Untitled

July 16, 2019

In [3]:
```
!pip install openpyxl
```

```
Collecting openpyxl
  Downloading https://files.pythonhosted.org/packages/ba/06/b899c8867518df19e242d8cbc82d4ba210
    100% || 174kB 307kB/s
Collecting jdcal (from openpyxl)
  Downloading https://files.pythonhosted.org/packages/f0/da/572cbc0bc582390480bbd7c4e93d14dc46
Collecting et_xmlfile (from openpyxl)
  Downloading https://files.pythonhosted.org/packages/22/28/a99c42aea746e18382ad9fb36f64c1c1f0
Building wheels for collected packages: openpyxl, et-xmlfile
  Building wheel for openpyxl (setup.py) ... done
  Stored in directory: /home/jovyan/.cache/pip/wheels/82/cd/05/2e1db4b561fda743444f9573f2c5090
  Building wheel for et-xmlfile (setup.py) ... done
  Stored in directory: /home/jovyan/.cache/pip/wheels/2a/77/35/0da0965a057698121fc7d8c5a7a9955
Successfully built openpyxl et-xmlfile
Installing collected packages: jdcal, et-xmlfile, openpyxl
Successfully installed et-xmlfile-1.0.1 jdcal-1.4.1 openpyxl-2.6.2
```

In [43]:
```
!pip install python-Levenshtein==0.12.0
```

```
Collecting python-Levenshtein==0.12.0
  Downloading https://files.pythonhosted.org/packages/42/a9/d1785c85ebf9b7dfacd08938dd028209c3
    100% || 51kB 126kB/s
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-packages (from pyth
Building wheels for collected packages: python-Levenshtein
  Building wheel for python-Levenshtein (setup.py) ... done
  Stored in directory: /home/jovyan/.cache/pip/wheels/de/c2/93/660fd5f7559049268ad2dc6d81c4e39
Successfully built python-Levenshtein
Installing collected packages: python-Levenshtein
Successfully installed python-Levenshtein-0.12.0
```

In [196]:
```
# Importando bibliotecas
import pandas as pd
import Levenshtein
df1 = pd.read_excel("planilha_questionarios.xlsx", index_col = 0)
df2 = pd.read_excel("avDireta.xls", index_col = 0)
```

In [197]:
```
str1 = "ANTONIO CLODOALDO CAMPOS ALCANTARA"
str2 = "ANTONIO CLODOALDO CAMPOS"

d = Levenshtein.distance(str1,str2)

print(d/len(str1))
```

```
0.29411764705882354
```

In [198]:
```
df1.head()
```

Out[198]:

Carimbo de data/hora	Nome da mãe, pai ou responsável:	Sexo:	Nome da criança:	Endereço:	Escola:	Série - Infantil	Turma (Ex.: A, B...)	Turno:	Telefone fixo:	Celular:	...
2019-06-03 17:50:50.341	FRANCISCA ALCENIR COSTA	F	NICOLLAS COSTA GOMES	PEDRINHAS RAFAEL ARRUDA	VICENTE ANTENOR	V (cinco)		NaN	NaN	8.899219e+10	...
2019-06-03 17:55:48.977	DIANA ARAUJO DE SOUZA OLIVEIRA	F	CARLOS DE SOUSA OLIVEIRA ARAUJO FILHO	RUA IPIRANGA	ANTENOR NASPOLINI	IV (quatro)		NaN	NaN	9.932827e+08	...
2019-06-03 17:57:32.779	NaN	NaN	RODRIGO LUCAS CAMPOS SANTOS	PEDRINHAS RAFAEL ARRUDA SOBRAL	VICENTE ANTENOR	IV (quatro)		NaN			
2019-06-03 18:12:10.086	ANTONIA LOPES NASCIMENTO	F	ANA VITORIA NASCIMENTO PAULA	RUA JACINTO ANTUNES	ANTENOR NASPOLINI	V (cinco)		NaN			
2019-06-03 18:16:39.701	ANTONIA LOPES NASCIMENTO	F	ANA VITORIA NASCIMENTO PAULA	RUA JANCINTO ANTUNES	ANTENOR NASPOLINI	V (cinco)		NaN			

```
2019-06-03 17:57:32.779            NaN  8.899358e+10  ...
2019-06-03 18:12:10.086            NaN  8.899383e+10  ...
2019-06-03 18:16:39.701            NaN  8.899383e+10  ...

                      A seguir apresentamos algumas perguntas sobre o que sua cria
Carimbo de data/hora
2019-06-03 17:50:50.341                                    Com frequência
2019-06-03 17:55:48.977                                    Com frequência
2019-06-03 17:57:32.779                                    Com frequência
2019-06-03 18:12:10.086                                      Sem resposta
2019-06-03 18:16:39.701                                      Sem resposta

                     A seguir apresentamos algumas perguntas sobre o que sua crian
Carimbo de data/hora
2019-06-03 17:50:50.341                                          Às vezes
2019-06-03 17:55:48.977                                           Não sei
2019-06-03 17:57:32.779                                          Às vezes
2019-06-03 18:12:10.086                                          Às vezes
2019-06-03 18:16:39.701                                          Às vezes

                      A seguir apresentamos algumas perguntas sobre o que sua cria
Carimbo de data/hora
2019-06-03 17:50:50.341                                    Com frequência
2019-06-03 17:55:48.977                                           Não sei
2019-06-03 17:57:32.779                                    Com frequência
2019-06-03 18:12:10.086                                      Sem resposta
2019-06-03 18:16:39.701                                      Sem resposta

                     A seguir apresentamos algumas perguntas sobre o que sua crian
Carimbo de data/hora
2019-06-03 17:50:50.341                                          Às vezes
2019-06-03 17:55:48.977                                           Não sei
2019-06-03 17:57:32.779                                          Às vezes
2019-06-03 18:12:10.086                                          Às vezes
2019-06-03 18:16:39.701                                          Às vezes

                     A seguir apresentamos algumas perguntas sobre o que sua crian
Carimbo de data/hora
2019-06-03 17:50:50.341                                    Com frequência
2019-06-03 17:55:48.977                                    Com frequência
2019-06-03 17:57:32.779                                    Com frequência
2019-06-03 18:12:10.086                                      Sem resposta
2019-06-03 18:16:39.701                                      Sem resposta

                     A seguir apresentamos algumas perguntas sobre o que sua crian
Carimbo de data/hora
2019-06-03 17:50:50.341                                    Com frequência
2019-06-03 17:55:48.977                                           Não sei
```

```
2019-06-03 17:57:32.779                                          Às vezes
2019-06-03 18:12:10.086                                          Às vezes
2019-06-03 18:16:39.701                                          Às vezes

                        A seguir apresentamos algumas perguntas sobre o que sua crian
Carimbo de data/hora
2019-06-03 17:50:50.341                                          Às vezes
2019-06-03 17:55:48.977                                   Com frequência
2019-06-03 17:57:32.779                                          Às vezes
2019-06-03 18:12:10.086                                    Sem resposta
2019-06-03 18:16:39.701                                    Sem resposta

                        Tabulação Questionário A criança é sorteada?
Carimbo de data/hora
2019-06-03 17:50:50.341       NaN          NaN                    NaN
2019-06-03 17:55:48.977       NaN          NaN                    NaN
2019-06-03 17:57:32.779       NaN          NaN                    NaN
2019-06-03 18:12:10.086       NaN          NaN                    NaN
2019-06-03 18:16:39.701       NaN          NaN                    NaN

[5 rows x 60 columns]
```

In [199]:
```
df2.head()
```

Out[199]:

nome_apl	id_apl	codesc_tablet	codesc	nome_turma	turma	\
Juliana	7	15	23264976.0	INFANTIL IV C	C	
Josiany	2	21	23229276.0	INFANTIL V B	B	
ISRAELA	5	4	NaN	INFANTIL IV U	U	
Juliana	7	35	23025300.0	INFANTIL V A	A	
DIEGO FERREIRA	3	33	23026014.0	INFANTIL V B	B	

nome_apl	serie	turno	pre_1_2	data_apl	dia_apl	...	dia_nasc	\
Juliana	INFANTIL IV	Tarde	1	29/04/19	29	...	25.0	
Josiany	INFANTIL V	Tarde	2	03/04/19	3	...	13.0	
ISRAELA	INFANTIL IV	Manhã	1	23/04/19	23	...	21.0	
Juliana	INFANTIL V	Tarde	2	24/04/19	24	...	2.0	
DIEGO FERREIRA	INFANTIL V	Manhã	2	29/04/19	29	...	16.0	

nome_apl	mes_nasc3	mes_nasc	ano_nasc3	ano_nasc	sabe_nome_mae	\
Juliana	NaN	2.0	NaN	2015.0	NaN	
Josiany	NaN	9.0	NaN	2013.0	NaN	
ISRAELA	NaN	9.0	NaN	2014.0	NaN	
Juliana	NaN	7.0	NaN	2013.0	NaN	
DIEGO FERREIRA	NaN	12.0	NaN	2013.0	NaN	

```
                                                                  nome_mae  \
nome_apl
Juliana                                                        ANGERLANE
Josiany                                                            LARA
ISRAELA                                  MARIA DA SAUDE AGUIAR CARNEIRO
Juliana          FRANCISCO ROBERTO DIOGO PAZ E                      ...
DIEGO FERREIRA                                KEYT FERNANDES DA SILVA

                 consentimento                 nome_escola  ra_crianca
nome_apl
Juliana                      1  CEI MARIA MENEZES CRISTINO         NaN
Josiany                      1               DINORAH RAMOS         9.0
ISRAELA                      1               ARAUJO CHAVES         NaN
Juliana                      1      MARIA DO CARMO ANDRADE         NaN
DIEGO FERREIRA               1      LEONILIA GOMES PARENTE         NaN

[5 rows x 31 columns]
```

In [200]:
```
# Renomeando o nome das colnas utilizadas para fazer o merge
df1 = df1.rename(columns = {"Nome da criança:":"nome_crianca","Escola:":"nome_escola"
```

In [201]:
```
import numpy as np
# Fazendo o merge do tipo inner
df_merge = pd.merge(df1,df2,on=["nome_escola","nome_crianca"],how="inner")
# Substituindo o valor das criancas vazias por nulos
df1 = df1.replace(np.nan,"Sem Nome")
```

In [202]:
```
#df_merge.to_excel("plan_merged.xlsx")
```

In [203]:
```
#df1["A criança é sorteada?"].value_counts()
```

In [204]:
```
nome_crianca = []
nome_crianca2 = []
# Fazendo dataframe a partir das semelhanças dos nomes das crianças
# Crianças da avaliacao direta
for nome1 in df2["nome_crianca"]:
    # Crianças do formulario
    for nome2 in df1["nome_crianca"]:
        T_max = max(len(nome1),len(nome2))
        # Calculando a distância de similaridade
        distance_LT = Levenshtein.distance(nome1,nome2)
        if distance_LT == 0:
            if nome_crianca.count(nome2) < 3:
                nome_crianca.append(nome2)
                nome_crianca2.append(nome1)
        else:
            # Normalizando a distância
            norm_dist = T_max/(distance_LT)
            # Selecionando apenas os nomes parecidos
```

```
                    if norm_dist <= 0.3:
                        if nome_crianca.count(nome2) < 3:
                            nome_crianca.append(nome2)
                            nome_crianca2.append(nome1)
```

```
In [205]: len(nome_crianca)
```

```
Out[205]: 649
```

```
In [206]: dicte = {keys:value for keys,value in zip(nome_crianca2,nome_crianca)}
```

```
In [207]: #dicte
```

```
In [208]: cont = 0
          removi = []
          for key,val in zip(dicte.keys(),dicte.values()):
              if key != val:
                  cont+=1
                  #print("key: {} e val: {}".format(key,val))
              else:
                  removi.append(val)
          print(cont)
```

```
0
```

```
In [209]: print(len(nome_crianca))
          #print(len(removi))
```

```
649
521
```

```
In [210]: # Removendo Pessoas sorteadas.
          df1.set_index("nome_crianca",inplace = True)
          nao_sorteado = df1.drop(nome_crianca)
```

```
In [211]: nao_sorteado.shape
```

```
Out[211]: (1492, 59)
```

```
In [212]: df1.shape
```

```
Out[212]: (2141, 59)
```

```
In [179]: len(remove)
```

```
Out[179]: 514
```

```
In [ ]:
```