# APM_4AI08_TP
# Linear Models
# Introduction to linear models

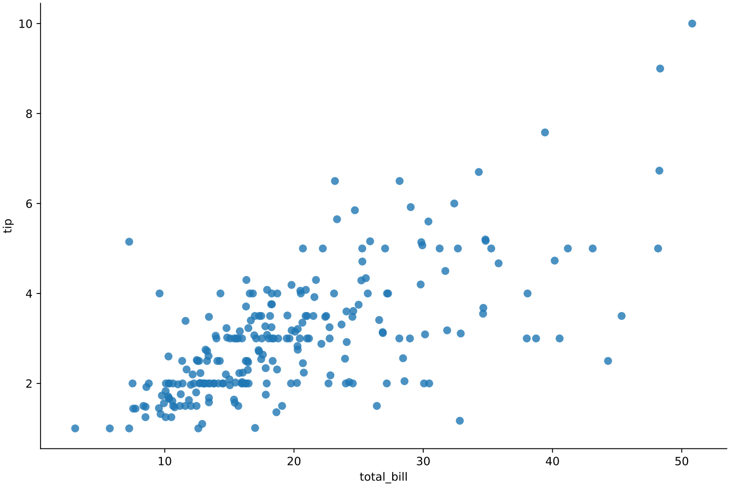
**Ekhiñe Irurozki**
Télécom Paris

# Intro

- Teaching team : Florence d'Alché, Nicolas Legouic, Mathilde Perez, Wen Yang, Thomas Sturma
- 1 TP, 2 TD, classes
- News are on Moodle
- In parallel with statistics
- Techniques here can be used in ML in general

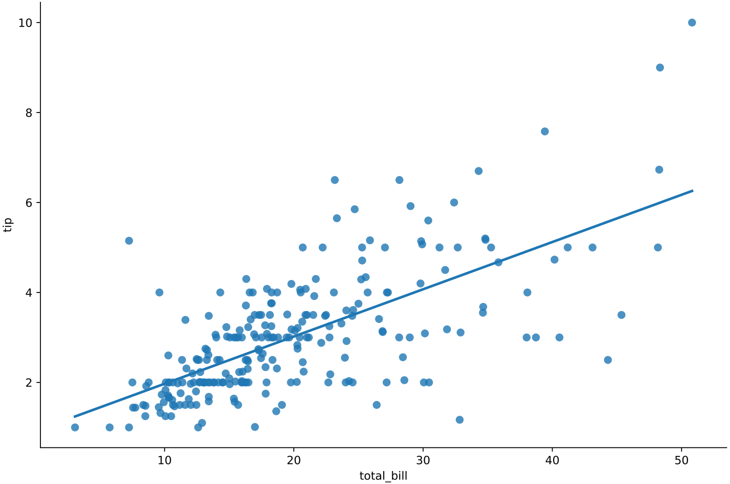
# A 2D starting example

Dependent-Independent variables, Regression. Assumption :linearity



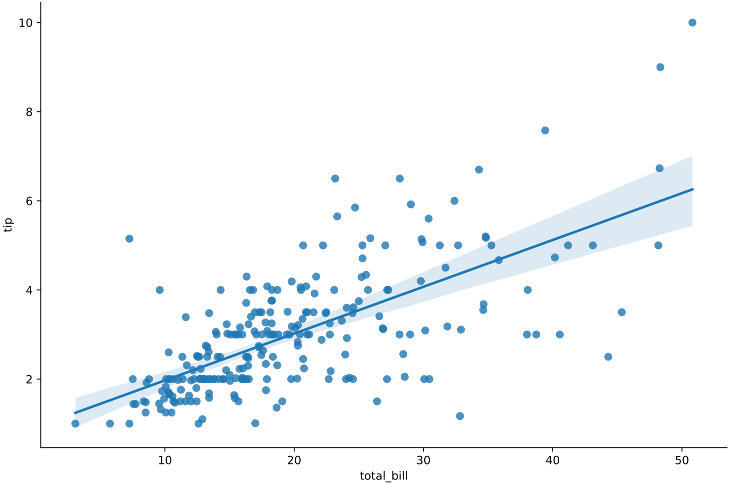
# A 2D starting example

Dependent-Independent variables, Regression. Assumption :linearity



# A 2D starting example

Dependent-Independent variables, Regression. Assumption :linearity



# Notation interpretation

- $n = 244$
- $p = 1$
- $y_i$ : tip let by the $i$-th customer
- $x_i$ : total bill payed by the $i$-th customer
- $y$ : the observation is the tips, dependent variable
- $x$ : the feature/covariate, price of the bill, independent variable

Linear model / Linear regression hypothesis : assume that the price of the bill and the tip let are linearly correlated

**Exo :** use `describe()` from `Pandas` to get a rough data summary

Three questions to be covered : modeling, learning and predicting

# Modeling I, the 1D case

Data

- $y$ is an **observation** or a variable to explain
- $x$ is a **feature** or a covariate

Given a sample : $\quad (x_i, y_i),\;$ for $i = 1, \ldots, n$

Linear model or linear regression hypothesis assume :

$$\boxed{y_i \approx \theta_0^\star + \theta_1^\star x_i}$$

Model coefficients

- **intercept** the scalar $\theta_0^\star$ ( : *ordonnée à l'origine*)
- **slope** the scalar $\theta_1^\star$ ( : *pente*)

<u>Rem</u>: both parameters are unknown from the statistician

# Modeling II

Probabilistic model. Let us give a precise meaning to the sign $\approx$ :

$$\begin{aligned} &y_i = \theta_0^\star + \theta_1^\star x_i + \varepsilon_i, \\ &\varepsilon_i \overset{i.i.d}{\sim} \varepsilon, \text{ for } i = 1, \dots, n \\ &\mathbb{E}(\varepsilon) = 0 \end{aligned}$$

where i.i.d. means “independent and identically distributed”

Interpretation : $\varepsilon_i = y_i - \theta_0^\star - \theta_1^\star x_i$ : represent the error between the theoretical model and the observations, represented by random variables $\varepsilon_i$ centered (often referred to as **white noise**).

<u>Rem</u>: motivation for the random nature of the noise – measurement noise, transmission noise, in-population variability, etc.

# Modeling III

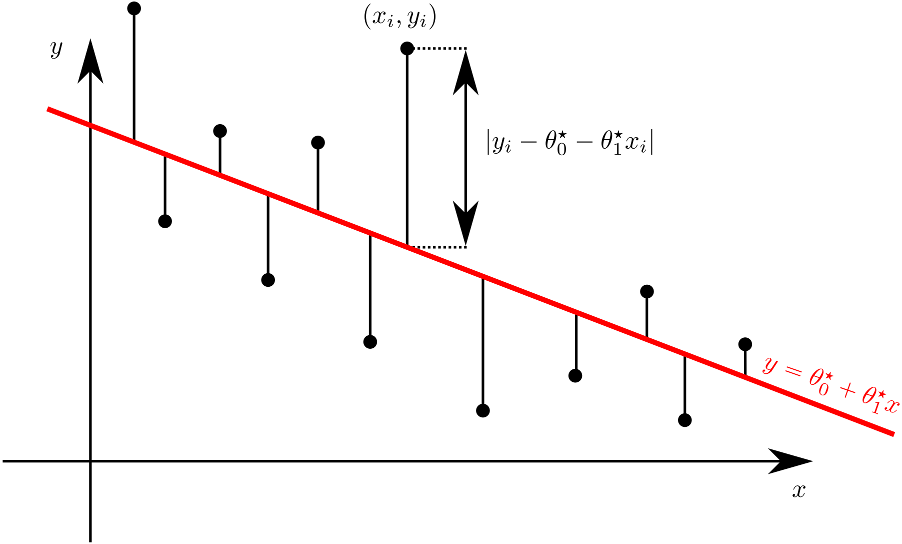
$$y_i = \theta_0^\star + \theta_1^\star x_i + \varepsilon_i$$

Our **goal in the learning stage** is to estimate $\theta_0^\star$ and $\theta_1^\star$ (unknown) by $\widehat{\theta}_0$ and $\widehat{\theta}_1$ relying on observations $(y_i, x_i)$ for $i = 1, \ldots, n$

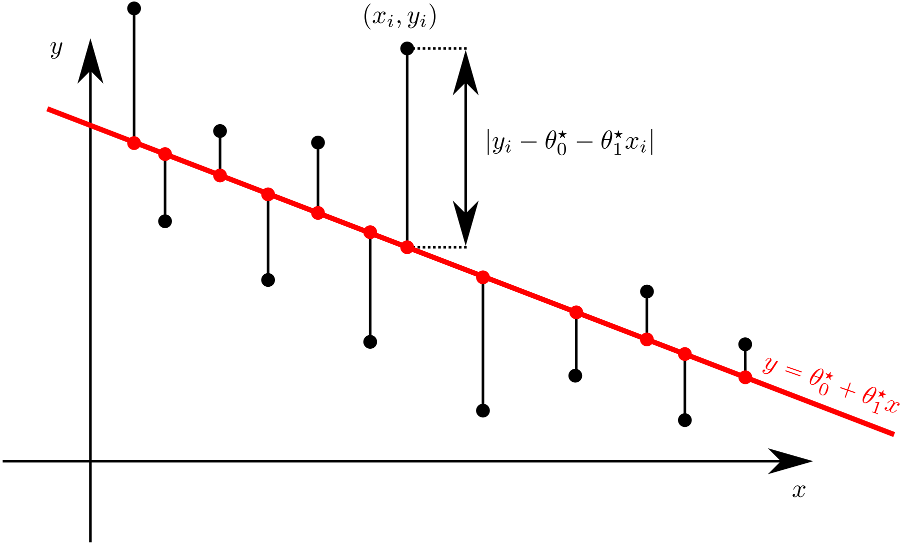
Rem: The "hat" notation is classical in statistics for referring to estimators

In **prediction time** $\widehat{y}_i = \widehat{\theta}_0 + \widehat{\theta}_1 x_i$

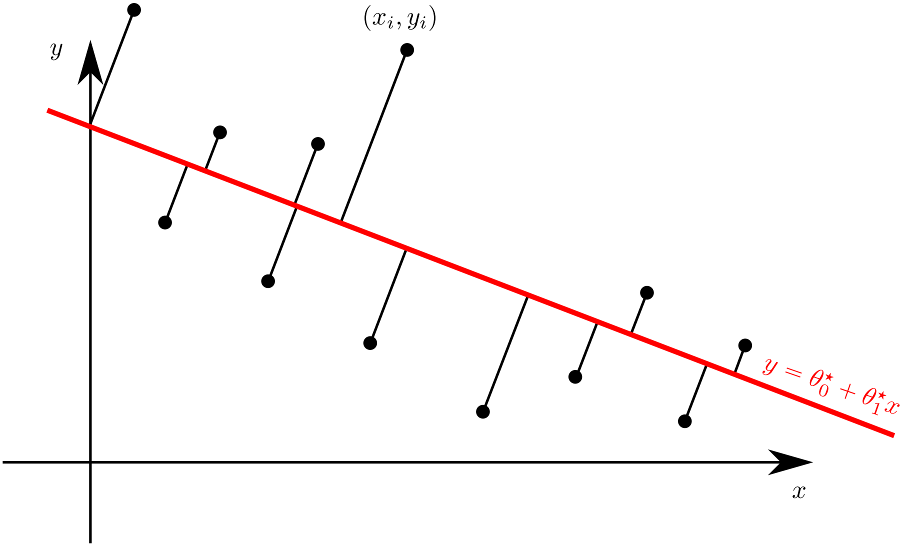
# Least squares : visualization



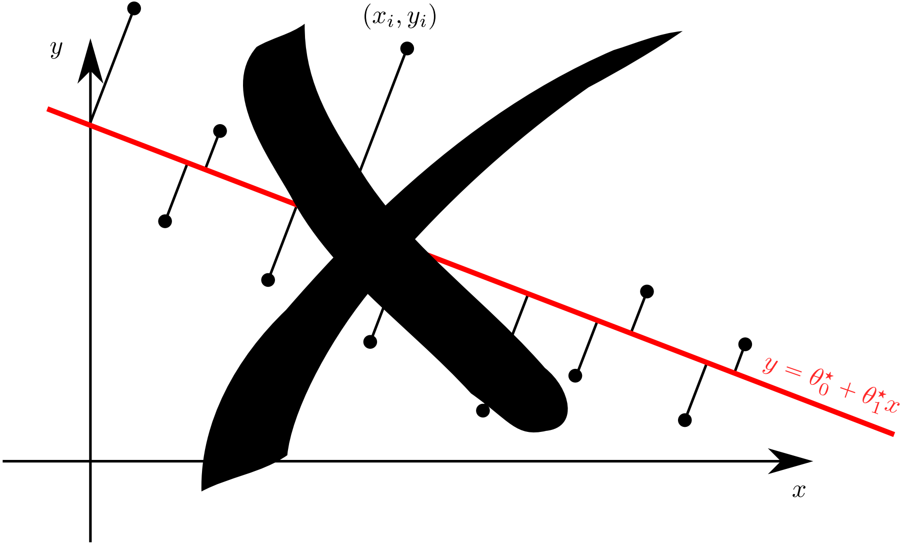
# Least squares : visualization



# (Total) Least squares : visualization



# (Total) Least squares : visualization



# Learning : mathematical formulation of Least squares

The **least squares** estimator is defined as :

$$(\widehat{\theta}_0, \widehat{\theta}_1) \in \operatorname{argmin}_{(\theta_0,\theta_1)\in\mathbb{R}^2} \sum_{i=1}^{n} \left(y_i - \theta_0 - \theta_1 x_i\right)^2$$

