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\_child\_Reallocated.csv')

all\_hh.drop(columns=["Unnamed: 0"], inplace=True)

print(all\_hh)

sectors\_names\_correspondance = pd.read\_csv("sector\_stat.csv", sep=";")

sectors\_names\_correspondance.drop(columns=["Name"], inplace=True)

sectors\_names\_correspondance.drop\_duplicates(inplace=True)

print(sectors\_names\_correspondance)

all\_hh = all\_hh.merge(sectors\_names\_correspondance, left\_on="SectorStatID", right\_on="Code")

print(all\_hh)

sectors = all\_hh['SectorStatID']

sectors.drop\_duplicates(inplace=True)

communes = all\_hh['Commune']

communes.drop\_duplicates(inplace=True)

print(communes)

jeune = all\_hh[(all\_hh.Age > 17) & (all\_hh.Age < 21)]

print(jeune.ChildOrParent)

jeune\_parents = jeune[(jeune.ChildOrParent == "parent")]

jeune\_parents = jeune\_parents[(jeune\_parents.HouseholdTypeID == 4)]

print(jeune\_parents)

jeune\_parents['WorkerID']=6

jeune\_parents['WorkerType']="Worker"

print(jeune\_parents)

jeune\_parents.to\_csv("18\_20\_jeune\_parent\_worker\_workId.csv")

jeune\_student = jeune[(jeune.ChildOrParent == "child")]

jeune\_student = jeune\_student[(jeune.HouseholdTypeID == 1) | (jeune.HouseholdTypeID == 4)]

print(jeune\_student)

avance\_retard = pd.read\_csv('avance\_retard\_scolaire.csv', sep=";")

avance\_retard.set\_index('Code', inplace=True, drop=True)

print(avance\_retard)

sectors = all\_hh['SectorStatID']

sectors.drop\_duplicates(inplace=True)

print(sectors)

#En retard d'un an: 18ans mais encore en secondaires

colnames = jeune\_student.columns.tolist()

colnames.extend(["WorkerID", "WorkerType"])

en\_retard\_1 = pd.DataFrame(columns=colnames)

#Garcons

i = 0

for s in sectors:

print('s', s)

possible = jeune\_student[(jeune\_student.GenderID == 0) & (jeune\_student.SectorStatID == s) & (jeune\_student.Age == 18)]

nb\_retard\_str = avance\_retard.loc[s, "#18 ans en secondaires garçon"]

if type(nb\_retard\_str) == str:

nb\_retard\_str = nb\_retard\_str.replace(",", ".")

nb\_retard = round(float(nb\_retard\_str))

while nb\_retard > 0 and len(possible) > 0:

print('i', i)

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

ind\_elu = possible.index[elu]

el = possible.loc[ind\_elu].tolist()

el.extend(["3", "Secondaires"])

en\_retard\_1.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_retard-=1

i+=1

#Filles

for s in sectors:

print("s f", s)

possible = jeune\_student[(jeune\_student.GenderID == 1) & (jeune\_student.SectorStatID == s) & (jeune\_student.Age == 18)]

nb\_retard\_str = avance\_retard.loc[s, "#18 ans en secondaires filles"]

if type(nb\_retard\_str) == str:

nb\_retard\_str = nb\_retard\_str.replace(",", ".")

nb\_retard = round(float(nb\_retard\_str))

while nb\_retard > 0 and len(possible) > 0:

print("i f", i)

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

ind\_elu = possible.index[elu]

el = possible.loc[ind\_elu].tolist()

el.extend(["3", "Secondaires"])

en\_retard\_1.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_retard-=1

i+=1

print(en\_retard\_1)

en\_retard\_1.to\_csv("jeune\_sec\_en\_retard\_1\_workId.csv")

#En retard de 2 ans: 19ans mais encore en secondaires

en\_retard\_2 = pd.DataFrame(columns=colnames)

#Garcons

i = 0

for s in sectors:

print("s r", s)

possible = jeune\_student[(jeune\_student.GenderID == 0) & (jeune\_student.SectorStatID == s) & (jeune\_student.Age == 19)]

nb\_retard\_str = avance\_retard.loc[s, "#19 ans en secondaires garçon"]

if type(nb\_retard\_str) == str:

nb\_retard\_str = nb\_retard\_str.replace(",", ".")

nb\_retard = round(float(nb\_retard\_str))

while nb\_retard > 0 and len(possible) > 0:

print("i r", i)

