Machine Learning
Regression

Charlotte Pelletier
Univ. Bretagne Sud – IRISA Vannes
Based on C. Friguet's lecture.

22 September 2021

Objectives:

- Understand what are linear and logistic regression models
- Know the principle and the properties of the gradient descent algorithm
- Be able to implement the optimisation algorithm with Python
- Be able to use a regularization technique

Course content

Content

Statistical Modelling

- **Data**: collect, description, analyse and information retrieval
  - ➜ Knowledge extraction from massive amount of data
  - ➜ biology, medical, marketing, geography, psychology, food (security), oceanography, *etc*.
- **Model**: design a simplified model of the observed reality to a controlled-level of approximation
- **Infer**: generalise a result from observations

Statistical Modelling

- **Modelling** the relationship between several variables
  - Explaining a phenomenon, interpreting the relationships between measurements
  - Predicting on new data
- **Target** variable, denoted by y
  - **quantitative** or **qualitative**
  - response variable, variable of interest, variable to be predicted, endogenous variable, dependent variable

Statistical Modelling

- **Modelling** the relationship between several variables
  - Explaining a phenomenon, interpreting the relationships between measurements
  - Predicting on new data
- **Target** variable, denoted by y
  - **quantitative** or **qualitative**
  - response variable, variable of interest, variable to be predicted, endogenous variable, dependent variable
- Explanatory variables, denoted $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_d$
  - quantitative, qualitative or both
  - predictor variables, exogenous variables, independent variables, factors, features
  - $\mathbf{X}_j = \{x_{ij}\}_{i=1}^m$ for j from 1 to d

Statistical Modelling

- **Modelling** the relationship between several variables
  - Explaining a phenomenon, interpreting the relationships between measurements
  - Predicting on new data
- **Target** variable, denoted by y
  - **quantitative** or **qualitative**
  - response variable, variable of interest, variable to be predicted, endogenous variable, dependent variable
- Explanatory variables, denoted $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_d$
  - quantitative, qualitative or both
  - predictor variables, exogenous variables, independent variables, factors, features
  - $\mathbf{X}_j = \{x_{ij}\}_{i=1}^m$ for j from 1 to d
- The analysis of a relationship between y and $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_d$ consists in defining a function f such that for each observation i :

$$y_i \approx f(x_{i1}, x_{i2}, \cdots, x_{id})$$

Data

- **Supervised learning**: we know the "true" value of the predictor variable, and we look for understanding/predicting the (presumed) relationship between explanatory variables and the target variable
- **What is the type of the explanatory variable** (Y) **?**
  - **quantitative**: **regression**
  - qualitative (2 or $>$2 modes): classification (binary / multiclasses)
- **What is the type and the number of explanatory variables** (X) **?**
  - type: **qualitative** and/or **quantitative**
  - **One** variable
    - Not frequent in practice, but it useful to understand how works the method $\Rightarrow$ visualisation
  - **Several** variables
    - Several = from a dozen to thousands $\Rightarrow$ variable selection

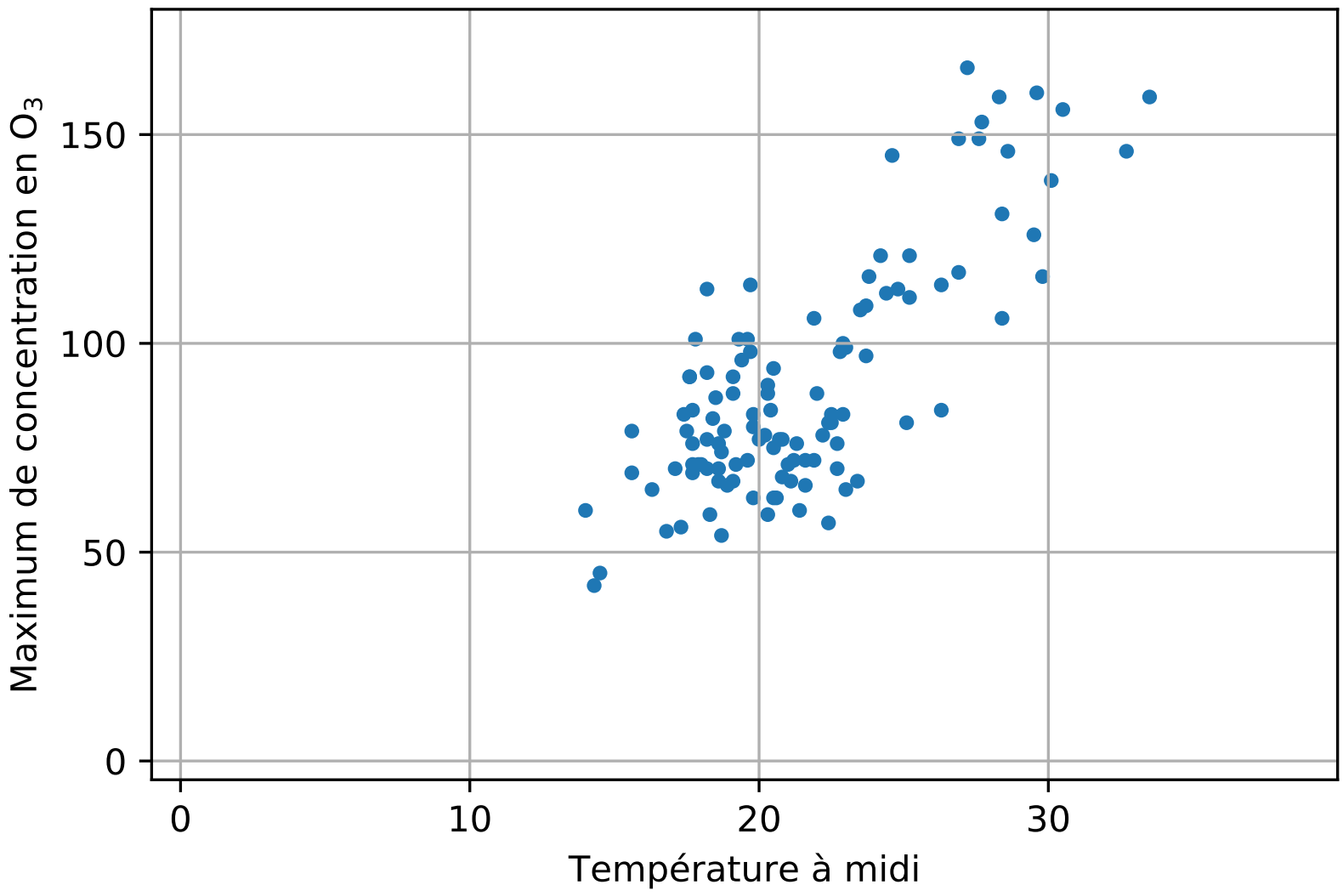
Exemple A

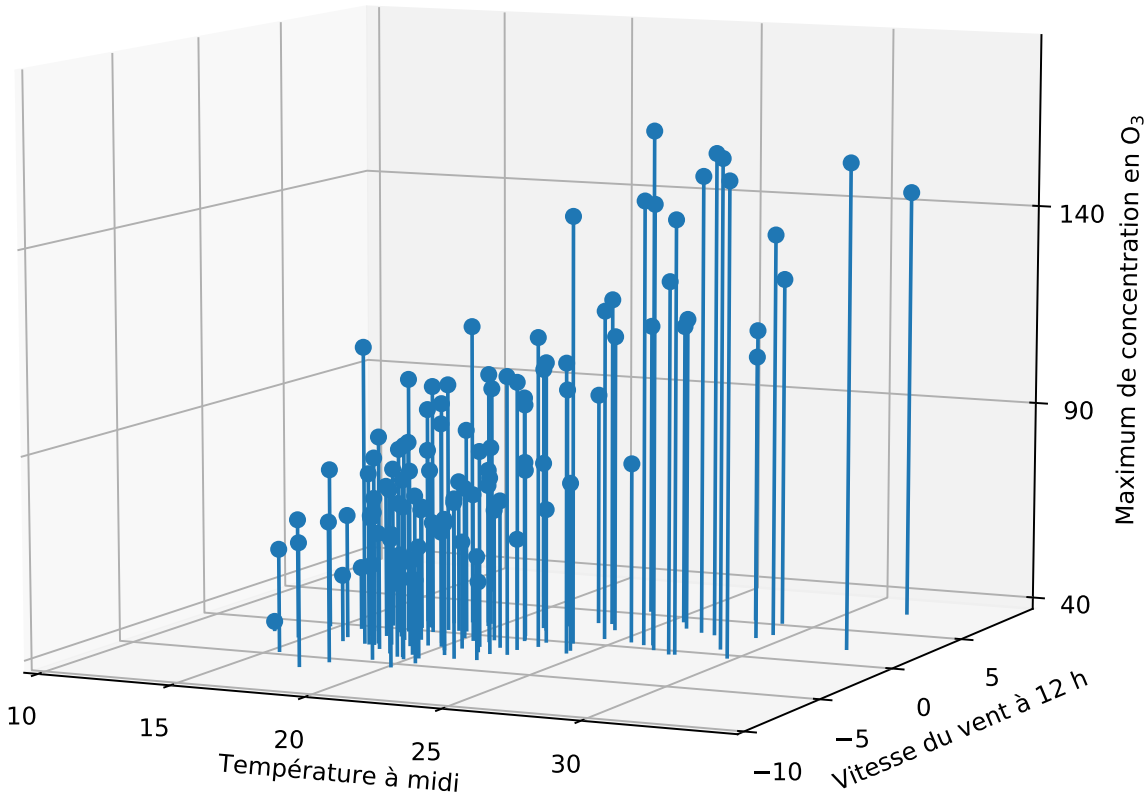
For public health reasons, we are interested in the concentration of ozone O_3 in the air. In particular, we want to know if we can explain the maximum ozone level for the day (in $\mu g/ml$) from other meteorological variables measured in the Rennes station.

Data subset:

O_3 max	Temperature at midday	Wind speed at midday	$\cdots$
87	18.5	-1.7101	$\cdots$
82	18.4	-4.0000	$\cdots$
92	17.6	1.8794	$\cdots$
114	19.7	0.3473	$\cdots$
94	20.5	-2.9544	$\cdots$
80	19.8	-5.0000	$\cdots$
79	15.6	-1.8794	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Air Breizh data (2001) are extracted fromRgression: Thorie et applications, Cornillon P.A. et Matzner-Lober E. (2006) Springer


Maximum de concentration en O_3
150
100
50
0
0
10
20
30
Température à midi


Maximum de concentration en O_3
140
90
40
10
15
20
25
30
Température à midi
−10
−5
0
5
Vitesse du vent à 12 h

Content

Notations

- Training data: $\{\mathbf{x}_i, y_i\}_{i=1}^m$
  - observations (inputs): $\mathbf{x}_i \in \mathbb{R}^d$
  - measure of interest (target variable): $y_i \in \mathcal{Y}$
- Prediction function: $f : \mathbb{R}^d \mapsto \mathcal{Y}$
  - regression: f predicts a real ($\mathcal{Y} = \mathbb{R}$)
  - multiclass classification : f predicts an integer between 1 and k ($\mathcal{Y} = \{1, ..., k\}$)

Notations

- Training data: $\{\mathbf{x}_i, y_i\}_{i=1}^m$
  - observations (inputs): $\mathbf{x}_i \in \mathbb{R}^d$
  - measure of interest (target variable): $y_i \in \mathcal{Y}$
- Prediction function: $f : \mathbb{R}^d \mapsto \mathcal{Y}$
  - **regression: f predicts a real ($\mathcal{Y} = \mathbb{R}$)**
  - multiclass classification : f predicts an integer between 1 and k ($\mathcal{Y} = \{1, ..., k\}$)

Matrix Notation

Data are usually represented by matrices

- Explanatory variables (observations in row and variables in columns). The following notations are equivalent:
  - $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_m]^T$ where $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th observation
  - $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_j, \cdots, \mathbf{X}_d]$ where $\mathbf{X}_j \in \mathbb{R}^m$ is the value of the j-th variable
- target variable: $\mathbf{Y} = [y_1, y_2, \cdots, y_i, \cdots, y_m]$

Matrix Notation

Data are usually represented by matrices

- Explanatory variables (observations in row and variables in columns). The following notations are equivalent:
  - $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_m]^T$ where $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th observation
  - $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_j, \cdots, \mathbf{X}_d]$ where $\mathbf{X}_j \in \mathbb{R}^m$ is the value of the j-th variable
- target variable: $\mathbf{Y} = [y_1, y_2, \cdots, y_i, \cdots, y_m]$

$$\begin{matrix} \mathbf{Y} \\ \\ \\ \\ \\ \\ \\ \end{matrix} \quad \begin{matrix} \mathbf{X} \\ \\ \\ \\ \\ \\ \\ \end{matrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_m \end{pmatrix} \quad \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{id} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mj} & \dots & x_{md} \end{pmatrix}$$

Model

- A model is a mathematical equation whose the goal is to describe a relationship between a target variable $\mathbf{Y} = [y_1, y_2, \cdots, y_m]$ and explanatory variables $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]^T$ (observations in row, variables in columns) :

$$y_i \approx f(\mathbf{x}_i) = f(x_{i1}, x_{i2}, \cdots, x_{id})$$

  with $x_{ij} \in \mathbb{R}$ the value of the variable j for the observation i
- The form of f depends on the context (data type) and on the desired level of simplification
  - We select a type of function: linear, polynomial, *etc*
  - Case of the linear model:

$$y_i \approx f_{\boldsymbol{\beta}}(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_d x_{id}$$

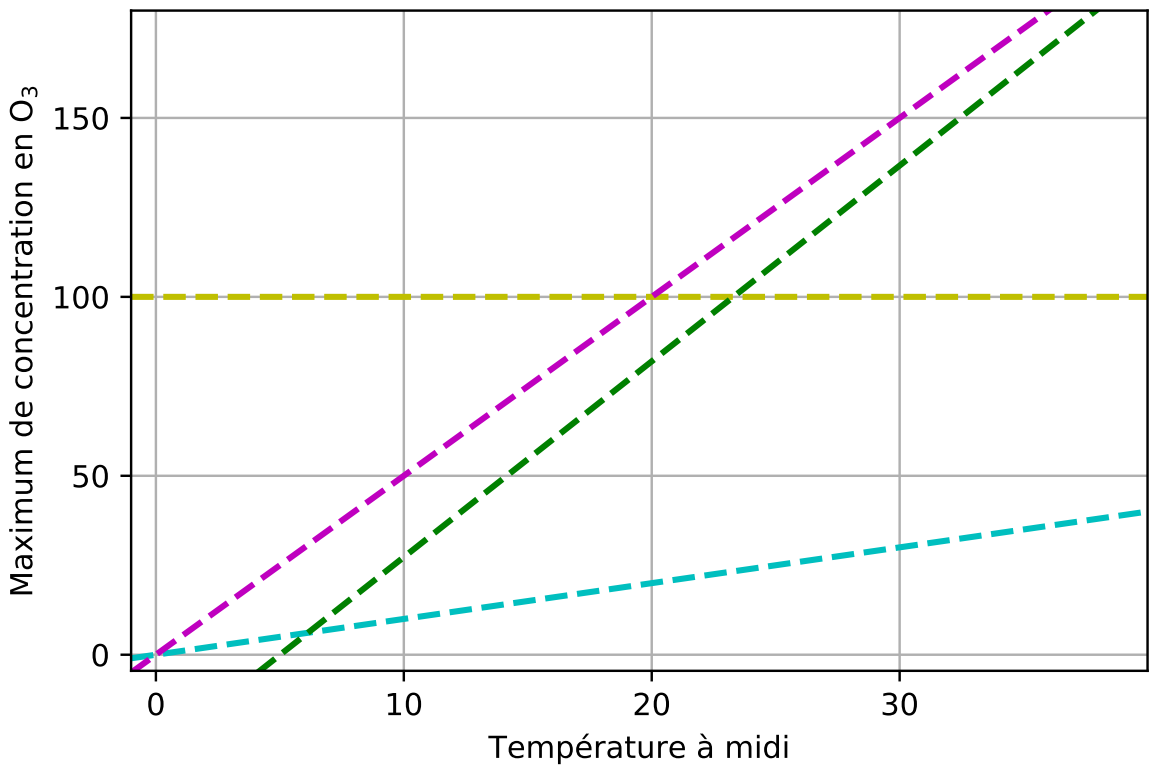
- Generally speaking, a model is defined from parameters (β_j)
  - they are unknown
  - they need to be estimated
  - but, how?