- Residual sum of squares (i.e., training error) is minimized
- Differentiate between $\theta^*$, $\theta$ and $\widehat{\theta}$ !!!!!
- it is also referred to as "ordinary least squares" (OLS)
- an original motivation for the squares is computational : first order conditions only require solving a linear system
- a solution always exists : minimizing a **coercive** continuous function (coercive : $\lim_{\|x\|\to+\infty} f(x) = +\infty$)

Rem: write « $\in$ argmin » as long as you do not know if the solution is unique

# Least square authorship (controversial)


Figure – Adrien-Marie Legendre and Carl Friedrich Gauss

# Historical / robust detour

The **least absolute deviation** (LAD) estimator reads :

$$(\widehat{\theta}_0, \widehat{\theta}_1) \in \operatorname{argmin}_{(\theta_0,\theta_1)\in\mathbb{R}^2} \sum_{i=1}^{n} |y_i - \theta_0 - \theta_1 x_i|$$

<u>Rem</u>: hard to compute without computer ; requires an optimization solver for non-smooth function (or a Linear Programming solver)

<u>Rem</u>: more robust to outliers ( : *données aberrantes*)

# Least absolute deviation authorship

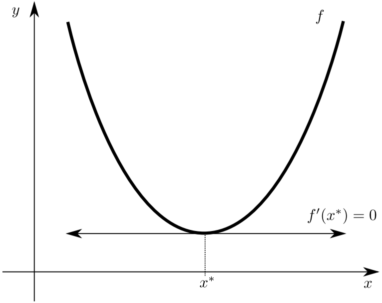
Figure – Ruđer Josip Bošković and Pierre-Simon de Laplace

# Existence and uniqueness of the solution

From now on, we consider the OLS and answer these question : Do the estimators $(\widehat{\theta}_0, \widehat{\theta}_1)$ exist ? Are the unique ?

Existence of a Local minimum : first order condition

**Fermat's rule Theorem** If $f$ is differentiable, then at a local minimum $x^*$ the gradient of $f$ vanishes at $x^*$, *i.e.* $\nabla f(x^*) = 0$.

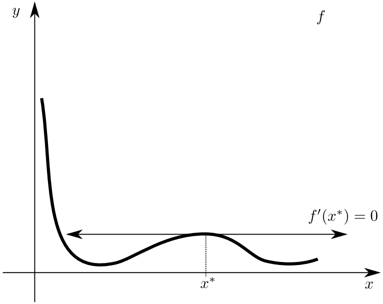


# Existence and uniqueness of the solution

From now on, we consider the OLS and answer these question : Do the estimators $(\widehat{\theta}_0, \widehat{\theta}_1)$ exist ? Are the unique ?

Existence of a Local minimum : first order condition

**Fermat's rule Theorem** If $f$ is differentiable, then at a local minimum $x^*$ the gradient of $f$ vanishes at $x^*$, *i.e.* $\nabla f(x^*) = 0$.



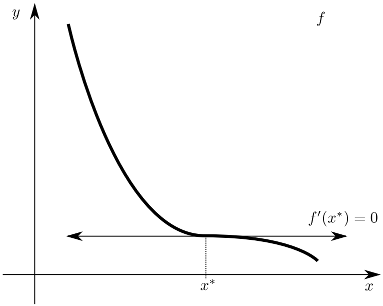

Rem: sufficient condition when $f$ is strongly convex !

# Existence and uniqueness of the solution

From now on, we consider the OLS and answer these question : Do the estimators $(\widehat{\theta}_0, \widehat{\theta}_1)$ exist ? Are the unique ?

Existence of a Local minimum : first order condition

**Fermat's rule Theorem** If $f$ is differentiable, then at a local minimum $x^*$ the gradient of $f$ vanishes at $x^*$, *i.e.* $\nabla f(x^*) = 0$.




Rem: sufficient condition when $f$ is strongly convex !

# The Hessian Matrix and Gradients

The **gradient** $\nabla f$ is a vector of first-order partial derivatives :

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

The **Hessian Matrix H** of $f$ is a square matrix of second-order partial derivatives :

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

# The minimizer is unique when $f$ its strictly convex

$f$ quadratic and nonnegative $\implies$ $f$ convex $\implies$ $\nabla^2 f(\widehat{\boldsymbol{\theta}})$ positive semi-definite

$\nabla^2 f(\widehat{\boldsymbol{\theta}})$ positive definite $\implies$ minimizer is unique

---

**Exo**: Derive the coefficients

---

# Back to least squares

$$\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_0, \widehat{\theta}_1) \in \operatorname{argmin}_{(\theta_0,\theta_1)\in\mathbb{R}^2} \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i)^2$$

For least squares, minimize the function of two variables :

$$f(\theta_0, \theta_1) = f(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i)^2$$

First order condition / Fermat's rule :

$$\begin{cases} \frac{\partial f}{\partial \theta_0}(\widehat{\boldsymbol{\theta}}) = \sum_{i=1}^{n}(y_i - \widehat{\theta}_0 - \widehat{\theta}_1 x_i) = 0 \\ \frac{\partial f}{\partial \theta_1}(\widehat{\boldsymbol{\theta}}) = \sum_{i=1}^{n}(y_i - \widehat{\theta}_0 - \widehat{\theta}_1 x_i)x_i = 0 \end{cases}$$

# Calculus continued

$$\text{Usual mean notation : } \overline{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ and } \overline{y}_n = \frac{1}{n}\sum_{i=1}^{n} y_i$$

With that, Fermat's rule states (dividing by $n$) :

$$\begin{cases} \frac{\partial f}{\partial \theta_0}(\widehat{\boldsymbol{\theta}}) = \sum_{i=1}^{n}(y_i - \widehat{\theta}_0 - \widehat{\theta}_1 x_i) = 0 \\ \frac{\partial f}{\partial \theta_1}(\widehat{\boldsymbol{\theta}}) = \sum_{i=1}^{n}(y_i - \widehat{\theta}_0 - \widehat{\theta}_1 x_i) x_i = 0 \end{cases}$$

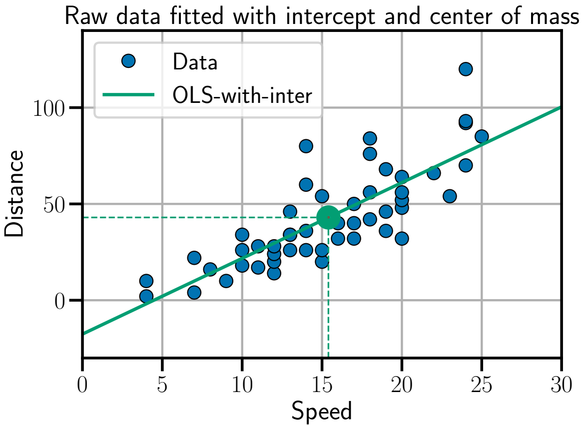
$$\Leftrightarrow$$

$$\begin{cases} \widehat{\theta}_0 = \overline{y}_n - \widehat{\theta}_1 \overline{x}_n & \text{(CNO1)} \\ \widehat{\theta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x}_n)(y_i - \overline{y}_n)}{\sum_{i=1}^{n}(x_i - \overline{x}_n)^2} & \text{(CNO2)} \end{cases}$$

**Exo :** Show that the solution to the OLS is unique iff $Var(x) \neq 0$

# Center of gravity and interpretation

$$(\text{CNO1}) \Leftrightarrow (\overline{x}_n, \overline{y}_n) \in \{(x,y) \in \mathbb{R}^2 : y = \widehat{\theta}_0 + \widehat{\theta}_1 x\}$$




- $\overline{speed} = 15.4$
- $\overline{dist} = 42.98$
- $\widehat{\theta}_0 = -17.579095$ intercept (negative !)
- $\widehat{\theta}_1 = 3.932409$ slope

Physical interpretation : the cloud of points' center of gravity belongs to the (estimated) regression line

# Vector formulation

<u>Notation</u> : $\mathbf{x} = (x_1, \dots, x_n)^\top$ and $\mathbf{y} = (y_1, \dots, y_n)^\top$

$$(\text{CNO2}) \Leftrightarrow \widehat{\theta}_1 = \frac{\sum_{i=1}^n (x_i - \overline{x}_n)(y_i - \overline{y}_n)}{\sum_{i=1}^n (x_i - \overline{x}_n)^2}$$

$$(\text{CNO2}) \Leftrightarrow \widehat{\theta}_1 = \text{corr}_n(\mathbf{x}, \mathbf{y}) \cdot \frac{\sqrt{\text{var}_n(\mathbf{y})}}{\sqrt{\text{var}_n(\mathbf{x})}}$$

where $$\text{corr}_n(\mathbf{x}, \mathbf{y}) = \frac{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x}_n)(y_i - \overline{y}_n)}{\sqrt{\text{var}_n(\mathbf{x})}\sqrt{\text{var}_n(\mathbf{y})}}$$

and $$\text{var}_n(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^n (z_i - \overline{z}_n)^2 \text{ (for any } \mathbf{z} = (z_1, \dots, z_n)^\top)$$

respectively **empirical correlation**, **empirical variances**

---

**Exo**: Derive this expression for $\widehat{\theta}_1$.

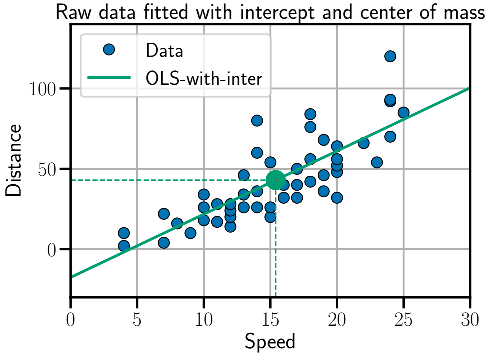
---

# *cars* example

This example plots the raking distance for cars as a function of the speed.
Line slope :

$$\mathrm{corr}_n(\mathbf{x}, \mathbf{y}) \cdot \frac{\sqrt{\mathrm{var}_n(\mathbf{y})}}{\sqrt{\mathrm{var}_n(\mathbf{x})}} = 3.932409.$$

Can the speed be negative ?
What if I shift the coordinate system so the centroid is at the origin ?



