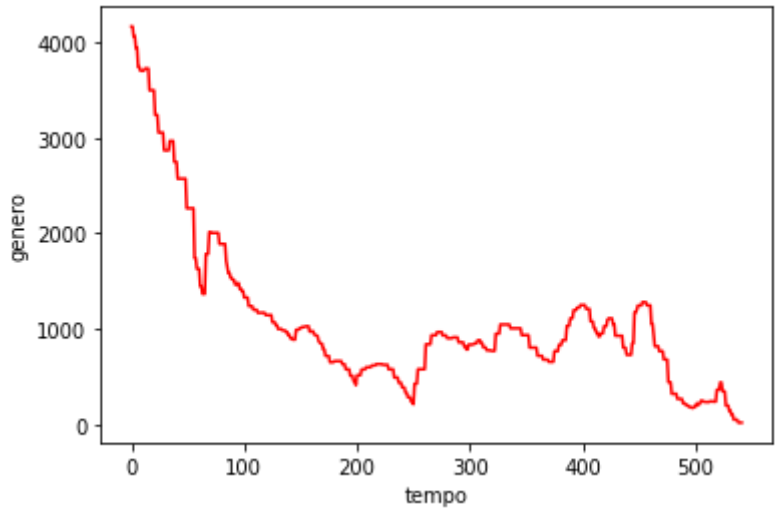
Comp4, CCM

EP2: Ciclos na diversidade da vida na Terra

(1):O arquivo da informações do tempo (MA) x diversidade, o arquivo .csv foi interpretado com algoritimos básicos de leitura e o gráfico da evolução temporal da variedade foi construído através do matplot

```
import matplotlib.pyplot as plt
import random
#Após upload, acesso pela função
arquivo = open('diversity[2].csv', 'r')
#Ler linhas por linhas com delimitadores e armazenamento
l=arquivo.readlines()
div=[]
t=[]
for i in range(1,len(l)):
  colunas=l[i].split(';')
  div.append(int(colunas[1]))
  t.append(int(colunas[0]))
#plot do gráfico de evolução temporal
plt.xlabel('tempo')
plt.ylabel('genero')
plt.plot(t, div, 'r')
```

```
[<matplotlib.lines.Line2D at 0x7f642663c1d0>]
```

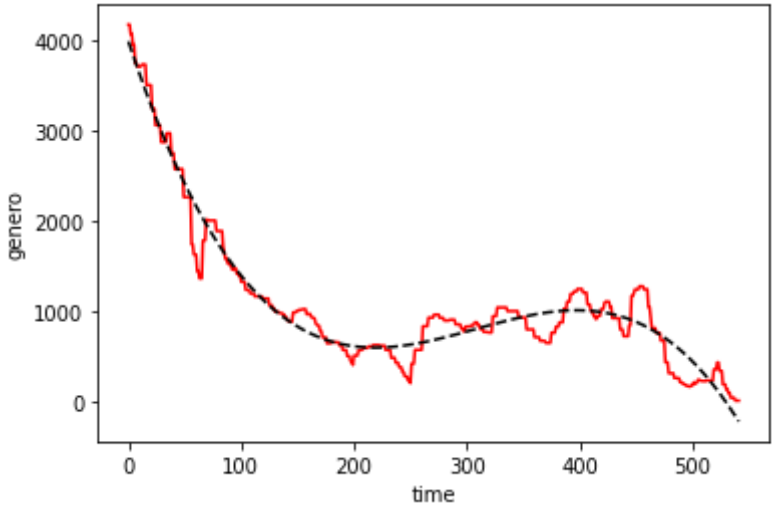


(2)Ajuste o polinômio de terceiro grau(faça uma regressão através de mínimos quadrados). Essa função já está disponível no python

```
#função de ajuste polinomial pelos metodos de minimos quadrados, no caso polinomio de grau
import numpy as np
```

```
pmin=np.polyfit(t,div,3)
#suas componentes
poli= np.poly1d(pmin)
#ajuste
#plot
plt.xlabel('time')
plt.ylabel('genero')
plt.plot(t,div,'r',t,poli(t),'k--')
```