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

ind\_elu = possible.index[elu]

el = possible.loc[ind\_elu].tolist()

el.extend(["3", "Secondaires"])

en\_retard\_2.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_retard-=1

i+=1

#Filles

for s in sectors:

print("s r f", s)

possible = jeune\_student[(jeune\_student.GenderID == 1) & (jeune\_student.SectorStatID == s) & (jeune\_student.Age == 19)]

nb\_retard\_str = avance\_retard.loc[s, "#19 ans en secondaires filles"]

if type(nb\_retard\_str) == str:

nb\_retard\_str = nb\_retard\_str.replace(",", ".")

nb\_retard = round(float(nb\_retard\_str))

while nb\_retard > 0 and len(possible) > 0:

print("i r f", i)

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

ind\_elu = possible.index[elu]

el = possible.loc[ind\_elu].tolist()

el.extend(["3", "Secondaires"])

en\_retard\_2.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_retard-=1

i+=1

print(en\_retard\_2)

en\_retard\_2.to\_csv("jeune\_sec\_en\_retard\_2\_workId.csv")

"""

jeune\_student['WorkerID']=5

jeune\_student['WorkerType']="Unif off campus"

"""

jeune\_student\_alone = jeune[(jeune.ChildOrParent == "parent")]

jeune\_student\_alone = jeune\_student\_alone[(jeune\_student\_alone.HouseholdTypeID == 1) | (jeune\_student\_alone.HouseholdTypeID == 2) |

(jeune\_student\_alone.HouseholdTypeID == 3) | (jeune\_student\_alone.HouseholdTypeID == 5)]

jeune\_student = pd.concat([jeune\_student\_alone, jeune\_student], ignore\_index = True)

#Etudiants from bxl to bxl nl

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

print(nb\_student\_nl)

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

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

i = 0

for age in range(18, 21):

print("age", age)

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

nb\_student\_nl\_age\_fe = int(nb\_student\_nl\_age.loc[:, "Fe"])

nb\_student\_nl\_age\_ho = int(nb\_student\_nl\_age.loc[:, "Ho"])

possible\_fe = jeune\_student[(jeune\_student.Age == age) & (jeune\_student.GenderID == 1)]

possible\_ho = jeune\_student[(jeune\_student.Age == age) & (jeune\_student.GenderID == 0)]

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

print("age i", i)

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

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

hh\_id = jeune\_student.loc[ind\_elu, "HouseholdTypeID"]

el = possible\_fe.loc[ind\_elu].tolist()

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

el.extend([5, "Unif off campus"])

students\_nl.loc[i] = el

elif hh\_id == 5:

el.extend(["4 or 5", "Unif on or off campus"])

students\_nl\_off\_or\_on\_campus\_to\_check.loc[i] = el #jeune\_student.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)

jeune\_student.drop([ind\_elu], inplace=True)

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

i+=1

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

print("age i g", i)

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

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

hh\_id = jeune\_student.loc[ind\_elu, "HouseholdTypeID"]

el = possible\_ho.loc[ind\_elu].tolist()

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

el.extend([5, "Unif off campus"])

students\_nl.loc[i] = el

elif hh\_id == 5:

el.extend(["4 or 5", "Unif on or off campus"])

students\_nl\_off\_or\_on\_campus\_to\_check.loc[i] = el

#TODO check whether same ss unif and logement

else:

error

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

jeune\_student.drop([ind\_elu], inplace=True)

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

i+=1

print(students\_nl)

students\_nl.to\_csv("student\_18\_20\_nl\_workid.csv")

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

students\_nl\_off\_or\_on\_campus\_to\_check.to\_csv("student\_18\_20\_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", sep=";")

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)

jeune\_student = jeune\_student.merge(corres\_commune\_ss, how='inner', left\_on='SectorStatID', right\_on='Code')

print(jeune\_student)

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

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

i = 0

for age in range(18, 21):

print("age g bxl out bxl", age)

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 = jeune\_student[(jeune\_student.Age == age) and (jeune\_student.GenderID == 1)]

possible\_ho = jeune\_student[(jeune\_student.Age == age) and (jeune\_student.GenderID == 0)]

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

print("i g bxl out bxl", i)

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

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

el = possible\_fe.loc[ind\_elu].tolist()

work\_ss\_id = closest\_peri.loc[jeune\_student.loc[ind\_elu, "SectorStatID"]]

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

el.extend([ 5, "Unif off campus", work\_ss\_id, work\_ss\_name])

students\_nl\_out.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

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

i+=1

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

print("i g bxl out bxl g", i)

