Clean _relg

November 29, 2021

1 Cleaning the data

In this notebook, coming from the raw data of the egos and their alteris, we will organize the data and extract two different dataframes. In this dataframes, we will save: > * Personal information about attributes of the ego and the alteri. We are talking about categories like sex, age, wealth... We will save this dataframe as *all_data_clean.csv*. > * For each ego, every connection with their alteri. This will be useful is order to build our ego networks. We will save this dataframe as *contactos.csv*. ***

The only change we have done in the religion case in this file is the reorganization and adaptation of the two different surveys of religion.

1.1 Load the .csv files

We import the packages numpy and pandas in order to manipulate the .csv files. We load two different dataframes, one named *df* that contains information about the different alteri connected to each ego. It also contains information about the closeness in each one of the relationships ego - alteri. The second dataframe (named *dfaux*) contains information about individual attributes of the egos.

[1]:
```
###Import the relevant packages
import numpy as np
import pandas as pd

###Load data, # replace with the working directory
df=pd.read_csv(r'/home/juan/Python/Acculturation/all_data(2).
 ↪csv',low_memory=False,encoding='cp1252')
dfaux=pd.read_csv(r'/home/juan/Python/Acculturation/all_updated_2.
 ↪csv',low_memory=False,encoding='cp1252')
```

1.2 Correct typos in the original dataframes

There are some mistakes in the .csv files. In the dataframe named *df* , we have to delete some columns that are loaded, but filled with *NaN* characters. There are also some special characters that need to be erased and a couple of errors in the anonymization. For both dataframes, we transform the whole dataframe to lowercase and reset their indexes.

[2]:
```
###Delete the irrelevant columns
del df['Unnamed: 0']
```

```
coldel=df.columns[0:60]
df=df[coldel]

###Correct the different typos and anonymization problems in the dataframe
df.set_index(['egoID','alterID'],inplace=True)
for col in df.columns:
    df[col]=df[col].replace({',':''},regex=True)
    df[col]=df[col].replace({'\(':''},regex=True)
    df[col]=df[col].replace({'\]':''},regex=True)
    df[col]=df[col].replace({'\[]':''},regex=True)
    df[col]=df[col].replace({'\'':''},regex=True)
    df[col]=df[col].replace({'HENRY>':'USA_PU_en_e53_a27'},regex=True)
    df[col]=df[col].replace({'Frederick>':'USA_PU_es_e1_a38'},regex=True)
dfaux.rename(columns={'EGOID':'egoID'},inplace=True)

###Fill the NaN in  both dataframes
df.fillna('',inplace=True)
dfaux.fillna('',inplace=True)

###Set to lowercase and reset indexes
df=df.apply(lambda x: x.astype(str).str.lower())
dfaux=dfaux.apply(lambda x: x.astype(str).str.lower())
df.reset_index(inplace=True)
```

1.3 Convert all the answers to the questionnaires to an unique language

In the dataframe *df* there are some personal questions that are answered in the mothertongue of each community. In the following cell we build up some Python dictionaries in order to translate the different answers to an unique language, in this case, it will be spanish. Then, we will use this dictionaries to replace the answers in our dataframes.