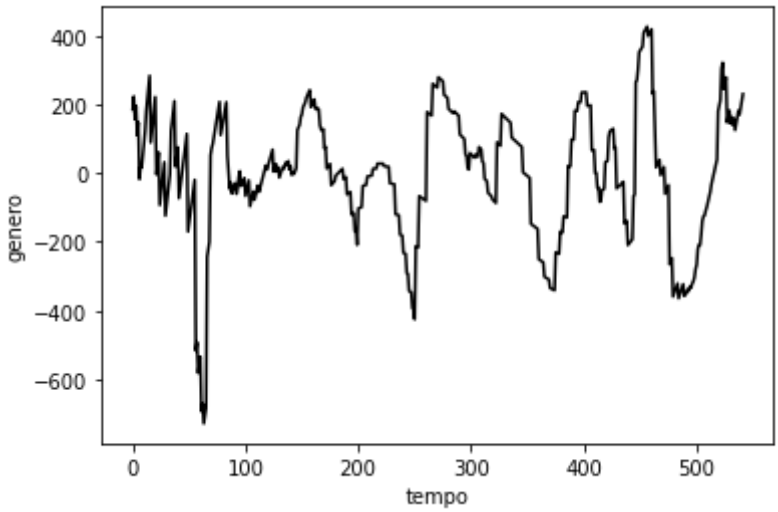
```
[<matplotlib.lines.Line2D at 0x7f64265f62b0>,
 <matplotlib.lines.Line2D at 0x7f64265f6400>]
```



(3)Subtraindo o polinômio e gerando uma nova série livre de tendência

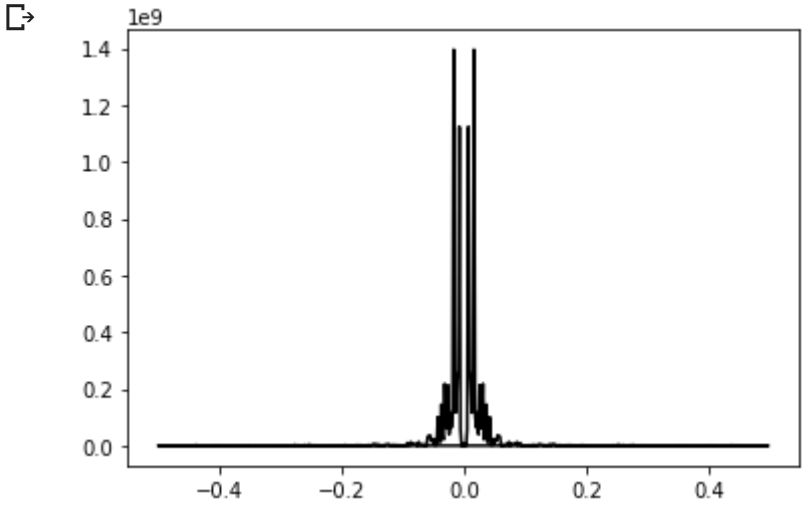
```
#novo plot da diferença
dif= div - poli(t)
#variáveis usadas posteriormente que já conhecemos seu valor
n=len(t)
tamadif=len(dif)
#plot da diferença
plt.xlabel('tempo')
plt.ylabel('genero')
plt.plot(t, dif, 'black')
```

```
[<matplotlib.lines.Line2D at 0x7f642654c390>]
```



(4)Espectro de fourier através de funções do python Scipy e aparecimento de dois picos

```
#função de fourier
def ff(f, t, N):
#dif entre steps
          b = t[1] - t[0]
        #calculado no fft
          sinal = np.fft.fft(f)
          frequencia = np.fft.fftfreq(N,b)
          freqq= np.linspace(0,1/b,tamadif)

          amps = []
          for k in sinal:
            amps.append(k*np.conjugate(k))
          return frequencia, amps

def esp(frequencia, amps): #refratad
     plt.plot(frequencia, np.real(amps), 'black')
f, c = ff(dif, t, n)
esp(f, c)
```



(5)