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

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

el = possible\_ho.loc[ind\_elu].tolist()

work\_ss\_id = closest\_peri.loc[jeune\_student.loc[ind\_elu, "SectorStatID"]]

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

el.extend([ 5, "Unif off campus", work\_ss\_id, work\_ss\_name])

students\_nl\_out.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

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

i+=1

print(students\_nl\_out)

students\_nl\_out.to\_csv("student\_18\_20\_nl\_to\_out\_workplace.csv")

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

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

#etudiant from bxl to bxl fr TODO

campus = pd.read\_csv("sup\_campus\_bxl\_fr\_with\_ss\_v2.csv")

print(campus.columns)

ss = pd.DataFrame()

ss['SectorStat'] = campus.sector\_stat

ss.drop\_duplicates(inplace=True)

print(ss)

sectors\_names\_correspondance = pd.read\_csv('sector\_stat.csv', sep=';')

#sectors\_names\_correspondance.set\_index("Code", inplace=True, drop=True)

print(sectors\_names\_correspondance)

ss = ss.merge(sectors\_names\_correspondance, left\_on="SectorStat", right\_on="Code", how='inner')

ss.drop(columns=["Code"], inplace=True)

print(ss)

ss.to\_csv('campus\_fr\_ss\_commune.csv')

communes = ss.Commune

dic = dict()

for com in communes:

list\_ss = ss[ss.Commune == com].SectorStat.tolist()

dic[com]=list\_ss #len(ss[ss.Commune==com])

print(dic)

colnames = jeune\_student.columns.tolist()

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

students = pd.DataFrame(columns=colnames)

age\_min = 18

age\_max = 20

unif\_fr = pd.read("unif\_fr\_bxl\_to\_bxl.csv")

print(unif\_fr)

i=0

for ind in unif\_fr.index:

print("ind", ind)

if unif\_fr.loc[ind, "Sexe"] == "Femme":

sex = 1

elif unif\_fr.loc[ind, "Sexe"] == "Homme":

sex = 0

else:

errorsex

age = unif\_fr.loc[ind, "Age"]

if age >= age\_min and age<= age\_max:

possible = jeune\_student[(jeune\_student.GenderID == sex) & (jeune\_student.Age == age) &

(jeune\_student.Commune == unif\_fr.loc[ind, "Commune\_from"])]

nb = unif\_fr.loc[ind, "Sum of Compte"]

com\_to = unif\_fr.loc[ind, "Commune\_to"]

list\_ss\_to = dic[com\_to]

nb\_per\_ss = round(nb/len(list\_ss\_to))

reste = nb%len(list\_ss\_to)

first\_ss = True

for ss in list\_ss\_to:

print("ss", ss)

if first\_ss:

nb\_per\_ss\_tmp = nb\_per\_ss + reste

first\_ss = False

else:

nb\_per\_ss\_tmp = nb\_per\_ss

while nb\_per\_ss\_tmp > 0 and len(possible) > 0:

print("i ss", i)

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

ind\_elu = possible.index[elu]

el = possible.loc[ind\_elu].tolist()

hh\_type\_id = possible.loc[ind\_elu, "HouseholdTypeID"]

if hh\_type\_id == 3 or hh\_type\_id == 5:

ss\_home = possible.loc[ind\_elu, "SectorStatID"]

if ss\_home == ss:

work\_id = 4

work\_type = "Unif on campus"

else:

work\_id = 5

work\_type = "Unif off campus"

else:

work\_id = 5

work\_type = "Unif off campus"

work\_ss\_name = sectors\_names\_correspondance.loc[ss]

el.extend([work\_id, work\_type, ss, work\_ss\_name])

students.loc[i]=el

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

jeune\_student.drop([ind\_elu], inplace = True)

nb\_per\_ss\_tmp -=1

i+=1

print(students)

students.to\_csv("student\_fr\_bxl\_bxl\_18\_20\_workplace.csv")

"""

for age in range(18, 21):

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 = jeune\_student.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', sep=';')

for age in range(18, 21):

print("age bxl out bxl fr", age)

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:

print("com", com)

nb\_to\_allocate\_fe = sup\_fr\_trajets\_hors\_bxl\_age\_fe[sup\_fr\_trajets\_hors\_bxl\_age\_fe["Commune\_from"]==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["Commune\_from"]==com]

possible\_fe = jeune\_student[(jeune\_student.GenderID == 1) & (jeune\_student.Commune == com) & (jeune\_student.Age == age)]

