import pandas as pd

import random

def replace\_only\_str(x, that, by):

if type(x)==str:

x = x.replace(that, by)

return x

def comma\_to\_dot(df):

for col in df.columns:

df[col] = df[col].apply(lambda x : replace\_only\_str(x, ',', "."))

return df

def clean\_col\_names(df):

for col in df.columns:

if 'Unnamed:' in col:

new\_col\_name = col[9:]

df[new\_col\_name]=df[col]

df.drop(columns=[col], inplace=True)

return df

all\_hh = pd.read\_csv('all\_hh.csv')

adultes = all\_hh[(all\_hh.Age > 20) and (all\_hh.Age < 65)]

adultes\_not\_collectif = adultes[adultes.HouseholdTypeID != 6]

#Etudiants from bxl to bxl nl

nb\_student\_nl = pd.read\_csv("sup\_nl\_bxl\_to\_bxl.csv")

print(nb\_student\_nl)

colnames = adultes\_not\_collectif.columns

colnames = colnames.append(["WorkID", "WorkType"])

students\_nl = pd.DataFrame(columns=colnames)

students\_nl\_off\_or\_on\_campus\_to\_check = pd.DataFrame(columns=colnames)

i = 0

for age in range(21, 65):

nb\_student\_nl\_age = nb\_student\_nl[nb\_student\_nl.Age == age]

nb\_student\_nl\_age\_fe = nb\_student\_nl\_age.Fe

nb\_student\_nl\_age\_ho = nb\_student\_nl\_age.Ho

possible\_fe = adultes\_not\_collectif[(adultes\_not\_collectif.Age == age) and (adultes\_not\_collectif.Gender == 1)]

possible\_ho = adultes\_not\_collectif[(adultes\_not\_collectif.Age == age) and (adultes\_not\_collectif.Gender == 0)]

while nb\_student\_nl\_age\_fe > 0:

elu = random.randrange(0, len(possible\_fe), 1)

ind\_elu = possible\_fe.index[elu]

hh\_id = adultes\_not\_collectif.loc[ind\_elu, "HouseholdTypeID"]

if hh\_id == 1 or hh\_id == 2 or hh\_id == 3 or hh\_id == 4:

students\_nl.loc[i] = [adultes\_not\_collectif.loc[ind\_elu], 5, "Unif off campus"]

elif hh\_id == 5:

students\_nl\_off\_or\_on\_campus\_to\_check.loc[i] = [adultes\_not\_collectif.loc[ind\_elu],"4 or 5", "Unif on or off campus"]

#TODO check whether same ss unif and logement

else:

error

possible\_fe.drop([ind\_elu], inplace=True)

adultes\_not\_collectif.drop([ind\_elu], inplace=True)

nb\_student\_nl\_age\_fe-=1

i+=1

while nb\_student\_nl\_age\_ho > 0:

elu = random.randrange(0, len(possible\_ho), 1)

ind\_elu = possible\_ho.index[elu]

hh\_id = adultes\_not\_collectif.loc[ind\_elu, "HouseholdTypeID"]

if hh\_id == 1 or hh\_id == 2 or hh\_id == 3 or hh\_id == 4:

students\_nl.loc[i] = [adultes\_not\_collectif.loc[ind\_elu], 5, "Unif off campus"]

elif hh\_id == 5:

students\_nl\_off\_or\_on\_campus\_to\_check.loc[i] = [adultes\_not\_collectif.loc[ind\_elu],"4 or 5", "Unif on or off campus"]

#TODO check whether same ss unif and logement

else:

error

possible\_ho.drop([ind\_elu], inplace=True)

adultes\_not\_collectif.drop([ind\_elu], inplace=True)

nb\_student\_nl\_age\_ho-=1

i+=1

print(students\_nl)

students\_nl.to\_csv("student\_21+\_nl\_workid.csv")

print(students\_nl\_off\_or\_on\_campus\_to\_check)

students\_nl\_off\_or\_on\_campus\_to\_check.to\_csv("student\_21+\_nl\_workid\_4\_or\_5\_to\_check.csv")

#Etudiants from bxl to out bxl nl

nb\_student\_nl\_out = pd.read\_csv("sup\_nl\_bxl\_to\_out\_bxl.csv")

print(nb\_student\_nl\_out)

closest\_peri = pd.read\_csv("closest\_peri.csv")

print(closest\_peri) #TODO check unnamed and index (should be ss id)

population = pd.read\_csv('population.csv')

corres\_commune\_ss = pd.DataFrame()

corres\_commune\_ss['Code'] = population.Code

corres\_commune\_ss['Territoire'] = population.Territoire

corres\_commune\_ss['Commune'] = population.Commune

communes = corres\_commune\_ss.Commune

communes.drop\_duplicates(inplace=True)

print(communes)

adultes\_not\_collectif = adultes\_not\_collectif.merge(corres\_commune\_ss, how='inner', left\_on='SectorStatID', right\_on='Code')

