Régression linéaire

- Rappels de python -

On pourra se servir du cours d'introduction aux statistiques pour les premières manipulations sous `python`. Au besoin, on peut également consulter les pages suivantes pour démarrer ou bien consulter quelques rappels utiles

- ⋆⋆⋆ http://perso.telecom-paristech.fr/~gramfort/liesse_python/1-Intro-Python.html
- ⋆⋆⋆ http://perso.telecom-paristech.fr/~gramfort/liesse_python/2-Numpy.html
- ⋆⋆⋆ http://perso.telecom-paristech.fr/~gramfort/liesse_python/3-Scipy.html
- ⋆⋆⋆ http://scikit-learn.org/stable/index.html
- ⋆⋆ http://www.loria.fr/~rougier/teaching/matplotlib/matplotlib.html
- ⋆⋆ http://jrjohansson.github.io/

Enfin des éléments de correction sont disponibles dans le fichier `TP_regression.py`.

- Rappels sur le modèle linéaire -

Nous considérons le modèle statistique suivant :

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \tag{1}$$

où

- $\mathbf{y} = (y_i)_{1\leq i\leq n} \in \mathbb{R}^n$ est un vecteur colonne $n \times 1$,
- $X = (X_{i,j})_{1\leq i\leq n, 0\leq j\leq p}$ est une matrice $n \times (p+1)$ de **rang plein** telle que $X_{i,0} = 1$ pour tout $1 \leq i \leq n$ (ce qui signifie que l'on prend en compte l'effet de la variable constante),
- $\boldsymbol{\theta} = (\theta_i)_{0\leq i\leq p} \in \mathbb{R}^{p+1}$ est un vecteur colonne $(p+1) \times 1$,
- $\boldsymbol{\varepsilon} = (\varepsilon_i)_{1\leq i\leq n} \in \mathbb{R}^n$ un vecteur colonne $n \times 1$ aléatoire.

On suppose de plus que le vecteur $\boldsymbol{\varepsilon}$ suit la distribution :

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I_n) \ .$$

On rappelle les notations suivantes :

- $\hat{\boldsymbol{\theta}} = (X^\top X)^{-1} X^\top \mathbf{y}$ l'estimateur par moindres carrés de $\boldsymbol{\theta}$ quand la matrice $(X^\top X)$ est inversible.
- $\hat{\mathbf{y}} = X\hat{\boldsymbol{\theta}}$, la prédiction sur les valeurs observées
- $\bar{x}_n = n^{-1}\sum_{i=1}^n x_i$ et $\bar{y}_n = n^{-1}\sum_{i=1}^n y_i$, les moyennes empiriques
- $\mathrm{Var}_n(\mathbf{x})$ et $\mathrm{Var}_n(\mathbf{y})$, les variances empiriques
- Le vecteur $r = \mathbf{y} - \hat{\mathbf{y}}$ est appelé vecteur des résidus.
- $\mathbf{1}_n = (1, \ldots, 1)^\top$ est le vecteur "tout à un" de taille $n \times 1$
- On note $\mathrm{RSS} = ||\mathbf{y} - \widehat{\mathbf{y}}||^2$ (*Residual Sum of Squares* en anglais).

- Régression linéaire simple -

On utilisera le langage `python` avec par exemple `Spyder` (prendre la version Anaconda) ou `ipython` pour faire ce TP. On se servira notamment des librairies `pandas` et `sklearn` (`statsmodels` peut aussi être une alternative) que l'on peut charger de la manière suivante :

```
import pandas as pd
from sklearn import linear_model
```

Le mot "régression" a été introduit par Sir Francis Galton (cousin de C. Darwin) alors qu'il étudiait la taille des individus au sein d'une descendance. On va s'intéresser à l'une de ces expériences statistiques.

1. Récupérer les données du fichier http://www.math.uah.edu/stat/data/Galton.txt. La seconde colonne contient la taille du parent "moyen", c'est-à-dire $\frac{1}{2}$ (taille(pere) + 1.08taille(mere)). La première colonne contient la taille d'un de leurs enfants (à l'âge adulte). On note x_i la taille du parent moyen pour la famille i et y_i la taille de l'enfant. On écrit $y_i = \theta_0 + \theta_1 x_i + \varepsilon_i$ et on modélise les variables ε_i comme gaussienne centrées, indépendantes de même variance σ^2 inconnue.
2. Tracer le nuage de points $(x_i.y_i)$ pour $1 \leq i \leq n$ où n est le nombre de familles figurant dans les données. Utiliser la fonction `plot` (voir par exemple `matplotlib.pyplot`) pour afficher les données.
3. Estimer θ_0, θ_1, par $\hat{\theta}_0$, $\hat{\theta}_1$ en utilisant la fonction `LinearRegression` de `sklearn.linear_model` Retrouver mathématiquement les formules pour calculer $\hat{\theta}_0$ et $\hat{\theta}_1$ dans le cas unidimensionnel. Vérifier les numériquement.
   Aide : il s'agit de retrouver la formule pour inverser une matrice 2×2.
4. Calculer et visualiser les valeurs prédites $\hat{y}_i = \hat{\theta}_1 x_i + \hat{\theta}_0$ et y_i sur un même graphique.
5. Quelle est la valeur prédite par la méthode si un point $x_{n+1} = 75$?
6. Trouver la valeur donnée par le modèle telle que l'enfant soit de même taille que le parent moyen.
7. Sachant que l'unité de mesure utilisée par Galton est le *inch* (2.54cm) comparer si les deux méthodes suivantes sont les même pour prédire la taille d'une personne dont le parent moyen mesure 196cm :
   (a) convertir 196cm en *inch* et utiliser les prédictions obtenues,
   (b) convertir toutes les données observées en cm et appliquer une régression linéaire sur ces données.
8. Visualiser l'histogramme des résidus $r_i = y_i - \hat{y}_i$. Proposer une estimation de σ à partir des résidus. L'hypothèse de normalité est-elle crédible ? Visualiser un histogramme des résidus avec la fonction `hist`. On pourra aussi s'appuyer sur la fonction `qqnorm` (on pourra utiliser `sm.qqplot` par exemple).
9. Régresser $\mathbf{x}$ sur $\mathbf{y}$ et comparer les coefficients $\hat{\alpha}_0$ et $\hat{\alpha}_1$ obtenus par rapport aux $\hat{\theta}_0$ et $\hat{\theta}_1$ du modèle original. Vérifier numériquement (et éventuellement en exercice formellement) que :

$$\begin{aligned}\hat{\alpha}_0 &= \bar{x}_n + \frac{\bar{y}_n}{\bar{x}_n} \frac{\mathrm{Var}_n(\mathbf{x})}{\mathrm{Var}_n(\mathbf{y})} (\hat{\theta}_0 - \bar{y}_n), \\ \hat{\alpha}_1 &= \frac{\mathrm{Var}_n(\mathbf{x})}{\mathrm{Var}_n(\mathbf{y})} \hat{\theta}_1.\end{aligned}$$

- Régression linéaire multiple : cas de deux variables -

Il s'agit dans cette partie de considérer deux variables explicatives. La base de donnée est disponible grâce au lien suivant :
http://vincentarelbundock.github.io/Rdatasets/csv/datasets/trees.csv
et les détails sur sa nature sont trouvables ci-dessous.
https://stat.ethz.ch/R-manual/R-patched/library/datasets/html/trees.html

10. On proposera un modèle linéaire comme dans la partie précédente : on s'attachera à expliquer et à illustrer graphiquement le lien entre le volume d'un arbre en fonction de sa hauteur et sa circonférence. Aide : pour l'affichage 3D on pourra consulter l'aide sur la fonction `meshgrid` de `numpy`.

- RÉGRESSION LINÉAIRE MULTIPLE : CAS GÉNÉRAL -

On travaille maintenant sur le fichier `auto-mpg.data` (disponible dans le même répertoire que le fichier source) et on cherche à régresser la consommation des voitures sur leurs caractéristiques : nombre de cylindres, cylindrés (*engine displacement* en anglais), puissance, poids, accélération, année, pays d'origine et le nom de la voiture. On utilise le modèle (1), où $\mathbf{y}$ est le vecteur contenant les consommations des voitures (plus précisément la distance parcourue en miles par gallon ou "mpg"), les colonnes de X sont les régresseurs quantitatifs [1].

11. Importer la base de données avec la commande `read_csv`.
12. Calculer $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{y}}$ sur une sous partie de la base : garder les 9 premières lignes et les 8 premières colonnes. Que constatez-vous ?
13. Calculer $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{y}}$ cette fois sur l'intégralité des données.
14. Calculer le carré de la norme du vecteur des résidus RSS $= \|r\|^2$ (r est ici le vecteur des résidus) puis la moyenne de ces écarts quadratiques : MSE $=$ RSS$/(n-p-1)$ (*Mean Square Errors* en anglais). Vérifier numériquement que :

$$\|\mathbf{y} - \bar{y}_n \mathbf{1}_n\|^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 + \|\hat{\mathbf{y}} - \bar{y}_n \mathbf{1}_n\|^2.$$

15. Supposons que l'on vous fournisse les caractéristiques suivantes d'un nouveau véhicule :

cylinders	displacement	horsepower	weight	acceleration	year	origin
6	225	100	3233	15.4	76	1

Prédire sa consommation [2].

16. Calculer de nouveau $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{y}}$ mais cette fois sur les données centrées-réduites (*i.e.,* quand on retranche leur moyenne aux colonnes, et que l'on fait en sorte que chaque colonne soit d'ecart-type 1).
Rem : Ce dernier point n'est visible que dans le code source hélas : *cf.* ligne 68 : https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/linear_model/base.py

Liens pour aller plus loin :

⋆⋆ http://perso.univ-rennes1.fr/bernard.delyon/regression.pdf (partie théorique)

⋆⋆ http://freakonometrics.hypotheses.org/ (pour des exemples ludiques sous `R`)

Pour plus d'aide sur les problèmes de pré-traitement de données, centrage, gestion des valeurs manquantes, etc. voir par exemple :
http://scikit-learn.org/stable/modules/preprocessing.html et
http://scikit-learn.org/stable/modules/feature_extraction.html#dict-feature-extraction

1. sauf la variable du nom, et la variable "origine".
Pour cette dernière, si on veut l'intégrer il faut introduire 3 nouvelles variables explicatives binaires (une pour chaque origine).
2. A titre d'information, la consommation effectivement mesurée sur cet exemple était de 22 mpg.