# Centering

**Centered** model :

$$\text{Write for any } i = 1, \ldots, n : \begin{cases} x_i^c = x_i - \overline{x}_n \\ y_i^c = y_i - \overline{y}_n \end{cases} \quad \Leftrightarrow \begin{cases} \mathbf{x}^c = \mathbf{x} - \overline{x}_n \mathbf{1}_n \\ \mathbf{y}^c = \mathbf{y} - \overline{y}_n \mathbf{1}_n \end{cases}$$

and $\mathbf{1}_n = (1, \ldots, 1)^\top \in \mathbb{R}^n$, then solving the OLS with $(\mathbf{x}^c, \mathbf{y}^c)$ leads to

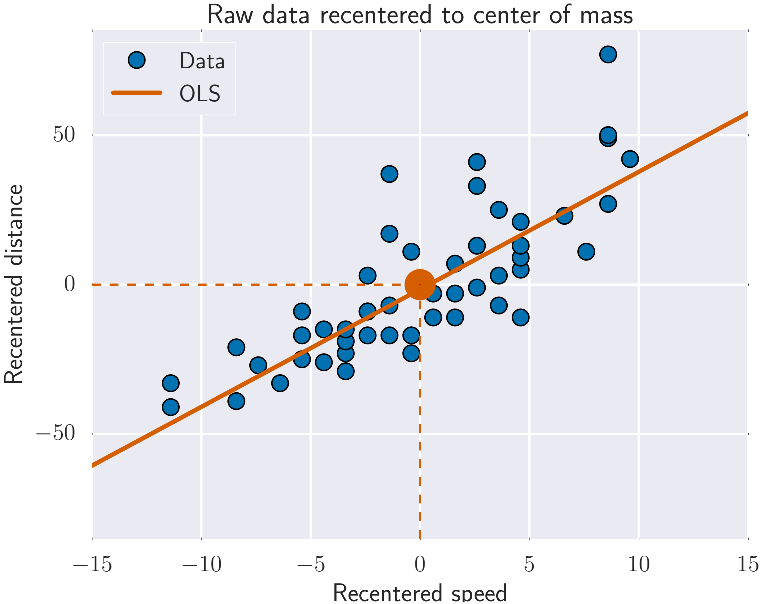
$$\widehat{\theta}_0^c = 0 \qquad \widehat{\theta}_1^c = \frac{\frac{1}{n}\sum_{i=1}^n x_i^c y_i^c}{\frac{1}{n}\sum_{i=1}^n (x_i^c)^2}$$

<u>Rem</u>: equivalent to choosing the cloud of points' center of mass as origin, *i.e.* $(\overline{x}_n^c, \overline{y}_n^c) = (0, 0)$

---

**Exo**: Derive this expression for $\widehat{\theta}_1^c$.

# Centering


Raw data recentered to center of mass
Data
OLS
50
0
−50
−15
−10
−5
0
5
10
15
Recentered distance
Recentered speed

# Centering and interpretation

Consider the coefficient $\widehat{\theta}_1^c$ ($\widehat{\theta}_0^c = 0$) for centered points $\mathbf{y}^c, \mathbf{x}^c$, then :

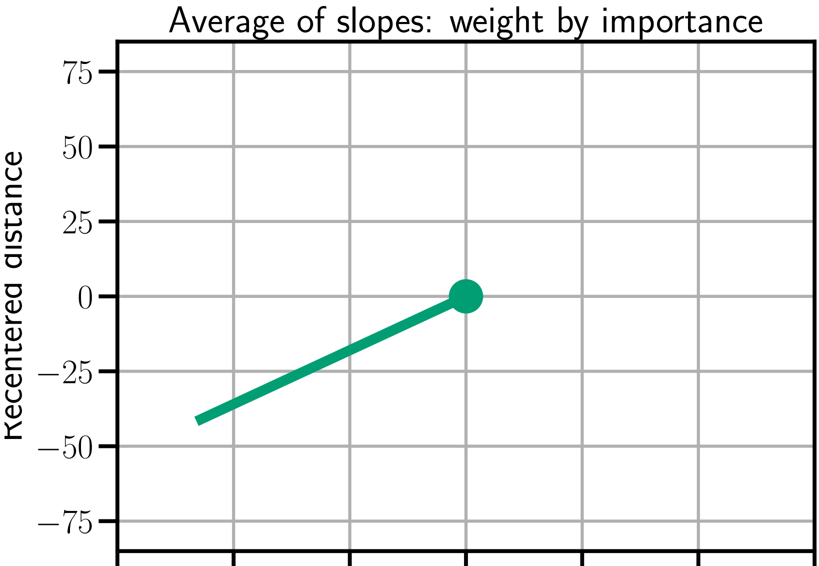
$$\widehat{\theta}_1^c \in \operatorname{argmin}_{\theta_1} \sum_{i=1}^{n} (y_i^c - \theta_1 x_i^c)^2 = \operatorname{argmin}_{\theta_1} \sum_{i=1}^{n} (x_i^c)^2 \left( \frac{y_i^c}{x_i^c} - \theta_1 \right)^2$$

Interpretation : $\widehat{\theta}_1^c$ is a weighted average of the slopes $\frac{y_i^c}{x_i^c}$

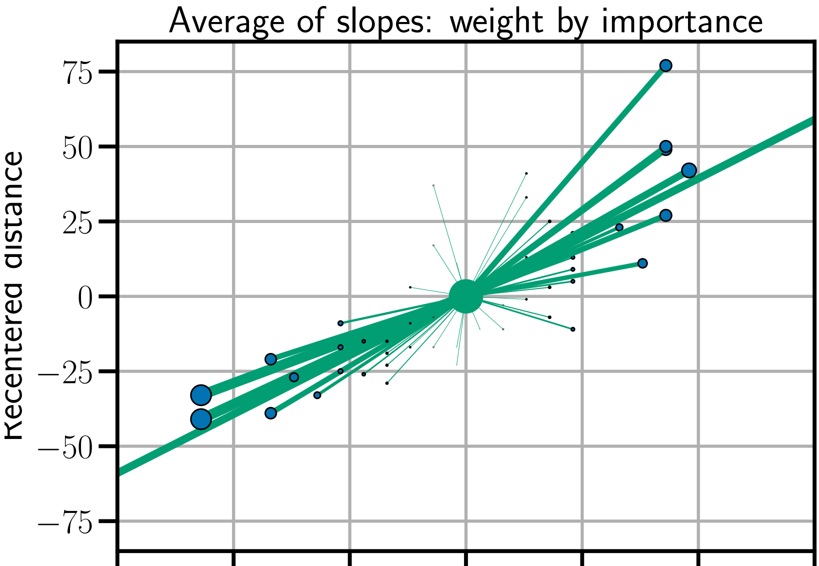
$$\widehat{\theta}_1^c = \frac{\displaystyle\sum_{i=1}^{n} (x_i^c)^2 \frac{y_i^c}{x_i^c}}{\displaystyle\sum_{j=1}^{n} x_j^{c2}}$$

Influence of extreme points : weights proportional to $(x_i^c)^2$ ; connected to the **leverage** ( : *levier*) effect

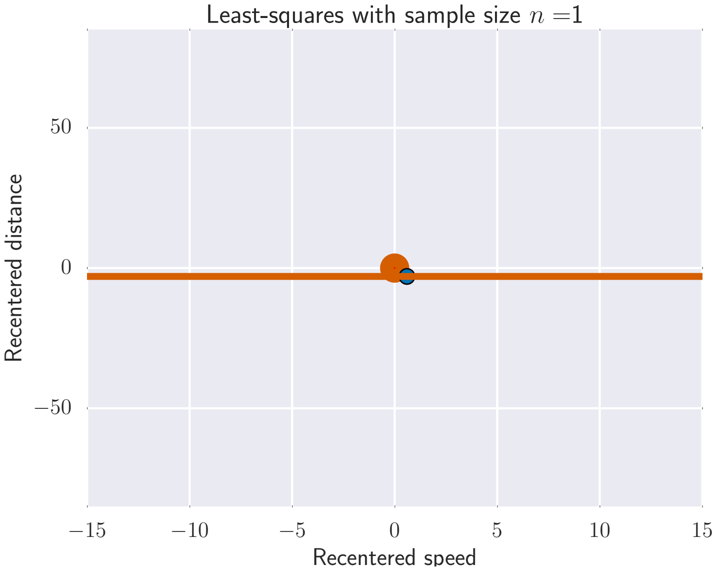
# Extreme points – leverage effect



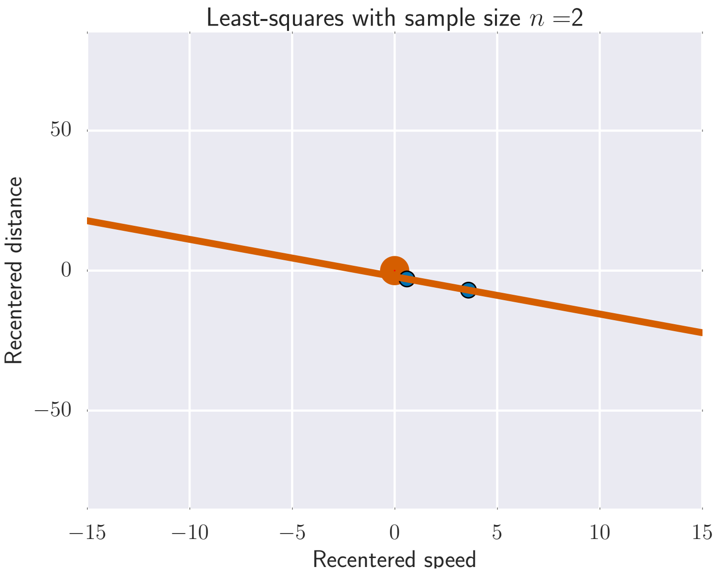
# Extreme points – leverage effect



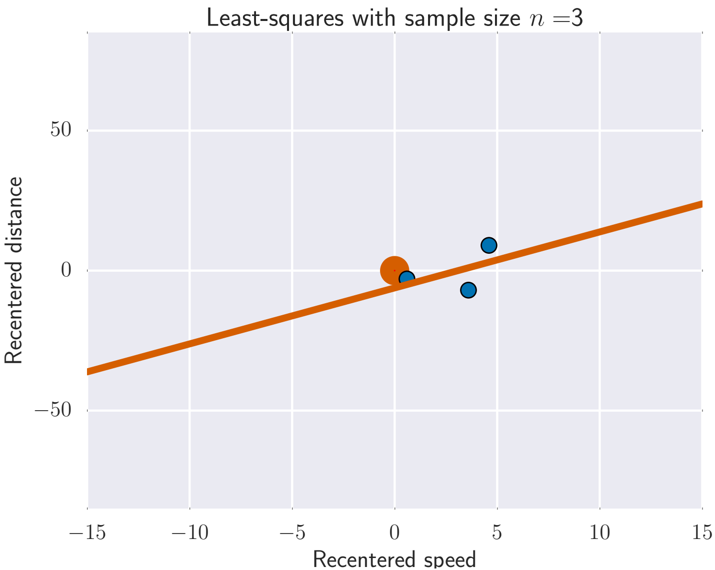
# Extreme points – leverage effect (II)



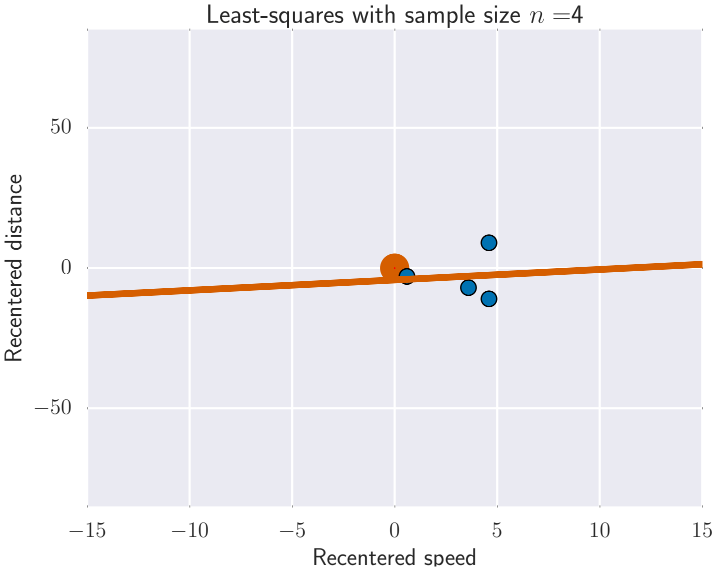
# Extreme points – leverage effect (II)



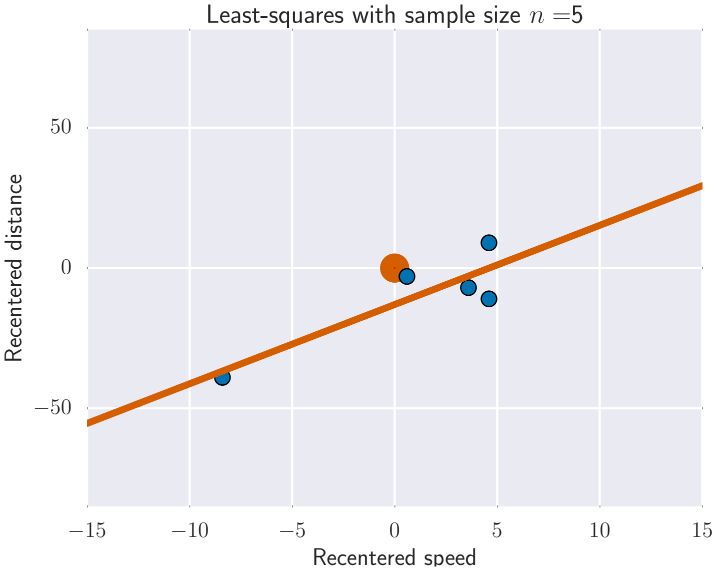
# Extreme points – leverage effect (II)