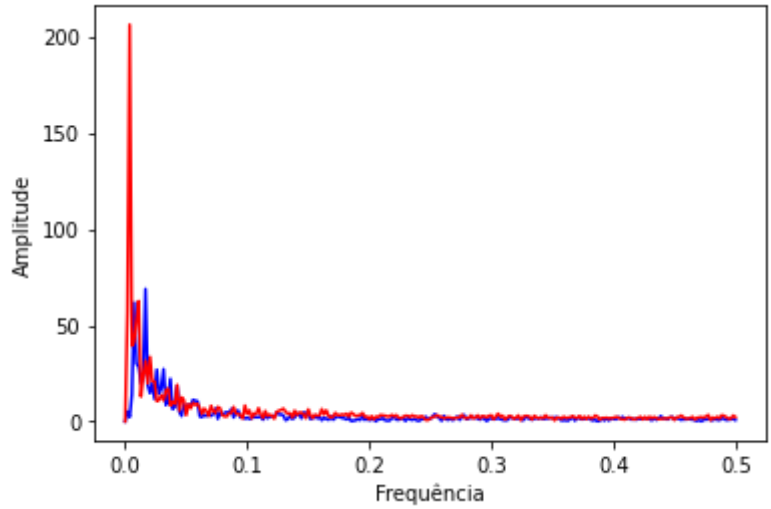
(a)Variações em cada passo da série temporal pode ser encontrada pela função„ $\triangle n(k)$. A seguir do embaralhamento afim de construir uma nova série temporal através do suffle. Para cada $div[i]$ repetimos a interpolação poli. em seguida sua esanálise livre de tendencias seguido do espectro de fourier.

```
#array que guardara as variações
variacoes = []
#interações
for i in range(len(t)-1):
  variacoes.append(div[i+1]-div[i])
```

```
from random import shuffle
from random import choice
#Função shuffle para começarmos o embaranhamento com n
tp= variacoes
tmp1=[tp]
for i in range(1000):
  random.shuffle(tp)
  tmp1.append(tp)
  tp=variacoes



from scipy.fft import fft
conj1,conj2=[],[]


tmp2=[tep]
tep=div


#array gerador
for i in range(1000):
  tep[0]=random.choice(div)
  amostra=tmp1[i]
  m=0
  for j in range(len(amostra)):
    m+=amostra.pop()
    tep[j+1]= m + tep[0]

  tmp2.append(tep)




#polo

for i in range(1000):
  d=np.array(tmp2[i])
  polo=np.polyfit( t , d, 3)
  Pol=np.poly1d(polo)
  v=d-Pol(t)
  ft=fft(v)

  s=np.abs(ft)[:tamadif//2]*(1/tamadif)
  conj2.append(s)


b=t[0]-t[1]
freqq= np.linspace(0,1/b,tamadif)
W=freqq[:tamadif//2]
W=W*(-1)
```

```
WL=conj2[0].size
tempo=[]
conj3=[]



for i in range(WL):
  tep=[]

  for j in range(1000):
    tep.append(conj2[j][i])

  conj3.append(tep)


tep=[]
conj4=[]
for i in range(WL):
  tep=conj3[i]
  m=np.max(tep)
  conj4.append(m)



print(conj4)


pia=fft(dif)


MA=np.array(conj4)
org=np.abs(pia)[:tamadif//2]*(1/tamadif)

plt.xlabel('Frequência')
plt.ylabel('Amplitude')
plt.plot(W,org,'b',W,MA,'r')
```

```
[<matplotlib.lines.Line2D at 0x7f642649f3c8>,
 <matplotlib.lines.Line2D at 0x7f642649f4e0>]
```



```
f_r=[]
for i in range(WL):
  mag=org[i]
  maj=MA[i]
  if(mag>maj or mag==maj):
     f_r.append(W[i])
```

(6) As frequências relevantes analisadas na 5 são:

```
print(f_r)
```

```
[0.0073937153419593345, 0.012939001848428836, 0.0166358595194085, 0.0258780036968576
```

(7)Após o embaralhamento de n vezes, geramos a partir dela uma série livre de tendências. Após isso é feito o calculo do espectro de fourier dos dados obtidos. No final temos resultados interessantes sobre frequências relevantes no problema.