6 mars 2018

EXERCICES D'AJUSTEMENT

1. INTRODUCTION

Le but de l'exercice est de mettre en place des algorithmes d'ajustement (fitting).

2. MOINDRES CARRES

Le code a été réalisé en Python en utilisant les librairies *numpy* pour les matrices et *matplotlib* pour le tracé des graphes.

A. Fonctions moindre carrés

Pour écrire la fonction moindre carré, on utilise la formule suivante :

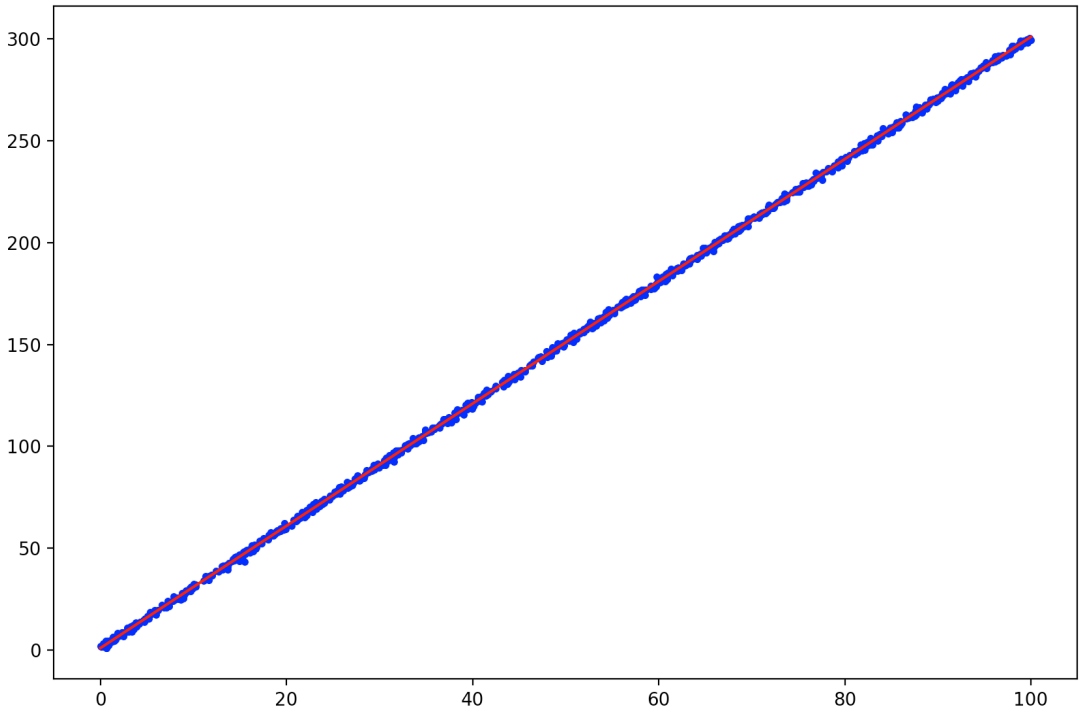
$$h = (X^T.X)^{-1}.X^TY$$

Avec X une matrice à 2 colonnes, la première colonne les valeurs en abscisses des points mesurés, la deuxième colonne une colonne de 1.

Avec Y la matrice des valeurs en ordonnées des points mesurés.

