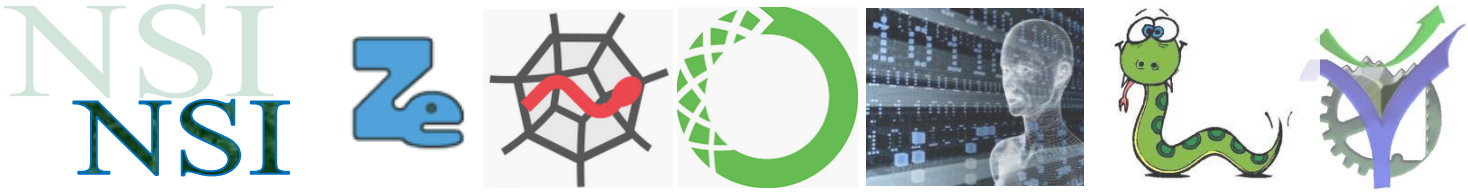


Machine Learning : Analyse en régression linéaire

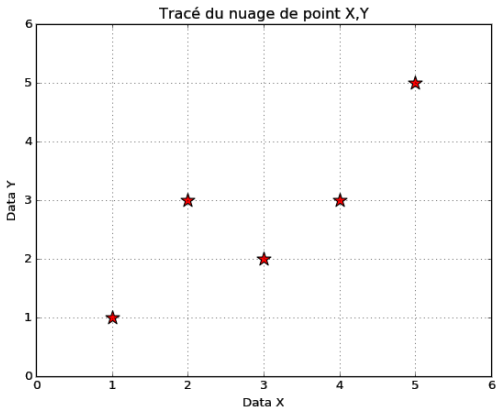
Une première étape en Machine Learning

Nom :	**Note : / 20**
	Classe :

Résumé :

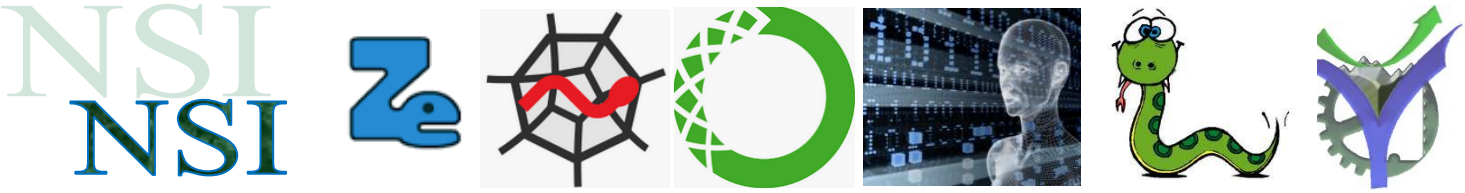
Première analyse du Machine Learning : la régression linéaire. Cette méthode ancienne a toute sa place comme première approche dès que la structuration des données le permet.

Un modèle est obtenu à partir d'un nuage de points puis celui-ci est utilisé pour réaliser des prédictions.



Sommaire

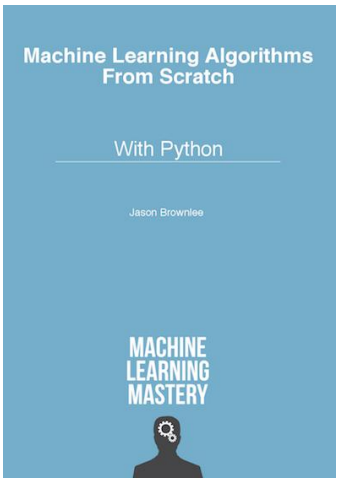
(1)



Remerciements

Un remerciement spécial à M. Jason Brownlee qui nous a donné l'autorisation d'utiliser ces travaux pour ces cours. En particulier son ouvrage disponible à l'adresse suivante :

https://machinelearningmastery.com/machine-learning-algorithms-from-scratch/



Dictionnaire anglais - français

standard deviation = écart-type
contrive = inventer
accuracy = précision
Mean Absolute Error MEA = erreur absolue moyenne
Root Mean Squared Error RMSE = Erreur moyenne quadratique
stream = flux
variance = variance
covariance = covariance
stopping distance = distance de freinage

