameters)

Small and realistic parameter values

Ridge

	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	0.87	95	-3e+02	3.8e+02	-2.4e+02	66	0.96	-4.8	0.64	0.15	-0.026	-0.0054	0.00086	0.0
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	-4.2
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	1.9
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	1.7
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	1.1
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	2.4
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	-1.3
alpha_10	18	0.4	-0.0	-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	-1.6
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	-1.4

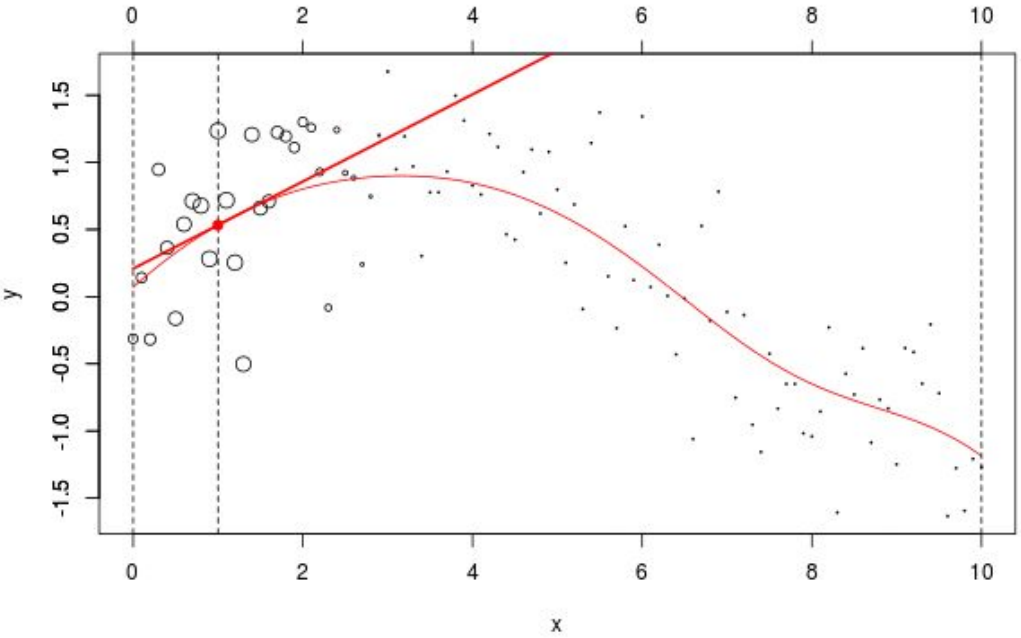
Simple model with few parameters

Lasso

	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

Nonlinear methods (local linear regression)



- Find the beta that minimizes locally:

$$\sum_{i=1}^{n} \omega_i (y_i - \beta_0 - \beta_1 x_i)^2$$

- With the weights given by local information:

$$\omega_i = K(x^\star - x_i)$$

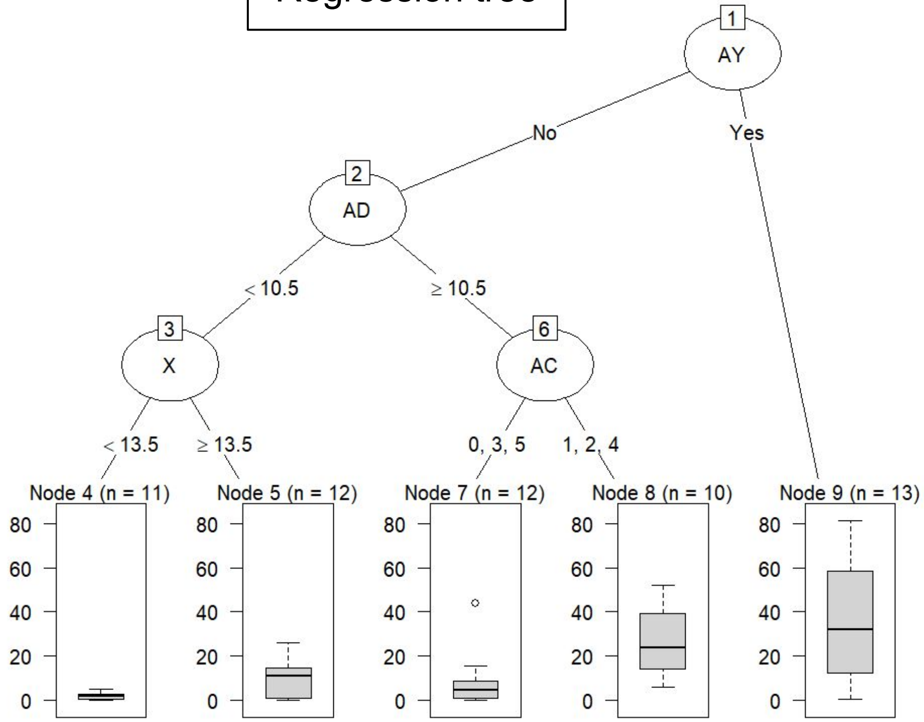
- Using for instance a Gaussian kernel:

$$K(x) = \exp\left(-\frac{x^2}{\lambda}\right)$$

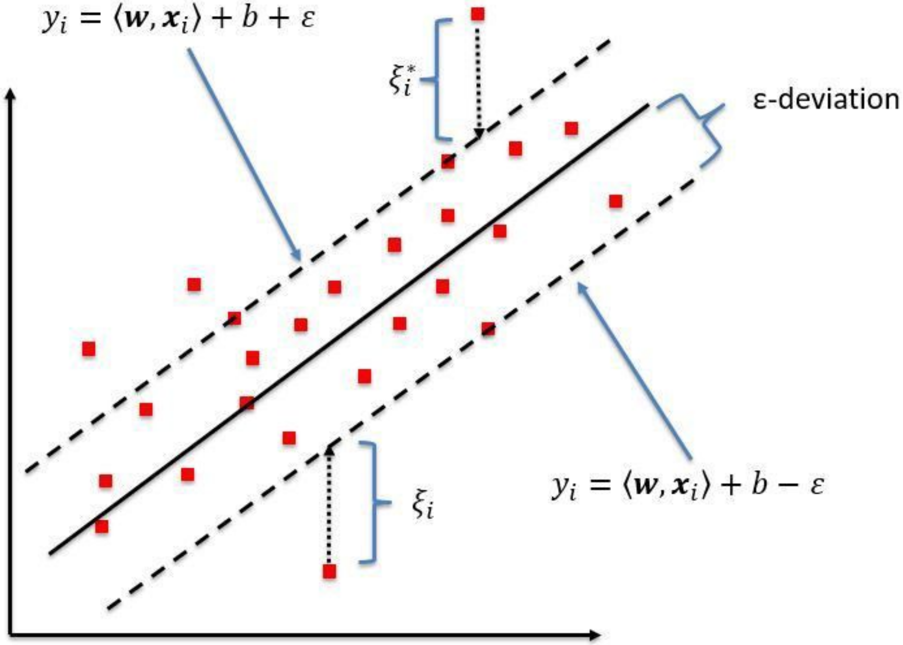
➩ play this Jupyter Notebook (chaotic Lorenz system)
https://github.com/DataScience4Geoscience/Toulouse2020/tree/master/Notebooks/Regression/ DSG_2020_regression_practice.ipynb

Other nonlinear methods

Regression tree



Support vector regression



Link between regression and classification

- In classification, y is **discrete**:
  - bimodal (0/1, heads/tails, etc…)
  - multimodal (large/medium/small, green/blue/red/green; etc…)
- In classification, different transfer and loss functions:
  - sigmoid transfer function f
  - cross-entropy loss function