Régression logarithmique pour parties imaginaires des 28 premiers zéros de ζ
Denise Vella-Chemla, 23 novembre 2024

On utilise la librairie python `limfit`, qui permet d'effectuer une régression logarithmique pour trouver par une formule de la forme $y = a \ \ln \ (x + b) + c$ les parties imaginaires des zéros non triviaux de la fonction ζ de Riemann.

On utilise le programme fourni un peu plus loin.

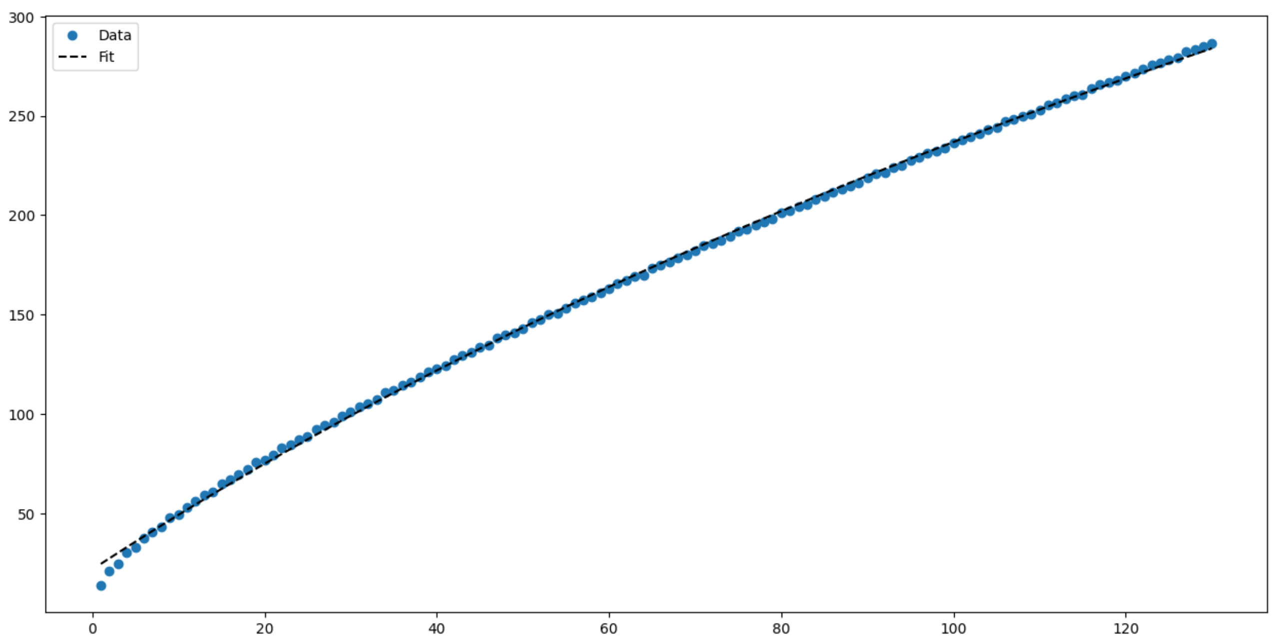
Ce programme dessine initialement le graphique de corrélation ci-après, il fournit en sortie les éléments caractéristiques de la régression logarithmique effectuée, qui sont reportés ci-dessous, puis on fait imprimer au programme, en dernier lieu, les valeurs trouvées en utilisant la fonction

$$f(x) = 402 \ln \ (x + 147) - 1978$$

à des fins de vérification.