print(adultes\_not\_collectif)

corres\_commune\_ss.set\_index("Code", inplace=True, drop=True)

students\_nl\_out = pd.DataFrame(columns=colnames)

i = 0

for age in range(21, 65):

nb\_student\_nl\_out\_age = nb\_student\_nl\_out[nb\_student\_nl\_out.Age == age]

nb\_student\_nl\_out\_age\_fe = nb\_student\_nl\_out\_age.Fe

nb\_student\_nl\_out\_age\_ho = nb\_student\_nl\_out\_age.Ho

possible\_fe = adultes\_not\_collectif[(adultes\_not\_collectif.Age == age) and (adultes\_not\_collectif.Gender == 1)]

possible\_ho = adultes\_not\_collectif[(adultes\_not\_collectif.Age == age) and (adultes\_not\_collectif.Gender == 0)]

while nb\_student\_nl\_out\_age\_fe > 0:

elu = random.randrange(0, len(possible\_fe), 1)

ind\_elu = possible\_fe.index[elu]

work\_ss\_id = closest\_peri.loc[adultes\_not\_collectif.loc[ind\_elu, "SectorStatID"]]

work\_ss\_name = corres\_commune\_ss.loc[work\_ss\_id, "Territoire"]

students\_nl\_out.loc[i] = [adultes\_not\_collectif.loc[ind\_elu], 5, "Unif off campus", work\_ss\_id, work\_ss\_name]

possible\_fe.drop([ind\_elu], inplace=True)

adultes\_not\_collectif.drop([ind\_elu], inplace=True)

nb\_student\_nl\_out\_age\_fe-=1

i+=1

while nb\_student\_nl\_out\_age\_ho > 0:

elu = random.randrange(0, len(possible\_ho), 1)

ind\_elu = possible\_ho.index[elu]

work\_ss\_id = closest\_peri.loc[adultes\_not\_collectif.loc[ind\_elu, "SectorStatID"]]

work\_ss\_name = corres\_commune\_ss.loc[work\_ss\_id, "Territoire"]

students\_nl\_out.loc[i] = [adultes\_not\_collectif.loc[ind\_elu], 5, "Unif off campus", work\_ss\_id, work\_ss\_name]

possible\_ho.drop([ind\_elu], inplace=True)

adultes\_not\_collectif.drop([ind\_elu], inplace=True)

nb\_student\_nl\_out\_age\_ho-=1

i+=1

print(students\_nl\_out)

students\_nl\_out.to\_csv("student\_21+\_nl\_to\_out\_workplace.csv")

sup\_fr\_trajets = pd.read\_csv('superieur\_fr\_trajets.csv')

sup\_fr\_trajets = sup\_fr\_trajets[(sup\_fr\_trajets["Bxl / hors bxl logement"]== "Brussels")]

colnames = adultes\_not\_collectif.columns

colnames = colnames.append(["WorkerID", "WorkerType", "WorkSectorStatID", "WorkSectorStatName"])

students = pd.DataFrame(columns=colnames)

i = 0

#etudiant from bxl to bxl fr TODO

for age in range(21, 65):

sup\_fr\_trajets = sup\_fr\_trajets[sup\_fr\_trajets.Age == age]

for com in communes:

possible = sup\_fr\_trajets[sup\_fr\_trajets.Commune == com]

hh\_id = adultes\_not\_collectif.loc[ind\_elu, "HouseholdTypeID"]

if hh\_id == 1 or hh\_id == 2 or hh\_id == 3:

work\_id = 5

work\_name = "Unif off campus"

elif hh\_id == 5:

#etudiants from bxl to out bxl fr

sup\_fr\_trajets\_hors\_bxl = pd.read\_csv('sup\_fr\_trajet\_hors\_bxl.csv')

for age in range(21, 65):

sup\_fr\_trajets\_hors\_bxl\_age = sup\_fr\_trajets\_hors\_bxl[sup\_fr\_trajets\_hors\_bxl.Age == age]

sup\_fr\_trajets\_hors\_bxl\_age\_fe = sup\_fr\_trajets\_hors\_bxl\_age[sup\_fr\_trajets\_hors\_bxl\_age.Sexe == "Femme"]

sup\_fr\_trajets\_hors\_bxl\_age\_ho = sup\_fr\_trajets\_hors\_bxl\_age[sup\_fr\_trajets\_hors\_bxl\_age.Sexe == "Homme"]

for com in communes:

nb\_to\_allocate\_fe = sup\_fr\_trajets\_hors\_bxl\_age\_fe[sup\_fr\_trajets\_hors\_bxl\_age\_fe["Libellé de la commune de domicile légal"]==com]

print(nb\_to\_allocate\_fe) #TODO verifier not null because of majuscules

nb\_to\_allocate\_ho = sup\_fr\_trajets\_hors\_bxl\_age\_ho[sup\_fr\_trajets\_hors\_bxl\_age\_ho["Libellé de la commune de domicile légal"]==com]

possible\_fe = adultes\_not\_collectif[(adultes\_not\_collectif.GenderID == 1) and (adultes\_not\_collectif.Commune == com) and