B. Tracé sur le jeu de données sans bruit

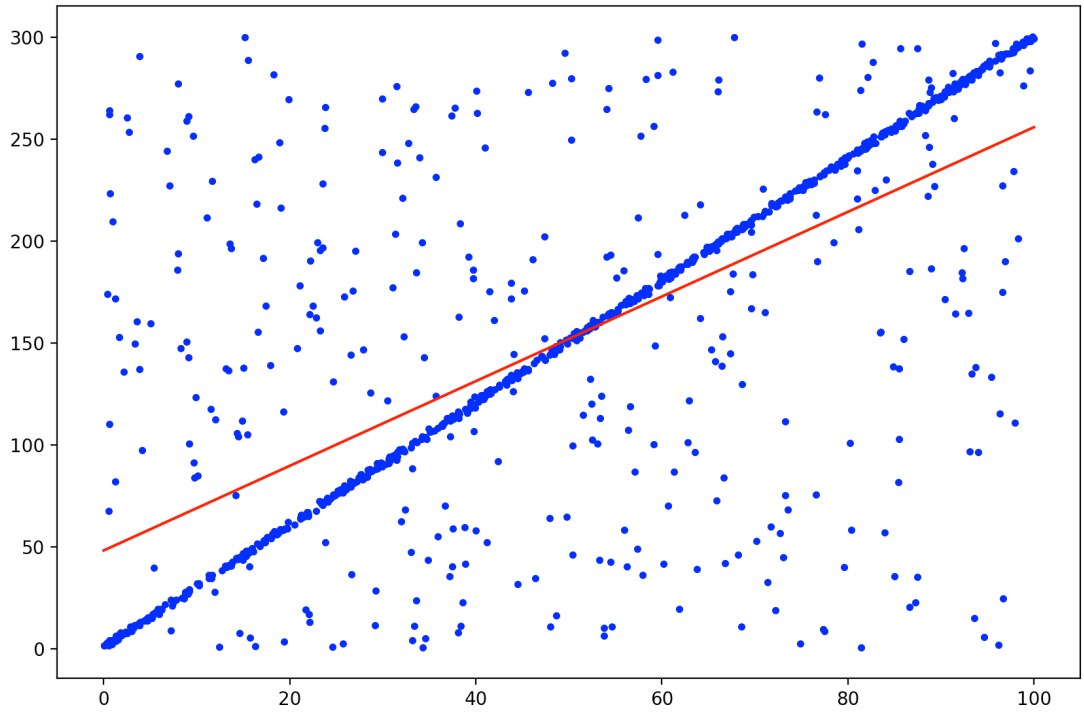
Sur le jeu de donnée *data.txt* qui ne contient pas de bruit, on obtient le résultat suivant :



Le résultat est très satisfaisant.

C. Tracé sur le jeu de données avec bruits

Sur le jeu de donnée *dataOutliers.txt* qui contient environ 30% de bruit, on obtient le résultat suivant :



Le défaut de cette approche est qu'elle considère tous les points donc est très influencée par le bruit.

3. RANSAC

A. Remarques sur l'implémentation

Pour écrire la fonction qui calcule les paramètres d'une droite à partir de 2 points, on reprend tout simplement la fonction des moindres carrés dans le cas particulier à 2 points.

On suit l'algorithme Ransac naïf pour son implémentation (le tirage des points est « complètement » aléatoire, sans mémoire ni optimisation).

B. Tracés et études des résultats

On trace la droite obtenue avec l'algorithme RANSAC :

