TP 20 pandas et élections russes

L'objectif du TP est d'analyser les statistiques des élections russes afin de démontrer que des anomalies statistiques sont présentes dans les résultats officiels. Ce TP reprend l'analyse de Dimitri Kobak, Sergey Shpilkin et Maxim Pshenichnikov dans une séries d'articles publiés entre 2016 et 2020 dans les revues Annals of Applied Statistics et Significance (vous pouvez trouver le premier article sur cahier de prépa). Les données ainsi que les modèles Python utilisés ici sont publiés sur Github à l'adresse `https://github.com/dkobak/elections`.

La librairie `pandas`

Pour manipuler des fichiers de type base de données, tableur, ou tout fichier structuré ayant des données réparties en colonnes, on peut utiliser la librairie `pandas`, que l'on importe à l'aide de la commande

```
import pandas as pd
```

. Sauvegardez votre script sous le nom `TP20.py` dans un dossier personnel sur votre machine (n'importe où dans « Documents » par exemple), puis téléchargez sur cahier de prépa le fichier `2021.csv` dans le même dossier. Chargez ce fichier dans une variable Python nommée `table` à l'aide de la commande

```
import pandas as pd

table = pd.read_csv('2021.csv')
```

Le nom du fichier à charger doit être une chaîne de caractères, correspondant exactement au nom du fichier tel qu'enregistré sur le disque. Le fichier doit impérativement être dans le même dossier que le script Python courant. Le nom de variable choisi n'importe pas. Nous avons choisi `table`, mais libre à vous d'en changer. La variable `table` est une instance d'une classe spécifique à `pandas` que nous ne détaillerons pas. Voici quelques méthodes utiles pour ces tables de données :

```
>>> type(table)
>>> table.shape
>>> table.columns
>>> type(table.columns)
>>> len(table.columns)
>>> [p, n] = table.shape
>>> n == len(table.columns)
>>> nom = table.columns[15]
>>> type(table[nom])
>>> table[nom].values
>>> table[nom].name
>>> type(table[nom].values)
>>> p == len(table[nom].values)
>>> table[nom].name == table.columns[15]
```

Vous l'aurez compris, un tableau de données `pandas` est une sorte de dictionnaire : chaque colonne dans le fichier de données a un nom (souvent une chaîne de caractères), nom auquel on accède grâce à la liste de ces noms `table.columns`. La colonne au nom `nom` est obtenue avec `table[nom]`. L'objet obtenu est encore une instance d'une classe `pandas`. Elle contient des valeurs, obtenues avec `table[nom].values` : c'est un vecteur `numpy` tout ce qu'il y a de plus classique cette fois. On retrouve le nom de la colonne en faisant `table[nom].name`.

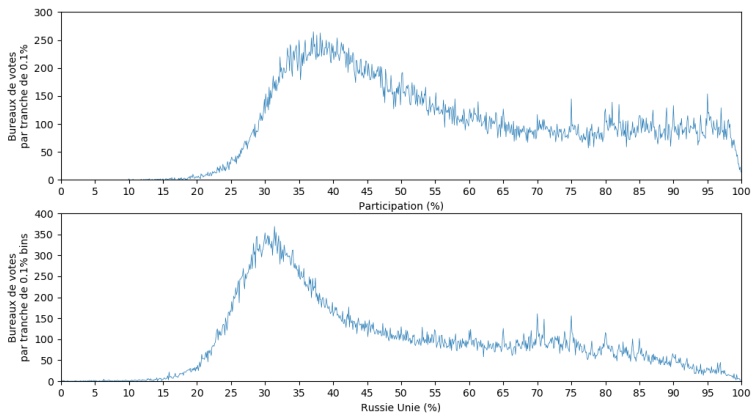
Observation du fichier des élections législatives Russes de 2021