Univariate analysis

- only one explanatory variable: $f_{\boldsymbol{\beta}}$ is an affine function (a straight line)

$$f_{\boldsymbol{\beta}}(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1}$$

  - β_0 is the intercept
  - β_1 is the slope

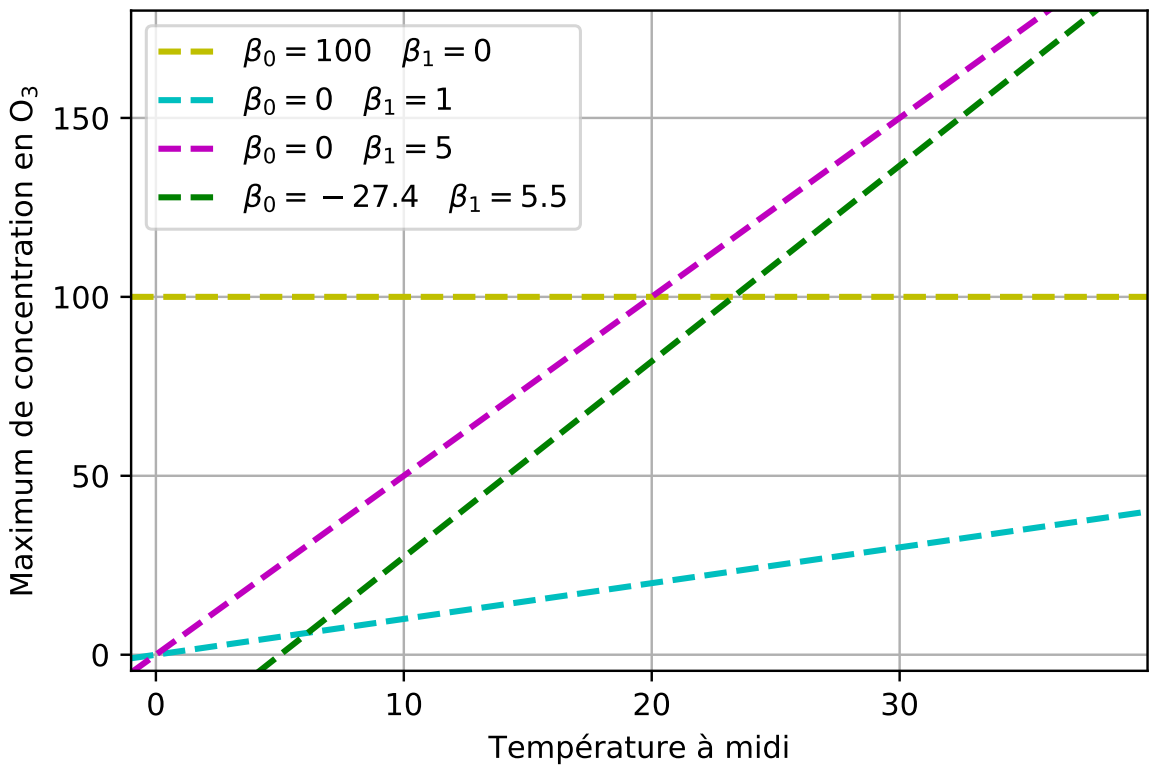


Univariate analysis

- only one explanatory variable: $f_{\boldsymbol{\beta}}$ is an affine function (a straight line)

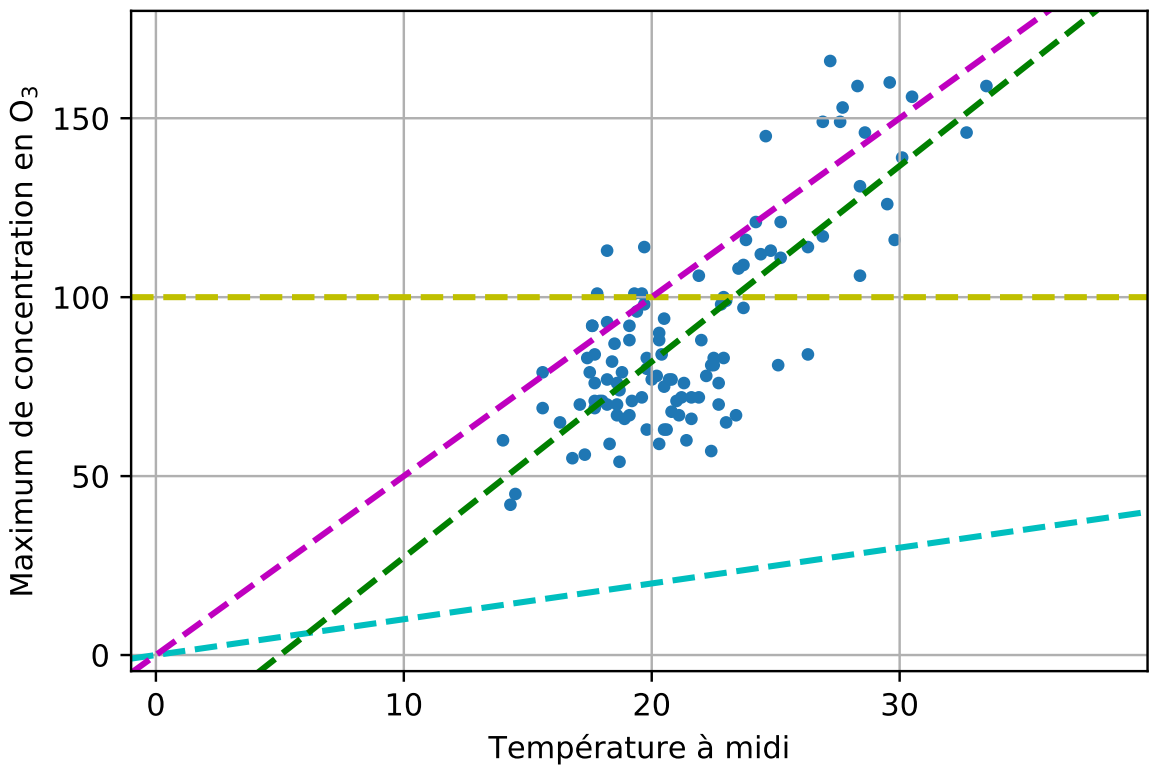
$$f_{\boldsymbol{\beta}}(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1}$$

  - β_0 is the intercept
  - β_1 is the slope



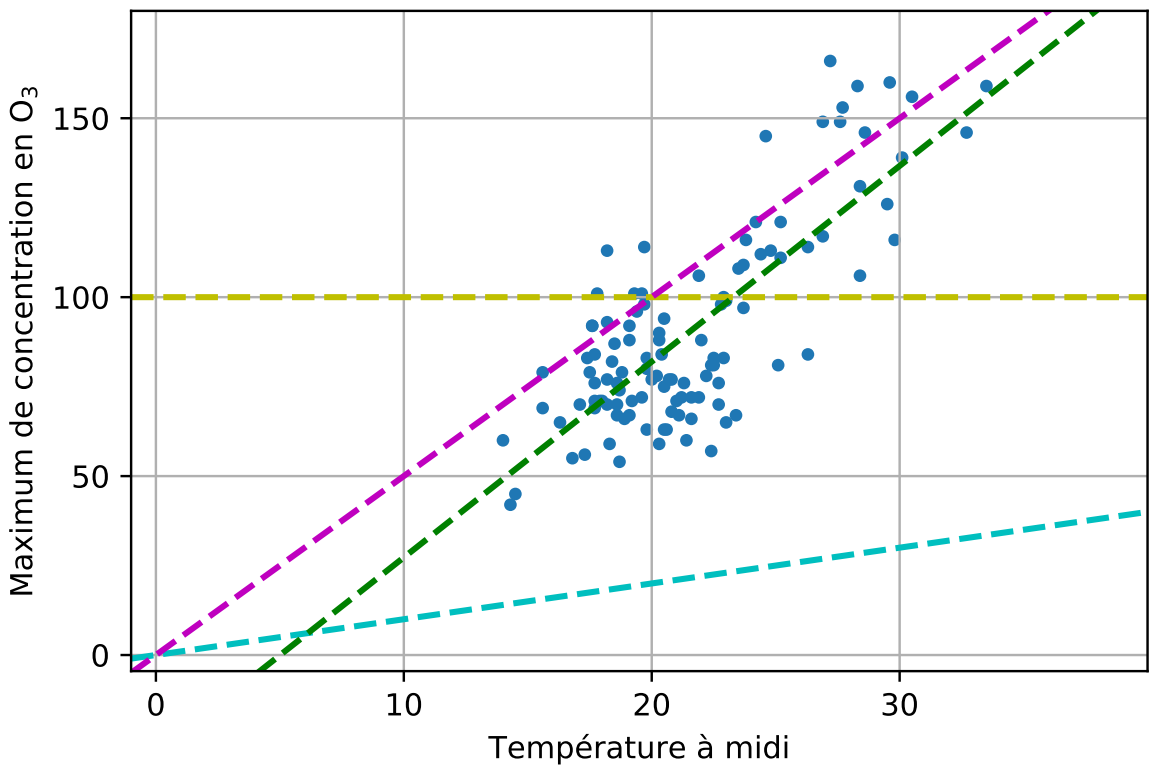
Univariate analysis

- How to select $\boldsymbol{\beta} = (\beta_0, \beta_1)$?



Univariate analysis

- How to select $\boldsymbol{\beta} = (\beta_0, \beta_1)$?
  - We search for $\boldsymbol{\beta}$ such that $f_{\boldsymbol{\beta}}(\mathbf{x})$ is closed from y for **all** the training data $\{\mathbf{x}_i, y_i\}_{i=1}^m$.



Cost function

- **Cost function** (or loss function or objective function or error function)
  - It is a function that measures the quality of a particular set of parameters based on how well a model performs on a given task,
  - used to guide the training process,
  - needs to be minimised (or maximised),
  - denoted by $\mathbf{J}(\boldsymbol{\beta})$.
- We search for $\boldsymbol{\beta}$ such that $f_{\boldsymbol{\beta}}(\mathbf{x})$ is closed from y for **all** the training data $\{\mathbf{x}_i, y_i\}_{i=1}^m$:

$$\hat{y}_i = f_{\boldsymbol{\beta}}(\mathbf{x}_i) \approx y_i \quad \forall i \in \{1, \cdots, m\}$$

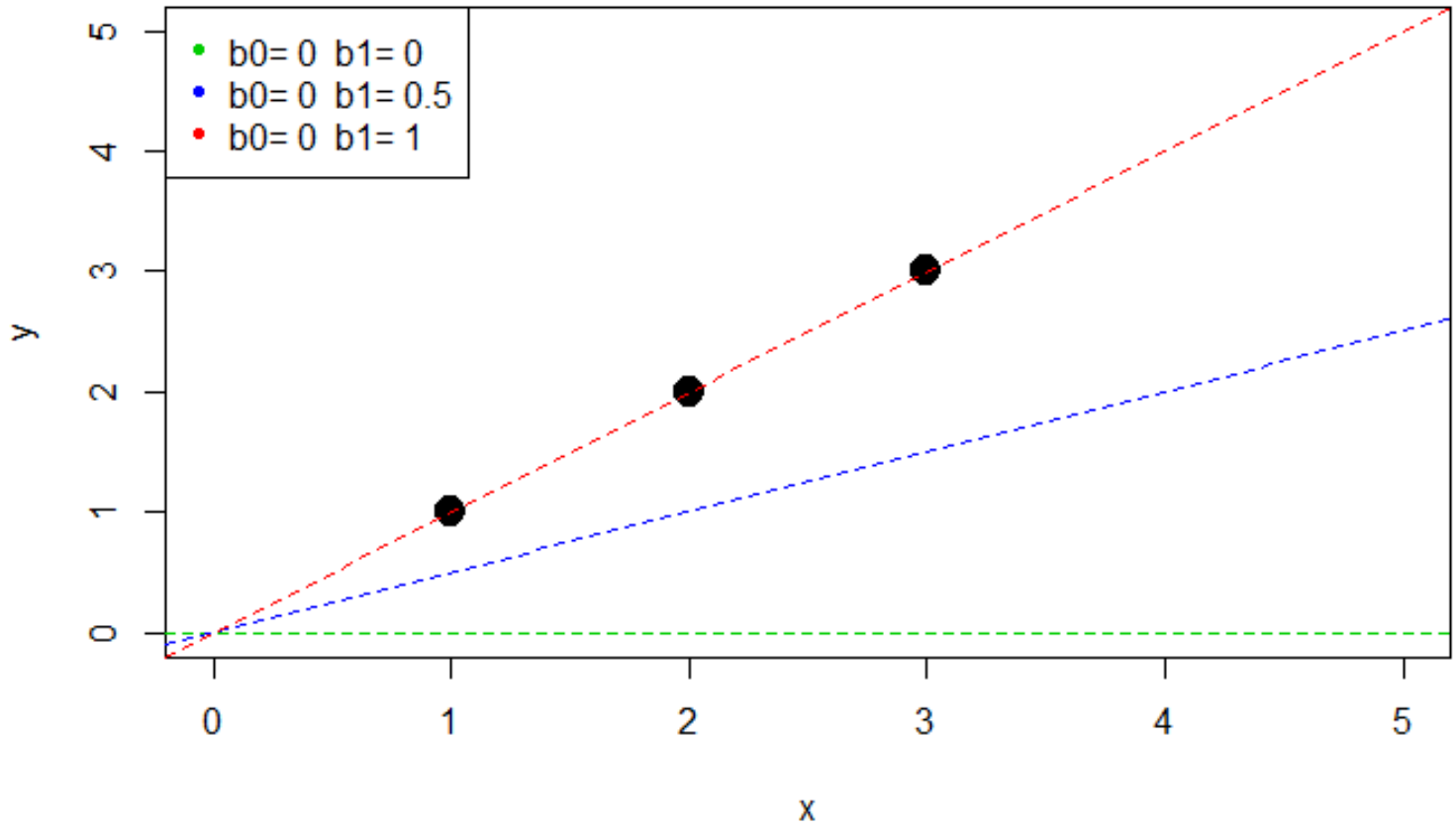
- Finding the best (β_0, β_1) is equivalent to minimise the (quadratic) global cost (*i.e.*, the squarred sum of errors):

$$\mathbf{J}(\boldsymbol{\beta}) = \frac{1}{2m} \sum_{i=1}^{m} \Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i) - y_i \Big)^2$$

