Formation pour l'Actuaire Data Scientist - Module d'introduction à Python

Examen à rendre au plus tard le 27 mars à minuit sous forme de pièce jointe (fichier texte, notebook) par mail à *xavier.dupre@gmail.com*.

Les réponses aux questions de cours devront être justifiées. Il y a 10 questions, chacun d'elles vaut 2 points.

1

1) Que fait le programme suivant ?

1. Il trie.
2. Il vérifie qu'un tableau est trié.
3. Rien car la boucle ne commence pas à 0.

```
l = [0,1,2,3,4,6,5,8,9,10]
res = True
for i in range(1,len (l)) :
    if l[i-1] > l[i]:
                res = False
```

2) La fonction suivante ne fonctionne pas sur

1. Le nombre 0.
2. La constante "123".
3. Les nombres strictement négatifs

```
def somme(n):
    return sum ( [ int(c) for c in str(n) ] )
```

3) Le programme suivant provoque une erreur. Quelle est l'exception qu'il va produire ?

1. `SyntaxError`
2. `TypeError`
3. `IndexError`

```
li = list(range(0,10))
sup = [0,9]
for i in sup :
    del li [i]
```

4) Entourer ce que est vrai à propos de la fonction suivante.

1. Elle est récursive.
2. Il manque une condition d'arrêt.

3. fibo(4) appelle récursivement 8 fois fibo : une fois fibo(3), deux fois fibo(2), trois fois fibo(1) et deux fois fibo(0)

```
def fibo (n) :
    if   n < 1  : return 0
    elif n == 1 : return 1
    else : return fibo (n-1) + fibo (n-2)
```

5) Combien de lignes comporte le dataframe `df2` ?

— 3
— 4
— 5
— 6
— 7
— 8
— 9
— Aucun, le code provoque une erreur.

```
import pandas
df = pandas.DataFrame([dict(x=1, t="e"), dict(x=3, t="f"), dict(x=4, t="e")])
df2 = df.merge(df, left_on="x", right_on="x")
```

6) Combien de lignes comporte le dataframe `df3` ?

— 3
— 4
— 5
— 6
— 7
— 8
— 9
— Aucun, le code provoque une erreur.

```
import pandas
df = pandas.DataFrame([dict(x=1, t="e"), dict(x=3, t="f"), dict(x=4, t="e")])
df3 = df.merge(df, left_on="t", right_on="t")
```

2

On suppose qu'on a un fichier de données trop gros pour être chargé en mémoire. On veut produire des statistiques simples. Pour tester votre code, vous pourrez utiliser le fichier *data.txt* construit comme suit :

```
import pandas
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00350/"
url += "default%20of%20credit%20card%20clients.xls"
df = pandas.read_excel(url, skiprows=1)
df.to_csv("data.txt", encoding="utf-8", sep="\t", index=False)
```

Les questions suivantes utiliseront ces données.

1) Ecrire une fonction qui agrège un dataset par **AGE** et calcule le mininum, maximum et la moyenne en une seule fois pour les variables **LIMIT_BAL**, **default payment next month** et qui calcule le nombre d'observations partageant le même **AGE**.

```
agg = df.groupby("age").agg ( ... )
```

La figure 1 donne un exemple ce qu'il faut obtenir. Les chiffres présents dans cette table ne sont qu'indicatifs.

Out[45]:

	LIMIT_BAL			default payment next month			ID
	min	max	mean	min	max	mean	len
AGE							
21	10000	60000	23846.153846	0	1	0.192308	26
22	10000	150000	34720.812183	0	1	0.279188	197
23	10000	500000	63718.750000	0	1	0.268750	320
24	10000	400000	71879.518072	0	1	0.306024	415
25	10000	440000	100143.540670	0	1	0.277512	418

Figure 1 : *Type de résultats qu'il faut obtenir pour la question 1.*

2) Lire la documentation de `read_csv`. On veut charger un fichier en plusieurs morceaux et pour chaque morceau, calculer l'agrégation ci-dessus. Le nom des colonnes n'est présent qu'à la première ligne du programme. Il suffit de compléter le programme à chaque fois qu'il contient **...**.

```
aggs = []
step = 10000
columns = None
for i in range(0, df.shape[0], step):
    part = pandas.read_csv("data.txt", encoding="utf-8", sep="\t",
                           skiprows= ...,
```

```
                        nrows= ...,
                        header= ...,
                        names= ...)

    agg = .... # voir première question

    aggs.append(agg)
    if columns is None:
        columns = ...

tout = pandas.concat( ... )
```

3) Le dataframe `tout` est la concaténation de deux dataframes contenant des informations aggrégées pour chaque morceau. On veut maintenant obtenir les mêmes informations agrégrées pour l'ensemble des données uniquement à partir du dataframe `tout`. Ecrire le code qui fait cette agrégation.

4) Tracer un histogramme avec la valeur moyenne de la variable `LIMIT_BAL`, on ajoutera deux lignes pour les valeurs *min* et *max*. Il devrait ressembler à la figure 2. Cette page pourrait vous donner quelques indications : http://stackoverflow.com/questions/19952290/how-to-align-the-bar-and-line-in-matplotlib-two-y-axes-chart.

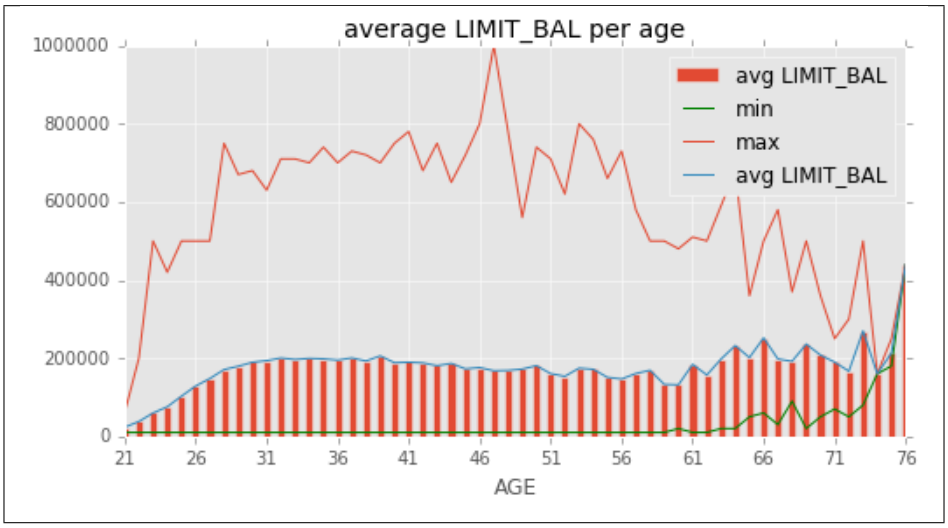

Figure 2 : *Type de graphe qu'il faut obtenir pour la question 4.*