[3]:
```
###Dictionary of answer to the closeness of an alteri
trad = {
    'close': 'proximo/a',
    'not at all close':'no me siento nada proximo/a',
    'not very close':'no muy proximo/a',
    'somewhat close':'bastante proximo/a',
    'very close':'muy proximo/a',
    'pa pwoch di tou':'no me siento nada proximo/a',
    'pa two pwoch':'no muy proximo/a',
    'pwoch':'proximo/a',
    'tre pwoch':'muy proximo/a',
    'yon ti jan pwoch':'bastante proximo/a'
}

###Dictionary of answers to the probability of two alteris knowing personally␣
 ↪each other.
```

```
trad4={
        'very likely': 'muy probablemente',
        'maybe'  : 'podria ser',
        'not at all likely':'poco probable',
        'anpil chans': 'muy probablemente',
        'petet': 'podria ser',
        'patap gen chans ditou':'poco probable'
}

###Dictionary of origins of the different egos
dict1={'USA_CO':'estados unidos,colombia',
         'USA_CU':'estados unidos,cuba',
         'USA_DO':'estados unidos,republica dominicana',
         'USA_HA':'estados unidos,haiti',
         'USA_ME':'estados unidos,mexico',
         'USA_PU':'estados unidos,puerto rico',
         'SP_AR':'espana,argentina',
         'SP_GU':'espana,guinea ecuatorial',
         'SP_MA':'espana,marruecos',
         'SP_RE':'espana,republica dominicana',
         'SP_SE':'espana,senegambia'}

###Dictionary of origins of the alteri
dict2={'colombia':'CO',
       'republica dominicana':'DO',
       'puerto rico':'PU',
       'senegal':'SE',
       'estados unidos':'USA',
       'ecuador':'EC',
        'otro':'OT',
       'mexico':'ME',
       'espana':'SP',
       'cuba':'CU',
       'gambia':'SE',
       'haiti':'HA',
       'other':'OT',
       'united states':'USA',
       'spain':'SP',
       'do not know':'DK',
       'dominican republic':'DO',
       'marruecos':'MA',
       'twice a week':'DK',
       'refused':'RF',
       'morocco':'MA',
       'puerto rico':'PU',
       'puerto rico ':'PU',
       'argentina':'AR',
```

```
       'un pais de america':'AM',
       'un pais de europa':'EU',
       'otro pais':'OT',
       'una vez al ano':'DK',
       'dos veces al ano':'DK',
       'no':'DK',
       'si':'DK'}

### Additional dictionary for answers that were misplaced.
dict3={ 'colombia':'CO',
        'estados unidos':'USA',
        'republica dominicana':'DO',
        'senegal':'SE',
        'marruecos':'MA',
        'espana':'SP',
        'na':'DK',
        'otro':'OT',
        'united states':'USA',
        'other':'OT',
        'cuba':'CU',
        'spain':'SP',
        'puerto rico':'PU',
        'ecuador':'EC',
        'mexico':'ME',
        '21 to 30':'DK',
        'dominican republic':'DO',
        'puerto rico':'PU',
        'puerto rico ':'PU',
        'morocco':'MA',
        'gambia':'GA',
        'haiti':'HA',
        'un pais de america':'AM',
        'argentina':'AR',
        'un pais de europa':'EU',
        'mujer':'DK',
        'otro pais':'OT',
        '21 a 30':'DK',
        'guinea ecuatorial':'GU',
        '31 a 40':'DK'
}

###Dictionary of closeness in the relationship ego-alteri
trad2={'no me siento nada proximo/a':5,
       'no muy proximo/a': 4,
       'proximo/a':3,
       'bastante proximo/a':2,
       'muy proximo/a':1,
```

```
        '4':0,
        'once a month':0,
        'dos veces al anyo':0,
        '13':0,
        '30':0,
        'na':0
}

###Dictionary for the columns that indicates how the ego met an alteri (with␣
 ↪wrong answers)
Acondict={'in person':'en persona',
          'by phone':'por telefono',
          'by email':'por correo electronico',
          'by mail':'por carta',