1 Introduction à la régression linéaire

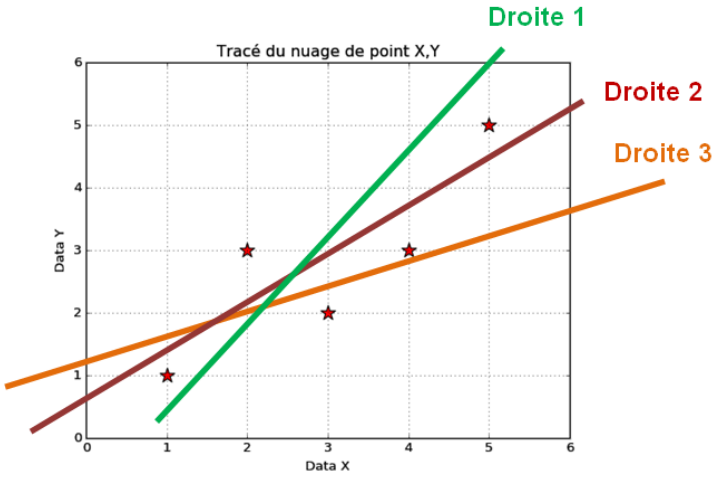
1.1 La régression et le Machine Learning

La régression linéaire est une méthode d'analyse qui date de plusieurs centaines d'années. Elle fait partie du Machine Learning tel qu'on le présente aujourd'hui car nous y retrouvons les principales caractéristiques à savoir :

- le dataset à analyser ici un nuage de points.
- à partir de ce dataset on réalise une modélisation.
- ce modèle est une droite y = a·x + b
- à partir de ce modèle on peut prédire de nouveaux résultats qui ne sont pas dans le jeu de données initial.

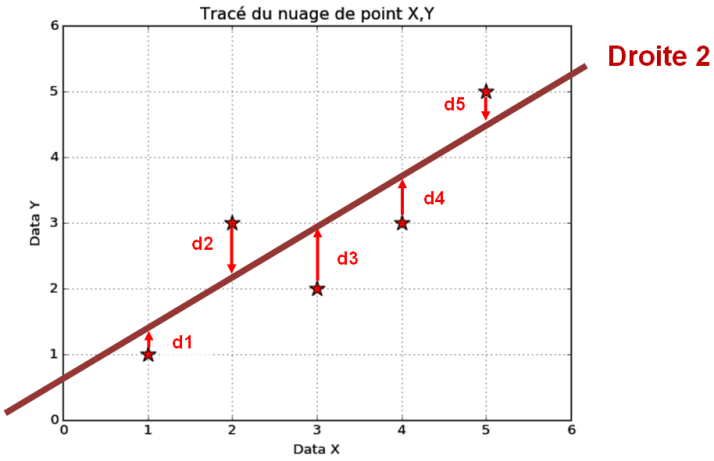
1.2 Présentation du principe de fonctionnement

Qualitativement la régression linéaire consiste à trouver où placer la droite qui se trouve la plus proche de tous les points du jeu de données. Voilà ce que cela donne sur un exemple simple :



Q1. Quelle est 'visuellement' la droite qui répond le mieux au critère spécifié ci-dessus ?

L'algorithme consiste donc à trouver les valeurs de a et b de telle manière que la distance entre tous les points du dataset et la droite y = a•x + b soit minimale.



1.3 Comment obtenir les valeurs a et b ?

Les résultats des études théoriques nous indiquent comment calculer les coefficients a et b :

$$a = \frac{\sum_{i=1}^{n}((x_i - mean(x)) \times (y_i - mean(y)))}{\sum_{i=1}^{n}(x_i - mean(x))^2}$$

$$b = mean(y) - a \times mean(x)$$

Nous retrouvons des grandeurs statistiques vues dans l'étude précédente sur le Machine Learning et la préparation des données. A savoir la moyenne et l'écart type.

La moyenne m(x)	$\frac{\sum_{i=1} x_i}{count(x)}$
La variance V(x) Avec $\sigma = \sqrt{V}$ $V = \sigma^2$ où σ est l'écart type (standard deviation)	$\frac{1}{n}\sum_{i=1}^{n}(x_i - mean(x))^2$
La covariance Cov(x,y)	$\frac{1}{n}\sum_{i=1}^{n}((x_i - mean(x)) \times (y_i - mean(y)))$

Sur cahier

Q2. Montrer littéralement à partir de la relation déterminant le coefficient a, avec les définitions rappelées ci-dessus, que l'on peut écrire :

$$a = \frac{Covariance(x,y)}{Variance(x)}$$

Q3. Après une rapide recherche sur Wikipédia ou un site équivalent rappelé la signification de la variance et de l'écart type d'une suite de valeurs.

