TP semaine7 :

Exercice 1 :

---------------------------------------------------------------CODE-----------------------------------------------------------

from pulp import \*

import numpy as np

import math

import matplotlib.pyplot as plt

file=open("./confiserie.csv",'r')

texte=file.readlines()

file.close()

#Exercice 1

x=[]

y=[]

for f in texte:

    a=f.split(',')

    x.append(float(a[1]))

    y.append(float(a[2]))

#creation du modele

model = LpProblem(name="Ex1TME\_4", sense=LpMinimize)

#nombre de variables

n=len(x)

#variable a

a = LpVariable("a")

#variable b

b = LpVariable("b")

#Changement de la population des USA par 5.27

for i in range(len(x)):

    if(y[i]==5142.2):

        x[i]=5.27

#n variables z

z = LpVariable.matrix("z", list(range(1,n+1))) #n variables

#Fonction objectif

model += lpSum(z) #on cherche à minimiser la somme des x

#contraintes

for i in range(len(x)):

    model += y[i]-a-b\*x[i] <= z[i]

    model += a+b\*x[i]-y[i] <= z[i]

#Résolution du problème

status = model.solve(solver=GLPK(msg=True,keepFiles=True)) #keepFiles permet de créer un fichier .sol contenant la solution

# Affiche le statut de la solution (optimale, non borné, etc.)

print("Status:", LpStatus[model.status])

# Affiche la valeur de la fonction objectif

print("objective=", value(model.objective))

#Affiche les valeurs optimales des variables de décision :

for i in range(0,n):

    print("z[",i,"]=",value(z[i]))

print("a =",value(a))

print("b =",value(b))

#Ancien tp

#equation de la droite linéaire est agale a y\*=a+bx

    #a = moyenneY-b\*moyenneX

    #b= Sxy/Sxx

#moyenne des x

moyenneX=0

i=0

for j in x:

    moyenneX+=j

    i=i+1

moyenneX=moyenneX/i

moyenneY=0

#moyenne des y

i=0

for j in y:

    moyenneY+=j

    i=i+1

moyenneY=moyenneY/i

#Sxy

Sxy=0

for p in range(len(x)):

    Sxy+=(x[p]-moyenneX)\*(y[p]-moyenneY)

#Sxx

Sxx=0

for j in x:

    Sxx+=((j-moyenneX)\*(j-moyenneX))

#Syy

Syy=0

for j in y:

    Syy+=((j-moyenneY)\*(j-moyenneY))

#b

b1=Sxy/Sxx

#a

a1=moyenneY-b1\*moyenneX

# affichage de la courbe

xx=np.linspace(0,max(x),100)

yy=value(a)+value(b)\*xx

yy1=a1+b1\*xx

plt.plot(xx,yy,'-r')

plt.plot(xx,yy1,'-b')

plt.scatter(x,y)

plt.show()

--------------------------------------------------------------------------------------------------------------------

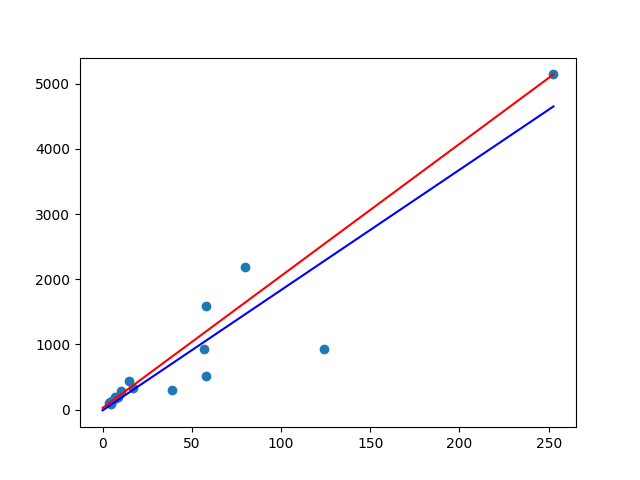
Q :1)

Valeur pour a et b :

a = 74.9407 b = 14.593

Q :2)

Nous pouvons observer que la droite de régression linéaire faites à partir de la méthode de (Least Absolute Deviation) en rouge est plus représentative des observations. On peut donc conclure que c’est la méthode de régression optimale.



1.3) Nous observons ici que la régression LAD en rouge est moins sujette au valeur absurde. Quant aux MCO une valeur absurde modifie grandement ca pente de régression.