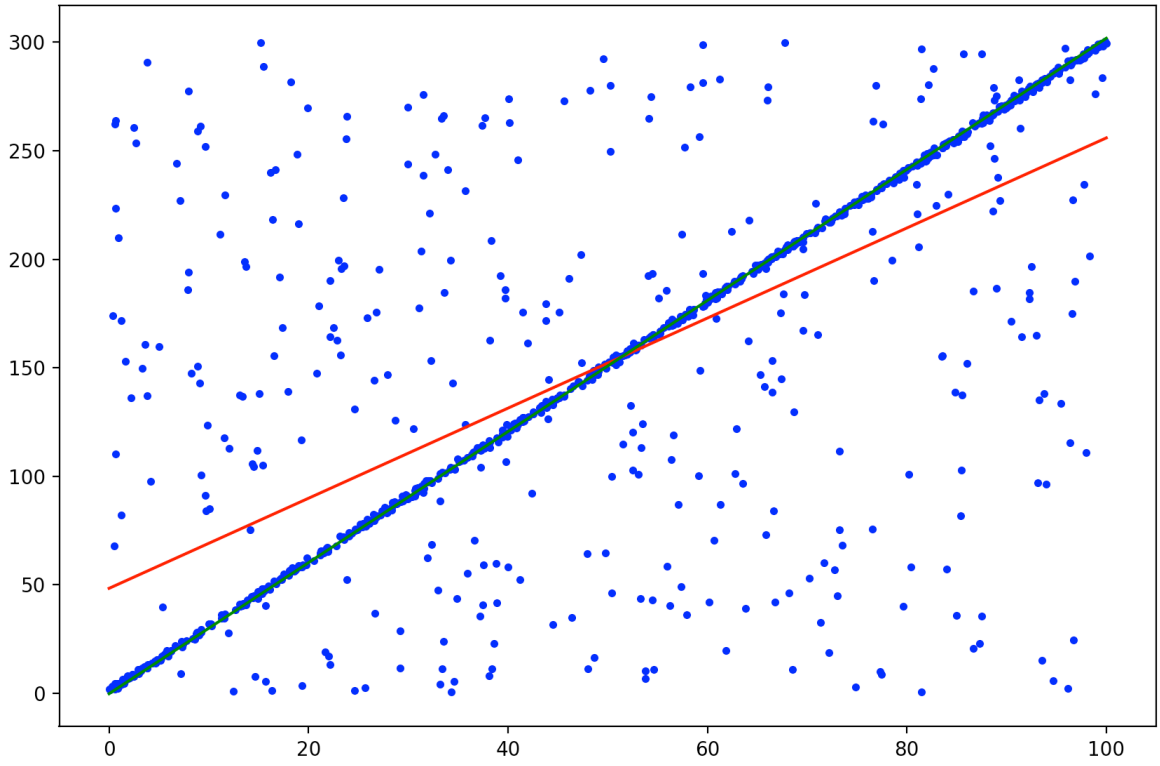


Figure 1 : Résultat obtenu par Ransac en vert pour delta = 0.1 et iter = 7

On a pris le nombre d'itérations 7 car il correspond à une probabilité de 99% d'obtenir une bonne estimation. La droite en vert est très satisfaisante, totalement confondues avec les points intéressants du jeu de données. Le résultat de l'algorithme des moindres carrés est constant, la droite est donc toujours aussi mauvaise.

Si on prend un nombre d'itérations trop petit, ce qui correspond à un nombre d'essais de modélisation trop petit, on peut avec le Ransac obtenir de mauvais résultats :

