

2 Modules et fichiers

July 26, 2024

```
[11]: import matplotlib.pyplot as plt
```

```
[12]: def file2str(nomfic:str)->str:
      """
      Retourne la chaîne que contient le fichier de nom nomfic.
      """
      f = open(nomfic, 'r')
      ch = f.read()
      f.close()
      return ch
```

```
[13]: ch = file2str('texte_reference.txt')
```

```
[14]: len(ch)
```

```
[14]: 70904
```

```
[15]: def occ(ch:str)->dict:
      """
      Retourne un dictionnaire qui associe
      à chaque lettre de la chaîne ch son
      nombre d'occurrences dans ch.
      """
      d = {}
      for car in ch:
          if car in d:
              d[car] +=1
          else:
              d[car] = 1
      return d
```

```
[16]: occ(ch)
```

```
[16]: {'p': 1936, 'r': 3955, 'e': 10745, 'a': 4127, 'm': 1458, 'b': 486, 'u': 3116,
      'l': 3848, ' ': 11666, 'f': 419, 'n': 4391, 'c': 2100, 'i': 4649, 's': 4977,
      'o': 3535, 't': 4514, 'h': 117, 'x': 341, 'd': 2628, 'v': 661, 'q': 484, 'g':
      508, 'y': 37, 'j': 180, 'z': 25, 'w': 1}
```

```
[17]: def freq(ch:str)->dict:
      """
      Retourne un dictionnaire qui associe
      à chaque lettre de la chaîne ch sa
      fréquence d'apparition dans ch.
      """
      d = occ(ch)
      d2 = {}
      for k in d:
          d2[k] = round(d[k]/len(ch), 3)
      return d2
```

```
[18]: f = freq(ch)
      f
```

```
[18]: {'p': 0.027, 'r': 0.056, 'e': 0.152, 'a': 0.058, 'm': 0.021, 'b': 0.007, 'u':
0.044, 'l': 0.054, ' ': 0.165, 'f': 0.006, 'n': 0.062, 'c': 0.03, 'i': 0.066,
's': 0.07, 'o': 0.05, 't': 0.064, 'h': 0.002, 'x': 0.005, 'd': 0.037, 'v':
0.009, 'q': 0.007, 'g': 0.007, 'y': 0.001, 'j': 0.003, 'z': 0.0, 'w': 0.0}
```

```
[19]: def tri_items(d:dict)->(list, list):
      """
      Retourne les listes des clés et des valeurs du dictionnaire
      d triées suivant l'ordre croissant des clés.
      """
      cles = []
      valeurs = []
      for k, v in sorted(d.items()):
          cles.append(k)
          valeurs.append(v)
      return cles, valeurs
```

```
[20]: cars, f = tri_items(f)
```

```
[21]: fig, ax = plt.subplots()
      ax.bar(cars, f)
      plt.show()
```

```
[22]: def csv(nomfic:str)->(list, list):
      f = open(nomfic, 'r')
      li_lignes = f.readlines()
      f.close()
      lix = []
      liy = []
      for ligne in li_lignes:
          x, y = ligne[:-1].split(',')
          lix.append(x)
```

```
    liy.append(float(y))
return lix, liy
```

```
[23]: lix, liy = csv('temp.csv')
      lix, liy
```

```
[23]: ('01/2016', '02/2016', '03/2016', '04/2016', '05/2016', '06/2016', '07/2016',
      '08/2016', '09/2016', '10/2016', '11/2016', '12/2016', '01/2017', '02/2017',
      '03/2017', '04/2017', '05/2017', '06/2017', '07/2017', '08/2017', '09/2017',
      '10/2017', '11/2017', '12/2017', '01/2018', '02/2018', '03/2018', '04/2018',
      '05/2018', '06/2018', '07/2018', '08/2018', '09/2018', '10/2018', '11/2018',
      '12/2018', '01/2019', '02/2019', '03/2019', '04/2019', '05/2019', '06/2019',
      '07/2019', '08/2019', '09/2019', '10/2019', '11/2019', '12/2019', '01/2020',
      '02/2020', '03/2020', '04/2020', '05/2020', '06/2020', '07/2020', '08/2020',
      '09/2020', '10/2020', '11/2020', '12/2020', '01/2021', '02/2021', '03/2021',
      '04/2021', '05/2021', '06/2021', '07/2021', '08/2021', '09/2021', '10/2021'],
      [7.27, 7.34, 7.72, 9.89, 14.51, 17.62, 20.03, 21.05, 19.1, 11.81, 8.79, 5.94,
      3.78, 8.04, 10.93, 10.84, 16.51, 20.27, 20.68, 19.95, 15.81, 14.66, 8.57, 6.7,
      8.62, 3.71, 8.29, 13.29, 16.17, 19.85, 22.49, 20.9, 17.78, 13.82, 8.93, 8.26,
      5.47, 7.98, 9.82, 11.64, 13.78, 19.13, 22.35, 20.37, 17.66, 14.29, 8.43, 7.94,
      7.74, 9.54, 9.5, 14.58, 16.62, 17.71, 20.09, 21.28, 18.63, 12.86, 10.44, 7.45,
      5.15, 7.66, 8.68, 10.34, 13.29, 19.14, 19.74, 18.69, 19.01, 13.03])
```

Taper dans la console l'instruction :

```
%matplotlib Qt
```

Pour faire apparaître le graphique dans une fenêtre séparée.

```
[28]: fig, ax = plt.subplots()
      ax.plot(lix, liy, marker='o')
      ax.tick_params(axis='x', rotation=90)
      plt.show()
```

Contenu du fichier `stats.py` :

```
[29]: def moyenne(li:[float])->float:
      """
      Retourne la moyenne arithmétique de la liste de nombres li.
      """
      S = 0
      for el in li:
          S += el
      return S/len(li)
```

```
[30]: from math import sqrt
```

```
[39]: def ecart_type(li:[float])->float:
      """
```

```

Retourne l'écart-type de la liste de nombres li.
"""
m = moyenne(li)
S = 0
for el in li:
    S += (el - m)**2
return sqrt( S / (len(li)-1) )

```

Contenu du fichier seq_search.py :

```

[40]: def maxi(li:[float])->float:
      """
      Retourne le max de la liste de nombres li.
      """
      M = li[0]
      for el in li:
          if el > M:
              M = el
      return M

```

```

[41]: def imax(li:[float])->int:
      """
      Retourne l'indice du max de la liste de nombres li.
      """
      iM = 0
      M = li[0]
      for i in range(len(li)):
          if li[i] > M:
              iM = i
              M = li[i]
      return iM

```

```

[43]: print('Température moyenne =', moyenne(liy))
      print('Variation moyenne de la température =', ecart_type(liy))
      print('Température max =', maxi(liy))
      print('Date de la température max =', lix[imax(liy)])

```

```

Température moyenne = 13.285000000000002
Variation moyenne de la température = 5.370781593728816
Température max = 22.49
Date de la température max = 07/2018

```

```
[ ]:
```