➜ The best (β_0, β_1) is given by:

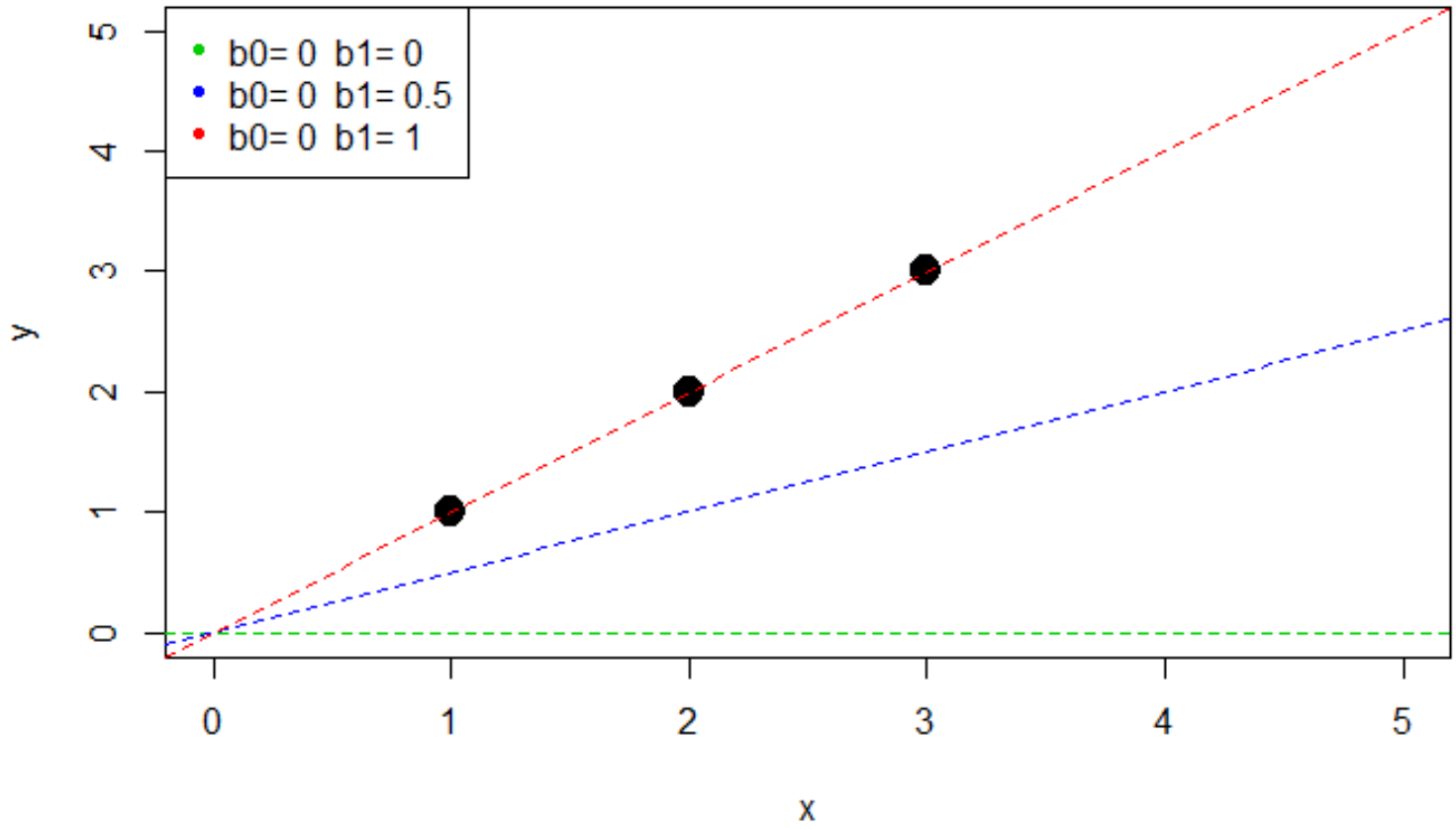
$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big(\mathbf{J}(\boldsymbol{\beta}) \Big)$$

Among all the possible lines, we search for the one that minimises the squared sum of errors over the training data.



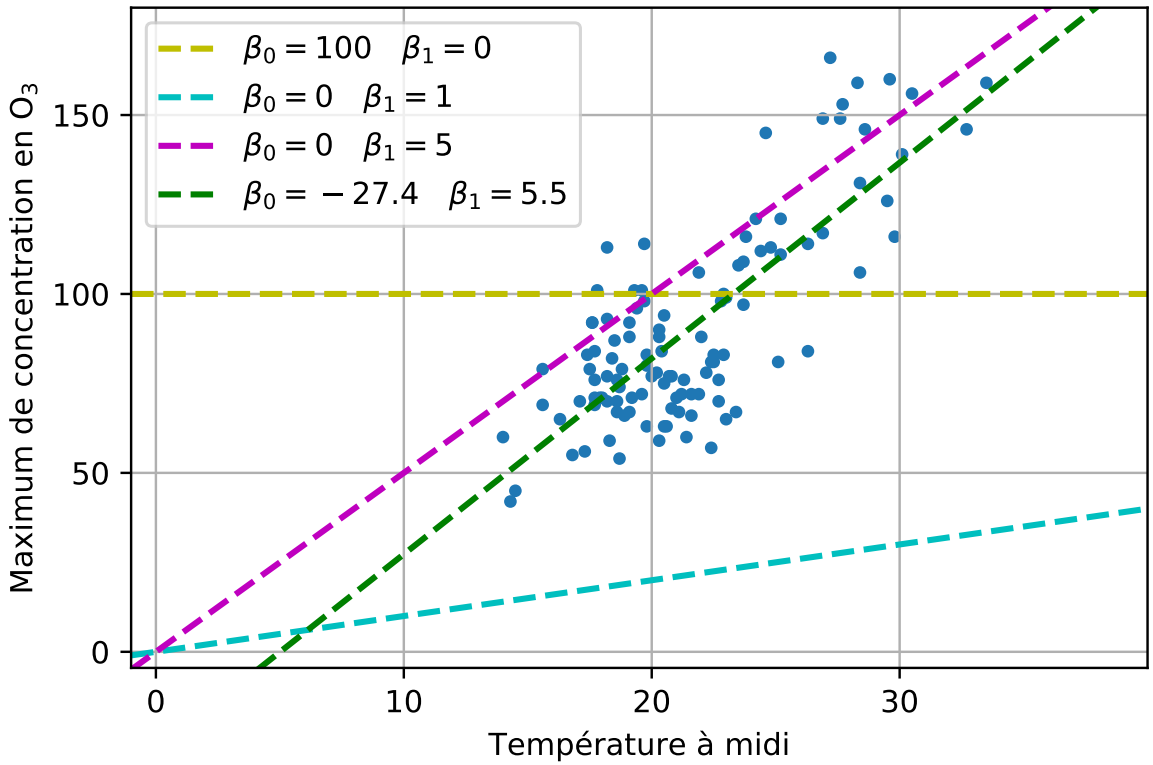
- How to compute $\mathbf{J}(\beta)$?

$\boldsymbol{\beta}$	(0,0)	(0,0.5)	(0,1)
$\mathbf{J}(\boldsymbol{\beta})$			



- How to compute $\mathbf{J}(\beta)$?

$\boldsymbol{\beta}$	(0,0)	(0,0.5)	(0,1)
$\mathbf{J}(\boldsymbol{\beta})$	2.33	0.58	0



β	(100,0)	(0,1)	(0,5)	(27.42,5.47)
$J(\beta)$	440.72	2678.37	303.50	151.80

Content

Principle

- Objective: find the minimum of a cost function
- Principle: iterative algorithm
  1. initialisation: $\beta^{(0)}$
  2. at each step k, modify $\boldsymbol{\beta}^{(k-1)}$ so as to decrease $\mathbf{J}(\boldsymbol{\beta}^{(k)})$
  3. end the process when a minimum is reached

Iteration k of the gradient descent algorithm

For the parameter β_j

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} \mathbf{J}(\boldsymbol{\beta}^{(k-1)})$$

avec:

- $\frac{\partial}{\partial \beta_j}$:
- α:

Principle

- Objective: find the minimum of a cost function
- Principle: iterative algorithm
  1. initialisation: $\beta^{(0)}$
  2. at each step k, modify $\boldsymbol{\beta}^{(k-1)}$ so as to decrease $\mathbf{J}(\boldsymbol{\beta}^{(k)})$
  3. end the process when a minimum is reached

Iteration k of the gradient descent algorithm

For the parameter β_j

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} \mathbf{J}(\boldsymbol{\beta}^{(k-1)})$$

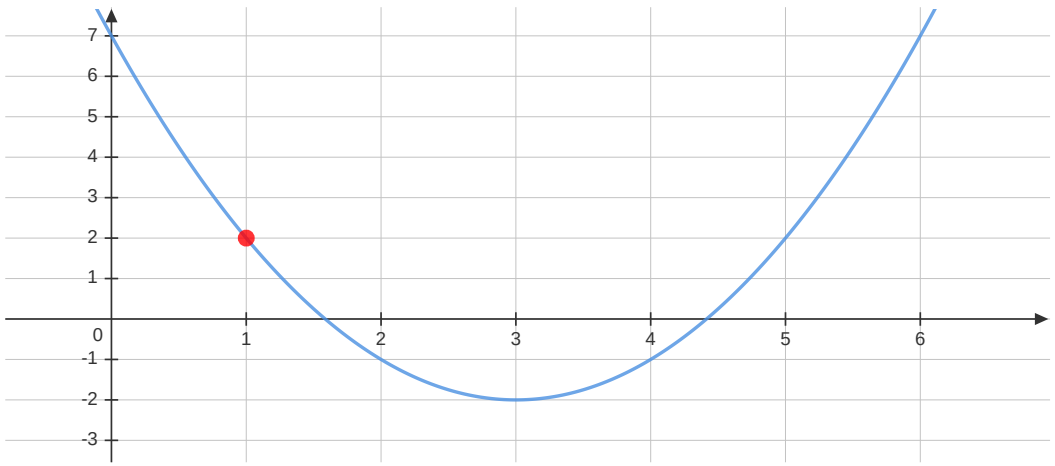
avec:

- $\frac{\partial}{\partial \beta_j}$: partial derivative
- α: learning rate

A toy example

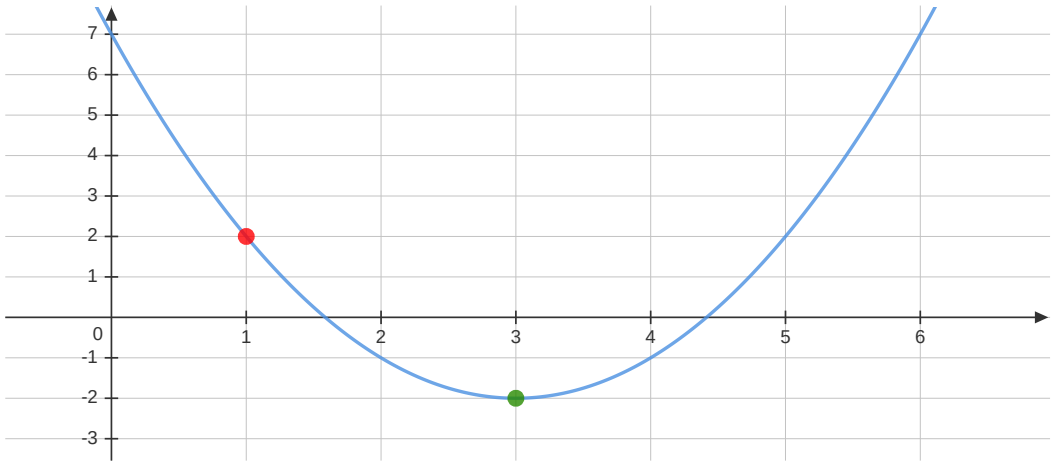
Let us perform the two first steps of the gradient descent algorithm

- $\mathbf{J}(\beta) = (\beta - 3)^2 - 2$
- initialisation: $\beta^0 = 1$
- learning rate: $\alpha = 0.25$



A toy example

If we continue we will converge to the green data point that corresponds to the global minimum of the function **J**.



Notes and properties

- Derivative sign: increase or decrease of $\boldsymbol{\beta}^{(k)}$
- Convergence criteria: decrease of $\mathbf{J}(\beta) < \varepsilon$ during an iteration (*e.g.*, $\varepsilon = 10^{-3}$)
- How to select the value of α?
  - If too small, then the algorithm is too slow.
  - If too big, then the algorithm might overshoot.
  - In practice, we test several values.
  - The gradient will decrease when it is close to the minimum
    one possible improvement of the gradien descent algorithm = big steps at the beginning, then small steps when we are close to the minimum
- Initialisation: issue if close to a local minimum
- Scale of X_j should be similar: no to diverge / converge faster
  - **Normalisation** (mean-centring data)
    $\forall j \in \{1, \cdots, d\}$): $\mathbf{X}_j := \frac{\mathbf{x}_j - \bar{\mathbf{x}}_j}{r_j}$, where $\bar{\mathbf{X}}_j$ is the mean of $\mathbf{X}_j$, and r_j the standard deviation.