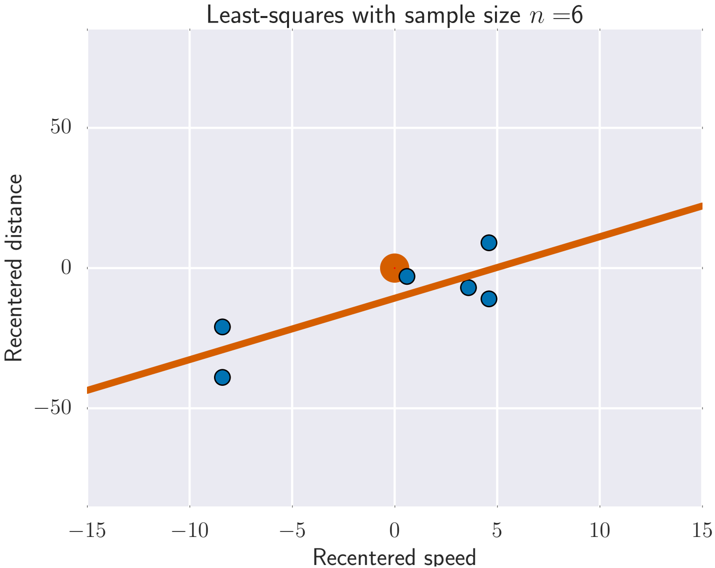
# Extreme points – leverage effect (II)



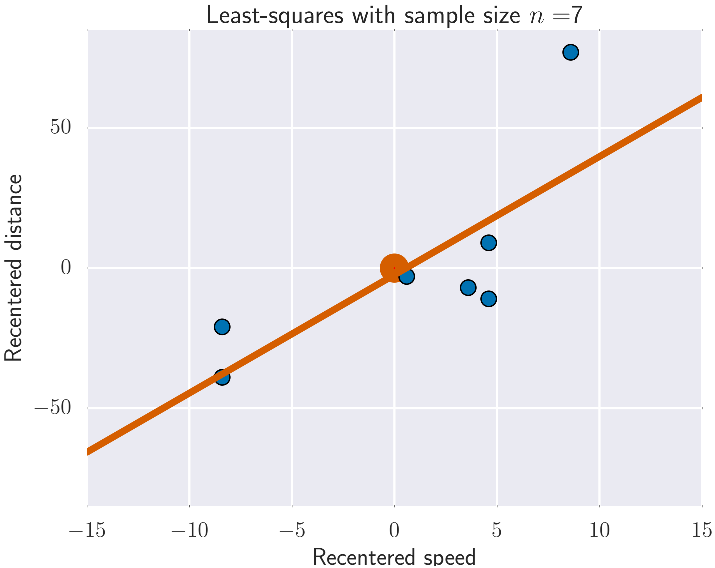
# Extreme points – leverage effect (II)



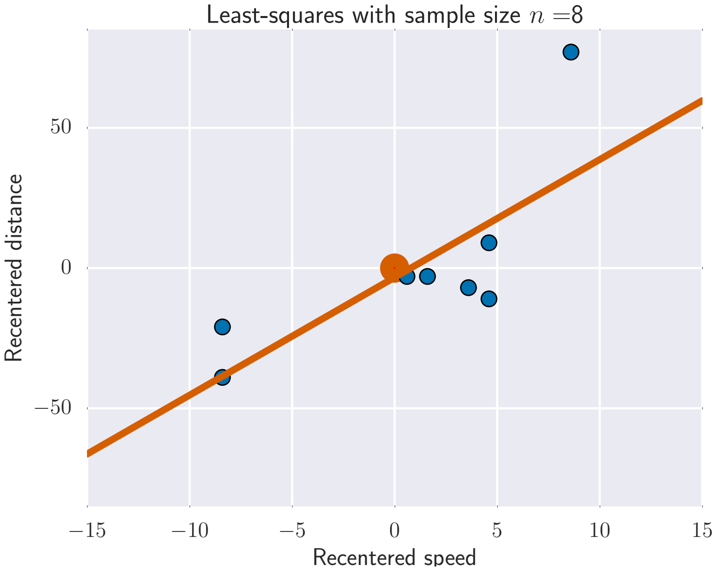
# Extreme points – leverage effect (II)



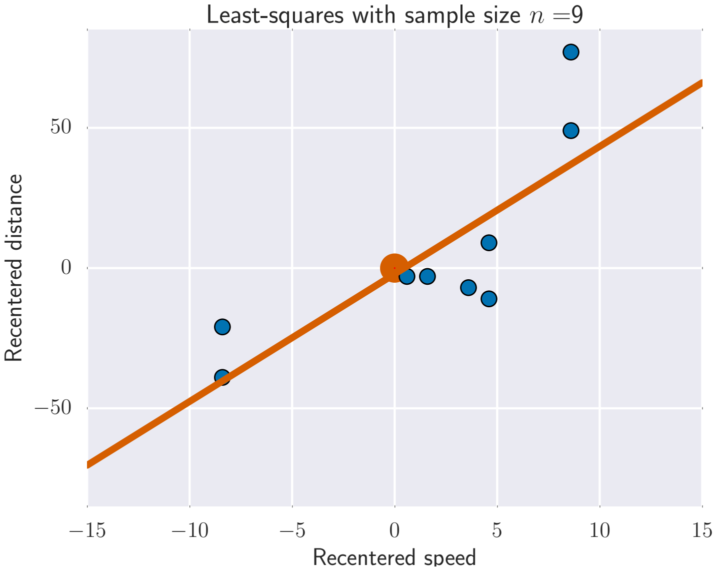
# Extreme points – leverage effect (II)



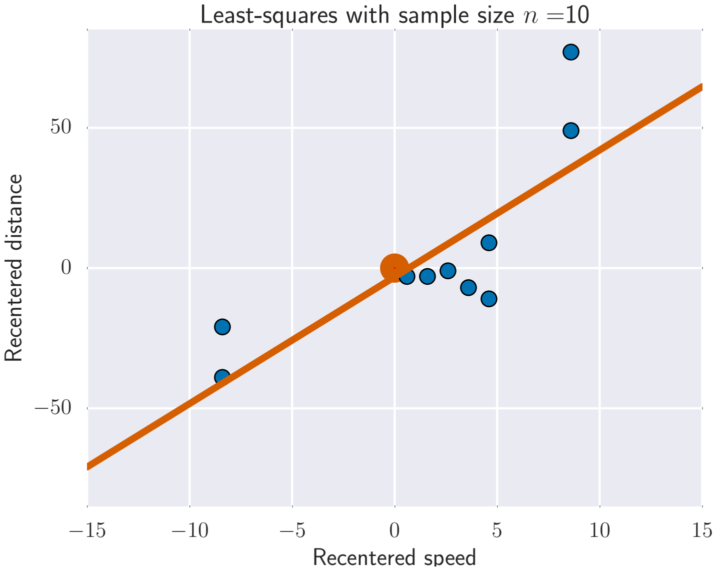
# Extreme points – leverage effect (II)



# Extreme points – leverage effect (II)

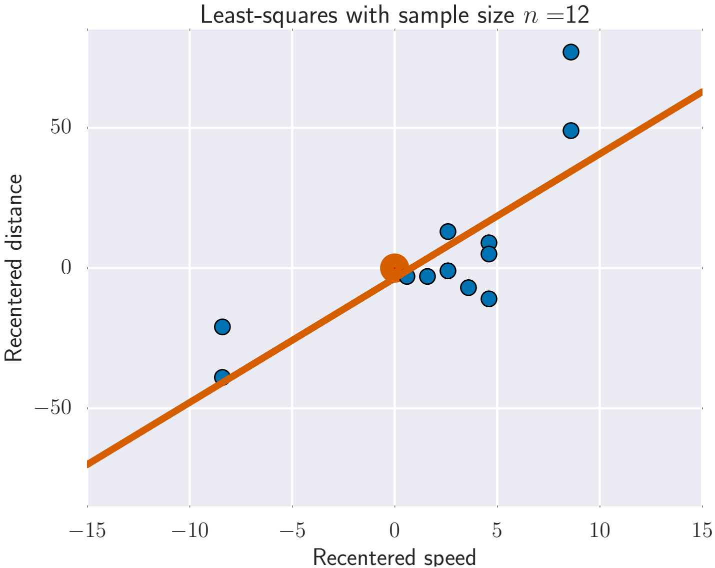


# Extreme points – leverage effect (II)



# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)



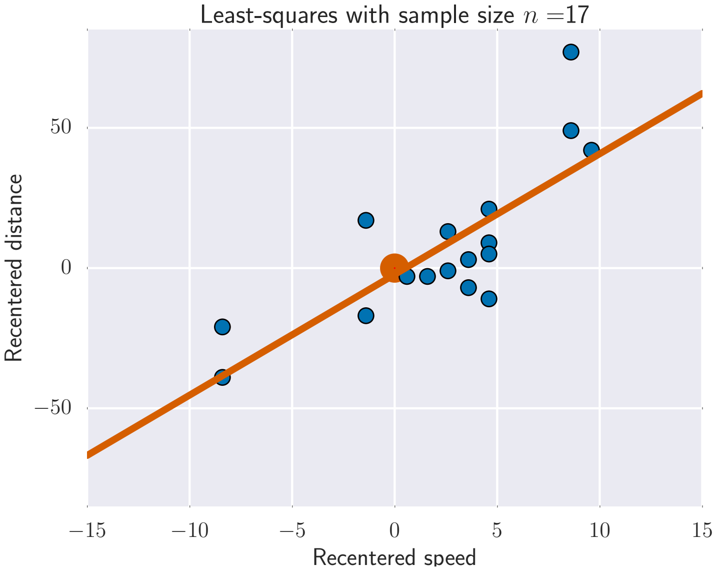
# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

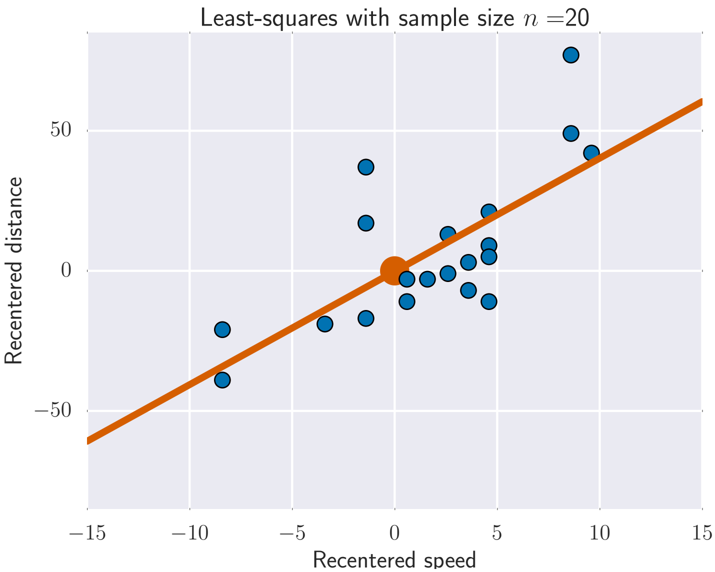
# Extreme points – leverage effect (II)



# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)



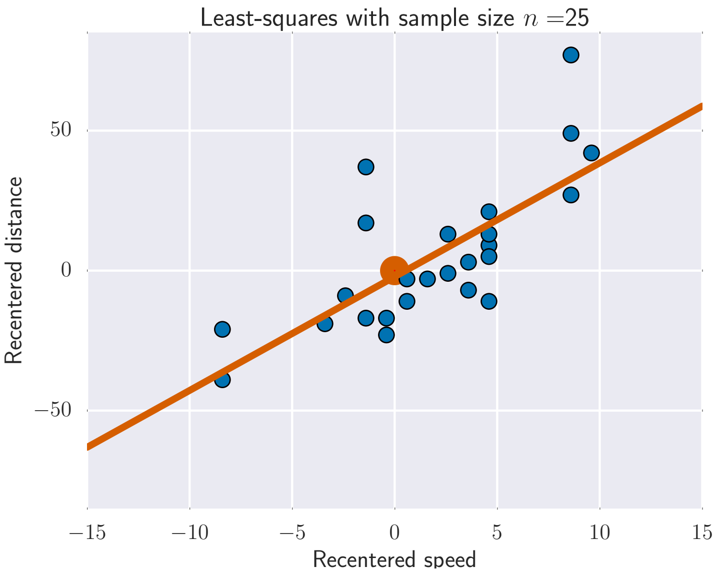
# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)



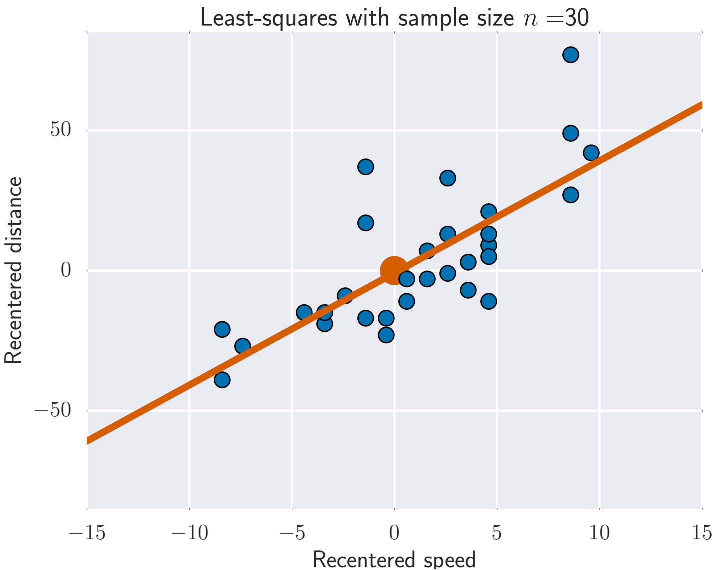
# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)