(adultes\_not\_collectif.Age == age)]

print(possible\_fe) #TODO verifier not null because of majuscules

possible\_ho = adultes\_not\_collectif[(adultes\_not\_collectif.GenderID == 0) and (adultes\_not\_collectif.Commune == com) and

(adultes\_not\_collectif.Age == age)]

while nb\_to\_allocate\_fe > 0:

elu = random.randrange(0, len(possible\_fe), 1)

ind\_elu = possible\_fe.index[elu]

work\_ss\_id = closest\_peri.loc[adultes\_not\_collectif.loc[ind\_elu, "SectorStatID"]]

work\_ss\_name = corres\_commune\_ss.loc[work\_ss\_id, "Territoire"]

students.loc[i] = [adultes\_not\_collectif.loc[ind\_elu], 5, "Unif off campus", work\_ss\_id, work\_ss\_name]

possible\_fe.drop([ind\_elu], inplace=True)

adultes\_not\_collectif.drop([ind\_elu], inplace=True)

nb\_to\_allocate\_fe-=1

i+=1

while nb\_to\_allocate\_ho > 0:

elu = random.randrange(0, len(possible\_ho), 1)

ind\_elu = possible\_ho.index[elu]

work\_ss\_id = closest\_peri.loc[adultes\_not\_collectif.loc[ind\_elu, "SectorStatID"]]

work\_ss\_name = corres\_commune\_ss.loc[work\_ss\_id, "Territoire"]

students.loc[i] = [adultes\_not\_collectif.loc[ind\_elu], 5, "Unif off campus", work\_ss\_id, work\_ss\_name]

possible\_ho.drop([ind\_elu], inplace=True)

adultes\_not\_collectif.drop([ind\_elu], inplace=True)

nb\_to\_allocate\_ho-=1

i+=1

print(students)

students.to\_csv("students\_21+\_unif\_workplace.csv")

#Adultes at home

colnames = adultes\_not\_collectif.columns

colnames = colnames.append(["WorkerID", "WorkerType"])

adultes\_at\_home = pd.DataFrame(columns=colnames)

adultes\_not\_collectif\_fe = adultes\_not\_collectif[adultes\_not\_collectif.Gender == 1]

act\_home\_fe = pd.read\_csv("activite\_home\_fe.csv")

print(act\_home\_fe)

sectors = all\_hh['SectorStatID']

act\_home\_fe.set\_index("Code", inplace=True, drop=True)

i = 0

for age in range(21, 65):

act\_home\_fe\_age= act\_home\_fe[age]

for ss in sectors:

nb\_act\_home\_fe = round(act\_home\_fe\_age.loc[ss])

possible\_act\_home\_fe = adultes\_not\_collectif\_fe[adultes\_not\_collectif\_fe.SectorStatID == ss]

while nb\_act\_home\_fe > 0:

elu = random.randrange(0, len(possible\_act\_home\_fe), 1)

ind\_elu = possible\_act\_home\_fe.index[elu]

adultes\_at\_home.loc[i] = [adultes\_not\_collectif.loc[ind\_elu], 7, "StayHome"]

possible\_act\_home\_fe.drop([ind\_elu], inplace=True)

adultes\_not\_collectif.drop([ind\_elu], inplace=True)

nb\_act\_home\_fe-=1

i+=1

adultes\_not\_collectif\_ho = adultes\_not\_collectif[adultes\_not\_collectif.Gender == 0]

act\_home\_ho = pd.read\_csv("activite\_home\_ho.csv")

print(act\_home\_ho)

act\_home\_ho.set\_index("Code", inplace=True, drop=True)

for age in range(21, 65):

act\_home\_ho\_age= act\_home\_ho[age]

for ss in sectors:

nb\_act\_home\_ho = round(act\_home\_ho\_age.loc[ss])

possible\_act\_home\_ho = adultes\_not\_collectif\_ho[adultes\_not\_collectif\_ho.SectorStatID == ss]

while nb\_act\_home\_ho > 0:

elu = random.randrange(0, len(possible\_act\_home\_ho), 1)

ind\_elu = possible\_act\_home\_ho.index[elu]

adultes\_at\_home.loc[i] = [adultes\_not\_collectif.loc[ind\_elu], 7, "StayHome"]

possible\_act\_home\_ho.drop([ind\_elu], inplace=True)

adultes\_not\_collectif.drop([ind\_elu], inplace=True)

nb\_act\_home\_ho-=1

i+=1

print(adultes\_at\_home)

adultes\_at\_home.to\_csv("adultes\_home\_workid.csv")

#TODO hopital ou prison ?