          'other':'otra',
          'no':'DK',
          '[':'DK',
          'puerto rico':'DK',
          'argentina':'DK',
          'marruecos':'DK'
}

###Dictionary for the column related to the contact frequency ego-alteri.
Afrqdict={'colombia':'DK',
          'estados unidos':'DK',
          'twice a year':'dos veces al anyo',
          'twice a week':'dos veces a la semana',
          'every day':'cada dia',
          'twice a month':'dos veces al mes',
          'once a week':'una vez a la semana',
          'once a month':'una vez al mes',
          'once a year':'una vez al anyo',
          'dos veces al ano':'dos veces al anyo',
          'una vez al ano':'una vez al anyo',
          'do not know':'DK',
          'republica dominicana':'DK',
          'puerto rico':'DK',
          '[':'DK',
          'united states':'DK',
          '1 fwa per semen':'una vez a la semana',
          '2 fwa per mwa':'dos veces al mes',
          '1 fwa pa mwa':'una vez al mes',
          '2 fwa pa an':'dos veces al anyo',
          '1 fwa pa an':'una vez al anyo',
          '2 fwa pa semen':'dos veces a la semana',
          'chak jou':'cada dia',
          'puerto rico':'DK',
```

```
          'na':'DK'
}

###Dictionary related to the column of "Ahlp" which tell us if the ego-alteri␣
 ↪couple talk about
###their health and feelings
Ahlpdict={'[':'DK',
          'yes':'si',
          'do not know':'DK',
            'na':'DK',
            'no fumado':'DK',
            'united states':'DK',
            'never smoked':'DK',
            'argentina':'DK',
            'marruecos':'DK'
}

###Dictionary related to the column of the age of the alteri.
Aol2dict={'21 to 30':'21 a 30',
          '31 to 40':'31 a 40',
          '41 to 50':'41 a 50',
          '11 to 20':'11 a 20',
          '51 to 60':'51 a 60',
          '61 to 70':'61 a 70',
          '71 to 80':'71 a 80',
          '81 to 90':'81 a 90',
          '91 to 100':'91 a 100',
          'puerto rico':'DK',
          'do not know':'DK',
          'puerto rico':'DK',
          'united states':'DK',
          'na':'DK',
          'other':'DK',
          'estados unidos':'DK',
          'dominican republic':'DK',
          'colombia':'DK',
          'republica dominicana':'DK',
          'ecuador':'DK',
          'no':'DK',
          'every day':'DK'
}

###Dictionary related to the "Apro" column
Aprodict={'yes':'si',
          'non':'no',
          'wi':'si',
          'man':'DK',
```

```
          'na':'DK',
          'do not know':'DK',
          'woman':'DK',
          'white':'DK',
          'moreno/a o mestizo/a':'DK',
          'blanco/a':'DK',
          '10':'DK'
}

###Dictionary related to the race of the alteri
Aracdict={'white':'blanco/a',
            'brown/mestizo':'moreno/a o mestizo/a',
            'black':'negro/a',
            'other':'otro/a',
            'nwa':'negro/a',
            'puerto rico':'DK',
            'senegal':'DK',
            'na':'DK',
            'otro':'otro/a',
            'united states':'DK',
            'familiar por descendencia':'DK',
            'school':'DK',
            'espana':'DK',
            'ecuador':'DK',
            'blan':'blanco/a'

}

###Dictionary related to the relationship between the ego and the alteri
Areldict={
    'familiar por descendencia':'sangre',
'familiar por matrimonio':'matrimonio',
'blood relative':'sangre',
'alguien con el que se encuentra a causa de una tercera persona':'tercera␣
 ↪persona',
'alguien con quien trabaja del mismo nivel':'compañero de trabajo',
'desde la juventud':'juventud',
'school':'escuela',
'someone you met through someone else':'alguien con el que se encuentra a causa␣
