# Supervised Learning - Linear Regression

Mark Asch - IMU/VLP/CSU

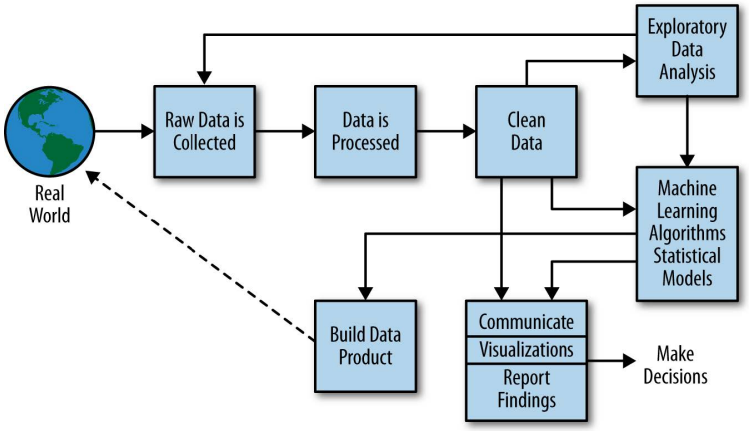
2023

Intro to ML

# Program

1. Data Analysis
   (a) Introduction: the 4 identifiers of "big data" and "data science"
   (b) Supervised learning methods: regression, k-NN, SVM, NN, decision trees.
   (c) Unsupervised learning methods: k-means, principal component analysis, clustering.

# The Data Science Process




1. Raw data: measurements, observations, web-crawling, etc.

2. Collection and cleaning: pipelines of data munging with tools such as Python, R, SQL, etc.

3. Exploratory data analysis (EDA).

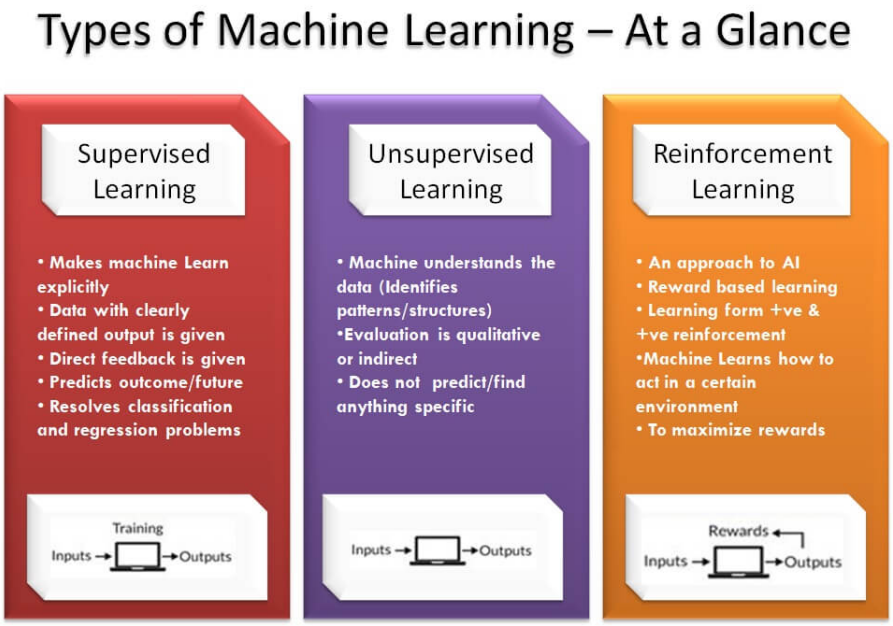
4. Choice of a model as a function of problem type: classification, prediction, description

   (a) Algorithms of statistical learning

(b) Statistical modeling

5. Interpret, visualize, report, communicate the results (or create an app !)

# Statistical Learning: what is it?




**Definition 1.** Statistical Learning ("machine learning") is a collection of tools for understanding data by seeking relations between them.