Q4. Après une rapide recherche sur Wikipédia ou un site équivalent rappelé la signification de la covariance et de l'écart type d'une suite de valeurs.

2 Tracés de nuages de points

2.1 Nécessité d'un tracé

Bien visualiser les données est nécessaire pour les appréhender facilement. Il faut utiliser la forme graphique la plus appropriée. Avec Python la bibliothèque idoine est Matplotlib que nous avons déjà rencontré.

https://matplotlib.org/

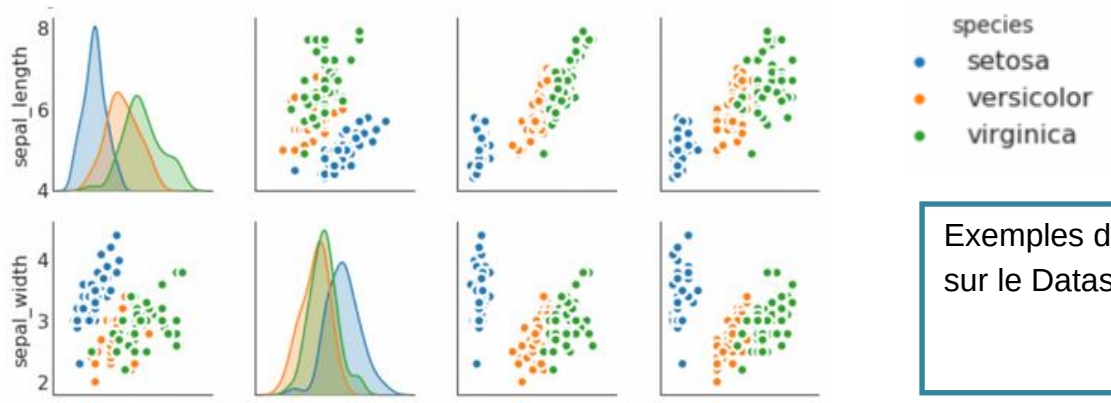
http://www.python-simple.com/

Et plus particulièrement pour les nuages de points scatter :

https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.scatter.html

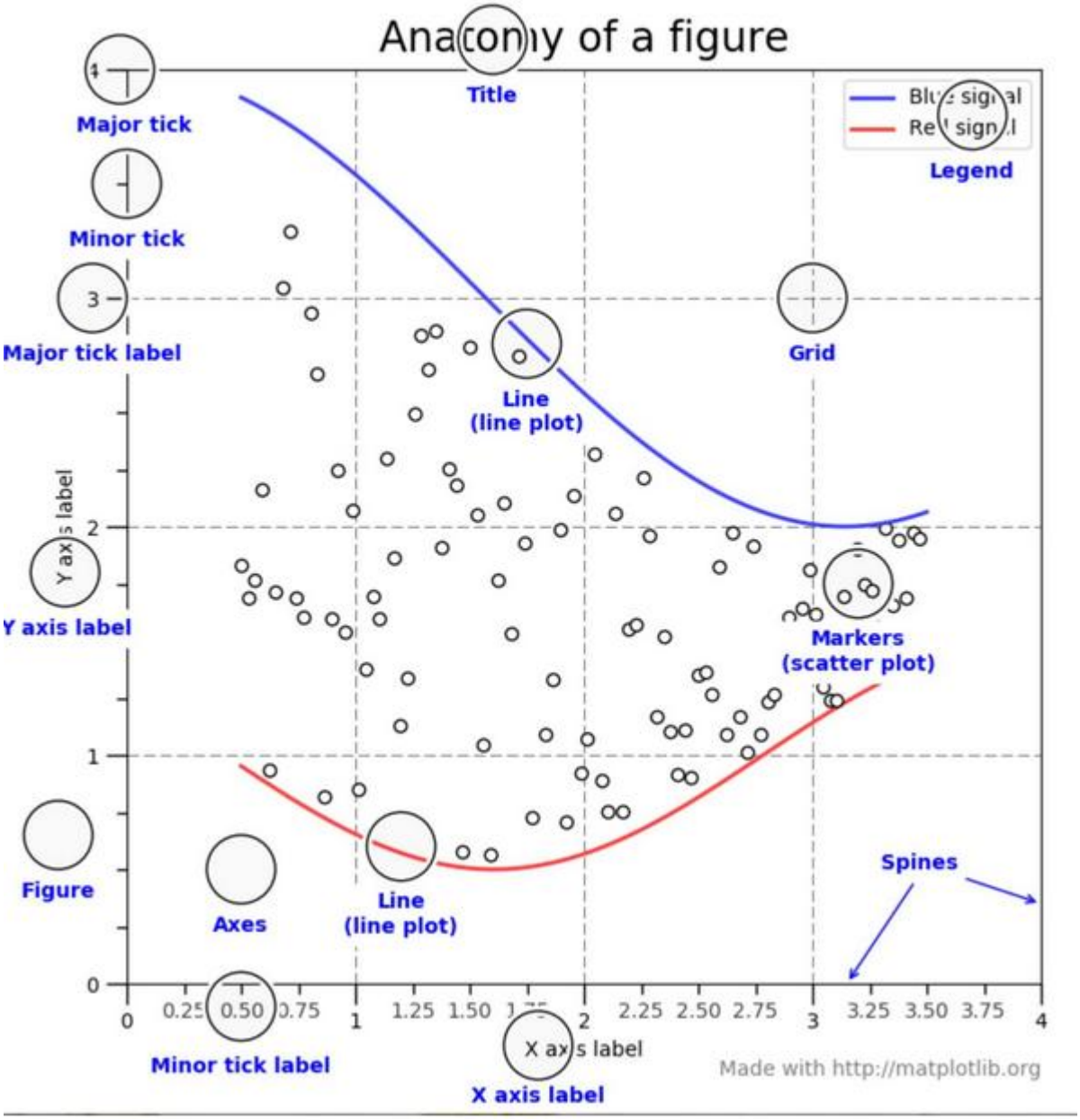
http://www.python-simple.com/python-matplotlib/scatterplot.php

