TP5

Ibrahim ALAME

9/12/2019

1 Parser la date

On considère une date t de l'année en cours formatée de la façon suivante : `"jj-mois. hh:mn"` , exemple t =`"26-oct. 20:53"` et on cherche à le convertir au format : `"yyyy-mm-jj hh:mn"` exemple : `"2019-10-26 20:53"`.

1. Quelles sont les valeurs de variables python x, y et z calculées par les instructions

```
[x,yz]=t.split("-")
[y,z]=yz.split(" ")
```

2. Expliquer la fonction suivante :

```
def parseDate(d):
    mois = {
        'sept.': '09',
        'oct.': '10',
        'nov.': '11',
        'déc.': '12',
    }
    [j,mh]=d.split("-")
    [m,h]=mh.split(" ")
    return  '2019-'+mois[m]+'-'+j+' '+h
```

2 DataFrame

Une DataFrame est une structure de données contenant une matrice de valeurs de types homogène par colonnes, les colonnes sont nommées par une liste `columns`, et les lignes sont indexées à l'aide d'une liste `index` sinon par défaut par la suite naturelle 0,1,2,... Soient la matrice A et la liste L :

```
A=[['a',2,3,4],['a',3,5,2],['b',1,2,3],['c',3,2,1],['b',0,1,2]]
L=['A','B','C','D']
```

La DataFrame est alors définie par :

```
df=pandas.DataFrame(A,columns=L)
```

1. Afficher la dataframe `df` et calculer sa transposée `df.T`.
2. Qu'affichent les commandes suivantes :

```
df['A']
df.loc[0]
df.loc[[1, 2], ['A', 'C']]
df.loc[:,['A', 'C']]
df.loc[1, 'C']
df.at[1, 'C']
df.iloc[1]
df.iloc[1:3,[0, 2]]
df.iloc[:,2:4]
df.iloc[1,2]
df.iat[1,2]
```

Quelle est la différence entre les propriétés `loc , at` et `iloc, iat`.

3. Quelle est la différence entre les commandes :

```
df.loc[:,['A']]
df.loc[:,'A']
```

Déterminer le type de chaque résultat.

4. Afficher à l'aide de `df.info()` : les infos sur le dataframe `df` (y compris place occupée).
5. Lister les 2 premières lignes et les deux dernières grâce à `df.head(2)` : renvoie un dataframe avec. Idem avec `df.tail(2)` . Peut-on appliquer les fonctions `head()` et `tail()` sans arguments ?
6. Donner les statistiques sur les valeurs (nombres de valeurs, moyenne, écart-type, ...). Que remarque-t-on ?
7. Qu'affichent les deux boucles suivantes

```
for x in df:
    print(x)
```

et

```
for x in df.itertuples():
    print(x.A)
```

8. Étudier les filtres suivants :

```
df[df['B'] > 2]
df2 = df[(df['C'] > 2) & - (df['B'] < 3)]
df[df['A'].isin(['a', 'c','m'])]
df.query('B > 2 and C < 10')
np.where(df['B'] < 3)[0]
```

9. Appliquer sur le dataframe *df* les fonctions suivantes :
   — `min(), max()`
   — `mean()` : la moyenne. (`df / df.sum()`)
   — `median()` : la médiane.
   — `std()` : la déviation standard (écart-type) qui par défaut est normalisée avec $N-1$.
   — `var()` : la variance normalisée avec $N-1$.
   — `sum(), prod()` : la somme, le produit.
10. Grouper les lignes de *df* selon une colonne en appliquant une fonction d'agrégation de la question précédente. exemple

```
>>> df.groupby('A').sum()

   B  C  D
A
a  5  8  6
b  1  3  5
c  3  2  1
```