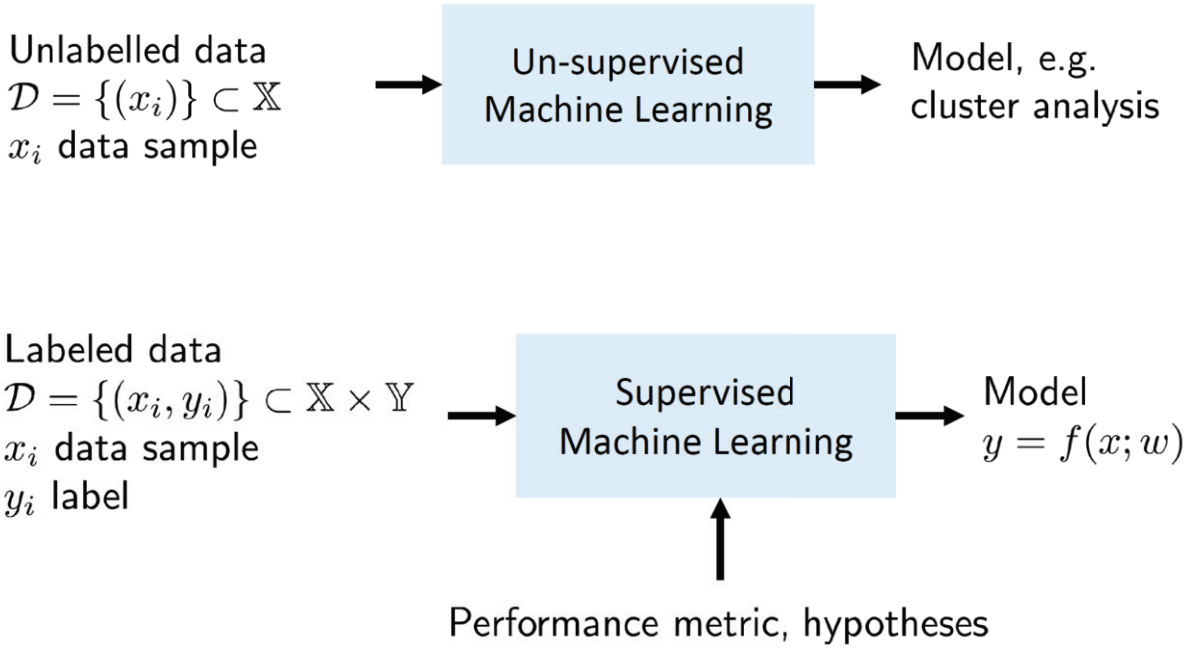
The tools can be classified into:

- supervised tools, where we construct a statistical model to predict or estimate an *output*, based on *inputs;*
- unsupervised tools, where we learn relations among *in-*

*puts* without any outputs (response variables) that supervise;

- tools for reinforcement learning, where an agent learns the environment by interacting with it, and by receiving rewards (process of maximization of the expected cumulative reward)

# Supervised and Unsupervised



# Which model for which task?

| Class | Model | Task |
|---|---|---|
| Supervised | linear regression | R |
| | CART (trees) | R&C |
| | SVM | R&C |
| | NN | R&C |
| | $k$-NN | C |
| | Naive Bayes | C |
| Unsupervised | $k$-means | Clustering |
| | dendrogram | Clustering |
| | PCA | pattern |

R = regression, C = classification

# Recall: the mathematical framework

- Suppose that we have:
  - ⇒ a response variable (to explain), $Y$,
  - ⇒ $p$ explanatory[1] variables, $X = (X_1, X_2, \ldots, X_p)$,
  - ⇒ a relationship between $Y$ and $X$ of the form

$$Y = f(X) + \epsilon$$

  - ⇒ where
    - → $f$ is an unknown function of $X_1, X_2, \ldots, X_p$
    - → $\epsilon$ is a random error term, independent of $X$, and with zero mean

- ML is then an ensemble of approaches for estimating $f$ with the objectives of
  - ⇒ Prediction: $\hat{Y} = \hat{f}(X)$ where $\hat{f}$ is an estimation for $f$ and $\hat{Y}$ is the resulting prediction
  - ⇒ Inference: to understand how $Y$ varies as a function of $X$ (correlations, importances, linearity, etc.)

