Carmilla

Projeto que gera números da Lotofacil da Caixa Econômica Federal

O projeto é para fins de estudos.

- História:
  - Porque não criar uma plataforma de *geração de números livres* para *possíveis jogos* e usar as *estatística para direcionar os casos*.
- Pré-requisitos:
  - Anaconda;
  - Jupyter;
  - Python 3.6 ou superior;
  - Bibliotecas iteration_utilities e pandas;
- Sistemas implementados:
  - Contador de ciclos
  - Verificação de pares e ímpares
  - Otimização de geração de números

Bibliotecas do Projeto

In [1]:
```
#!pip install iteration_utilities pandas selenium
from pandas import pandas as pd                    #-> Biblioteca para manipular o *.xlsx.
import random                                      #-> Biblioteca para gerar sequências de números.
from iteration_utilities import duplicates         #-> Biblioteca para remover as duplicatas.
from selenium import webdriver as wb               #-> Biblioteca para baixar a planilha.
from selenium.webdriver.common.by import By        #-> Biblioteca para passar como chave de procura o XPATH.
```

In [2]:
```
navegador = wb.Chrome() #Abre o navegador
navegador.get("https://asloterias.com.br/download-todos-resultados-lotofacil") #Acessa o link
navegador.find_element(By.XPATH, '/html/body/main/div[2]/div/div/div[1]/div[5]/a').click() # Localizador do botão para
#baixar a planilha loto_facil_asloterias_ate_concurso_2741_sorteio.xlsx -> loto_facil_asloterias_ate_concurso_*_sorteio.xlsx
```