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

possible\_ho = jeune\_student[(jeune\_student.GenderID == 0) & (jeune\_student.Commune == com) & (jeune\_student.Age == age)]

while nb\_to\_allocate\_fe > 0:

print("i com bxl out bxl fr", i)

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

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

work\_ss\_id = closest\_peri.loc[jeune\_student.loc[ind\_elu, "SectorStatID"]]

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

el = possible\_fe.loc[ind\_elu].tolist()

el.extend([5, "Unif off campus", work\_ss\_id, work\_ss\_name])

students.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_to\_allocate\_fe-=1

i+=1

while nb\_to\_allocate\_ho > 0:

print("i com bxl out bxl fr g", i)

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

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

work\_ss\_id = closest\_peri.loc[jeune\_student.loc[ind\_elu, "SectorStatID"]]

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

el = possible\_ho.loc[ind\_elu].tolist()

el.extend([5, "Unif off campus", work\_ss\_id, work\_ss\_name])

students.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_to\_allocate\_ho-=1

i+=1

print(students)

students.to\_csv("students\_18\_20\_unif\_workplace.csv")

#Jeune at home

colnames = jeune\_student.columns

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

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

jeune\_student\_fe = jeune\_student[jeune\_student.GenderID == 1]

jeune\_student\_fe\_18\_19 = jeune\_student\_fe[(jeune\_student\_fe.Age == 18) | (jeune\_student\_fe.Age == 19)]

to\_allocate\_fe = {"Anderlecht":2, "Evere":1, "Forest":1, "Ganshoren":1, "Jette":1, "Uccle":1} #From activites home fe pivot

i=0

for com in to\_allocate\_fe.keys():

print("com home", com)

nb\_to\_allocate\_fe = to\_allocate\_fe[com]

possible\_fe = jeune\_student\_fe\_18\_19[jeune\_student\_fe\_18\_19.Commune == com]

while nb\_to\_allocate\_fe > 0:

print("com home i", i)

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

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

el = possible\_fe.loc[ind\_elu].tolist()

el.extend([7, "StayHome"])

jeune\_at\_home.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_to\_allocate\_fe-=1

i+=1

jeune\_student\_ho = jeune\_student[jeune\_student.GenderID == 0]

jeune\_student\_ho\_18\_19 = jeune\_student\_ho[(jeune\_student\_ho.Age == 18) | (jeune\_student\_ho.Age == 19)]

to\_allocate\_ho = {"Anderlecht":53, "Bruxelles":4, "Forest":1, "Ganshoren":1, "Saint-Josse-ten-Noode":3, "Schaerbeek":3, "Molenbeek Saint-Jean":4}

for com in to\_allocate\_ho.keys():

print("com home g", com)

nb\_to\_allocate\_ho = to\_allocate\_ho[com]

possible\_ho = jeune\_student\_ho\_18\_19[jeune\_student\_ho\_18\_19.Commune == com]

while nb\_to\_allocate\_ho > 0:

print("com home g i", i)

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

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

el = possible\_ho.loc[ind\_elu].tolist()

el.extend([7, "StayHome"])

jeune\_at\_home.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_to\_allocate\_ho-=1

i+=1

jeune\_student\_fe = jeune\_student[jeune\_student.GenderID == 1]

jeune\_student\_fe\_20 = jeune\_student\_fe[(jeune\_student\_fe.Age == 20)]

to\_allocate\_fe = {"Anderlecht":30, "Auderghem": 3, "Bruxelles":44, "Etterbeek":4, "Evere":15, "Forest":11, "Ganshoren":7, "Ixelles":10, "Jette":11,

"Koekelberg": 5, "Saint-Gilles": 7, "Saint-Josse-ten-Noode":10, "Schaerbeek":31, "Uccle": 7, "Watermael-Boitsfort": 4,

"Berchem Sainte-Agathe": 4, "Molenbeek Saint-Jean":32, "Woluwe Saint-Lambert":4, "Woluwe Saint-Pierre": 3}

for com in to\_allocate\_fe.keys():

print("com home 20:", com)

nb\_to\_allocate\_fe = to\_allocate\_fe[com]

possible\_fe = jeune\_student\_fe\_20[jeune\_student\_fe\_20.Commune == com]

while nb\_to\_allocate\_fe > 0:

print("com home 20 i", i)