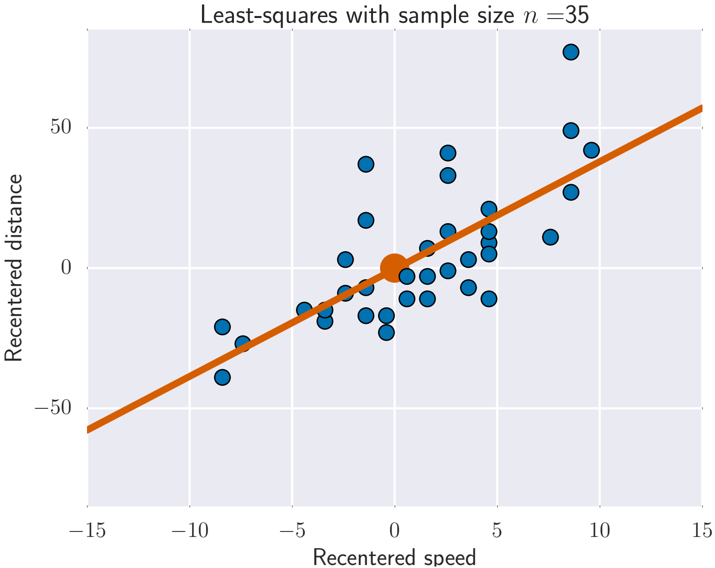
# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

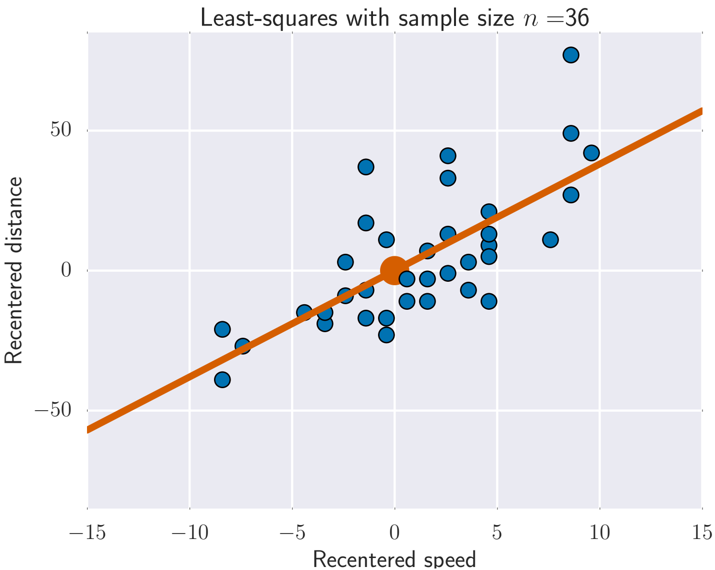
# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

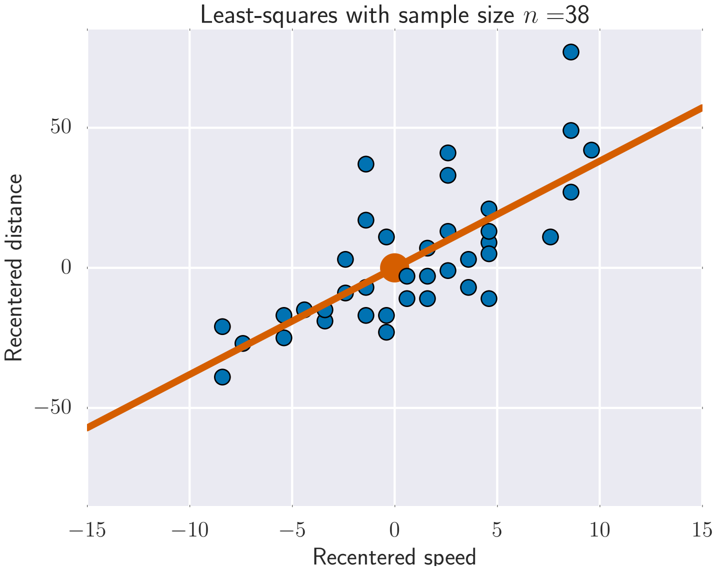


# Extreme points – leverage effect (II)

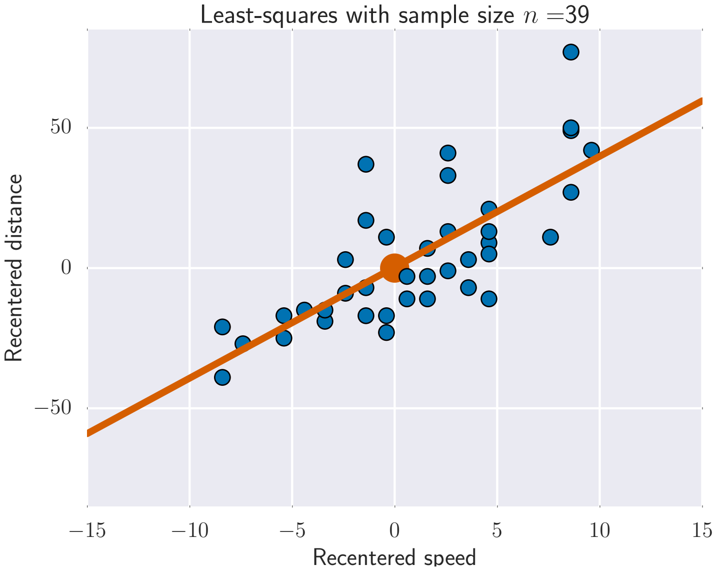


# Extreme points – leverage effect (II)

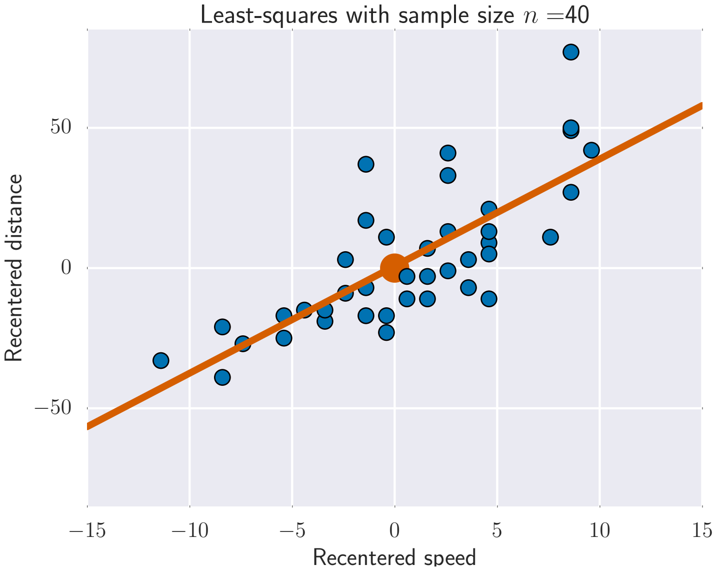
# Extreme points – leverage effect (II)



# Extreme points – leverage effect (II)



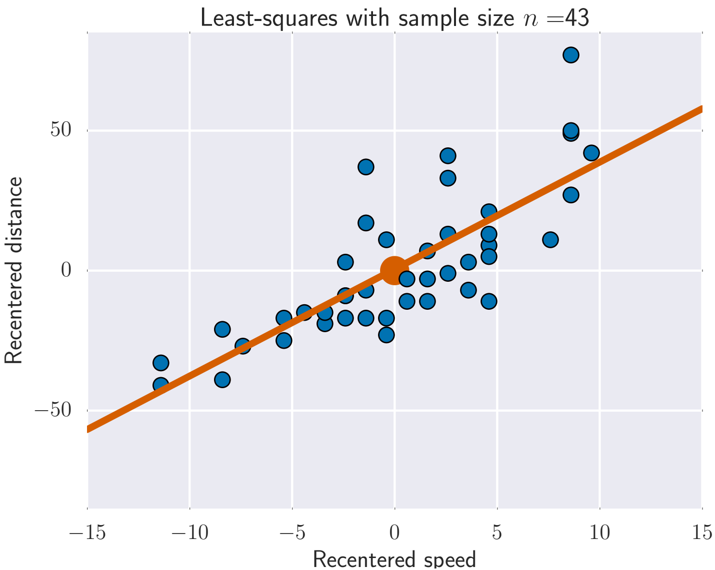
# Extreme points – leverage effect (II)



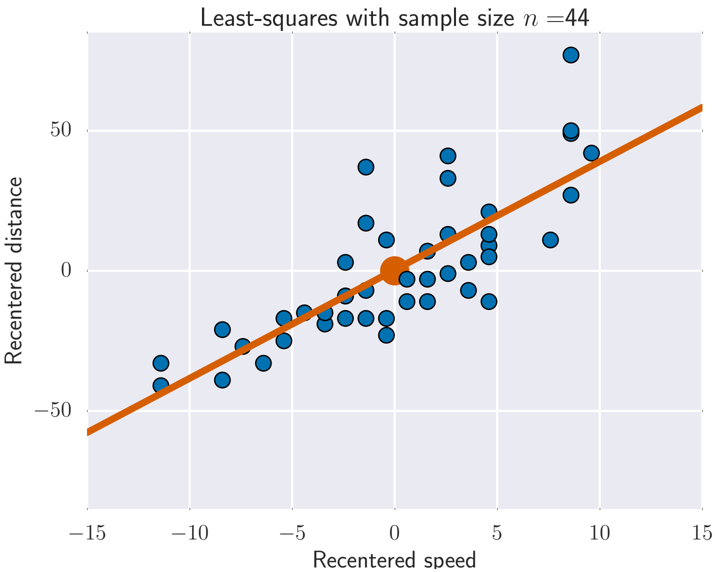
# Extreme points – leverage effect (II)

# Extreme points – leverage effect (II)

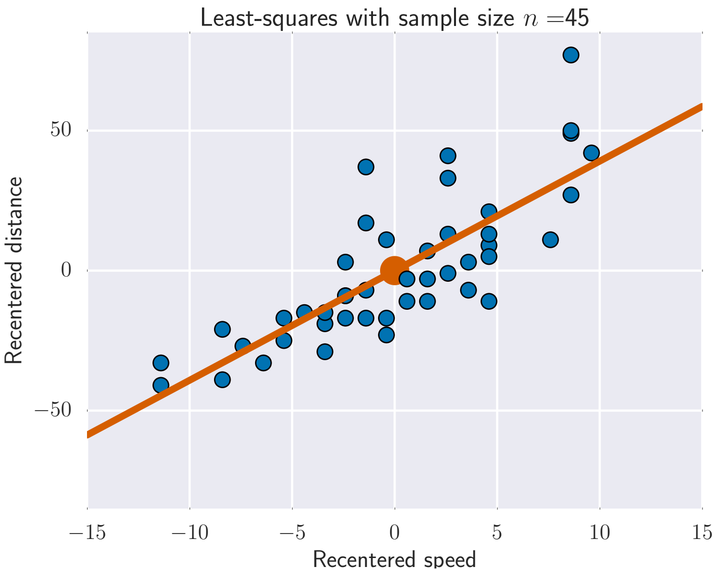
# Extreme points – leverage effect (II)



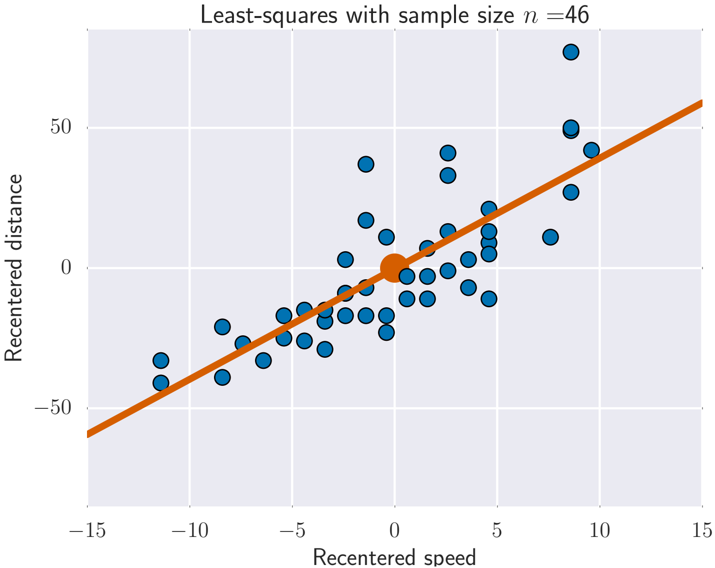
# Extreme points – leverage effect (II)



# Extreme points – leverage effect (II)

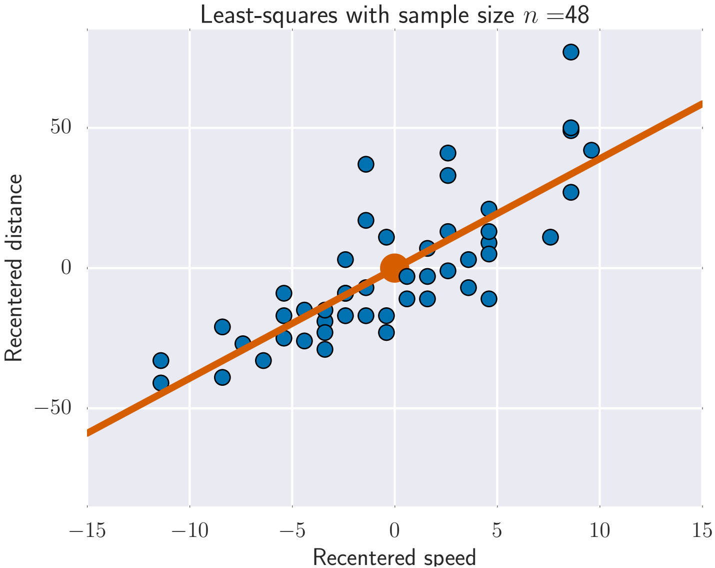


# Extreme points – leverage effect (II)

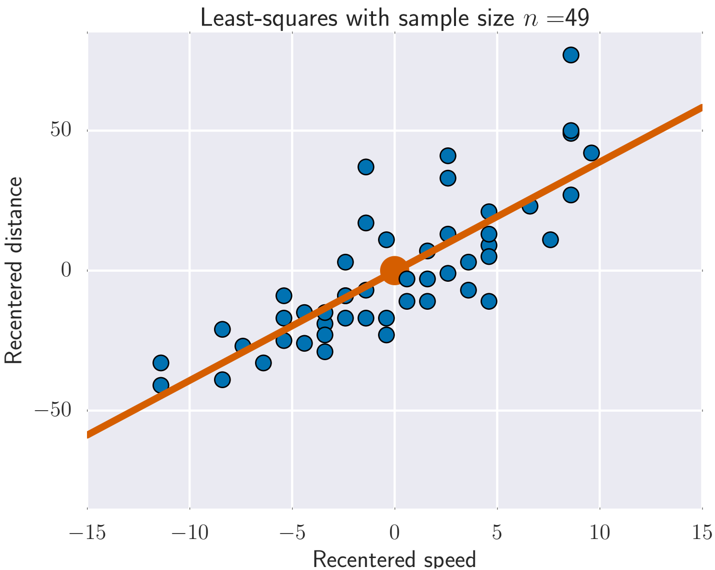


# Extreme points – leverage effect (II)

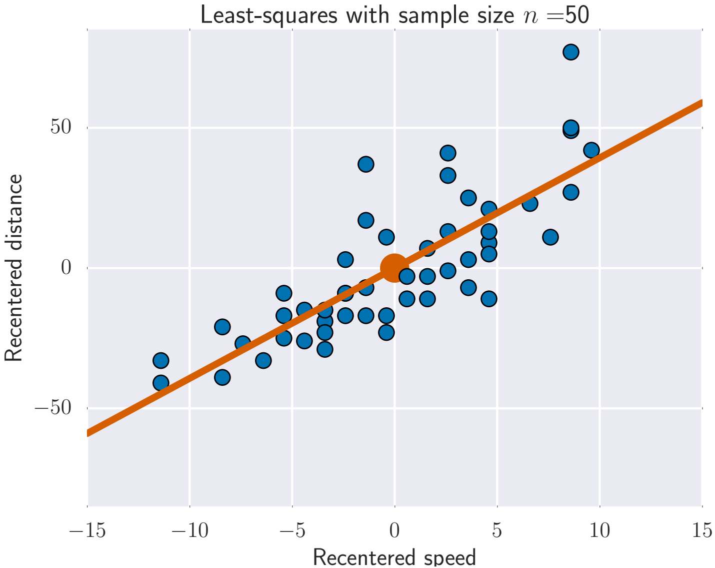
# Extreme points – leverage effect (II)



# Extreme points – leverage effect (II)



# Extreme points – leverage effect (II)



# Centering + scaling (standardization)

Centered-scaled model :

$$\forall i = 1, \dots, n : \begin{cases} x_i^s = (x_i - \overline{x}_n)/\sqrt{\mathrm{var}_n(\mathbf{x})} \\ y_i^s = (y_i - \overline{y}_n)/\sqrt{\mathrm{var}_n(\mathbf{y})} \end{cases} \quad \Leftrightarrow \begin{cases} \mathbf{x}^s = \dfrac{\mathbf{x} - \overline{x}_n \mathbf{1}_n}{\sqrt{\mathrm{var}_n(\mathbf{x})}} \\ \mathbf{y}^s = \dfrac{\mathbf{y} - \overline{y}_n \mathbf{1}_n}{\sqrt{\mathrm{var}_n(\mathbf{y})}} \end{cases}$$

Solving OLS with $(\mathbf{x}^s, \mathbf{y}^s)$ then

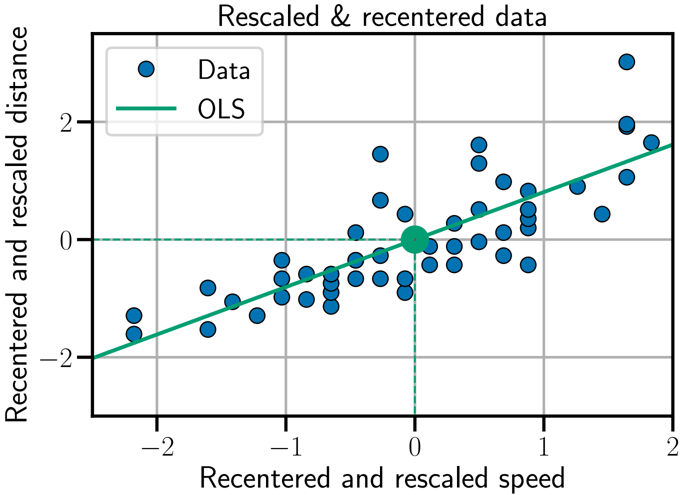
$$\begin{cases} \widehat{\theta}_0^s = 0 \\ \widehat{\theta}_1^s = \dfrac{1}{n} \displaystyle\sum_{i=1}^n x_i^s y_i^s \end{cases}$$

<u>Rem</u>: equivalent to choosing the points cloud center of mass as origin and normalize $\mathbf{x}$ and $\mathbf{y}$ to have unit **empirical norm** $\|\cdot\|_n$ :

$$\|\mathbf{x}^s\|_n^2 = \frac{1}{n} \sum_{i=1}^n (x_i^s)^2 = 1$$

$$\|\mathbf{y}^s\|_n^2 = \frac{1}{n} \sum_{i=1}^n (y_i^s)^2 = 1$$

# Centering + scaling



# When/why preprocessing ?

Centering **y** or using an intercept (or adding a constant feature) is equivalent

Rem: for sparse (🇫🇷 : *creux*) cases centering **y** adding a constant feature could be preferred

Scaling features is important :

- if you want to interpret the coefficients' amplitude in regression (better solution : t-tests)
- if you want to penalize or regularize coefficients (*c.f.* Lasso, Ridge, etc.) a single scale is needed
- for computing reasons (*e.g.* store scaling to improve efficiency, etc.)

Rem: in practice centering/scaling is useful for **estimation** not so much for **prediction** (see next courses)

What happens with the logarithm scaling ?

# Centering with Python

Use centering classes from **sklearn**, see **preprocessing** :

http://scikit-learn.org/stable/modules/preprocessing.html

```python
from sklearn import preprocessing

scaler = preprocessing.StandardScaler().fit(X)

print(np.isclose(scaler.mean_, np.mean(X)))

print(np.array_equal(scaler.std_, np.std(X)))

print(np.array_equal(scaler.transform(X),
                     (X - np.mean(X)) / np.std(X)))

print(np.array_equal(scaler.transform([26]),
                     (26 - np.mean(X)) / np.std(X)))
```

Rem:most valuable with **pipeline**

http://scikit-learn.org/stable/modules/pipeline.html

# Prediction

We call **prediction** function the function that associates an estimation of the variable of interest to a new sample. For least squares the prediction is given by :

$$\text{pred}(x_{n+1}) = \widehat{\theta}_0 + \widehat{\theta}_1 x_{n+1}$$

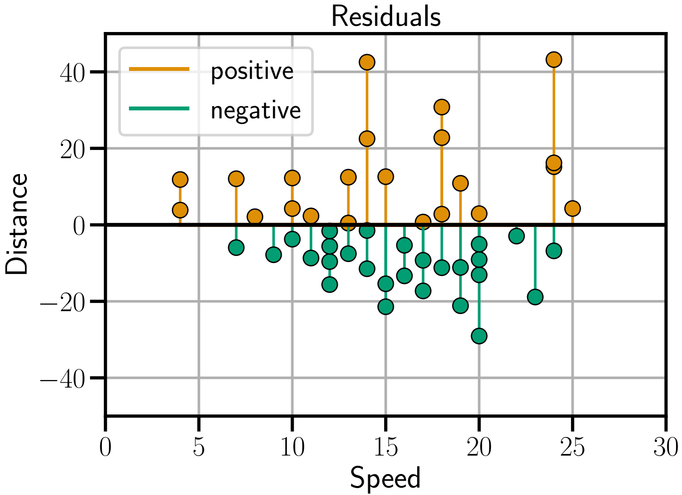
Rem: often written $\widehat{y}_{n+1}$ (implicit dependence on $x_{n+1}$)

The **residual** : difference between observations and predicted values

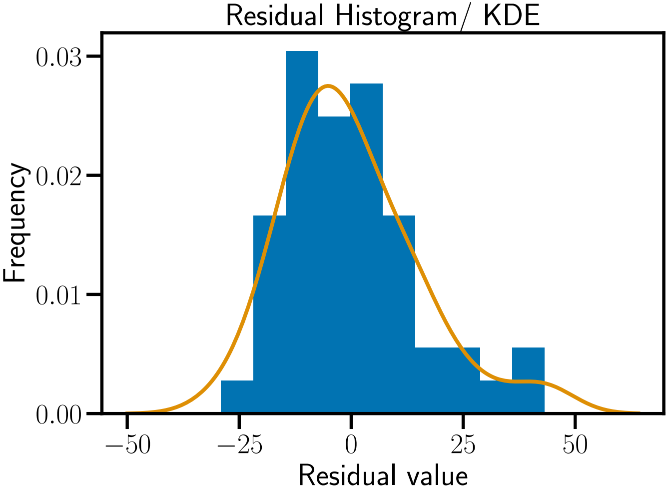
$$\epsilon_i = y_i - \text{pred}(x_i) = y_i - \widehat{y}_i = y_i - (\widehat{\theta}_0 + \widehat{\theta}_1 x_i)$$

Rem: observable estimate of the unobservable statistical error

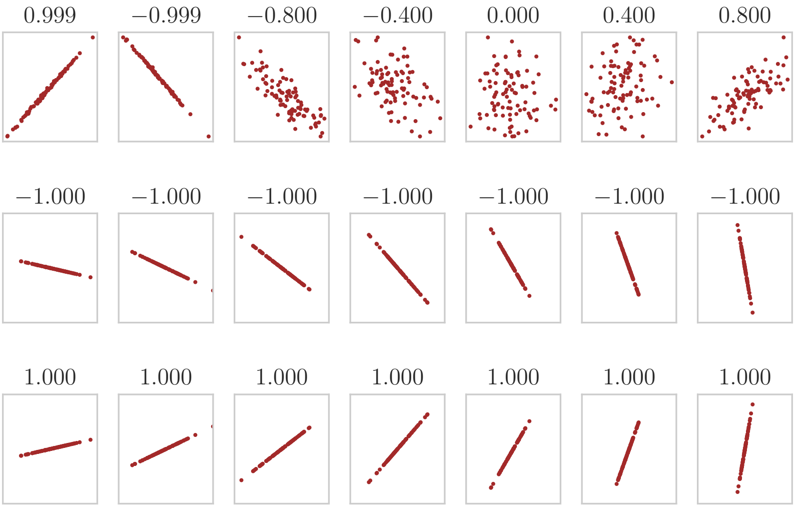
# Residuals (on cars, heteroscedasticity)



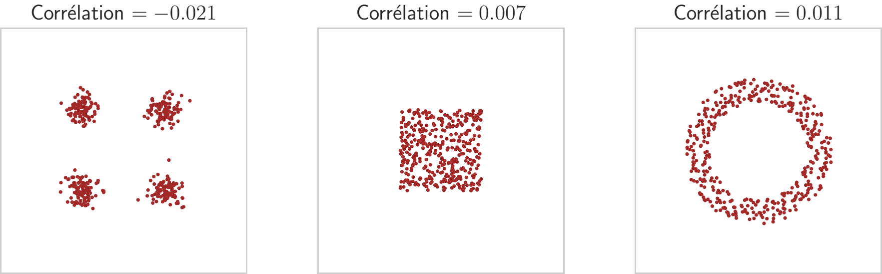
# Residual histograms



# Model Fit : Correlation, variance



# Model Fit : Correlation, variance




Always visualize the data https://www.research.autodesk.com/publications/same-stats-different-graphs/

# Model Fit : $R^2$ and Variance Decomposition

The coefficient of determination, denoted $R^2$, is defined as the ratio of the explained sum to the total sum :

$$R^2 = \frac{\sum_{i=1}^{n} (\widehat{y_i^c})^2}{\sum_{i=1}^{n} (y_i^c)^2}$$

- Scale : Residuals depend on the units of $Y$, while $R^2$ is dimensionless and normalized between 0 and 1.
- Comparability : Residuals cannot be compared across datasets with different scales of $Y$, but $R^2$ can.
- Interpretability : Residual measures the discrepancy between predictions and observations, whereas $R^2$ quantifies the proportion of variance in $Y$ explained by the model.