3 Série statistique

On rappelle le programme python du dernier TP permettant de déterminer les ventes de cartes pokemon sur internet :

```
import sys
from bs4 import BeautifulSoup
import openpyxl
import urllib

def parseDate(d):
    mois = {
        'sept.': '09',
        'oct.': '10',
        'nov.': '11',
        'déc.': '12',
        'dÃ©c.': '12'
```

```
    }
    [j,mh]=d.split("-")
    [m,h]=mh.split(" ")
    return  '2019-'+mois[m]+'-'+j+' '+h

# chargement du classeur excel: workbook
wb = openpyxl.load_workbook('cartes.xlsx')

# Noms des feuilles de calculs
sheets = wb.sheetnames
# Première feuille de calcul
ws = wb[sheets[0]]

# Le classeur excel de sortie qui contiendera les données
workbook = openpyxl.Workbook()
sheet = workbook.active
# Les entêtes de la feuille excel de sortie
sheet.append(["Produit","Titre","Prix","Date","Pays"])
for row in ws.iter_rows(min_row=3,max_row=50, values_only=True):
    print(row[1]+" "+row[2])
    carte=row[1]+" "+row[2]
    #carte="Spectrum 29/102"
    carteCodee = urllib.parse.quote_plus(carte)
    url="https://www.ebay.fr/sch/i.html?_from=R40&_sacat=0&_nkw="+\
        carteCodee+"&LH_Complete=1&LH_Sold=1&rt=nc"
    f = urllib.request.urlopen(url)
    html_doc = f.read()

    soup = BeautifulSoup(html_doc, "html.parser")
    N=soup.select("span.rcnt")[0].get_text().strip();
    N=int(N)
    for k in soup.find_all("li",_sp="p2045573.m1686.l0",attrs={"class": "sresult"}):
        p = k.select("h3.lvtitle")[0]
        titre = p.select_one("a").get_text().strip()
        p = k.select("li.lvprice")[0]
        prix=p.select_one("span").get_text().strip()[:-4]
        i = prix.find("EUR")
        if i>0:
            prix=prix[0:i-1]
        p = k.select("ul.lvdetails")[0]
        lis=p.select("li")
        date=str(lis[0].get_text().strip())
        date = parseDate(date)
        pays = "France"
        if len(lis)>1:
            result = lis[1].get_text().find('Provenance')
            if result>0:
                pays = lis[1].get_text().strip()[13:]

        print(prix)
        prix= float(prix.replace(",","."))
        print(titre,prix,date,pays)
        sheet.append([carte,titre,prix,date,pays])
        print("--------------------------------------")
        N = N - 1
        if N==0:
            break
workbook.save(filename="ventes.xlsx")
```

1. Commenter rapidement les lignes de ce programme.
2. Que contient le ficher `ventes.xlsx` après l'exécution de ce programme ?
3. Charger ce fichier dans un `pandas.DataFrame` de la façons suivante en précisant le type réel de Prix :

   ```
   import pandas as pd
   pokemon = pd.read_excel("ventes.xlsx",dtype ={'Prix':np.float64} )
   ```

4. Grouper la DataFrame `pokemon` selon la colonne `Produit` à l'aide de la fonction `groupby` tout en calculant les moyennes des articles groupés. Soit X la série de prix moyen par produit.

   ```
   X=pokemon.groupby('Produit').mean()['Prix']
   ```

5. Tracer la courbe et l'histogramme de la série X.
6. Déterminer ses caractéristiques statistiques : moyenne `mean()` et écart-type `std()`.
7. Quelles sont les particularités des produits affichés par les deux commandes suivantes :

   ```
   X.argmax()
   X.argmin()
   ```

4 Série temporelle

Dans cette partie nous allons étudier la série des prix en fonction du temps. On commence par charger la feuille excel `ventes.xlsx` dans une *DataFrame* python, en précisant le type `float` de la colonne `Prix` et en considérant la colonne Date comme index de la *DataFrame* :

```
import pandas as pd
df=pd.read_excel("ventes.xlsx",dtype ={'Prix':float},index_col='Date',parse_dates=True )
```

