

Projet 4

Zouiche Omar

20 Février 2021

0.0.1 Projet 4

Zouiche Omar On souhaite modéliser la tendance d'un ensemble de données en fonction d'années qu'on trouve dans le fichier temperature.txt. On utilisera la méthodes des moindres carrées et la droite exprimée par la fonction $T = at + b$ avec T étant la donnée et t la période, donc on année. Dans le fichier on a des données sur 133 ans, de 1880 à 2013. On veut trouver a et b tel que $J(a, b) = \sum_{i=0}^n |X[i] - b - a * year[i]|^2$ soit minimal. Nous commencerons par extraire les données du fichier temperature.txt, après le traitement des données et la minimisation, on comparera les résultats avec ceux de np.polyfit

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

### Importation du fichier au format Pandas Dataframe
read_file = pd.read_csv (r'https://imag.umontpellier.fr/~mohamadi/MOML/python/
    ↳temperature.txt', 'r')
read_file.columns=['col']

### Séparation des variables pour définir trois grandeurs différentes
read_file = read_file.col.str.split(expand=True)
read_file.columns = ['Year', 'X', 'Y']

### Changement du type des données, de "string" à "float"
Year = read_file.Year.astype(float)
X = read_file.X.astype(float)
Y = read_file.Y.astype(float)

### Application de la fonction numpy polyfit sur X et Y

mx, bx = np.polyfit(Year, X, 1)
mxx, bxx, cxx = np.polyfit(Year, X, 2)
my, by = np.polyfit(Year, Y, 1)
myy, byy, cyy = np.polyfit(Year, Y, 2)

'''
```

```

print('Le modèle linéaire optimal pour les données X est y = ax+b avec a=', mx,
      ' et b=', bx)
print('Le modèle linéaire optimal pour les données Y est y = ax+b avec a=', my,
      ' et b=', by)
print('Le modèle quadratique optimal pour les données X est y = ax2+bx+c avec
      ' a=', mxx, ' , b=', bxx, ' et c =', cxx)
print('Le modèle quadratique optimal pour les données Y est y = ax2+bx+c avec
      ' a=', myy, ' , b=', byy, ' et c=', cyy)
'''
### Tableau complet
data = pd.concat([Year,X,Y], axis = 1)
data

```