 ↪de una tercera persona',
'alguien que conocio a traves de alguna asociacion o club':'asociacion',
'someone you work with':'compañero de trabajo',
'alguien para el que trabaja':'superior laboral',
'relative through marriage':'matrimonio',
'neighbor':'vecino/a',
'other':'otra',
'esposa_o o pareja':'pareja',
```

```
'someone you met through an organization':'asociacion',
'from childhood':'juventud',
'alguien con el que se encuentra en la iglesia o centro de culto':'centro de␣
 ↪culto',
'someone you work for':'superior laboral',
'spouse or significant other':'pareja',
'alguien con quien trabaja al mismo nivel':'compañero de trabajo',
'alguien con que se encuentra a causa de una tercera persona':'tercera persona',
'alguien que trabaja para usted':'subordinado laboral',
'spouse of significant other':'pareja',
'someone you met at religious service':'centro de culto',
'famni pa san':'sangre',
'someone that works for you':'alguien que trabaja para usted',
'esposa/o o pareja':'pareja',
'yon moun ke yon lot moun te prezante-w':'tercera persona',
'lekol':'escuela',
'famni pa maryaj':'matrimonio',
'yon moun ke ou te fe konesans nan yon oganizasyon':'asociacion',
'alguien con el que se encuentra en la iglesia o cento de culto':'centro de␣
 ↪culto',
'na':'DK',
'yon moun ke ou te fe konesans nan legliz la':'centro de culto',
'lot':'DK',
'madam/mari':'pareja',
'someone you met at a religious service':'centro de culto',
'hombre':'DK',
'depi ou pitit ou konnen moun nan':'juventud',
'man':'DK',
'do not know':'DK',
'vwazen':'vecino/a',
'51 a 60':'DK',
'41 a 50':'DK',
'11 to 20':'DK',
'11 a 20':'DK'
}

###Dictionary related to the sex of the alteri
Asexdict={'man':'hombre',
'woman':'mujer',
'fanm':'mujer',
'gason':'hombre',
'dk':'DK',
'barcelona':'DK',
'st louis':'DK',
'do not know':'DK',
'new york':'DK',
'na':'DK',
```

```
'milano':'DK',
'salamanca':'DK',
'no':'DK',
'dakar':'DK',
'marsella':'DK',
'paris':'DK',
'bilbao':'DK',
'port a prince':'DK',
'girona':'DK',
'jersey city':'DK',
'lerida':'DK',
'madrid':'DK',
'nueva york':'DK',
'lloret':'DK'
}

###Dictionary related to the smoking frequency of the alteri.
Asmodict={
'no fumador':'no',
'no fumado':'no',
'yes':'si',
'never smoked':'no',
'fumo cada dia':'diario',
'former smoker':'exfumador',
'fumaba pero lo deje':'exfumador',
'smoke everyday':'diario',
'smoke occasionally':'ocasional',
'fumo ocasionalmente':'ocasional',
'non':'no',
'no me siento nada proximo/a':'DK',
'muy proximo/a':'DK',
'bastante proximo/a':'DK',
'no muy proximo/a':'DK',
'wi':'si',
'na':'DK',
'proximo/a':'DK',
'somewhat close':'DK'
}



###Replace all the different dictionaries in the original dataframe and␣
 ↪renaming some columns asociated
###to them.
df['Clos'].replace(trad,inplace=True)
df.replace(trad4,inplace=True,regex=True)
df['Clos'].replace(trad2,inplace=True)
```

```
df['Afrm'].replace(dict2,inplace=True)
df['Aliv'].replace(dict3,inplace=True)
col_int = df.columns[16:len(df.columns)]
for j in col_int:
    df[j]=df[j].str.replace('muy probablemente','Muyprobablemente')
    df[j]=df[j].str.replace('podria ser','Podriaser')
    df[j]=df[j].str.replace('poco probable','Pocoprobable')

df.rename(columns={"Afrm": "alter_origin", "Aliv": "alter_residence"}, inplace␣
 ↪= True)
df['Acon'].replace(Acondict,inplace=True)
df['Afrq'].replace(Afrqdict,inplace=True)
df['Ahlp'].replace(Ahlpdict,inplace=True)
df['Aol2'].replace(Aol2dict,inplace=True)
df['Apro'].replace(Aprodict,inplace=True)
df['Arac'].replace(Aracdict,inplace=True)
df['Arel'].replace(Areldict,inplace=True)
df['Asex'].replace(Asexdict,inplace=True)
df['Asmo'].replace(Asmodict,inplace=True)
```