Simple Linear Regression

- Analysing the relationship between $\mathbf{Y}$ and $\mathbf{X}_1$ with f a linear function such that $\forall i \in \{1, \cdots, m\}$:

$$y_i \approx f_{\boldsymbol{\beta}}(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1}$$

- Iteration k of the gradient descent algorithm?

- Analysing the relationship between $\mathbf{Y}$ and $\mathbf{X}_1$ with f a linear function such that $\forall i \in \{1, \cdots, m\}$:

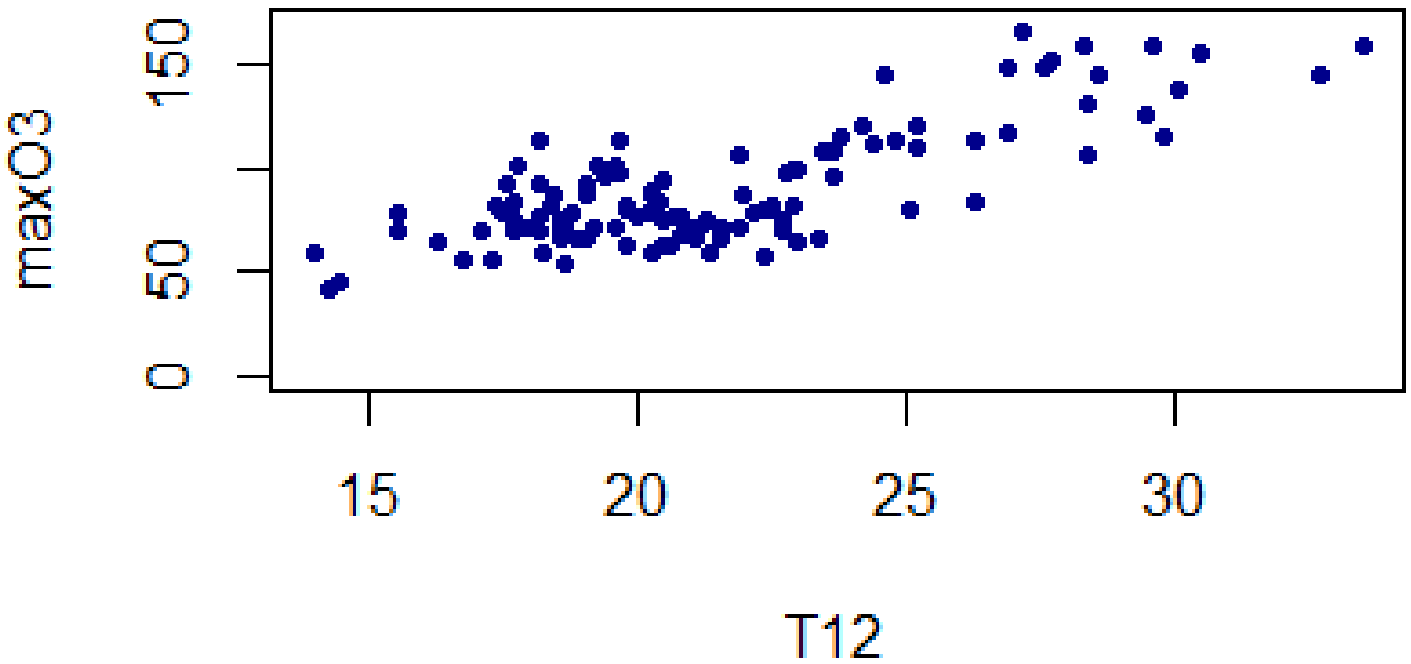
$$y_i \approx f_{\boldsymbol{\beta}}(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1}$$

- Iteration k of the gradient descent algorithm?

Iteration k of the gradient descent algorithm - simple linear regression

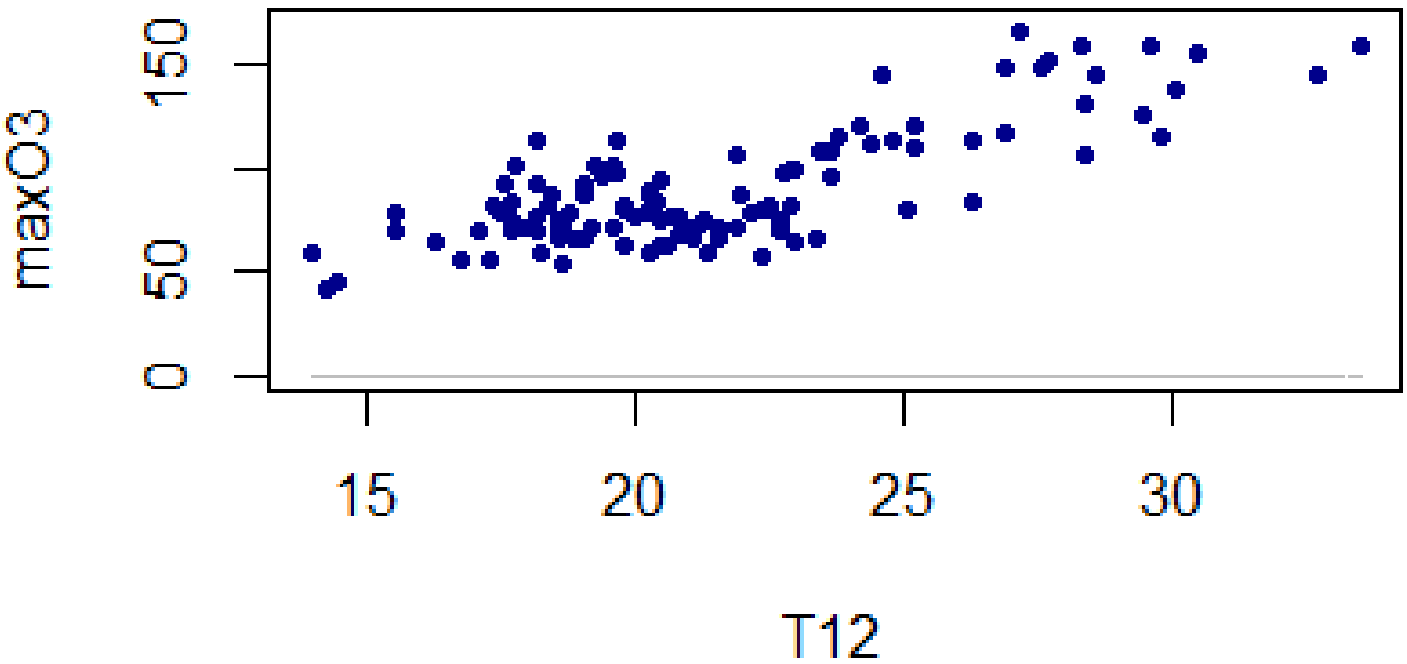
$$\begin{cases} \beta_0^{(k)} := \beta_0^{(k-1)} - \alpha \dfrac{1}{m} \displaystyle\sum_{i=1}^{m} \left(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i \right) \\ \beta_1^{(k)} := \beta_1^{(k-1)} - \alpha \dfrac{1}{m} \displaystyle\sum_{i=1}^{m} \left(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i \right) x_{i1} \end{cases}$$

Simple Linear Regression



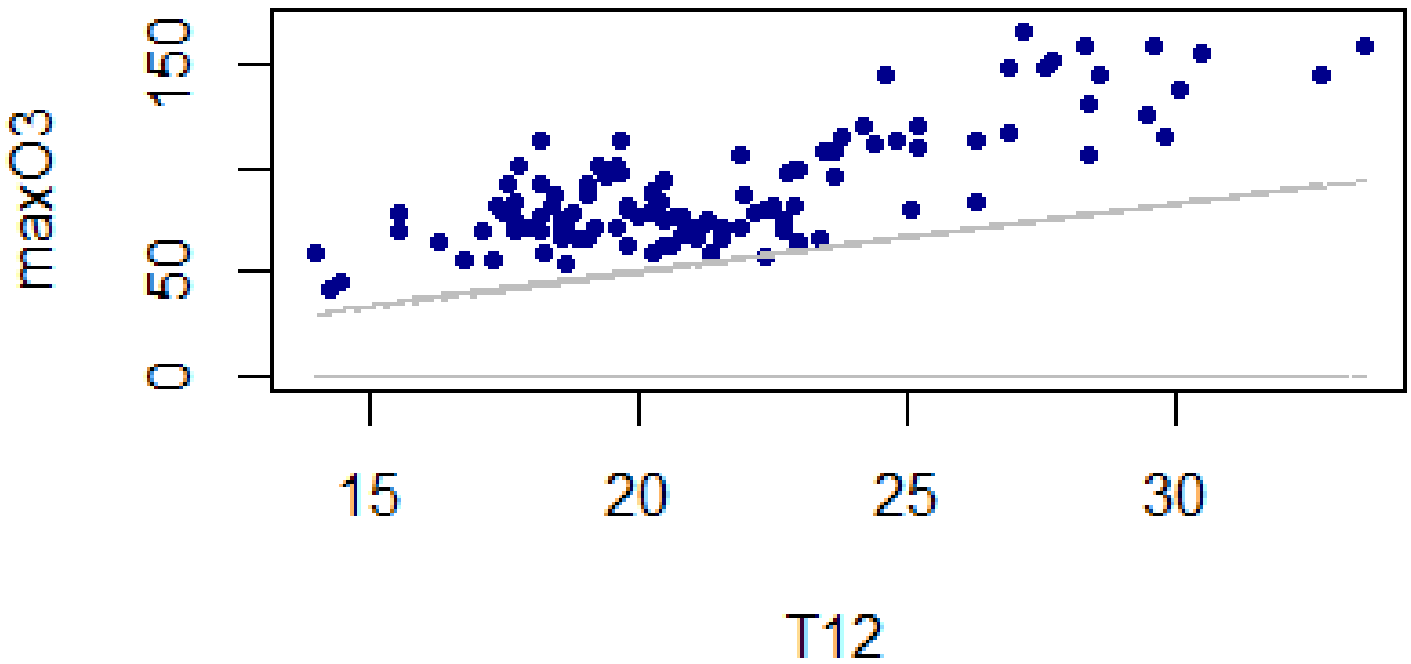
it. β	1	10	20	30	40	50	100
$J(\beta)$							

Simple Linear Regression



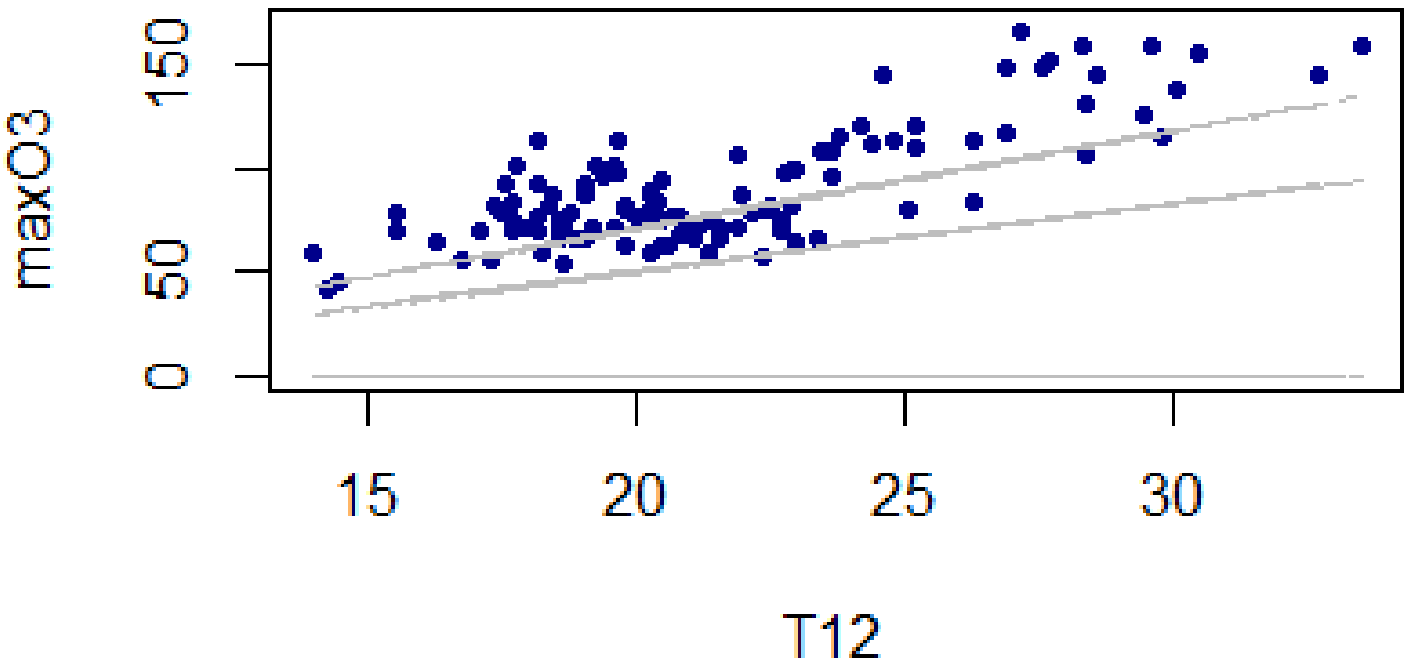
it. β	1 (0,0)	10	20	30	40	50	100
$J(\beta)$	3650.76						

Simple Linear Regression



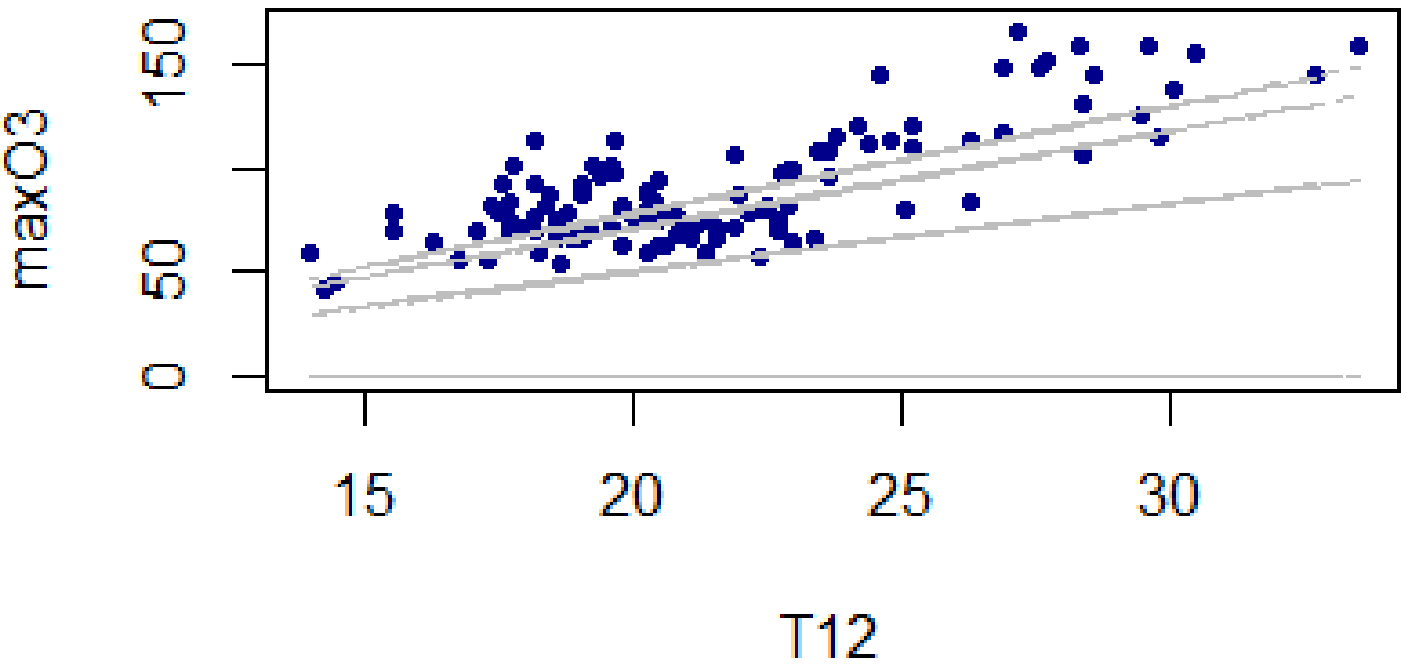
it.	1	10	20	30	40	50	100
β	(0,0)	(-16.39,3.33)					
$J(\beta)$	3650.76	677.30					

Simple Linear Regression



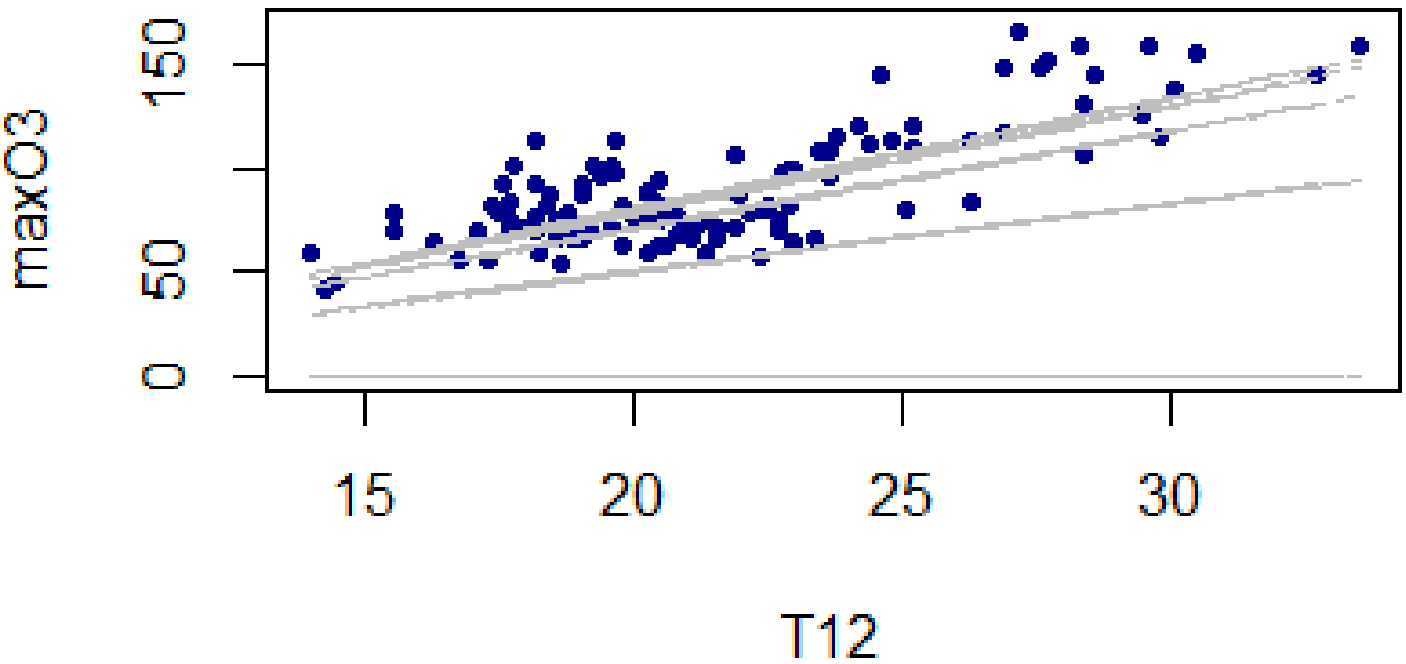
it.	1	10	20	30	40	50	100
β	(0,0)	(-16.39,3.33)	(-23.41,4.72)				
$J(\beta)$	3650.76	677.30	215.54				

Simple Linear Regression



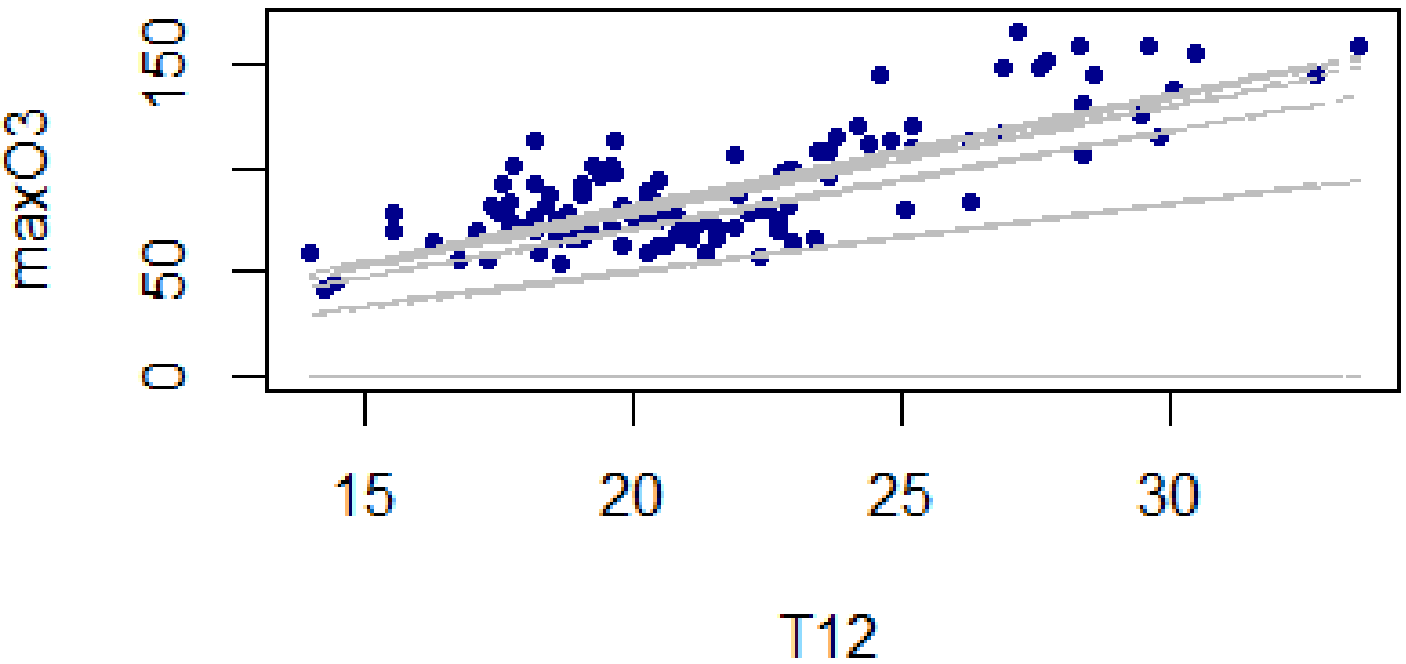
it.	1	10	20	30	40	50	100
β	(0,0)	(-16.39,3.33)	(-23.41,4.72)	(-25.97,5.20)			
$J(\beta)$	3650.76	677.30	215.54	159.34			

Simple Linear Regression



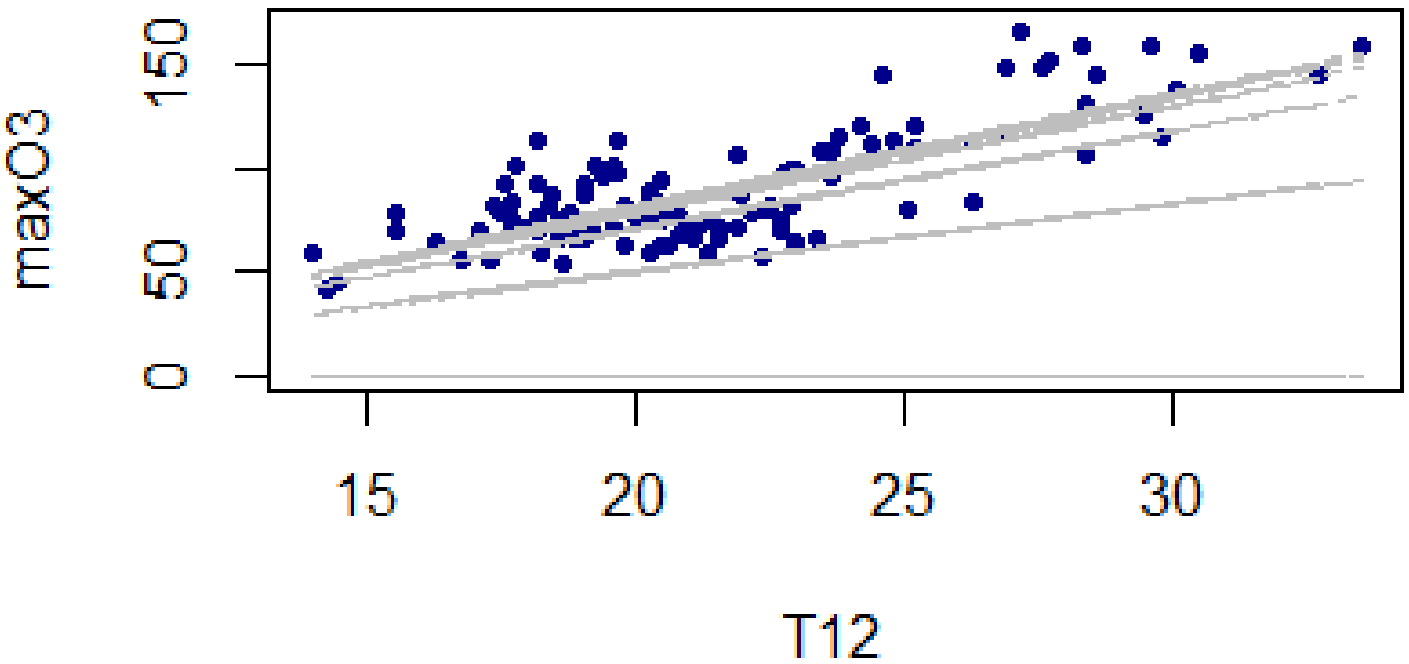
it.	1	10	20	30	40	50	100
β	(0,0)	(-16.39,3.33)	(-23.41,4.72)	(-25.97,5.20)	(-26.89,5.38)		
$J(\beta)$	3650.76	677.30	215.54	159.34	152.50		