https://www.section.io/engineering-education/matplotlib-visualization-python/

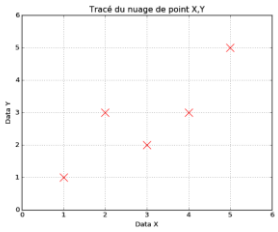


Exemples d'utilisation sur le Dataset_Iris

Aperçu d'un graphe Matplotlib



2.2 Tracé du premier nuage de points

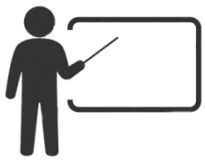


Exécuter le script : Script_Regression_Lineaire_1.py

Avec le jeu de données : Test_Dataset.csv

Script_Régression_Linéaire_1. a) Compléter le script précédent pour améliorer l'apparence des points comme ci-dessus,

b) Améliorer le script pour choisir automatiquement le fichier de données Test_Dataset.csv par défaut.



Exemple du fonctionnement, si aucun nom de fichier n'est entré alors c'est le nom par défaut qui est utilisé :

```
>>> (executing lines 1 to 84 of "Script_Regression_Lineaire_1
Indiquez le nom complet du fichier à traiter :
Choix par défaut : Test_Dataset.csv
```

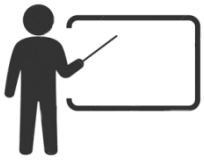
Q5. Mettez un commentaire sur toutes les instructions de Matplotlib utilisées dans le script.

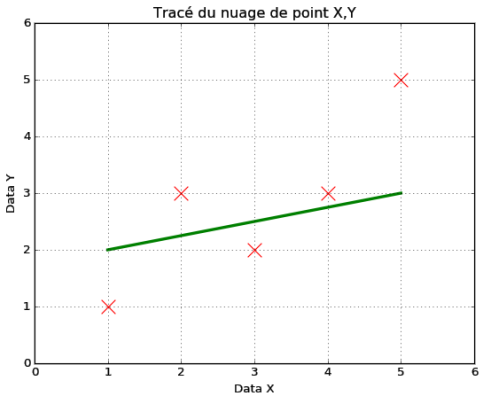
Quelques instructions de Matplotlib	Commentaires
`import matplotlib.pyplot as plt`	
`plt.scatter(valeurs_x, valeurs_y)`	
`plt.xlim(0,6)` `plt.ylim(0,6)`	
`plt.xlabel("Data X")` `plt.ylabel("Data Y")`	
`plt.title("Tracé du nuage de point X,Y")`	
`plt.grid()`	
`plt.show()`	

2.3 Tracé d'une droite en superposition

Quand nous aurons déterminé la droite de régression il nous faudra alors la tracer et l'ajouter sur notre graphique de nuages de points.

Script_Régression_Linéaire_2. Compléter le script précédent pour tracé une droite entre les points (x,y) = (1,2) et (5,3).





Le plus simple pour tracer une ligne droite entre les points (x1,y1) et (x2,y2) est d'écrire la commande suivante: plt.plot([x1, x2], [y1, y2]).

Vous pouvez alors ajouter des arguments supplémentaires comme la couleur c='r', l'épaisseur de la ligne lw=2, etc. (voir matplotlib pyplot pour avoir l'ensemble des arguments possibles).

https://moonbooks.org/Articles/Tracer-une-ligne-droite-avec-matplotlib/

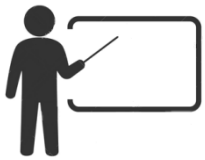
3 Calcul de la régression linéaire

Nous allons pouvoir maintenant calculer les valeurs a et b permettant de tracer la droite de régression linéaire. Sans surprise nous allons décliner toutes les étapes du calcul :

3.1 Calcul des moyennes moy_x et moy_y

Script_Regression_Lineaire_3.py Test_Dataset.csv

Script_Régression_Linéaire_3. Compléter le script pour calculer les moyennes moy_x et moy_y



Résultats :

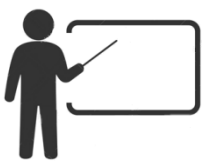
```
Indiquez le nom complet du fichier à traiter :
Choix par défaut : Test_Dataset.csv

Moyenne X=      3.000
Moyenne Y=      2.800
```

3.2 Calcul des variances V_x et V_y

Script_Regression_Lineaire_4.py Test_Dataset.csv

Script_Régression_Linéaire_4. Compléter le script pour calculer les variances V_x V_y



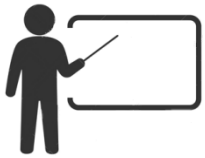
Suite des résultats :

```
Variance X=     2.000
Variance Y=     1.760
```

3.3 Calcul de la Covariance Cov_{xy}

Script_Regression_Lineaire_5.py Test_Dataset.csv

Script_Régression_Linéaire_5. Compléter le script pour calculer la Covariances Cov_{xy}



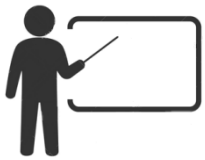
Suite des résultats :

```
Coariance XY=     1.600
```

3.4 Calcul des coefficients a et b

Script_Regression_Lineaire_6.py Test_Dataset.csv

Script_Régression_Linéaire_6.
Compléter le script pour calculer les coefficients de la droite de régression linéaire.



Suite des résultats :

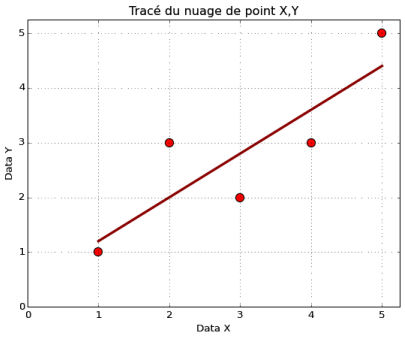
```
Droite de régression linéaire y = a.x + b
Coef.a X=     0.800
Coef.b Y=     0.400
```

3.5 Tracé de la droite du modèle de régression linéaire

Script_Regression_Lineaire_7.py Test_Dataset.csv

Script_Régression_Linéaire_7. Compléter le script pour tracer la droite à partir des coefficients a et b déterminés précédemment.

Suite des résultats :



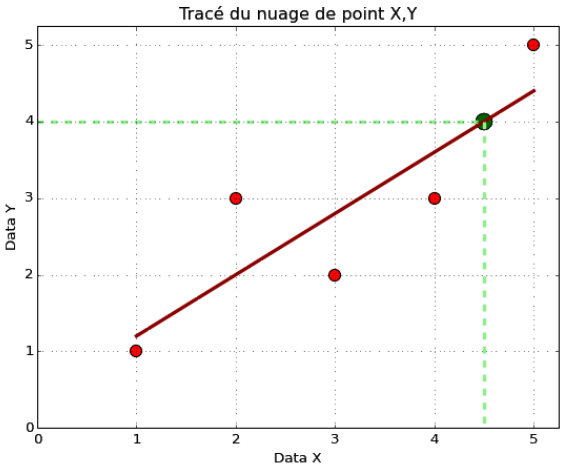
Le plus simple pour tracer une ligne droite entre les points (x1,y1) et (x2,y2) est d'écrire la commande suivante: plt.plot([x1, x2], [y1, y2]).

Vous pouvez alors ajouter des arguments supplémentaires comme la couleur c='r', l'épaisseur de la ligne lw=2, etc. (voir matplotlib pyplot pour avoir l'ensemble des arguments possibles).

https://moonbooks.org/Articles/Tracer-une-ligne-droite-avec-matplotlib/

3.6 Utilisation du modèle pour faire une prédiction

A ce stade nous avons réalisé une première analyse du Machine Learning. A partir de quelques points nous avons obtenu une modélisation qui permet maintenant d'effectuer des prédictions. C'est-à-dire de prédire des résultats pour des points qui ne font pas partie des valeurs initiales.

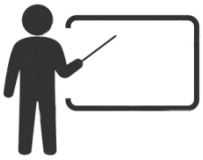


Prédiction

Nous pouvons en utilisant l'équation de la droite de régression connaître l'ordonnée y de n'importe quelle valeur de départ x.

Le tracé est ajouté sur le graphique en traçant deux segments de droites et un point.

Script_Régression_Linéaire_8. Compléter le script précédent pour calculer et tracer comme ci-dessus la prédiction pour x=4.5

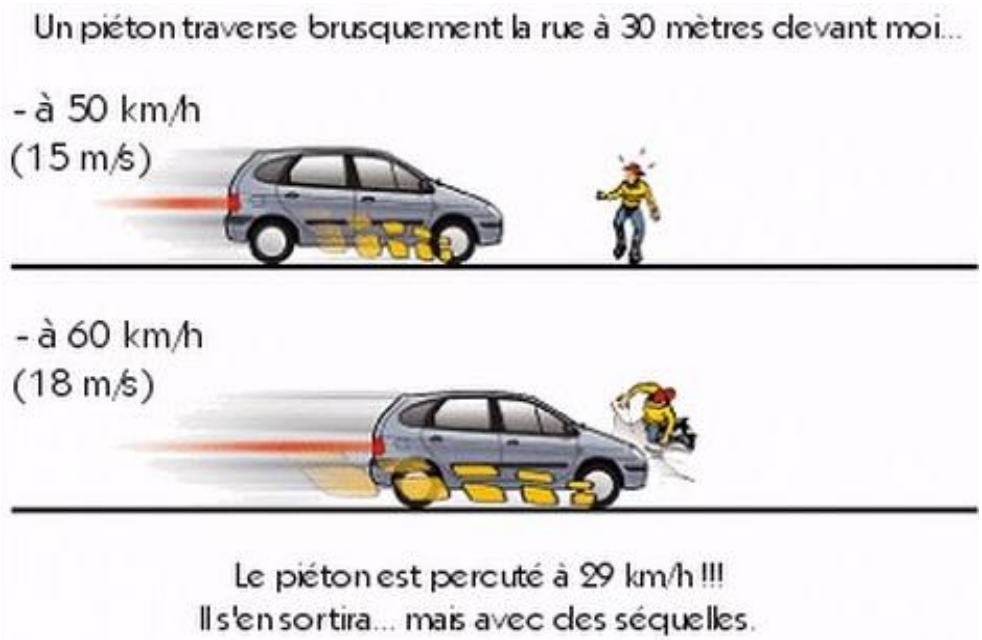


4 Application : Analyse de la distance de freinage

Nom :	**Note : / 20**
	Classe :

Vous pouvez maintenant réaliser une analyse complète basée sur la méthode détaillée aux paragraphes précédents.

L'étude porte sur l'analyse de la distance de freinage en fonction de la vitesse initiale.



Le fichier dataset est au format csv :

Cars_Dataset.csv

```
1  "","speed","dist"
2  "1",4,2
3  "2",4,10
4  "3",7,4
5  "4",7,22
6  "5",8,16
7  "6",9,10
8  "7",10,18
9  "8",10,26
10 "9",10,34
```

Dans ce fichier la vitesse est donnée en mph et la distance est donnée en pieds.

Vous devrez convertir ces données dans le système métrique.

Q6. Indiquez la relation permettant de passer d'une donnée de vitesse en mph en km•h^{-1}

Q7. Indiquez de même comment convertir les distances de pied en mètres ?

Compléter le script suivant pour réaliser l'étude du freinage :

Script Projet Freinage à compléter :

Script_Regression_Lineaire_Freinage.py

Ajouter ensuite les deux prédictions sur le graphe pour
x = 28 km•h^{-1} et x = 37 km•h^{-1}

Noter ici vos résultats numériques :

X	Y
28 km•h^{-1}	
37 km•h^{-1}	

Exemple de résultats à obtenir :