1.4 Merge both dataframes

At this point, we will create a dataframe based on merging the individual attributes from the egos and the ones from the alteris. This will be our primary source of information about different social categories of the individuals in our dataset. We will also split the format of the identification of the ego and the alteri. In the original .csv file an ego was identified by: *residence of the ego_origin of the ego_language of the ego_ego identification number*, and the same for the alteri. We split this format into several fields. At the end, we unify the different names used for each origin/language.

[4]:
```
###Merge both dataframes and create an updated df
df=df.apply(lambda x: x.astype(str).str.lower())
dfaux['ego_number'] = range(len(dfaux))
dfaux2=pd.merge(df, dfaux, on='egoID')
df = dfaux2

###Create new IDs for the merged dataframe
dict_ego_num = dict(zip(df['ego_number'].unique(),range(1,len(df['ego_number'].
 ↪unique())+1)))
df['ego_number'].replace(dict_ego_num,inplace=True)

###Split the format of the identification
ego_residence = [0]*len(df)
ego_origin = [0]*len(df)
ego_language = [0]*len(df)
alter_language = [0]*len(df)
alter_number = [0]*len(df)
for i in range(len(df)):
```

```
    current_ego = df['egoID'].iloc[i].split('_')
    current_alter = df['alterID'].iloc[i].split('_')
    ego_residence[i] = current_ego[0]
    ego_origin[i] = current_ego[1]
    ego_language[i] = current_ego[2]
    alter_language[i] = current_alter[2]
    alter_number[i] = int(current_alter[-1].replace('a',''))
###Use this new identification in the updated df
df['ego_residence'] = ego_residence
df['ego_origin'] = ego_origin
df['ego_language'] = ego_language
df['alter_number'] = alter_number
df['alter_language'] = alter_language

###Merge the several names used for the same origin and/or language
origins = df['ego_origin'].unique()
languages = df['ego_language'].unique()
df.loc[df.ego_origin=='re', 'ego_origin'] = 'do'
df.loc[df.ego_origin=='re', 'alter_origin'] = 'do'
df.loc[df.ego_origin=='re', 'alter_residence'] = 'do'
```

1.5 Creation of a dataframe for the structure of the ego networks

We recall the merged dataframe from above and change the format the final columns of it, the ones that contain the information about the intensity of the relationships betweeen alteri. We separate this information and extract from it introducing a numerical encoding for the intensity of this alteri-alteri relationships.

[5]:
```
###Creation of the use that will be use to set the dataframe
alter_1 = []
alter_2 = []
intensity = []
sub_origin = []
sub_residence = []
sub_num = []
sub_language = []
###Append the information from the merged data
for i in df['ego_number'].unique():
    ego_relationships =df[df['ego_number'] == i].iloc[:,16:60]
    for row in ego_relationships.iterrows():
        for i in row[1]:
            if i!='':
                sub_origin.append(df['ego_origin'].loc[row[0]])
                sub_residence.append(df['ego_residence'].loc[row[0]])
                sub_num.append(df['ego_number'].loc[row[0]])
                sub_language.append(df['ego_language'].loc[row[0]])
                alter_1.append(df['alterID'].loc[row[0]].split('_')[-1])
```

```
                alter_2.append(i.split('_')[-1].split(' ')[0])
                intensity.append(i.split(' ')[-1])
###Creation of the dataframe
contactos=pd.DataFrame({'Alter':alter_1,'Alter2':alter_2,'Value':intensity,'sub/
↪origin':sub_origin,
                        'sub/residence':sub_residence,'sub/num':sub_num,'sub/
↪language':sub_language})

###Dictionary to translate and apply the numerical encoding
trad5={'muyprobablemente':3,
       'podriaser':2,
       'pocoprobable':1,
}
contactos.replace(trad5,inplace=True,regex=True)
###Solve the typos introduced in the encoding
contactos.dropna(inplace=True)
contactos['Alter'] = contactos['Alter'].str.replace('a', '')
contactos['Alter2'] = contactos['Alter2'].str.replace('a', '')
contactos=contactos.apply(lambda x: x.astype(str).str.lower())
#Otro@s,Cristianos, musulmanes
dic_usa = {1:2,2:2,3:2,4:3,5:1,6:1,-99:1}
dic_spa = {1:3,2:2,3:2,4:1,5:1,6:1,-99:1}
dic_global =[dic_usa,dic_spa]
resid = ["usa","sp"]
for k1 in range(len(df)):
    k2 = resid.index(df["ego_residence"].iloc[k1])
    df["RELG"].iloc[k1] = dic_global[k2][int(df["RELG"].iloc[k1])]
```

```
/home/juan/.local/lib/python3.8/site-packages/pandas/core/indexing.py:1732:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
```

1.6 Save the two dataframes obtained

In order to save a clean result, we delete the columns that have been used to set the *contacts* dataframe and introduce some changes in the data types. Then we save this two dataframes as two different .csv files.