Simple Linear Regression

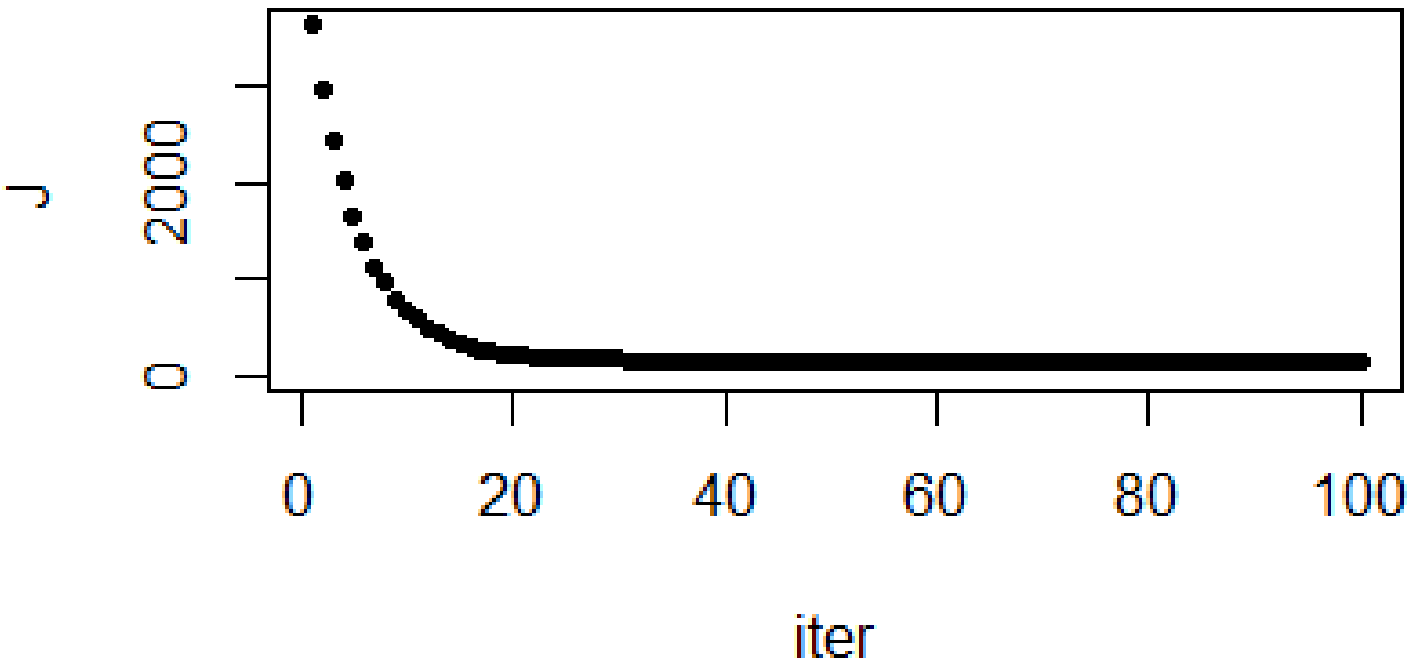


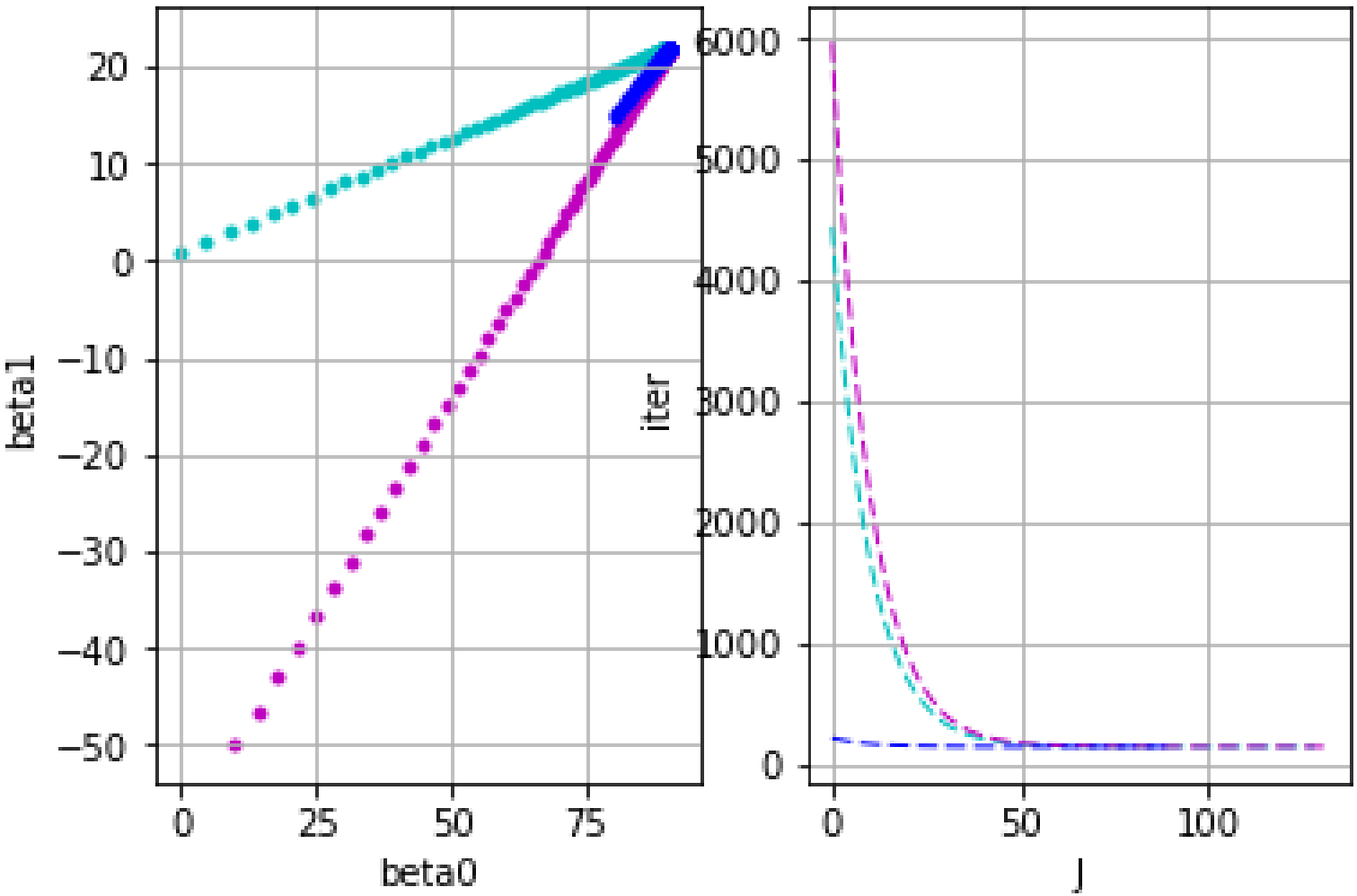
it.	1	10	20	30	40	50	100
β	(0,0)	(-16.39,3.33)	(-23.41,4.72)	(-25.97,5.20)	(-26.89,5.38)	(-27.23,5.44)	
$J(\beta)$	3650.76	677.30	215.54	159.34	152.50	151.67	

Simple Linear Regression



it.	1	10	20	30	40	50	100
β	(0,0)	(-16.39,3.33)	(-23.41,4.72)	(-25.97,5.20)	(-26.89,5.38)	(-27.23,5.44)	(-27.42,5.47)
$J(\beta)$	3650.76	677.30	215.54	159.34	152.50	151.67	151.55


J
0
2000
0
20
40
60
80
100
iter


beta1
beta0
iter
J
20
10
0
−10
−20
−30
−40
−50
0
25
50
75
6000
5000
4000
3000
2000
1000
0
0
50
100

Multiple Linear Regression

- Analysing the relationship between $\mathbf{Y}$ and all the explanatory variables $[\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_d]$ with a linear function f such as $\forall i \in \{1, \cdots, m\}$:

$$y_i \approx f_{\boldsymbol{\beta}}(\mathbf{x}_i) = f_{\boldsymbol{\beta}}(x_{i1}, x_{i2} \cdots, x_{id}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_d x_{id}$$

- Matrix notation:

$$f_{\boldsymbol{\beta}}(\mathbf{X}) \approx \tilde{\mathbf{X}}\boldsymbol{\beta}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \\ y_m \end{pmatrix} \approx \begin{pmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1d} \\ 1 & x_{21} & x_{22} & \ldots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \\ \\ 1 & x_{m1} & x_{m2} & \ldots & x_{md} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \\ \beta_d \end{pmatrix}$$

where $\mathbf{X} \in \mathbb{R}^{m \times d}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{m \times (d+1)}$ et $\boldsymbol{\beta} \in \mathbb{R}^{d+1}$

- Iteration k of the gradient descent algorithm?

- Iteration k of the gradient descent algorithm?

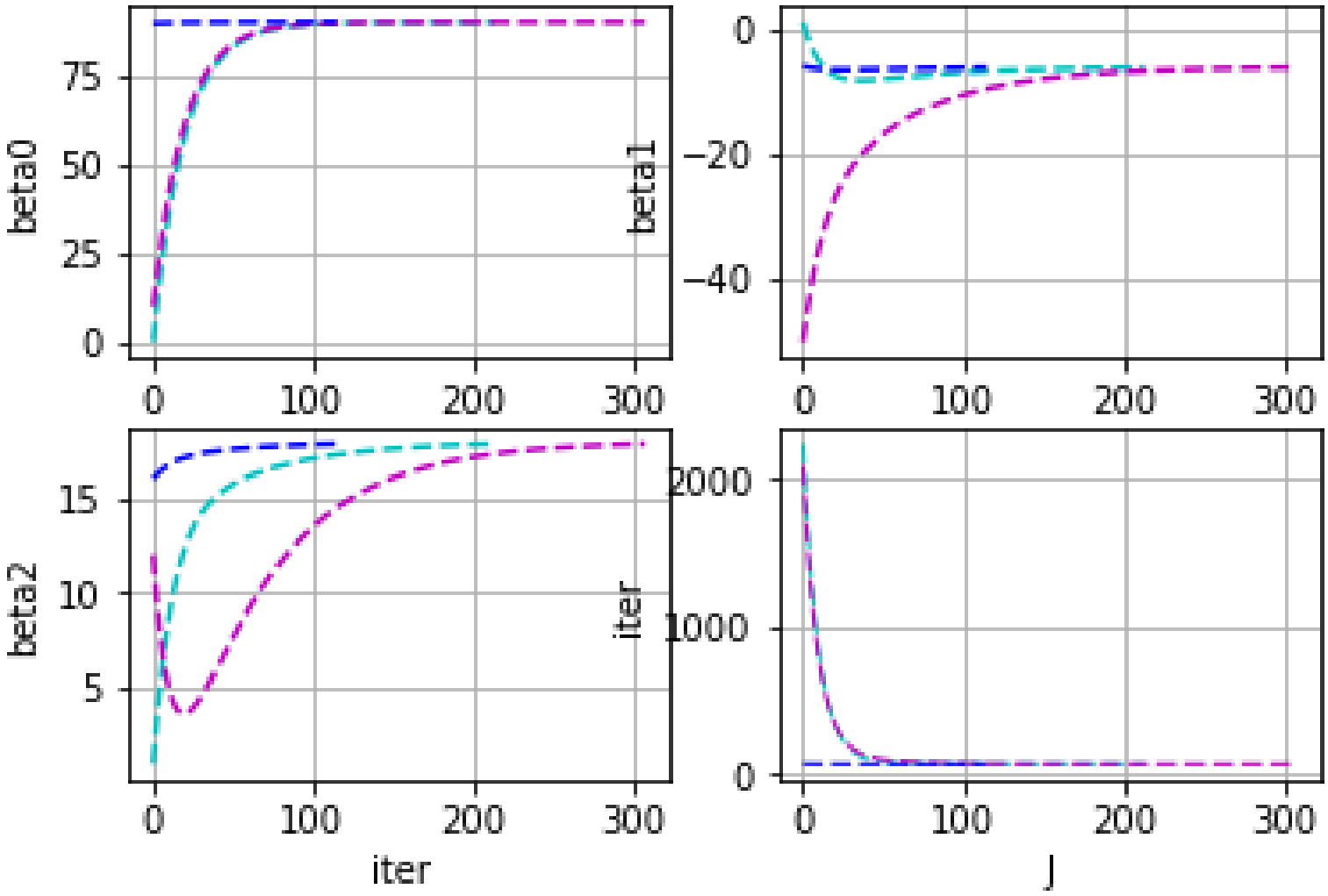
Iteration k of the gradient descent algorithm - multiple linear regression

For each parameter β_j :

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i \right) x_{ij}$$

Note: $\forall i, x_{i0} = 1$

Multiple Linear Regression



To wrap-up

- Iterative optimisation algorithm
  - at each iteration, we decrease the cost function in the opposite direction of the gradient?
- ⚠ local/global minimum $\Rightarrow$ requires several initialisations
- Learning rate α fixed
  - possible improvement with some variants (for example conjugated gradient, BFGS[1]),
  - but these variants are more complex
- Other models (hypotheses), other cost function

[1] Broyden-Fletcher-Goldfarb-Shanno

Content

In the case of the linear regression, the cost function is always convex $\Rightarrow$ there exists a global minimum.

In the case of the linear regression, the cost function is always convex $\Rightarrow$ there exists a global minimum.

To find this minimum, it is required to find the point where the gradient of the function is zero :

$$\frac{\partial \mathcal{J}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0$$

We can demonstrate for the linear regression problem that this minimum is reached for :

$$\begin{aligned} \beta_0 &= \bar{y} - \beta 1 \bar{x} \\ \beta_1 &= \frac{cov(x, y)}{var(x)} \end{aligned}$$

Ordinary Least Squares

- **Least Squares**: minimisation of *squared differences* between observations and the model

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(y_i - f_{\boldsymbol{\beta}}(\mathbf{x}_i) \right)^2 &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_d x_{id}) \right)^2 \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, ||\mathbf{Y} - \tilde{\mathbf{X}}\boldsymbol{\beta}||^2 \end{aligned}$$

Gauss-Markov's Theorem

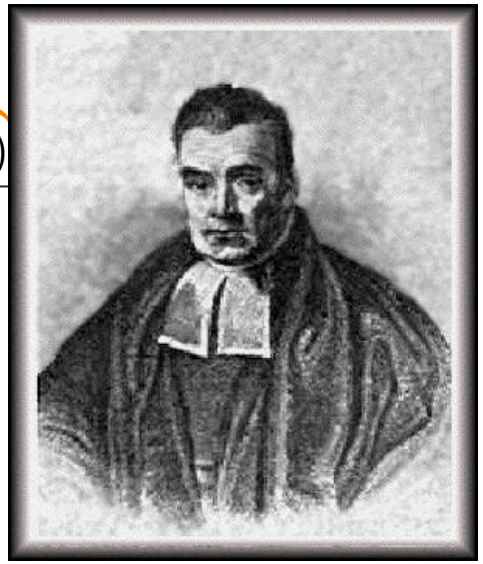
- **Hypotheses** : errors are centred (the expected value is zero), non-correlated and of the same variance (homoscedasticity)
- **Explicit solution** for $\boldsymbol{\beta}^*$:

$$\boldsymbol{\beta}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

- It is the best unbiased estimator.

Bayes's theorem

$$\underbrace{\Pr(Y = y|X = \mathbf{x})}_{\textit{A posteriori} \text{ probability}} = \frac{\overbrace{\Pr(Y = y)}^{\textit{A priori} \text{ probability}} \cdot \overbrace{\Pr(X = \mathbf{x}|Y = y)}^{\text{Likelihood}}}{\Pr(X = \mathbf{x})}$$

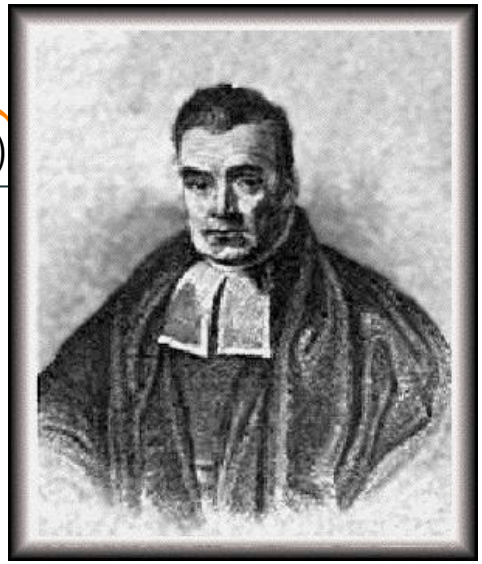


where

- $\mathbf{x} \in \mathbb{R}^d$ an observation
- $y \in Y$ a label

Bayes's theorem

$$\underbrace{\Pr(Y = y|X = \mathbf{x})}_{\textit{A posteriori} \text{ probability}} = \frac{\overbrace{\Pr(Y = y)}^{\textit{A priori} \text{ probability}} \cdot \overbrace{\Pr(X = \mathbf{x}|Y = y)}^{\text{Likelihood}}}{\Pr(X = \mathbf{x})}$$



where

- $\mathbf{x} \in \mathbb{R}^d$ an observation
- $y \in Y$ a label

In a supervised learning problem, we search for $\Pr(Y = y|X = \mathbf{x})$: the probability of observed the y label given the observation $\mathbf{x}$.

Gaussian linear model

Hypotheses

- explanatory variables are non-collinear
- errors are centred, non-correlated, and has the same variance σ^2

$$\epsilon_i = f_{\boldsymbol{\beta}}(\mathbf{x}_i) - y_i$$

- errors follow a $\mathcal{N}$ormal law (of parameters 0 and σ^2):

$$\begin{cases} \epsilon_i \sim \mathcal{N}(0, \sigma^2) \\ \epsilon_i \text{ indpendants} \end{cases}$$

and so: $\mathcal{Y} \sim \mathcal{N}(\tilde{\mathbf{X}}\boldsymbol{\beta}, \sigma^2 \mathbb{I})$

Gaussian linear model

Hypotheses

- explanatory variables are non-collinear
- errors are centred, non-correlated, and has the same variance σ^2

$$\epsilon_i = f_{\boldsymbol{\beta}}(\mathbf{x}_i) - y_i$$

- errors follow a $\mathcal{N}$ormal law (of parameters 0 and σ^2):

$$\begin{cases} \epsilon_i \sim \mathcal{N}(0, \sigma^2) \\ \epsilon_i \text{ indpendants} \end{cases}$$

and so: $\mathcal{Y} \sim \mathcal{N}(\tilde{\mathbf{X}}\boldsymbol{\beta}, \sigma^2 \mathbb{I})$

Note: other hypotheses are here required

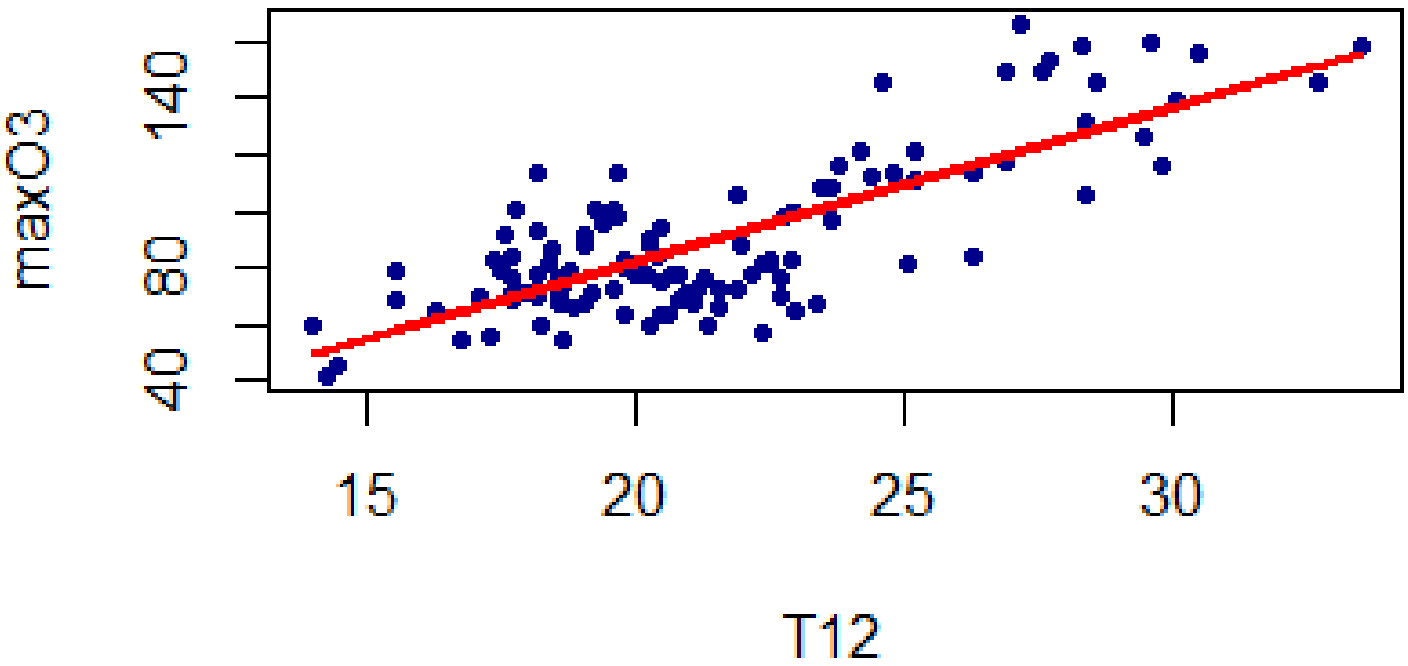
Maximum Likelihood Estimation

- Maximisation of the log-likelihood of the model

$$\begin{aligned}\underset{\boldsymbol{\beta},\sigma^2}{\text{argmax}}\,\mathcal{L}(\mathbf{Y},\boldsymbol{\beta},\sigma^2) &= \underset{\boldsymbol{\beta}\sigma^2}{\text{argmax}}\, log\left(\prod_{i=1}^{m}\phi(\mathbf{Y},\boldsymbol{\beta})\right)\\ &= \underset{\boldsymbol{\beta},\sigma^2}{\text{argmax}}\left(-\frac{m}{2}log\sigma^2-\frac{m}{2}log2\pi-\frac{1}{2\sigma^2}||\mathbf{Y}-\tilde{\mathbf{X}}\boldsymbol{\beta}||\right)\end{aligned}$$

- **Explicit solution** (and identical) for $\boldsymbol{\beta}^*$:

$$\boldsymbol{\beta}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$



$$\beta = (-27.420; 5.469)$$

it.	1	10	20	30	40	50	100
β	(0;0)	(-16.39;3.33)	(-23.41;4.72)	(-25.97; 5.20)	(-26.89; 5.38)	(-27.23; 5.44)	(-27.42;5.47)
$\mathbf{J}(\beta)$	3650.76	77.30	215.54	159.34	152.50	51.67	151.55

Comparison

- **Gradient descent algorithm**
  - hyperparameter α to be fixed by the user
  - iterative algorithm, approximated solution
  - fast for big value of d (100,1000,10000...)
- **Ordinary Least Squares / Gaussian Linear Model**
  - No parameter to be fixed
  - No iteration, exact solution
  - Complexity in d^3 (due to the computation of $X^T X^{-1}$) (slow or impossible for high value of d): variable selection, regularisation

Content

Evaluation measures

- Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - f(\mathbf{x}_i)|$$

  $|y_i - f(\mathbf{x}_i)|^2$ is the absolute difference between the actual and predicted value (namely residuals or errors). MAE indicates the average error in unit of y.

- Root-Mean-Square Error (RMSE)

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(y_i - f(\mathbf{x}_i)\big)^2}$$

  RMSE indicates the average error in unit of y too, but penalises larger errors more severely than MAE. RMSE also tends to be bigger than MAE as the sample size increases.

[2] We take the absolute difference to remove the sign on the error value! If we don't, the positive and negative errors will tend to cancel each other out, giving a misleadingly small value for the evaluation metric.

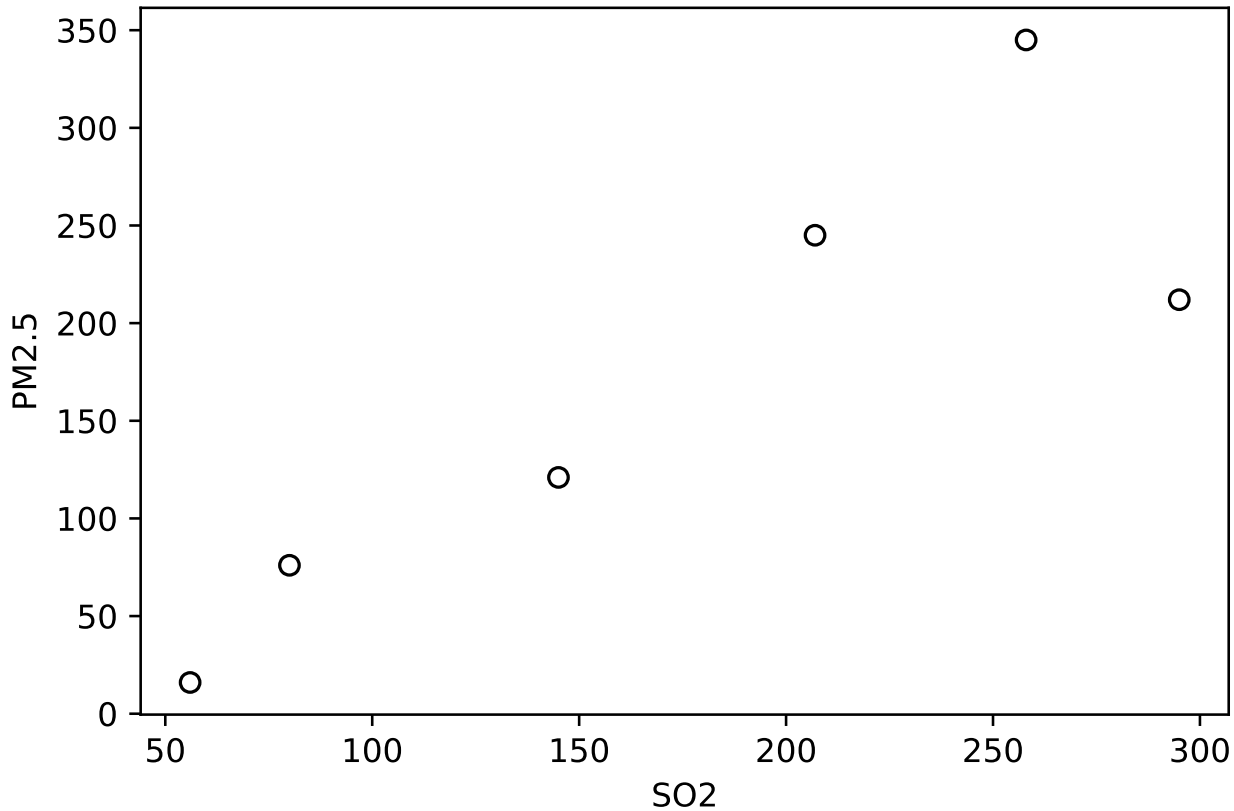
- R-Squared

$$R^2 = \frac{\sum_{i=1}^{N} \left(y_i - \bar{y}\right)^2 - \sum_{i=1}^{N} \left(y_i - f(\mathbf{x}_i)\right)^2}{\sum_{i=1}^{N} \left(y_i - \bar{y}\right)^2}$$

  R^2 computes how much better the regression line fits the data than the mean line: how much better is regression than just predicting the mean? It indicates the degree to which the model explains the variance in the data.

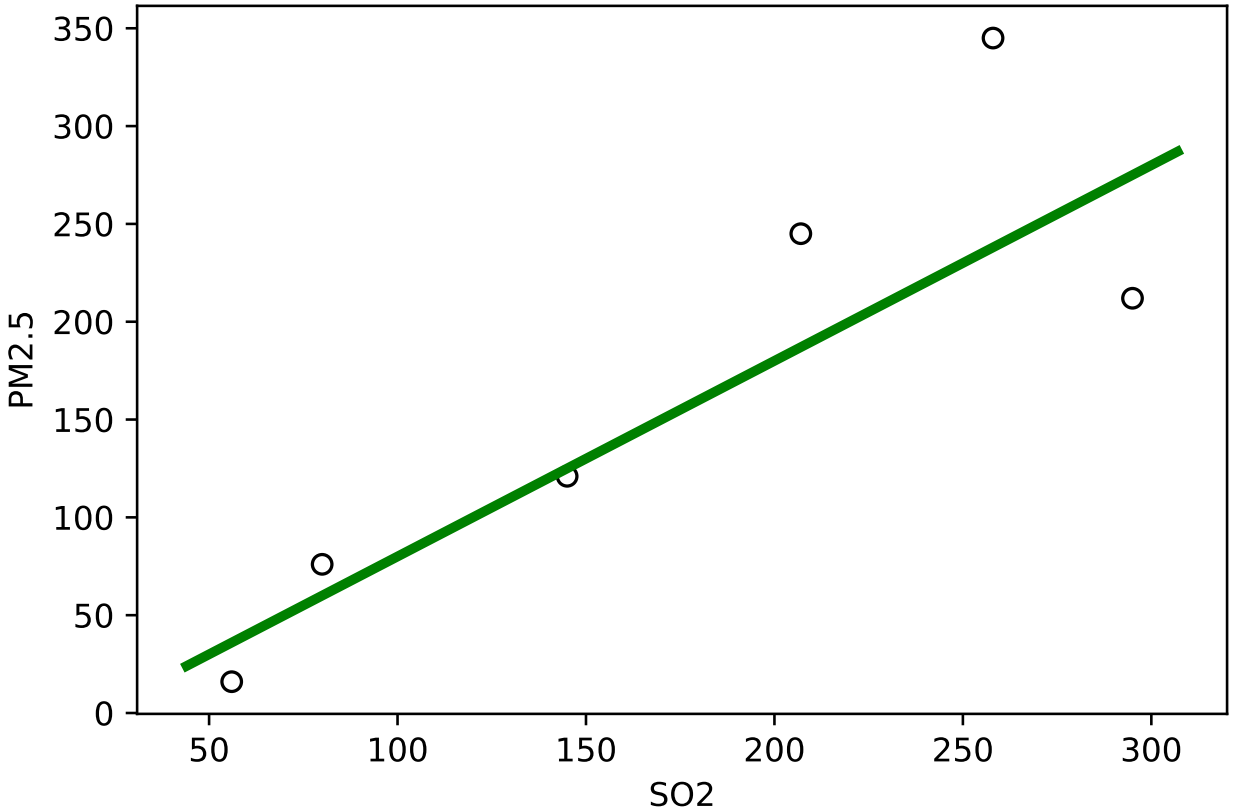
  - if $R^2 = 1$, the model perfectly fits the data
  - if $R^2 = 0$, the model is not better or worse than predicting the mean
  - if $R^2 < 0$, the model is worse than just predicting the mean