---

**Exo**: Show that

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \widehat{\varepsilon_i}^2}{\sum_{i=1}^{n} (y_i^c)^2}.$$

# Least squares motivation

- Computing advantage : computationally heavy methods avoided before computers (*e.g.* iterative methods)
- Theoretical advantage : least square analysis easy under simple hypothesis
- Interpretability : how much does the regressor increase with the features.

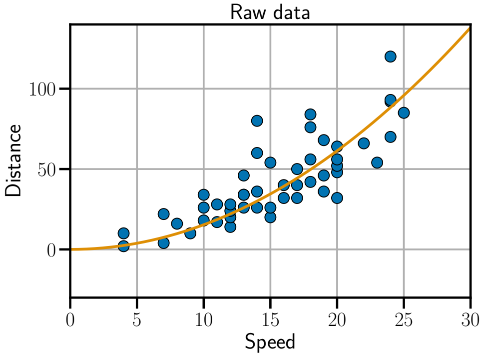
Example : under additive white Gaussian noise assumption *i.e.*, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ the maximum likelihood is equivalent to solving least squares to estimate $(\theta_0^\star, \theta_1^\star)$

Rem: for another noise model and/or to limit outliers influence one can solve (see *e.g.* `QuantReg` in `statsmodels`)

$$\boxed{\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_0, \widehat{\theta}_1) \in \mathrm{argmin}_{(\theta_0,\theta_1)\in\mathbb{R}^2} \sum_{i=1}^{n} |y_i - \theta_0 - \theta_1 x_i|}$$

# Discussion : toward multivariate cases

Physical laws (or your driving school memories) would lead to rather pick a **quadratic** model instead of a **linear** one : the OLS can be applied by choosing $x_i^2$ as features instead of $x_i$ :

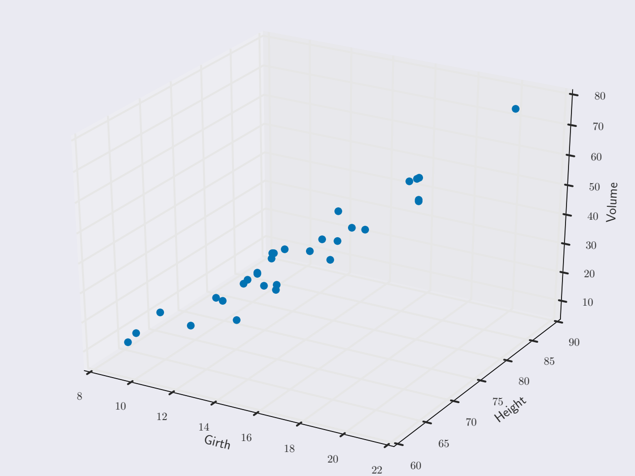


# Web sites and books to go further

- Datascience in general : Blog + videos by Jake Vanderplas
  http://jakevdp.github.io/
  **Homework for next lesson** : watch the following videos http://jakevdp.github.io/blog/2017/03/03/reproducible-data-analysis-in-jupyter/
- A few notebooks of OLS with `statsmodels`
- McKinney (2012) about `Python` for statistics
- Lejeune (2010) about linear models (in French)
- Regression course by B. Delyon (in French, more technical)

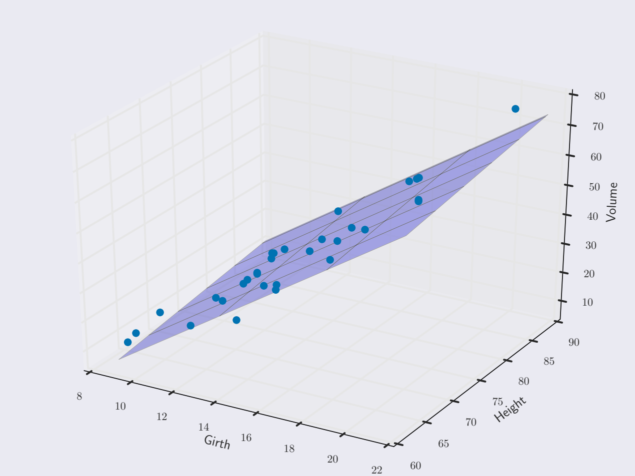
# Toward multivariate models

Tree volume as a function of height / girth (🇫🇷 : *circonférence*)



# Toward multivariate models

Tree volume as a function of height / girth (🇫🇷 : *circonférence*)



# Python commands

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Load example data
...