```

[1]:
      Year      X      Y
0   1881.0 -0.10 -0.17
1   1882.0 -0.08 -0.16
2   1883.0 -0.19 -0.19
3   1884.0 -0.26 -0.22
4   1885.0 -0.30 -0.27
..      ...      ...      ...
128  2009.0  0.64  0.63
129  2010.0  0.72  0.62
130  2011.0  0.60  0.65
131  2012.0  0.63  0.67
132  2013.0  0.65  0.70

```

[133 rows x 3 columns]

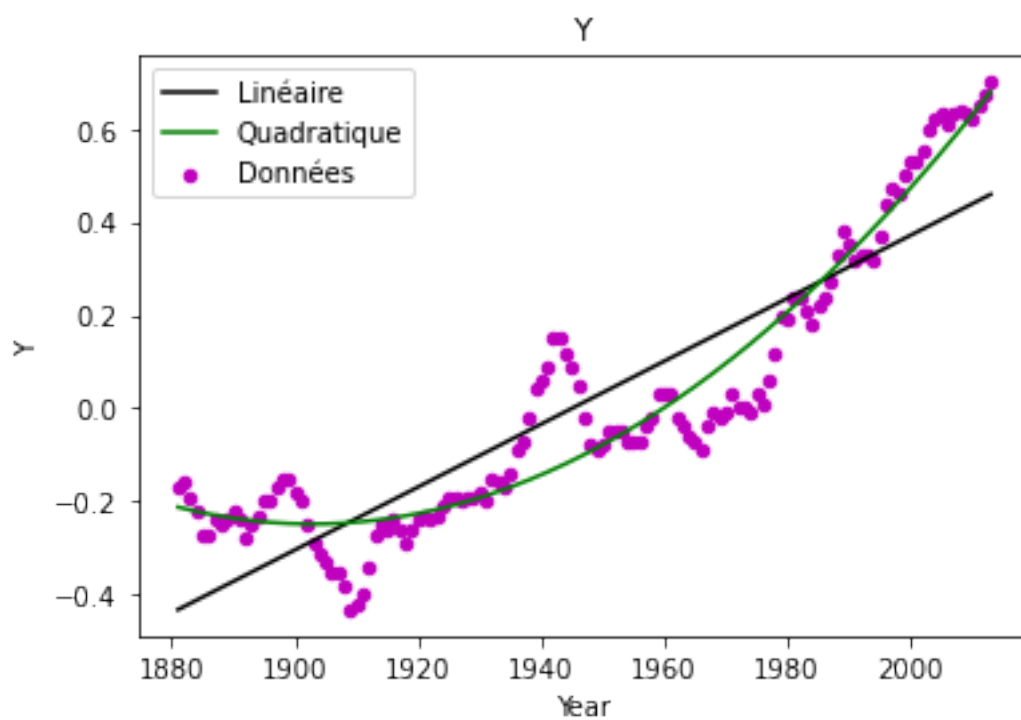
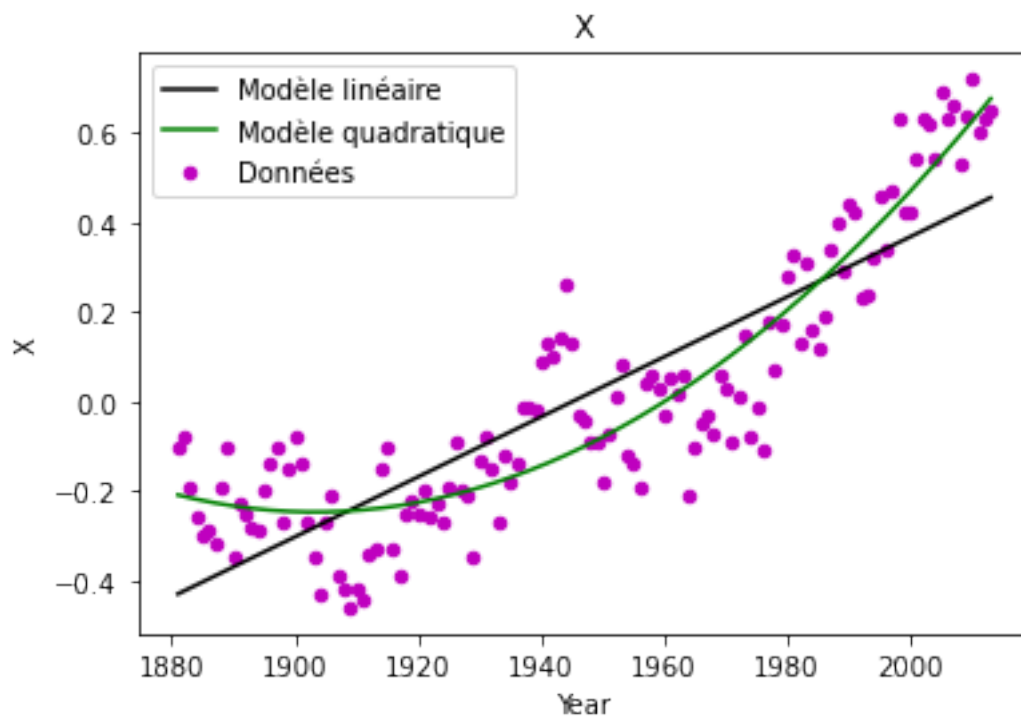
```

[6]: plt.figure(1)
data.plot.scatter('Year','X', label = 'Données', c='m')
plt.plot(Year, mx*Year + bx, label = 'Modèle linéaire', c= "k")
plt.plot(Year,mxx*Year**2 + bxx*Year + cxx, label = 'Modèle quadratique', c =
      'g')
plt.title('X')
plt.legend()

plt.figure(2)
data.plot.scatter('Year','Y', label = 'Données', c='m')
plt.plot(Year, my*Year + by, label = 'Linéaire', c= "k")
plt.plot(Year, myy*Year**2 + byy*Year + cyy, label = 'Quadratique', c = 'g')
plt.title('Y')
plt.legend()
plt.show()

```

<Figure size 432x288 with 0 Axes>



```
[4]: erra = np.zeros(134)
      errb = np.zeros(134)

      for i in range(0,134):
          erra[i] = abs(np.linalg.solve(mat1,mat2)[1] - mx)
          errb[i] = abs(np.linalg.solve(mat1,mat2)[0] - bx)

      erreura = np.linalg.norm(erra)
      erreurb = np.linalg.norm(errb)

      print('Erreur a : ',erreura,'Erreur b : ', erreurb)
```

Erreur a : 6.616648651670449e-15 Erreur b : 1.2790072207621109e-11