An example evaluation



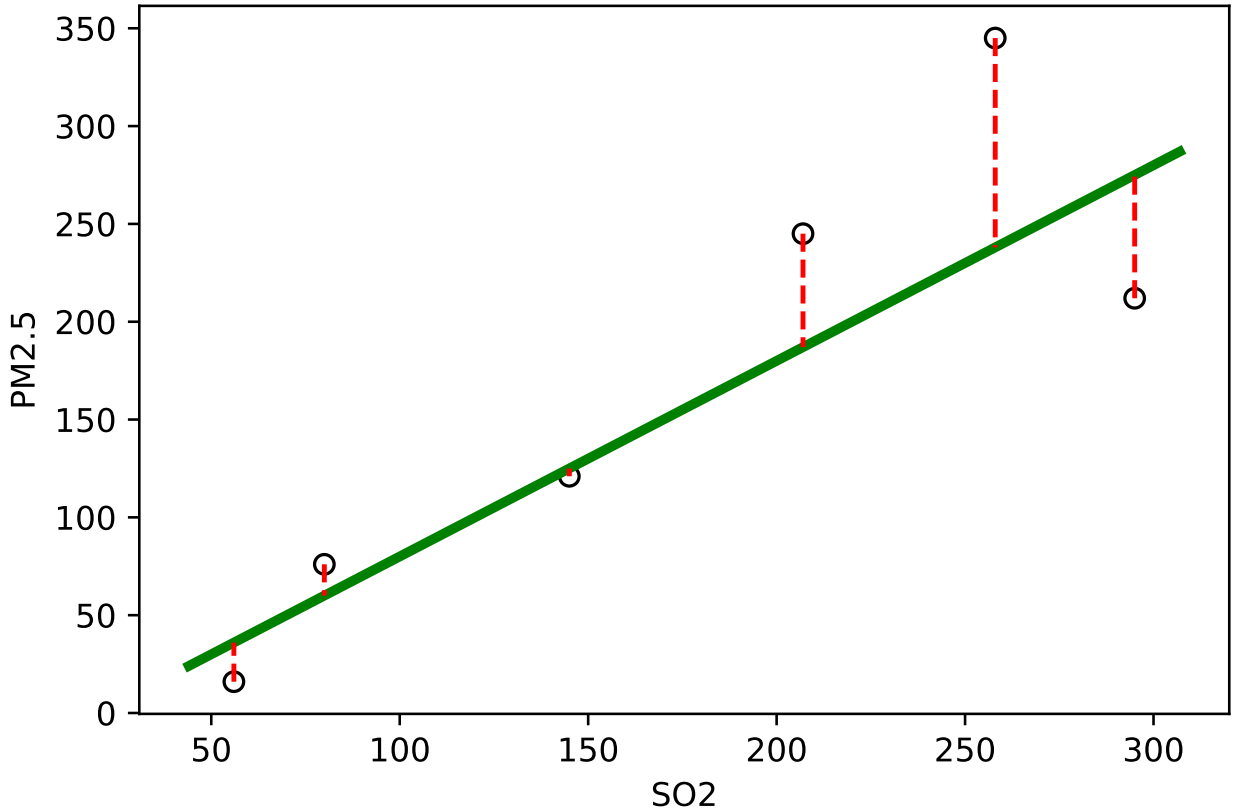
x	y					
56	16					
80	76					
145	121					
207	245					
258	345					
295	212					

An example evaluation



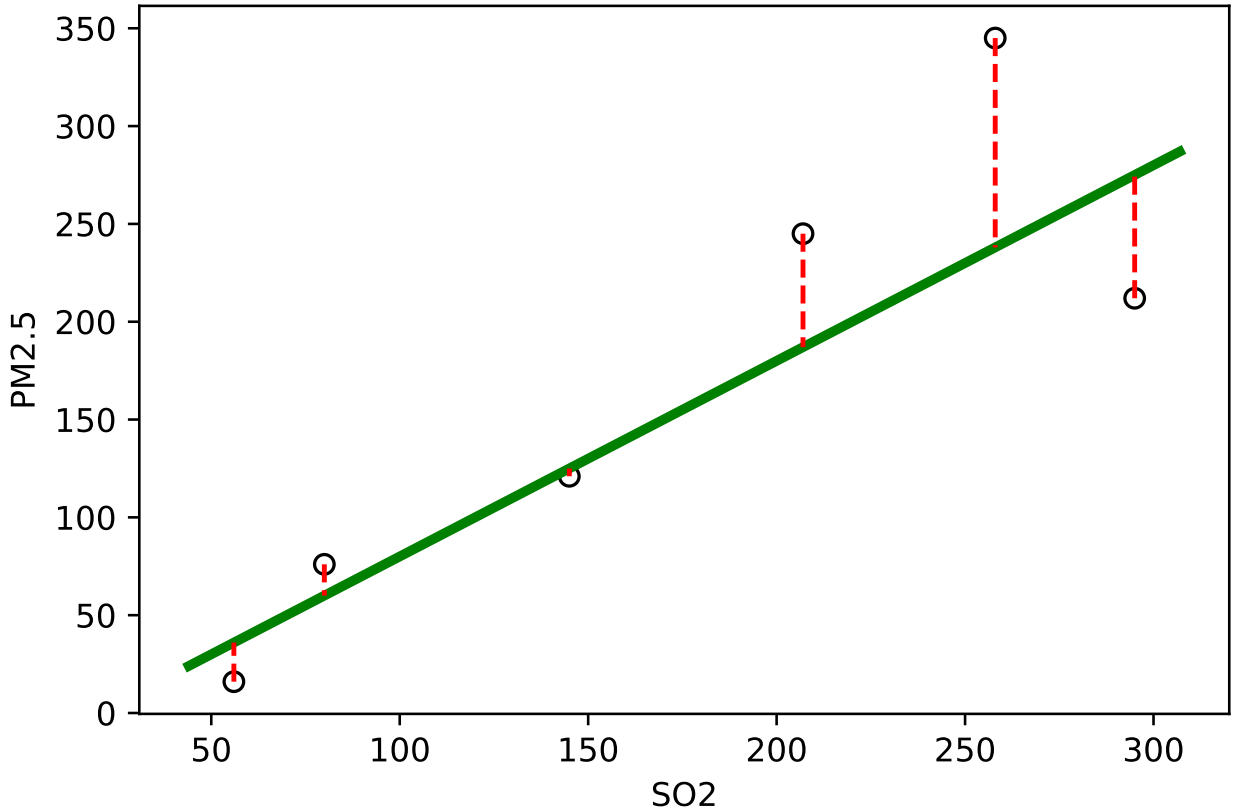
x	y	$\hat{y}$				
56	16	36				
80	76	60				
145	121	125				
207	245	187				
258	345	238				
295	212	275				

An example evaluation



x	y	$\hat{y}$	$y - \hat{y}$			
56	16	36	-20			
80	76	60	16			
145	121	125	-4			
207	245	187	58			
258	345	238	107			
295	212	275	-63			

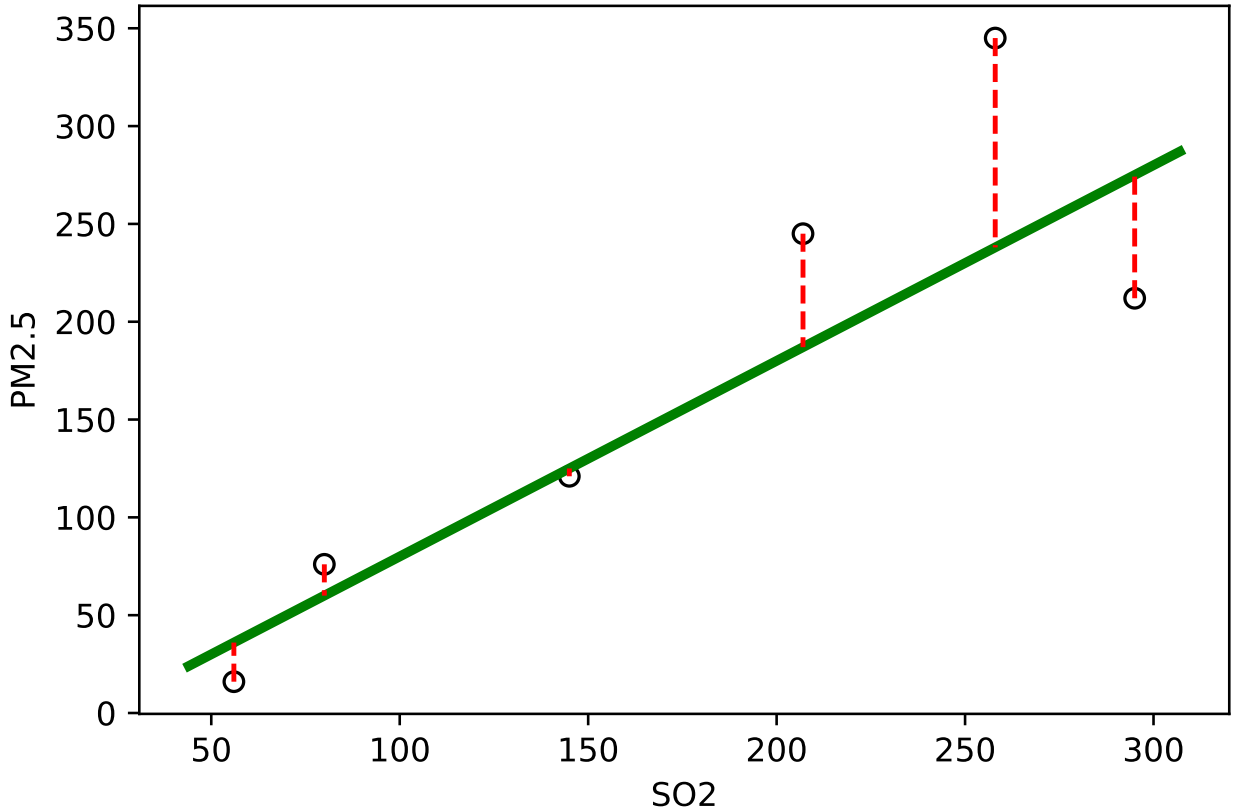
An example evaluation



x	y	$\hat{y}$	$y - \hat{y}$	$\lvert y - \hat{y} \rvert$		
56	16	36	-20	20		
80	76	60	16	16		
145	121	125	-4	4		
207	245	187	58	58		
258	345	238	107	107		
295	212	275	-63	63		

- $MAE = 44.67$

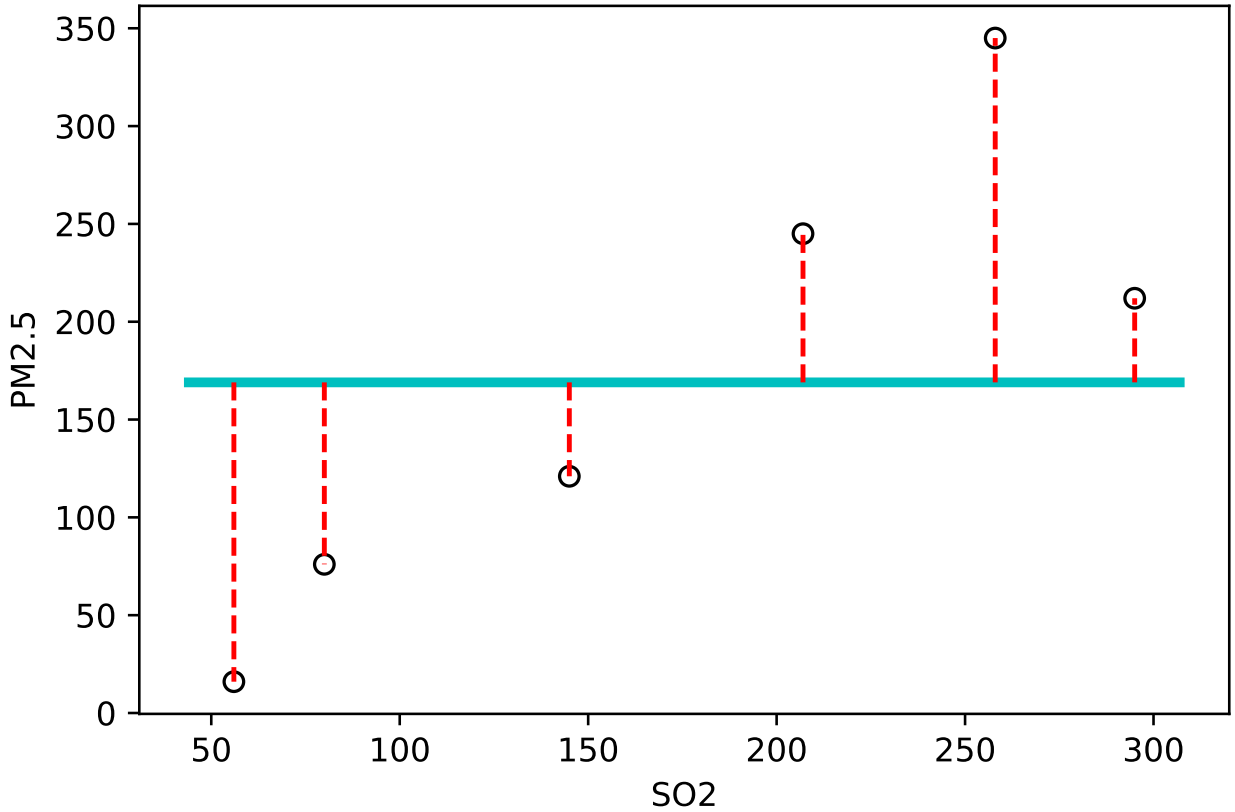
An example evaluation



x	y	$\hat{y}$	$y - \hat{y}$	$\lvert y - \hat{y} \rvert$	$(y - \hat{y})^2$	
56	16	36	-20	20	400	
80	76	60	16	16	276	
145	121	125	-4	4	16	
207	245	187	58	58	3364	
258	345	238	107	107	11,449	
295	212	275	-63	63	3969	

- $MAE = 44.67$
- $RMSE = 56.97$

An example evaluation



x	y	$\hat{y}$	$y - \hat{y}$	$\lvert y - \hat{y} \rvert$	$(y - \hat{y})^2$	$(y - \bar{y})^2$
56	16	36	-20	20	400	23,409
80	76	60	16	16	276	8,649
145	121	125	-4	4	16	2,304
207	245	187	58	58	3364	5,776
258	345	238	107	107	11,449	30,976
295	212	275	-63	63	3969	1,849

- $MAE = 44.67$
- $RMSE = 56.97$
- $R^2 = 0.73$

Content

Application

The goal of the application is to predict PM2.5 air quality in the city of Beijing. The dataset contains 17532 time series with 9 dimensions. This includes hourly air pollutants measurements (SO2, NO2, CO and O3), temperature, pressure, dew point, rainfall and windspeed measurements from 12 nationally controlled air quality monitoring sites. The air-quality data are from the Beijing Municipal Environmental Monitoring Center. The meteorological data in each air-quality site are matched with the nearest weather station from the China Meteorological Administration. The time period is from March 1st, 2013 to February 28th, 2017.
For this lab session, only the information collected at midday is used.

Objective: study the relationship between the PM2.5 air quality measurement and the explanatory variables.

The data was originally published by Zhang, S., Guo, B., Dong, A., He, J., Xu, Z. and Chen, S.X. (2017) Cautionary Tales on Air-Quality Improvement in Beijing. Proceedings of the Royal Society A, vol. 473, no. 2205, pp. 20170457

Lab Session

- Download the data
- Implement the functions: model f, cost function, gradient computation, and then the gradient descent algorithm
  - Simple linear regression
  - Extension to multiple linear regression
  - Return the model coefficients and the values of the cost function for all the iterations
- Analyse the algorithm sensitivity to its hyperparameters (initialisation, learning rate), and comment.
- Compare the results with the one obtained by the Gaussian linear model.
- Evaluate your regression model by using the `sklearn.metrics`.