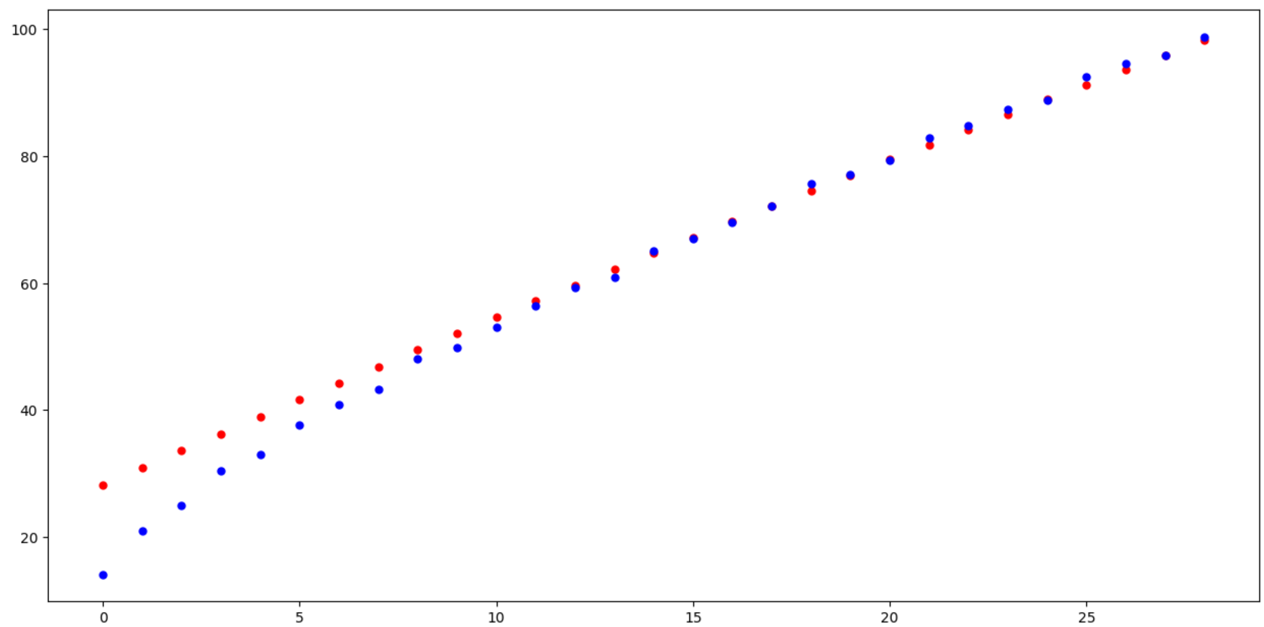
```
"""
Soit (z_k)_{1<=k<=130} la liste des parties imaginaires des 130 premiers zeros non triviaux de la
fonction zeta de Riemann.
On souhaite ajuster ces donnees a un modele logarithmique de parametres a,b,c reels, avec b0,
soit :

        y_k = f(x_k) = a ln(x_k + b) + c,    x_k = k,    1 <= k <= 130.

"""

import lmfit
import matplotlib.pyplot as plt
import numpy as np

z = np.array([  14.134725141734693790457251983562 5,  21.022039638771554992628479593896 9,
                25.010857580145688763213790992562 8,  30.424876125859513210311897530584 0,
                32.935061587739189690662368964074 7,  37.586178158825671257217763480705 3,
                40.918719012147495187398126914633 4,  43.327073280914999519496122165406 8,
                48.005150881167159727942472749427 7,  49.773832477672302181916784678563 8,
                52.970321477714460644147296608880 8,  56.446247697063394804367759476706 0,
                59.347044002602353079653648674992 2,  60.831778524609809844259901824524 1,
                65.112544048081606660875054253183 6,  67.079810529494173714478828896522 1,
                69.546401711173979252926857526554 7,  72.067157674481907582522107969826 1,
                75.704690699083933168326916762030 5,  77.144840068874805372682664856304 7,
                79.337375020249367922763592877116 1,  82.910380854086030183164837494770 6,
                84.735492980517050105735311206827 6,  87.425274613125229406531667850919 1,
                88.809111207634465423682348079509 5,  92.491899270558484296259725241810 5,
                94.651344040519886966597925815208 0,  95.870634228245309758741029219246 7,
                98.831194218193692233324420138622 4]

def f(x, a, b, c):
     return a*np.log(x+b) + c
```

```
model = lmfit.Model(f)
params = model.make_params(a=1, b=dict(value=0.1, min=0.001), c=0.1)
x = np.arange(1, len(z)+1)
r = model.fit(z, params, x=x)
y = r.best_fit
print(r.fit_report())
_, ax = plt.subplots(figsize=(15, 5))
plt.plot(x, z, "o", label="Data")
plt.plot(x, y, "k--", label="Fit")
plt.legend()
plt.show()
print('Verification')
import math
from math import log

for x in range(29):
     res1 = 402*log(x+147)-1978
     res2 = z[x]
     print(x,' --> ',res1,' ',res2)
     plt.scatter(x,res1,s=25,color='red')
     plt.scatter(x,res2,s=25,color='blue')
plt.show()
```



```
[[Model]]
Model(f)
[[Fit Statistics]]
fitting method = leastsq
function evals = 102
data points = 130
variables = 3
chi-square = 457.447882
reduced chi-square = 3.60195183
Akaike info crit = 169.556706
Bayesian info crit = 178.159310
R-squared = 0.99936890
[[Variables]]
a:  402.423515 +/- 10.6902515 (2.66%) (init = 1)
b:  141.683692 +/- 5.22597956 (3.69%) (init = 0.1)
c:  -1971.54485 +/- 67.3104952 (3.41%) (init = 0.1)
[[Correlations]] (unreported correlations are < 0.100)
C(a, c) = -0.9999
C(b, c) = -0.9975
C(a, b) = +0.9965

  0  -->  28.153899885051942  14.134725142
  1  -->  30.879334053174034  21.022039639
  2  -->  33.58641499007467   25.01085758
  3  -->  36.27538822669476   30.424876126
  4  -->  38.94649439959949   32.935061588
  5  -->  41.59996938020322   37.586178159
  6  -->  44.236044399759066  40.918719012
  7  -->  46.85494617027916   43.327073281
  8  -->  49.4568970015373    48.005150881
  9  -->  52.04211491431397   49.773832478
 10  -->  54.61081375001959   52.970321478
 11  -->  57.163203276840704  56.446247697
 12  -->  59.699489292533144  59.347044003
 13  -->  62.21987372399826   60.831778525
 14  -->  64.72455472375418   65.112544048
 15  -->  67.21372676341821   67.079810529
 16  -->  69.68758072431842   69.546401711
 17  -->  72.1463039853279    72.067157674
 18  -->  74.59008050803322   75.704690699
 19  -->  77.01909091933067   77.144840069
 20  -->  79.43351259153587   79.33737502
 21  -->  81.83351972010996   82.910380854
 22  -->  84.21928339907572   84.735492981
 23  -->  86.59097169420556   87.425274613
 24  -->  88.94874971406944   88.809111208
 25  -->  91.29277967900816   92.491899271
 26  -->  93.62322098810728   94.651344041
 27  -->  95.9402302842409    95.870634228
 28  -->  98.24396151725296   98.831194218
```

Vérification : on applique la formule pour la vérifier en plottant les parties imaginaires effectives des premiers zéros non triviaux de ζ (en bleu) et les valeurs calculées par la formule logarithmique (en rouge). Une seule couleur apparaît quand les zéros se chevauchent.



Mais en allant jusqu'à 1000 zéros, on obtient le graphique ci-dessous, la régression logarithmique $y = 402 \ln(x + 147) - 1978$ n'est pas une bonne idée du tout.