[1] Also called: features, attributes

# Errors

**Example 1.** Let $X_1, X_2, \ldots, X_p$ be characteristics of a patient's blood sample, easily measured in a laboratory. Let $Y$ be a variable that describes the patient's risk of an adverse reaction to a given drug. It is natural to seek to predict $Y$ from $X$ - then we can avoid to give the drug to high-risk patients.

The precision of $\hat{Y}$ as a prediction of $Y$ depends on 2 quantities:

- the reducible error - $\hat{f}$ is not a perfect estimate $f$ and can be improved

- the irreducible error - $\epsilon$ cannot be predicted by $f$ ($\epsilon$ can contain effects of non-measured variables - for example, the risk of an adverse reaction can depend on the patient's health status on the given day, or the variability in the manufacture of the drug)

We can show that

$$E(Y - \hat{Y})^2 = E\left[f(X) + \epsilon - \hat{f}(X)\right]^2$$
$$= \underbrace{\left[f(X) - \hat{f}(X)\right]^2}_{\text{reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{irreducible}}$$

✔ The objective of Statistical Leaning is to study techniques for the estimation of $f$ while minimizing the reducible error...

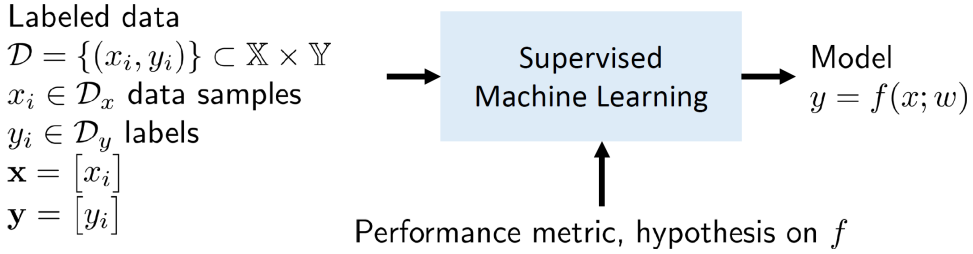
# Regression and Classification

Variables can be characterized as:

- ✔ quantitative, taking on numerical values
- ✔ qualitative (or categorical), that take values in one of $K$ different classes (or categories).

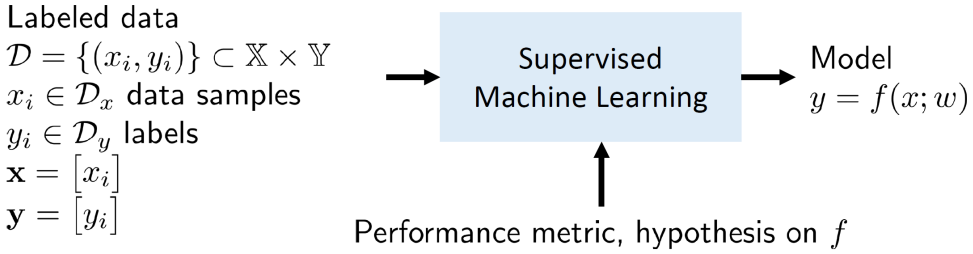
The problems are then of type:

- ✔ regression when we have quantitative variables,
- ✔ classification for qualitative variables.




**Model purpose – Regression**

- The model $f$ shall map $x \mapsto y$ and approximate an unknown function $\hat{f} : \mathbb{X} \to \mathbb{Y}$
- $y_i \in \mathbb{Y} \subseteq \mathbb{R}^{n_y}$
- Examples: data-driven modeling, energy forecasting, ...




**Model purpose – Classification**

- The model $f$ shall map $x \mapsto y$ and approximate an unknown function $\hat{f} : \mathbb{X} \to \mathbb{Y}$
- $y_i \in \mathbb{Y} \subseteq \mathbb{N}^{n_y}$
- Examples: spam filter, fraud detection, fault detection, ...

- the only difference is the space in which $y_i$ takes its

values:

$\Rightarrow$ continuous space, $\mathbb{R}^n$, for regression

$\Rightarrow$ discrete space, $\mathbb{N}^n$, for classification

# Linear Regression

**Hypothesis** : there exists a linear relation between the output (response, dependent) variable and the input (explanatory, independent, feature) variable(s)

✔ LR is one of the most widely-used statistical learning methods

✔ LR implies a linear correlation between the changes in an explanatory variable and its output

We start with simple linear regression (SLR) :

$$Y \approx \beta_0 + \beta_1 X$$

where

- $\beta_0$ is the intercept and $\beta_1$ is the slope;
- $\{\beta_0, \beta_1\}$ are the parameters of the model that we will estimate by $\left\{\hat{\beta}_0, \hat{\beta}_1\right\}$

We obtain the prediction model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

# Estimating the parameters

- In practice, $\beta_0$ and $\beta_1$ are unknown
- We must use the data to estimate the coefficients...
- Let the observations be

$$(x_1, y_1),\ (x_2, y_2),\ \cdots, (x_n, y_n)$$

- Our objective: obtain the best fit possible between a linear model and the data
- We will use the least squares criterion (there are others... see lecture on "Other Regression Methods")

# Least Squares

**Definition 2.** The residue of the $i$-th response is

$$e_i = y_i - \hat{y}_i$$

and the sum of squares of residues is defined by

$$\begin{aligned} \mathrm{RSS}(\beta) &= e_1^2 + e_2^2 + \cdots + e_n^2 \\ &= \sum_{i=1}^{n} \left( y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i \right)^2 \end{aligned}$$

The least squares criterion: choose $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the $\mathrm{RSS}$.

# Coefficients of SLR

✔ To minimize the RSS, we differentiate with respect to $\beta$ and we set the derivatives equal to zero[2]...

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} = \frac{s_{xy}}{s_x^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad \text{and} \qquad \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

are the empirical averages.

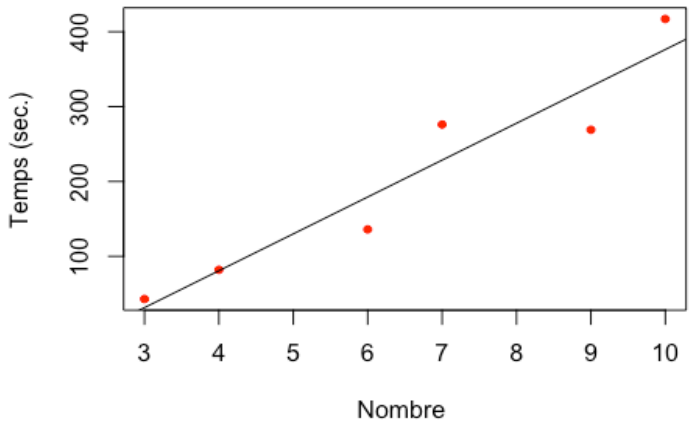
[2]This is known as the "necessary condition for optimality".

# How to do this with R?

```r
# Régression Linéaire Simple
x <- c(7, 3, 4, 6, 10, 9)
y <- c(276, 43, 82, 136, 417, 269)
SLRmodel <- lm(y~x)
SLRmodel

## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##     -116.23        49.24

coefs <- coef(SLRmodel)
plot(x, y, pch=20,col="red", xlab="Nombre ", ylab="Temps (sec.)")
abline(coefs[1],coefs[2])
```




```r
# diagnostics
summary(SLRmodel)
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -116.227     54.792  -2.121  0.10120
## x             49.240      7.868   6.258  0.00332 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 48.18 on 4 degrees of freedom
## Multiple R-squared:  0.9073, Adjusted R-squared:  0.8842
## F-statistic: 39.17 on 1 and 4 DF,  p-value: 0.003325
```

# How to do this with Python (statsmodels)?

```
In [5]: import statsmodels.api as sm

        x = [7, 3, 4, 6, 10, 9]
        y = [276, 43, 82, 136, 417, 269]

        x = sm.add_constant(x)
        model   = sm.OLS(y,x)
        results = model.fit()
        results.summary()

Out[5]: <class 'statsmodels.iolib.summary.Summary'>
        """
                                    OLS Regression Results
        ==============================================================================
        Dep. Variable:                      y   R-squared:                       0.907
        Model:                            OLS   Adj. R-squared:                  0.884
        Method:                 Least Squares   F-statistic:                     39.17
        Date:                Mon, 06 Jan 2020   Prob (F-statistic):           0.00332
        Time:                        10:20:05   Log-Likelihood:                -30.547
        No. Observations:                   6   AIC:                             65.09
        Df Residuals:                       4   BIC:                             64.68
        Df Model:                           1
        Covariance Type:            nonrobust
        ==============================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
        ------------------------------------------------------------------------------
        const       -116.2267     54.792     -2.121      0.101    -268.355      35.901
        x1            49.2400      7.868      6.258      0.003      27.396      71.084
        ==============================================================================
        Omnibus:                        nan   Durbin-Watson:                   2.175
        Prob(Omnibus):                  nan   Jarque-Bera (JB):                0.578
        Skew:                        -0.268   Prob(JB):                        0.749
        Kurtosis:                     1.577   Cond. No.                         19.7
        ==============================================================================

        Warnings:
        [1] Standard Errors assume that the covariance matrix of the errors is correctly
        specified.
        """
```

# How to do this with Python (sklearn)?

```
In [1]: import numpy as np
        from sklearn import linear_model

        x = np.array([7, 3, 4, 6, 10, 9])
        y = np.array([276, 43, 82, 136, 417, 269])

        lm    = linear_model.LinearRegression()
        model = lm.fit(x.reshape(-1, 1),y.reshape(-1, 1))

In [2]: print('Coefficients : \n', model.coef_)

Coefficients :
 [[49.24]]


In [8]: print('Intercepte : %0.2f' % model.intercept_)

Intercepte : -116.23


In [10]: print('Coefficient de determination : %.3f' %
             model.score(x.reshape(-1, 1),y.reshape(-1, 1)))

Coefficient de determination : 0.907
```

# SLR: Analysis of results

- The estimated straight line is

$$\hat{y} = -116,23 + 49,24\,x$$

- But, if a new measurement arrives, with an $x$-value of $5$, with what confidence can we claim that the response is

$$-116,23 + 49,24 * 5 = 129,97\ ?$$

- For this, we need to extend the model by:

  ✔ Adding hypotheses of error modeling.
  ✔ Adding predictive variables.
  ✔ Transforming the predictive variables.

# I. Error Modeling Hypotheses

- If we use a model to predict $y$ for a given value $x$, then the prediction is deterministic and does not take into account the variability in the observed data...

- We generalize the model to

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

  where the new term $\epsilon$ is a random "noise" and is called the error term

- Modeling hypotheses on $\epsilon$ are:

  $\Rightarrow$ the error follows a Gaussian distribution, with zero mean and variance $\sigma$,$^2$that is $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$
  $\Rightarrow$ the error is independent of $x$ and
  $\Rightarrow$ the errors $\epsilon_i$ are uncorrelated and of equal variance (i.i.d.)

- Mathematically, the model tells us that for any given value of $x$, the conditional distribution of $y$ for $x$ given,

is

$$p\left(y|x\right) \sim \mathcal{N}\left(\beta_0 + \beta_1 x, \sigma^2\right)$$

# Noise Model

- A noise model is essential for any analysis, particularly Bayesian (of which regression is an example)

- Neutrality of the Gaussian hypothesis for the noise model:

  $\Rightarrow$ the noise is centered : its mean value is zero, but it can take any value (small or big)
  $\Rightarrow$ large amplitudes/deviations are less and less probables: the variance is finite
  $\Rightarrow$ independence between observations (otherwise, this is part of the trends of the model)

- We neither suppose, nor impose that the noise really follows a Gaussian law...

# Estimation of parameters

- How do we fit such a model? How do we compute the parameters $\beta_0$, $\beta_1$ and $\sigma$ from the data?

- Theorem: the least squares estimate for $\beta_0$, $\beta_1$ is optimal, being unbiased and of minimal variance ( BLUE estimator)

- Estimation of the variance : the mean squared error, defined by

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n} e_i^2}{n-2} = \frac{\mathrm{RSS}}{n-2}$$

  (quantifies the variation of the predicted value with respect to the observation) is an unbiased estimator of the variance $\sigma^2$.

- But how can we measure the confidence ?

# Evaluation metrics for SLR

✔ In the output of the function `lm` of R : p-value and R-squared

```
> summary(model)
Call: lm(formula = y ~ x)
Residuals:       1       2       3       4       5       6
              47.547  11.507   1.267 -43.213  40.827 -57.933
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -116.227     54.792  -2.121  0.10120
x             49.240      7.868   6.258  0.00332 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’
Residual standard error: 48.18 on 4 degrees of freedom
Multiple R-squared:  0.9073,  Adjusted R-squared:  0.8842
F-statistic: 39.17 on 1 and 4 DF,  p-value: 0.003325
```

# R-squared

**Definition 3.** The proportion of the error/variance explained by the model is

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}} = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}},$$

where $\mathrm{TSS}$ is the total sum of squares that measures the inherent variability in the response before the regression is done.

*Remark.* $R^2$ (coefficient of determination) measures the proportion of the variability in $y$ that can be explained using $x$. $\mathrm{RSS}$, by contrast, measures the quantity of variability in the response that is left unexplained after performing the regression. $\mathrm{TSS} - \mathrm{RSS}$ measures the quantity of variability that is explained (or removed) by performing the regression.

- ✔ A value of $R^2$ close to zero indicates that the regression has not explained/captured much of the variability in the response...
  - ✔ either a linear model is not suitable,
  - ✔ or the inherent $\sigma^2$ is too high,
  - ✔ or both.

- ✔ Warning : the value itself of $R^2$ is not always reliable, and the MSE should also be taken into account—see examples in `R-squared-dangers.html`

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

## p-Value

- The estimations of $\beta$ can be found in the column "Estimate"
- The p-values are in the column, Pr(>|t|)

**Definition 4.** The p-value is the probability, under the null hypothesis ($\beta_1 = 0$), to observe a value greater than $|t|$, where the $t$-statistic is given by

$$t = \frac{\hat{\beta}_1 - 0}{\mathrm{SE}\left(\hat{\beta}_1\right)}$$

with standard error,

$$\mathrm{SE}\left(\hat{\beta}_1\right)^2 = \frac{\sigma^2}{\sum_i \left(x_i - \bar{x}\right)^2}.$$

# Interpretation of the p-values

- If the p-value is low (threshold that is context-dependent...), it is very unlikely to observe such a test statistic under the null hypothesis (where no trend is supposed)

  $\Rightarrow$ it is this highly probable that the coefficient is non-zero and thus significant (statistically speaking).

- If the p-value is high we cannot reject the null hypothesis, since the observed value of the test statistic is probably not due exclusively to chance (resulting from the intrinsic variability)

  $\Rightarrow$ the coefficient could be zero and non-significant

# Graphical diagnostics

- Let the linear model be:
  `try <- lm(Amax ~ SLA, data=photo, na.action=na.omi`

- The command `plot(try)` displays 4 plots:

  1. The plots ("Residuals vs Fitted" and "Scale-Location") should not give any clear trends (they should neither be all increasing or all decreasing). This shows,
     (a) that on average, the regression line is well fitted to the data, and thus the hypothesis of linearity is acceptable,
     (b) that the variance is constant and of the same value for all the observations.
  2. The "normal Q-Q" plot should show points distributed around the dashed line and that follow the line approximately, without marked deviations (especially at the extremities). This shows that the hypothesis of the residues having a normal distribution, is satisfied.

3. The last "Cook distances", should not show any point that exceeds 1 on the abscissa. This shows the presence of influential data.

# Cross Validation

✔ A learning approach, used systematically, for model evaluation... (see lecture "Other methods" for full details)

- Divide/Split the data into:

  ⇒ a training subset (80%)
  ⇒ a test subset (20%)

- Fit the model to the training subset

- Calculate the mean-squared error on the test subset

- Compare with that of the training subset

- Vary the sample size and repeat.

# II. Adding predictive variables

- We have so far studied simple linear regression with

  $\Rightarrow$ 1 output
  $\Rightarrow$ 1 predictor

- We can easily extend this model by adding predictive variables,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$$

- The model is now called multiple linear regression.

- Key idea to construct the model:

  $\Rightarrow$ plot all scatterplots of $y$ against each of the predictive variables
  $\Rightarrow$ plot all histograms of $y|x$ for different values of each predictive variable

- Evaluation (as before) :

  $\Rightarrow$ $R^2$, p-values

  $\Rightarrow$ training and test subsets

# III. Transformation of predictive variables

- Why suppose a linear relation?
- Possible to test other relations (non linear)

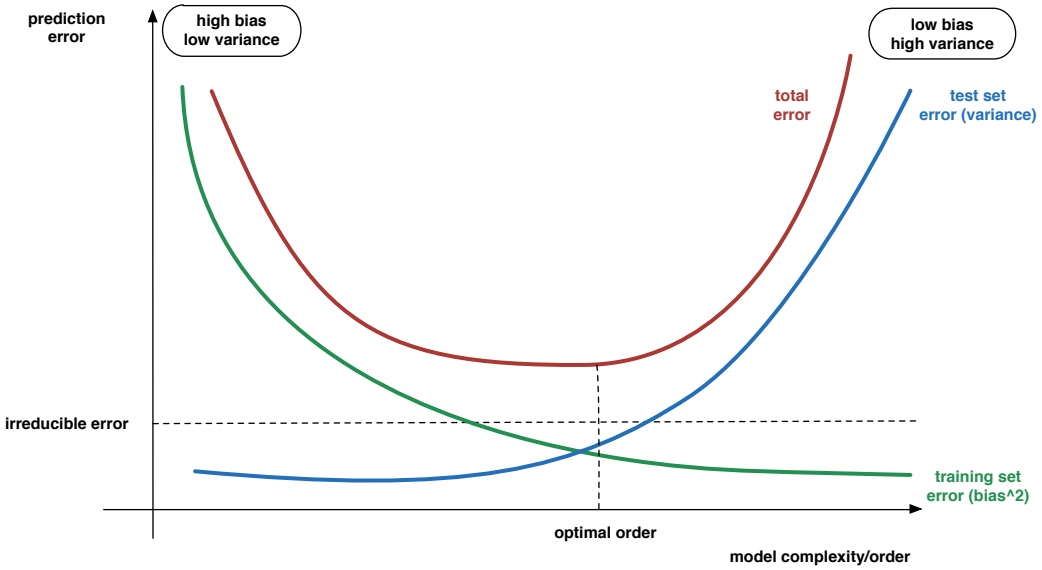
  $\Rightarrow$ by defining, for example, $z = x^2$
  $\Rightarrow$ then performing a linear regression on $z$

✘ The biggest challenge is that we never know the "truth"!

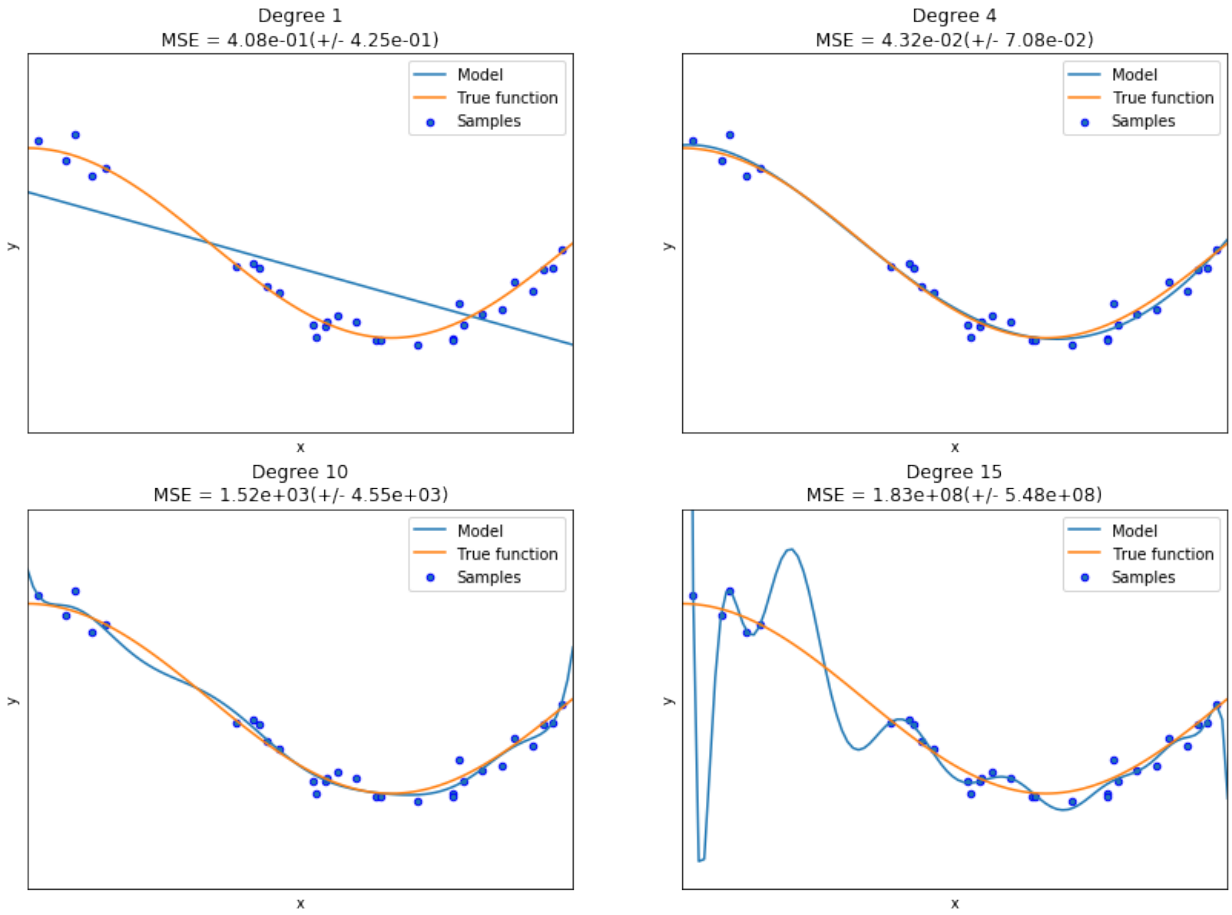
# Over- and Under-fitting and the Bias-Variance Tradeoff

- A general rule (theorem...) states that by reducing the bias of a model (by adding variables or parameters), we increase its variance, which implies greater estimation errors and a rigidity/fragility of the model obtained.

- We seek then, in the parametrization of our statistical models, a compromise between the bias and the variance.



# Example of Regression of a Cosine



# References


1. M. DeGroot, M. Schervish, *Probability and Statistics*, Addison Wesley, 2002.

2. Spiegel, Murray and Larry Stephens, *Schaum's Outline of Statistics,* 6th edition, McGraw Hill. 2017.

3. G. James, D. Witten, T. Hastie, R. Tibshirani. *An Introduction to Statistical Learning with Applications in R.* Springer. 2013.

4. Rachel Schutt and Cathy O'Neil. *Doing Data Science.* O'Reilly. 2014.