1. Observez les différentes colonnes du fichier.
2. La commande `pd.describe` permet d'avoir un coup d'oeil sur les statistiques de base d'un objet généré par `pandas` : moyenne, écart-type, médiane, quartiles. En tapant `table.describe()`, Python fait la moyenne/médiane/etc. sur les lignes, et ce pour chaque colonne de `table`. On peut aussi faire ces statistiques sur une seule colonne, par exemple `table[table.columns[3]].describe()`. Calculez ainsi le premier quartile du nombre de bulletins de votes attribués au parti « Plateforme civique » dans les différents bureaux de votes.
3. Stockez dans un vecteur numpy `votants` le nombre d'électeurs inscrits dans chaque bureau de vote. En déduire le nombre total d'électeurs inscrits à ce scrutin. Vous pourrez utiliser `np.sum` pour faire la somme des éléments d'un vecteur `numpy`. Vérifiez que vous obtenez 109204662 électeurs.
4. Calculez le nombre de bulletins de votes émis : d'abord dans chaque bureau de vote (stockez le résultat dans un vecteur numpy `exprimés`), puis au total.
   Pour sélectionner plusieurs colonnes de `table`, on peut faire `table[liste_colonnes]`.
5. En déduire la participation au scrutin. Vérifiez que vous obtenez environ 51.72%.
6. Calculez le pourcentage de vote pour le parti « Russie Unie » dans chaque bureau de vote. Vous stockerez le résultat dans un vecteur numpy `leader`.
7. Calculez le nombre de bulletins pris en compte (on prend en compte les bulletins valides ainsi que les bulletins nuls) : dans chaque bureau de vote (à stocker dans un vecteur `recus`), puis au total.
8. Calculer le score national du parti « Russie Unie » . Vérifiez que vous obtenez environ 48.82%.

Anomalies statistiques dans les résultats des élections

Certains bureaux de vote n'ont reçu aucun bulletin valide. Dans ces bureaux, impossible de calculer le score d'un parti : on diviserait par 0. On a de même certains bureaux qui n'ont aucun inscrits sur les listes électorales : on peut les trouver en tapant `[k for k in range(len(votants)) if votants[k] == 0]`. Pour régler le problème, on va utiliser une méthode très classique de nettoyage d'un vecteur.

9. Créez un vecteur `ind` qui contient un booléen pour chaque bureau de vote : `True` si le nombre de bulletins reçus est non nul, `False` sinon.
10. À l'aide de `100*leader[ind]/recus[ind]`, on obtient un vecteur donnant le pourcentage de vote pour « Russie Unie » dans les différents bureaux de vote. On stockera ce vecteur dans une variable `score`.
11. On va créer un histogramme de ces résultats. Importer le module `matplotlib.pyplot`. Créez un histogramme des valeurs de score à l'aide de la commande `plt.hist(score, 100, density = True)`. On a précisé qu'on découpe l'ensemble des valeurs prises par `score` en 100 cases de même taille. Les scores allant de 0 à 100%, chaque case a pour longueur $0,1$.
12. Faire de même pour le taux de participation dans chaque bureau de vote (en éliminant les bureaux dans lesquels le nombre de votants était nul).



Formulaire :

Vous aurez besoin des commandes suivantes (dans le désordre)

```
### LES MODULES USUELS
from math import *
import numpy as np
import matplotlib.pyplot as plt
import numpy.random as rd

### LES VARIABLES ALEATOIRES
rd.randint(a,b,N)
rd.random(N)
rd.binomial(n,p,N)
rd.geometric(p,N)
rd.poisson(lambda,N)

## LES LISTES
np.linspace(a,b,n)
for compteur in liste :
for compteur in range(a,b):
[ f(compteur) for compteur in liste]
[ f(compteur) for compteur in range(a,b)]

### COMMANDES CLASSIQUES
while condition:
def nom_fonction( variables ) :
return resultat
print( 'message', variable )
if condition :
else :
log(x)
plt.plot( abscisses , ordonnees )

### CLASSES EN PYTHON
class nom_de_classe :
class nom_de_classe(classe_d_heritage):
def __init__(self, autres_parametres):
def __str__(self):

### MATRICES EN PYTHON
np.array([[...],...,[...]])
A.shape
np.zeros([p,n])
np.eye(n)
np.ones([p,n])
A.dot(B)
np.dot(A,B)
A.transpose()
A.all()
```