In [3]:
```
planilha = 'loto_facil_asloterias_ate_concurso_2743_sorteio.xlsx' #Ultima planilha atualizada.

df = pd.read_excel(planilha, header=6)#Saber em qual concurso está para atualizar o número no nome da planilha.

def transformacao(df: pd)  -> list[list]:
    aux = ["bola 1", "bola 2","bola 3","bola 4","bola 5","bola 6","bola 7","bola 8","bola 9","bola 10","bola 11","bola 12","bola 13","bola 14","bola 15"]

    lista, sorteio = [], []

    for j in range(0,len(df[0:])):
        for i in aux: lista.append(df[i][j])

        lista = sorted(lista)
        sorteio.append(lista)
        lista = []
    return sorteio

 # valores
#[[1, 2, 3, 4, 8, 9, 10, 11, 13, 15, 19, 20, 23, 24, 25], [2, 3, 4, 7, 8, 10, 11, 12, 13, 19, 20, 21, 22, 23, 24], [3, 6, 7, 8, 9, 11, 12, 13, 15, 16, 18, 19,


def equi_par_impar(lista:list) -> bool:
    p, i = 0, 0
    for j in lista:
        if(j % 2 == 0): p += 1
        else: i += 1
    if((p == 8 and i == 7) or (p == 6 and i == 9) or (p == 7 and i == 8)): return True
    else: return False

#quantidade de impares e pares
    #40% tiveram 8 pares e 7 impares.
    #40% tiveram 6 pares e 9 impares.
    #20% tiveram 7 pares e 8 impares.


def sort_num_apost(valores: list, transf: list[bool,list]) -> list:
    #inicio das variaveis
    response = []
    num_sort = range(1,26)

    while(True):
        #tecnica de otimizacao, se eu sei quais numeros falta, vou inserilos. isso se a lista não for vazia
        if(not(transf[0])):
            response = random.sample(num_sort,15)
            return(response)

        numDeNumerosQueVaiSerAdicionado = random.randint(1, len(transf[1]))
        response = random.sample(transf[1], numDeNumerosQueVaiSerAdicionado)# min 1, max len(transf[1])(lista que falta para sortear)

        #adicione o restante dos numeros para formar 15 dezenas
        for i in random.sample(num_sort,(15 - len(response))):
            response.append(i)

        #precisamos verificar se não veio repetido
        aux = len(list(duplicates(response)))
        while(aux != 0):
            response = list(duplicates(response))

            for i in random.sample(num_sort, aux):#a nova quantidade a ser inserida é o resultado do aux("auxiliar")
                response.append(i)
            aux = len(list(duplicates(response)))

        if(not(response in valores)):# se os números gerados já foi sortiado as chances de ser sorteado novamente são muito baixas então descartamos.
            if(equi_par_impar(response)):# se os números gerados não bater com a estatística é melhor descartar.
                break
    response = sorted(response)
    return(response)
#valores = transformacao(df)
#teste = sort_num_apost(valores)
#teste
#[22, 1, 16, 5, 15, 19, 24, 13, 6, 2, 8, 10, 21, 4, 9]

def transformacaoComConcursos(df: pd) -> list[dict]:
    aux = ["Concurso", "bola 1", "bola 2","bola 3","bola 4","bola 5","bola 6","bola 7","bola 8","bola 9","bola 10","bola 11","bola 12","bola 13","bola 14","bol
    lista, sorteio = [], []
    dicio = {}

    for j in range(0,len(df[0:])):
        for i in aux:
            lista.append(df[i][j])
        dicio = {"Concurso" : lista[0], "sorteio" : sorted(lista[1:])}
        sorteio.append(dicio)
        lista = []
        dicio = {}

    return sorteio
#[{'Concurso': 2703,
#  'sorteio': [1, 2, 5, 7, 9, 10, 12, 13, 14, 17, 19, 20, 21, 23, 24]},
# {'Concurso': 2702,
#  'sorteio': [1, 2, 4, 6, 7, 10, 14, 15, 16, 17, 18, 19, 21, 23, 25]},
# {'Concurso': 2701,
#  'sorteio': [1, 3, 4, 5, 7, 9, 12, 14, 16, 18, 19, 21, 22, 23, 25]},...]

def calculoDeCiclo(df: pd) -> list[bool,list]:
    trans = transformacaoComConcursos(df)
    ciclo = [i for i in range(1, 26)]
    control = False # imagine que o ciclo seja vazio([]), significa que não há elementos que podem ser os "favoritos" para os próximos concursos, ou seja o cic

    for i in trans[::-1]:
        if (ciclo == []):
            ciclo = [i for i in range(1, 26)] # se o ciclo for vazio é porque precisa ser sorteado tudo novamente
            control = False
        else: control = True

        for j in i["sorteio"]:
            if(j in ciclo):
                ciclo.remove(j)

    return [control, ciclo]

#[3, 8, 11, 22]
#True
# A ideia é que se for verdade, os jogos gerados tem que conter esses números

def main():
    valores = transformacao(df)
    transf = calculoDeCiclo(df)
    a = 0
    aux = []
    while(True):
        a = int(input("Quantas apostas quer fazer?\n"))
        print("\n")
        for i in range(a):
            aux = sort_num_apost(valores, transf)

            print(sorted(aux))
            print("\n")
        break
```

In [4]:
```
if __name__ == "__main__":
    main()
```

```
Quantas apostas quer fazer?
10


[2, 3, 4, 5, 6, 8, 9, 13, 14, 16, 17, 19, 20, 21, 22]


[2, 3, 5, 6, 8, 9, 11, 12, 13, 15, 17, 19, 20, 21, 22]


[2, 3, 4, 6, 7, 9, 10, 11, 16, 17, 19, 21, 23, 24, 25]


[1, 3, 6, 8, 9, 11, 14, 15, 16, 19, 20, 21, 22, 24, 25]


[3, 4, 5, 8, 9, 11, 12, 14, 15, 17, 19, 20, 21, 22, 24]


[4, 7, 9, 11, 12, 14, 16, 17, 18, 19, 20, 21, 23, 24, 25]


[1, 2, 4, 9, 10, 11, 12, 13, 15, 16, 20, 21, 22, 23, 24]


[3, 4, 6, 8, 9, 10, 11, 13, 14, 16, 17, 18, 21, 23, 25]


[3, 4, 8, 9, 10, 11, 13, 14, 15, 17, 18, 21, 22, 24, 25]


[2, 3, 4, 6, 7, 8, 9, 11, 13, 16, 17, 18, 21, 23, 25]
```