# Fit linear regression model
model = LinearRegression()
model.fit(X, y)
```

# Model

One observes $p$ features $(\mathbf{x}_1, \dots, \mathbf{x}_p)$. Model in dimension $p$

$$
\begin{aligned}
y_i &= \theta_0^\star + \sum_{j=1}^{p} \theta_j^\star x_{i,j} + \varepsilon_i \\
\varepsilon_i &\overset{i.i.d}{\sim} \varepsilon, \text{ pour } i = 1, \dots, n \\
\mathbb{E}[\varepsilon] &= 0
\end{aligned}
$$

<u>Rem</u>: we assume (frequentist point of view) there exists a “true” parameter $\boldsymbol{\theta}^\star = (\theta_0^\star, \dots, \theta_p^\star)^\top \in \mathbb{R}^{p+1}$

# Dimension $p$

## Matrix model

$$\underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,p} \end{pmatrix}}_{X} \underbrace{\begin{pmatrix} \theta_0^\star \\ \vdots \\ \theta_p^\star \end{pmatrix}}_{\boldsymbol{\theta}^\star} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}}_{\boldsymbol{\epsilon}}$$

$$\text{Equivalently :} \quad \boxed{\mathbf{y} = X\boldsymbol{\theta}^\star + \boldsymbol{\epsilon}} \tag{1}$$

Column notation : $X = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$ with $\mathbf{x}_0 = \mathbf{1}_n = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$.

Line notation : $X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{pmatrix} = (x_1, \dots, x_n)^\top$

# Matrix Notation and $L_2$ Norm

Matrix notation is a powerful way to represent mathematical operations involving vectors and matrices.

The **Inner Product** (dot product) of two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined as :

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{n} u_i v_i = \mathbf{u}^T \cdot \mathbf{v}$$

Let $\mathbf{A}$ be an $m \times n$ matrix and $\mathbf{B}$ be an $n \times p$ matrix. The **matrix product** $\mathbf{C} = \mathbf{AB}$ is an $m \times p$ matrix with elements :

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

The $L_2$ **Norm** (Euclidean norm) of a vector $\mathbf{v}$ is defined as :

$$\|\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}$$

Matrix notation simplifies operations and equations involving vectors and matrices.

# Vocabulary

$$\boxed{\mathbf{y} = X\boldsymbol{\theta}^\star + \boldsymbol{\epsilon}}$$

- $\mathbf{y} \in \mathbb{R}^n$ : observations vector
- $X \in \mathbb{R}^{n\times(p+1)}$ : **design** matrix (with features as columns and a first column of 1s)
- $\tilde{X} \in \mathbb{R}^{n\times(p)}$ : **reduced design** matrix (with features as columns and NO column of ones)
- $\boldsymbol{\theta}^\star \in \mathbb{R}^{p+1}$ : (unknown) **true** parameter to be estimated
- $\boldsymbol{\epsilon} \in \mathbb{R}^n$ : noise vector

# Vocabulary (and abuse of terms)

We call **Gram matrix** the matrix

$$X^\top X$$

whose general term is $[X^\top X]_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$

If the design matrix $X$ is centered and scaled, the Gram matrix is proportional to the correlation between columns. $X^\top X$ is often referred to as the feature correlation matrix

<u>Rem</u>: when columns are scaled such that $\forall j \in [\![0, p]\!], \|\mathbf{x}_j\|^2 = n$, the Gramian diagonal is $(n, \ldots, n)$

The vector $X^\top \mathbf{y} = \begin{pmatrix} \langle \mathbf{x}_0, \mathbf{y} \rangle \\ \vdots \\ \langle \mathbf{x}_p, \mathbf{y} \rangle \end{pmatrix}$ represents the correlation between the observations and the features

# (Ordinary) Least squares

**A** least square estimator is **any** solution of the following problem :

$$\widehat{\boldsymbol{\theta}} \in \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^{p+1}} \left( \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \right)$$

$$\widehat{\boldsymbol{\theta}} \in \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left[ y_i - \left( \theta_0 + \sum_{j=1}^{p} \theta_j x_{i,j} \right) \right]^2$$

$$\widehat{\boldsymbol{\theta}} \in \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left[ y_i - \langle x_i, \boldsymbol{\theta} \rangle \right]^2$$

- Does the solution exist ? A solution always exists, as we are minimizing a coercive continuous function (**coercive** : $\lim_{\|x\| \to +\infty} f(x) = +\infty$)
- Is the solution unique ? not guaranteed

**Exo** how do we make the prediction ?

# Row / column interpretation

**Row interpretation**

Let $\tilde{x}_1^\top, \ldots, \tilde{x}_{p+1}^\top$ be the rows of $X$. The residuals are $r_i = y_i - \tilde{x}_i\boldsymbol{\theta}$ and the OLS is equivalent to minimizing the sum of squares residuals

**Column interpretation**

Let $\mathbf{x}_0, \ldots, \mathbf{x}_p$ be the columns of $X$. Then $\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 = \|(\theta_0\mathbf{x}_0, + \ldots + \theta_p\mathbf{x}_p) - \mathbf{y}\|_2^2$, so OLS is to find a linear combination of columns of $X$ that is closest to $\mathbf{y}$.

# Hilbert projection theorem (HPT)

The HPT states that :

Let $C \subset \mathbb{R}^d, Y \in \mathbb{R}^d$. Let $\widehat{z} = \arg\min_{z \in C} \|Y - z\|_2^2$. Then $\widehat{z}$ always exists and is given by

$$\boxed{< Y - \widehat{z}, z >= 0 \qquad \forall z \in C}$$

We can use this theorem to characterize the solutions for the OLS

# Hilbert projection theorem (HPT) and application to OLS

$$\widehat{\boldsymbol{\theta}} \in \operatorname{argmin}_{\theta \in \mathbb{R}^{p+1}} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$$

Note $col(X) = span([\mathbf{x}_0, ..., \mathbf{x}_p]) = \sum_{j=0}^{p} \mathbf{x}_j \theta_j = X\boldsymbol{\theta}$

OLS can be written as : $\widehat{W} \in \operatorname{argmin}_{W \in col(X)} \left(\|\mathbf{y} - W\|_2^2\right)$ and the HPT can be directly applied

$$\begin{aligned}
< \mathbf{y} - \widehat{W}, W > &= 0 \\
(\mathbf{y} - \widehat{W})^\top W &= 0 \\
(\mathbf{y} - \widehat{W})^\top X\boldsymbol{\theta} &= 0 \\
(\mathbf{y} - \widehat{W})^\top X &= 0 \\
(\mathbf{y} - X\widehat{\boldsymbol{\theta}})^\top X &= 0 \\
X^\top(\mathbf{y} - X\widehat{\boldsymbol{\theta}}) &= 0 \\
X^\top X\widehat{\boldsymbol{\theta}} &= X^\top \mathbf{y}
\end{aligned} \tag{2}$$

# OLS normal equations

The solution to the OLS problem is given by the solution to the normal equation

$$\textbf{Normal equation :} \; \boxed{X^\top X \widehat{\boldsymbol{\theta}} = X^\top \mathbf{y}}$$

As a consequence,

- a solution always exists.
- its unique if the solution to the normal equations is unique

# Hilbert projection theorem, geometric interpretation

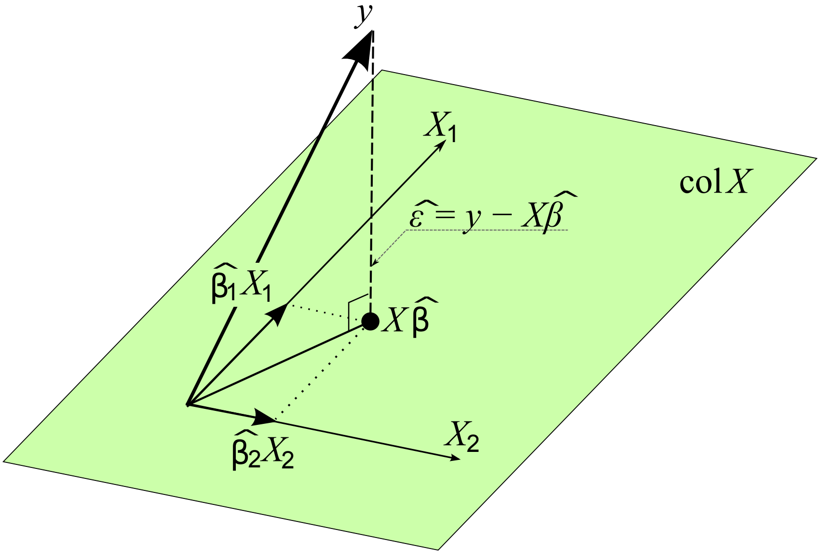



Figure – Souce : Wikipedia

# Least Squares and Uniqueness

Let $\widehat{\boldsymbol{\theta}}$ be a solution of $X^\top X\widehat{\boldsymbol{\theta}} = X^\top \mathbf{y}$.

**Proposition**:Non-uniqueness in OLS occurs when the design matrix $X$ has a non-trivial kernel, i.e.

$$\boxed{\ker(X) \neq \{0\}}.$$

<u>Rem</u>: $\ker(X) = \{\boldsymbol{\theta} \in \mathbb{R}^{p+1} : X\boldsymbol{\theta} = 0\}$

To see this, assume $\boldsymbol{\theta}_K \in \ker(X)$ with $\boldsymbol{\theta}_K \neq 0$. Then

$$X(\widehat{\boldsymbol{\theta}} + \boldsymbol{\theta}_K) = X\widehat{\boldsymbol{\theta}},$$

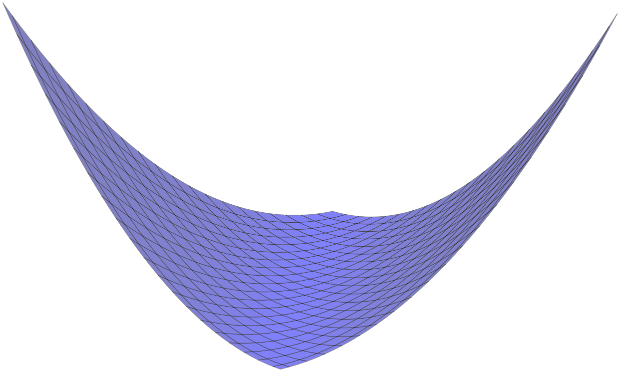
$$(X^\top X)(\widehat{\boldsymbol{\theta}} + \boldsymbol{\theta}_K) = X^\top \mathbf{y}.$$

<u>Conclusion</u> : the set of least squares solutions is an affine subspace :

$$\boxed{\widehat{\boldsymbol{\theta}} + \ker(X)}.$$

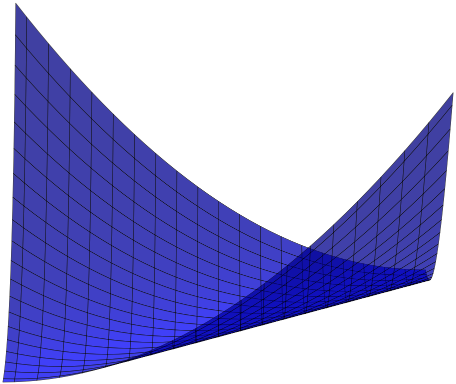
# Optimization in $\mathbb{R}^d$

Convex case, $f(\boldsymbol{\theta}) = \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$, where the set of minimizers is non-unique :



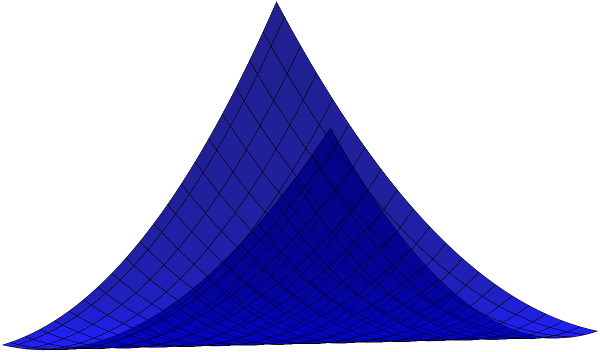
# Optimization in $\mathbb{R}^d$

Convex case, $f(\boldsymbol{\theta}) = \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$, where the set of minimizers is non-unique :



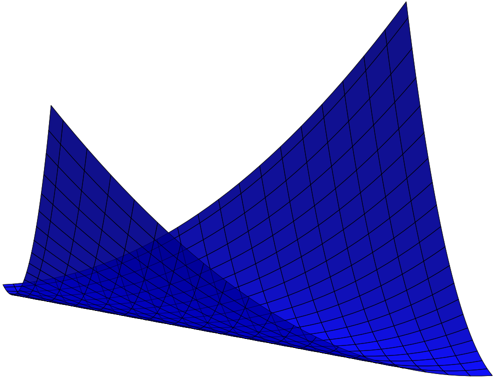
# Optimization in $\mathbb{R}^d$

Convex case, $f(\boldsymbol{\theta}) = \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$, where the set of minimizers is non-unique :



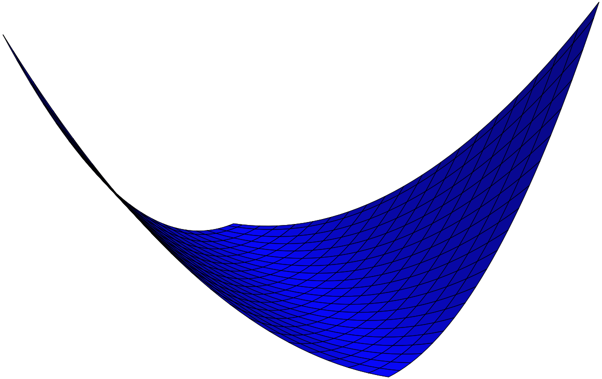
# Optimization in $\mathbb{R}^d$

Convex case, $f(\boldsymbol{\theta}) = \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$, where the set of minimizers is non-unique :



# Optimization in $\mathbb{R}^d$

Convex case, $f(\boldsymbol{\theta}) = \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$, where the set of minimizers is non-unique :



# Interpretation for multivariate cases

Reminder : we write $X = (\mathbf{1}_n, \mathbf{x}_1, \dots, \mathbf{x}_p)$, the features being column-wise (each are of length $n$)

The property $\ker(X) = \{\boldsymbol{\theta} \in \mathbb{R}^{p+1} : X\boldsymbol{\theta} = 0\} \neq \{0\}$ means that there exists a linear dependence between the features $\mathbf{1}_n, \mathbf{x}_1, \dots, \mathbf{x}_p$,

Reformulation : $\exists \boldsymbol{\theta} = (\theta_0, \dots, \theta_p)^\top \in \mathbb{R}^{p+1} \setminus \{0\}$ s.t.

$$\theta_0 \mathbf{1}_n + \sum_{j=1}^{p} \theta_j \mathbf{x}_j = 0$$

# Algebra reminder

**Rank of a matrix :** $\text{rank}(X) = \dim(\text{span}(\mathbf{1}_n, \mathbf{x}_1, \ldots, \mathbf{x}_p))$ ; $\text{span}(\cdot)$ : the space generated by $\cdot$

<u>Property</u> : $\text{rank}(X) = \text{rank}(X^\top)$

Rank–nullity theorem :

- $\text{rank}(X) + \dim(\ker(X)) = p + 1$
- $\text{rank}(X^\top) + \dim(\ker(X^\top)) = n$

<u>Property</u> : $\boxed{\text{rank}(X) \leq \min(n, p + 1)}$

See Golub and Van Loan (1996) for details

# Algebra reminder (continued)

Matrix inversion : A square matrix $A \in \mathbb{R}^{m \times m}$ is invertible

- if and only if its kernel is trivial : $\ker(A) = \{0\}$
- if and only if it is full rank $\text{rank}(A) = m$

OLS is unique iff $X^\top X$ is invertible

$\Leftrightarrow \ker(X^\top X) = \{0\}$

$\Leftrightarrow \ker(X) = \{0\}$

$\Leftrightarrow X$ has full rank

---

**Exo**: $\ker(X) = \ker(X^\top X)$

---

# Non uniqueness : single feature case

Reminder : $$X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$$

If $\ker(X) = \{\boldsymbol{\theta} \in \mathbb{R}^2 : X\boldsymbol{\theta} = 0\} \neq \{0\}$ there exists $(\theta_0, \theta_1) \neq (0, 0)$ :

$$\begin{cases} \theta_0 + \theta_1 x_1 & = 0 \\ \vdots \qquad \vdots & = \vdots \\ \theta_0 + \theta_1 x_n & = 0 \end{cases} \qquad (\star)$$

1. If $\theta_1 = 0$ : $(\star) \Rightarrow \theta_0 = 0$, so $(\theta_0, \theta_1) = (0, 0)$, **contradiction**
2. If $\theta_1 \neq 0$ :
    2.1 If $\forall i, x_i = 0$ then $X = (\mathbf{1}_n, 0)$ and $\theta_0 = 0$
    2.2 Otherwise there exists $x_{i_0} \neq 0$ and $\forall i, x_i = -\theta_0/\theta_1 = x_{i_0}$, *i.e.* $X = [\mathbf{1}_n \quad x_{i_0} \cdot \mathbf{1}_n]$

Interpretation : $\mathbf{x}_1 \propto \mathbf{1}_n$, *i.e.* $\mathbf{x}_1$ is constant

# The determination coefficient $R^2$

The ratio of the variation explained by the model and the total variation of the data

$$R^2 = \frac{\|\widehat{\mathbf{y}} - \bar{\mathbf{y}}\mathbf{1}_n\|^2}{\|\mathbf{y} - \bar{\mathbf{y}}\mathbf{1}_n\|^2}$$

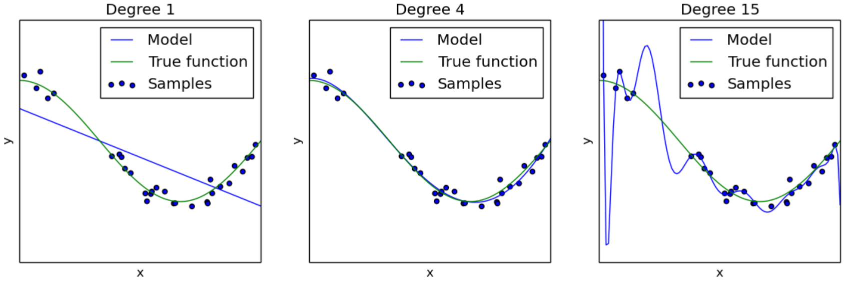
---

**Exo**: Show that $0 \leq R^2 \leq 1$ and

$$R^2 = 1 - \frac{\|\mathbf{y} - \widehat{\mathbf{y}}\|^2}{\|\mathbf{y} - \bar{\mathbf{y}}\mathbf{1}_n\|^2} \tag{3}$$

---

# Polynomial regression and overfitting




Source : sklearn

# References I


B. Delyon.
Régression, 2015.
https://perso.univ-rennes1.fr/bernard.delyon/regression.pdf.

G. H. Golub and C. F. van Loan.
*Matrix computations.*
Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

M. Lejeune.
*Statistiques, la théorie et ses applications.*
Springer, 2010.

W. McKinney.
*Python for Data Analysis : Data Wrangling with Pandas, NumPy, and IPython.*
O'Reilly Media, 2012.