1. On désigne par X la série temporelle des prix $X = df$['Prix']. Tracer deux représentations graphiques de X : une courbe $X = X(t)$ et un histogramme.
   Pour cela, on importe le package matplotlib :

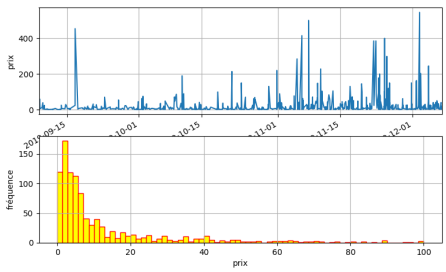
   ```
   import matplotlib.pyplot as plt
   ```

   Pour regrouper les deux graphes verticalement on subdivise la zone graphique en deux sur deux lignes et une colonne. Le motif 21 indique 2 ligne et 1 colonne. le premier graphe est désigné par 211 en ajoutant 1 au motif de découpage et le deuxième graphe est indiqué par 212, en ajoutant 2. D'où le code :

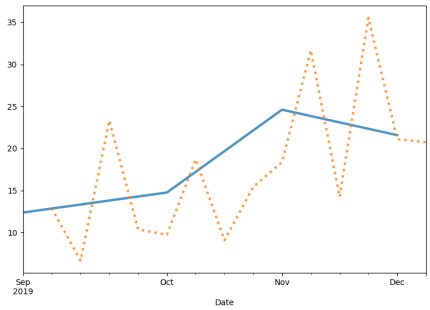
   ```
   X=df['Prix']
   plt.figure(figsize=(9,6))
   plt.subplot(211)
   X.plot()

   plt.xlabel('date')
   plt.ylabel('prix')
   plt.grid(True)

   plt.subplot(212)
   X.hist(range = (0, 100), bins = 70, color = 'yellow',edgecolor = 'red')
   plt.xlabel('prix')
   plt.ylabel('fréquence')
   plt.show()
   ```



2. Sachant que la DataFrame permet de sélectionner une période dans le temps en utilisant les crochets, par exemple $Y = df$['2019-09']['Prix'] est la série des prix de vente en septembre 2019. et $Z = df$['2019-9' :'2019-11']['Prix'] est la série des prix de vente entre septembre et novembre 2019. Tracer Y et Z. et déterminer le produit le mieux vendu dans chaque période.
3. Soit $V = df$['2019']['Prix'] la série des prix de vente en 2019. La fonction `resample()` permet de calculer à partir d'une série temporelle X la série des moyennes par mois `resample('M')` ou par semaine `resample('W')` ou par quinzaine `resample('2W')` par exemple `X.resample('M').mean()`. Tracer sur le même graphique l'évolution des ventes par mois et par semaine. Le graphique doit ressembler à ceci :



4. Ecrire un programme python permettant de calculer pour chaque semaine, la moyenne μ, l'écart type σ, le minimum m et le maximum M des ventes en 2019. Le resultat doit ressambler à quelque chose comme ceci :

Date	mean	std	min	max
2019-09-08	12.888889	17.431612	5.00	59.00
2019-09-15	6.659048	8.675557	0.45	38.80
2019-09-22	23.322581	80.401451	1.17	454.34
2019-09-29	10.374375	14.903380	1.00	69.99
2019-10-06	9.742769	15.266585	0.30	75.00
2019-10-13	18.702059	31.627977	1.00	189.79
2019-10-20	9.089718	15.551405	1.00	99.00
2019-10-27	15.433649	32.908969	0.20	214.11
2019-11-03	18.397917	41.006100	0.49	220.00
2019-11-10	31.670367	71.490914	0.45	500.58
2019-11-17	14.263729	20.765722	1.00	130.47
2019-11-24	35.538211	68.058012	0.59	400.00
2019-12-01	21.108235	41.459441	1.00	299.00
2019-12-08	20.736455	59.875106	1.00	545.67