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

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

el = possible\_fe.loc[ind\_elu].tolist()

el.extend([7, "StayHome"])

jeune\_at\_home.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_to\_allocate\_fe-=1

i+=1

jeune\_student\_ho = jeune\_student[jeune\_student.GenderID == 0]

jeune\_student\_ho\_20 = jeune\_student\_ho[(jeune\_student\_ho.Age == 20)]

to\_allocate\_ho = {"Anderlecht":36, "Auderghem": 4, "Bruxelles":62, "Etterbeek":5, "Evere":9, "Forest":11, "Ganshoren":6, "Ixelles":14, "Jette":18,

"Koekelberg": 8, "Saint-Gilles": 14, "Saint-Josse-ten-Noode":10, "Schaerbeek":40, "Uccle": 11, "Watermael-Boitsfort": 4,

"Berchem Sainte-Agathe": 6, "Molenbeek Saint-Jean":30, "Woluwe Saint-Lambert":6, "Woluwe Saint-Pierre": 3}

for com in to\_allocate\_ho.keys():

print("com home 20 g :", com)

nb\_to\_allocate\_ho = to\_allocate\_ho[com]

possible\_ho = jeune\_student\_ho\_20[jeune\_student\_ho\_20.Commune == com]

while nb\_to\_allocate\_ho > 0:

print("com home 20 i g", i)

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

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

el = possible\_ho.loc[ind\_elu].tolist()

el.extend([7, "StayHome"])

jeune\_at\_home.loc[i] = el

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_to\_allocate\_ho-=1

i+=1

print(jeune\_at\_home)

jeune\_at\_home.to\_csv("jeune\_at\_home\_workId.csv")

jeune\_student["WorkID"]=6

jeune\_student["WorkType"]="Worker"

print(jeune\_student)

jeune\_student.to\_csv("jeune\_wroker\_workId.csv")

"""

ado\_hopital = pd.DataFrame(columns=colnames)

place\_hopital = 401 #- somme des places pediatrie utilisées

"""

jeune\_collectif = jeune[jeune.HouseholdTypeID == 6]

print(jeune\_collectif)

jeune\_collectif.to\_csv("jeune\_collectif\_hopi\_or\_prison.csv")

"""

i = 0

while len(ado\_collectif) > 0 and place\_hopital > 0 :

elu = random.randrange(0, len(ado\_collectif), 1)

ind\_elu = ado\_collectif.index[elu]

ado\_hopital.loc[i] = [ado\_collectif.loc[ind\_elu], "10", "Hospital"]

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

place\_hopital-=1

i+=1

print(ado\_hopital)

ado\_hopital.to\_csv("ado\_16\_17\_hopital\_workid.csv")

ado\_collectif["WorkID"]=9

ado\_collectif["WorkType"]="Prison"

print(ado\_collectif)

ado\_collectif.to\_csv("ado\_16\_17\_prison\_workid.csv")

#TODO hh\_id 9 prison or hopital

for others years

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

print(act\_home\_fe)

act\_home\_fe= act\_home\_fe["20"]

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

for ss in sectors:

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

possible\_act\_home\_fe = jeune\_student\_fe\_20[jeune\_student\_fe\_20.SectorStatID == ss]

while nb\_act\_home\_fe > 0:

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

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

jeune\_at\_home.loc[i] = [jeune\_student.loc[ind\_elu], 7, "StayHome"]

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_act\_home\_fe-=1

i+=1

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

print(act\_home\_fe)

act\_home\_fe= act\_home\_fe["20"]

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

for ss in sectors:

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

possible\_act\_home\_fe = jeune\_student\_fe\_20[jeune\_student\_fe\_20.SectorStatID == ss]

while nb\_act\_home\_fe > 0:

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

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

jeune\_at\_home.loc[i] = [jeune\_student.loc[ind\_elu], 7, "StayHome"]

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

jeune\_student.drop([ind\_elu], inplace=True)

nb\_act\_home\_fe-=1

i+=1

"""

"""

ado\_sec = ado\_not\_collectif[(ado\_not\_collectif.HouseholdTypeID == 1) or (ado\_not\_collectif.HouseholdTypeID == 4)]

ado\_sec['WorkerID']=3

ado\_sec['WorkerType']="Secondaires"

print(ado\_sec)

ado\_sec.to\_csv("ado\_sec\_workId.csv")

ado\_colloc\_worker = ado\_not\_collectif[ado\_not\_collectif.HouseholdTypeID == 5]

ado\_colloc\_worker['WorkerID']=6

ado\_colloc\_worker['WorkerType']="Worker"

print(ado\_colloc\_worker)

ado\_colloc\_worker.to\_csv("ado\_worker\_colloc\_workId.csv")

#TODO hopital ou prison ?

"""