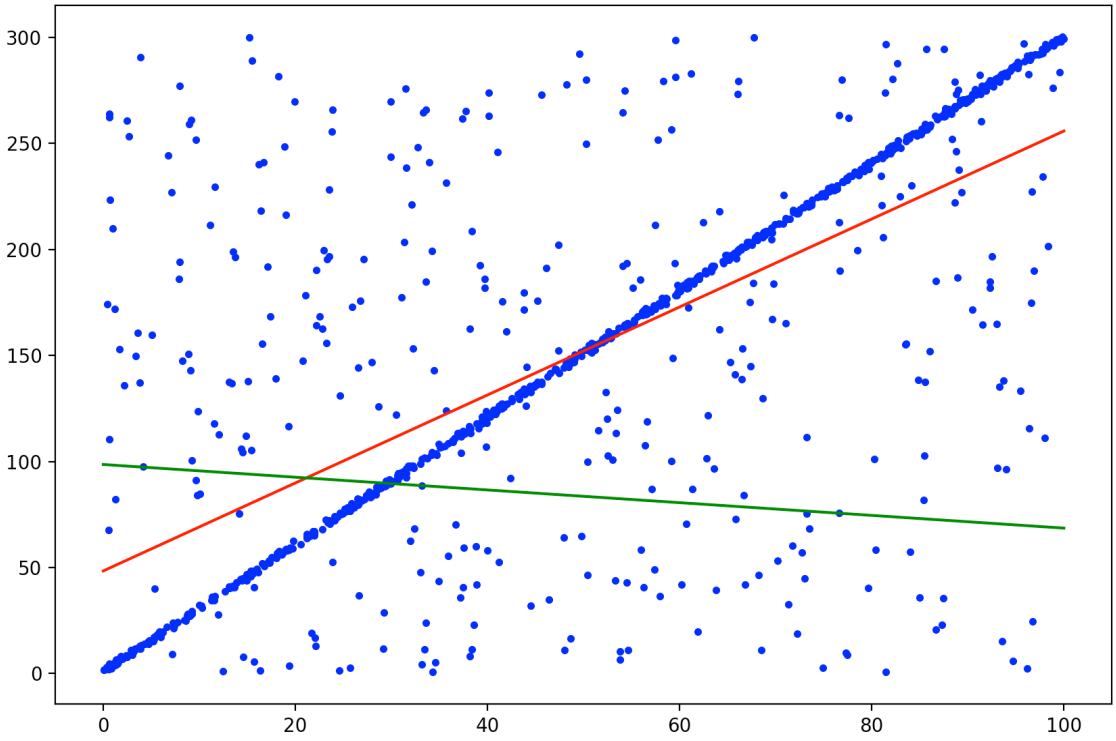


Figure 2 : Résultat obtenu par Ransac en vert pour delta = 0.1 et iter = 7

Si on prend un seuil de distance trop permissif, ce qui correspond à « accepter » le bruit, on peut aussi obtenir de mauvais résultats :

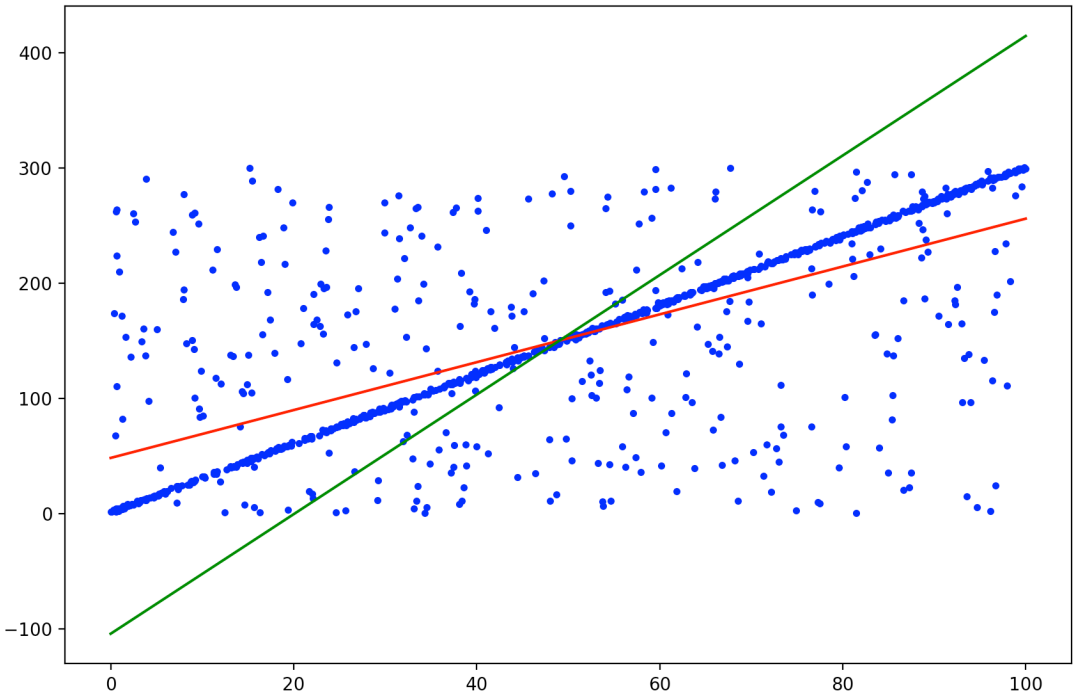


Figure 3 : Résultat obtenu par Ransac en vert pour delta = 300 et iter = 7

4. CONCLUSION

L'algorithme stochastique semble être très performant si l'on a une idée du seuil delta que l'on doit considérer. Il donne de plus de très bons résultats avec un nombre faibles d'itérations comme montrés dans le calcul de probabilité dans le cours.