[6]:
```
###Delete the columns that has been used to build the previous dataframe
del df['Rating']
for i in df.columns:
    if ('Unnamed' in i): del df[i]
del df['egoID']
del df['alterID']
```

```
###Change some data types and rename columns
df['ego_number'] = pd.to_numeric(df['ego_number'])
contactos['sub/num'] = pd.to_numeric(contactos['sub/num'])
df_copy = df.copy()
df_copy.rename(columns = {"ego_number":"sub/num","ego_residence":"sub/
↪residence",
                      "ego_origin":"sub/origin","ego_language":"sub/
↪language"},inplace=True)
df_copy.rename(columns={'alter_origin':'alter/origin','alter_residence':'alter/
↪residence',
                                  'alter_number':'alter/num'},inplace=True)


df_copy.to_csv('all_data_clean_relg.csv')
contactos.to_csv('Contactos_relg.csv')
```

1.7 Inspect the final results

We show the final result, a sample of 10 rows, for the two dataframes.

[7]:
```
df_copy.sample(10)
```

[7]:

	Acit	Acon	alter/origin	Afrq	Ahlp	\
12217	california	dk	dk	dos veces a la semana	dk	
14715	vic	dk	ma	cada dia	dk	
12195	barcelona	dk	ot	una vez a la semana	dk	
17943	do not know	dk	do	una vez al anyo	dk	
12028	barcelona	dk	am	una vez a la semana	dk	
20841	dakar	dk	se	una vez al mes	dk	
17675	barcelona	dk	sp	una vez a la semana	dk	
8993	vic	dk	sp	una vez a la semana	dk	
14352	vic	dk	ma	una vez a la semana	dk	
6381	new york	dk	sp	dos veces a la semana	dk	

	alter/residence	Aol2	Apro	Arac	\
12217	ot	21 a 30	si	negro/a	
14715	sp	1 a 10	si	blanco/a	
12195	gu	31 a 40	no	negro/a	
17943	do	31 a 40	no	moreno/a o mestizo/a	
12028	sp	21 a 30	si	blanco/a	
20841	se	1 a 10	no	negro/a	
17675	sp	31 a 40	no	moreno/a o mestizo/a	
8993	sp	41 a 50	no	blanco/a	
14352	sp	31 a 40	no	moreno/a o mestizo/a	
6381	usa	51 a 60	si	dk	

```
                      Arel  …  MOS  AOS ACCULTUR ALEVEL sub/num  \
12217              juventud  …                                272
14715                sangre  …                                328
12195                  otra  …                                272
17943              vecino/a  …                                399
12028                  otra  …                                268
20841                sangre  …                                464
17675            asociacion  …                                393
8993   compañero de trabajo  …                                200
14352            matrimonio  …                                319
6381                 pareja  …  0.0  0.0      0.0    0.0     142

      sub/residence sub/origin sub/language alter/num alter_language
12217            sp         gu           es        23             es
14715            sp         ma           es         1             es
12195            sp         gu           es         1             es
17943            sp         do           es        34             es
12028            sp         ar           es        14             es
20841            sp         se           es         7             es
17675            sp         do           es        36             es
8993             sp         ar           es        39             es
14352            sp         ma           es        43             es
6381            usa         pu           en        37             en

[10 rows x 373 columns]
```

[8]:
```
df_copy["RELG"].value_counts()
```

[8]:
```
2    11790
3     5625
1     3915
Name: RELG, dtype: int64
```

This is the distribution of individuals according to religion, where 1 means control group, 2 means christian and 3 means muslim.

[ ]:

[ ]: