GIRACS - Analyses

July 10, 2020

1 Geospatial analysis of the determinants of cancer screening participation

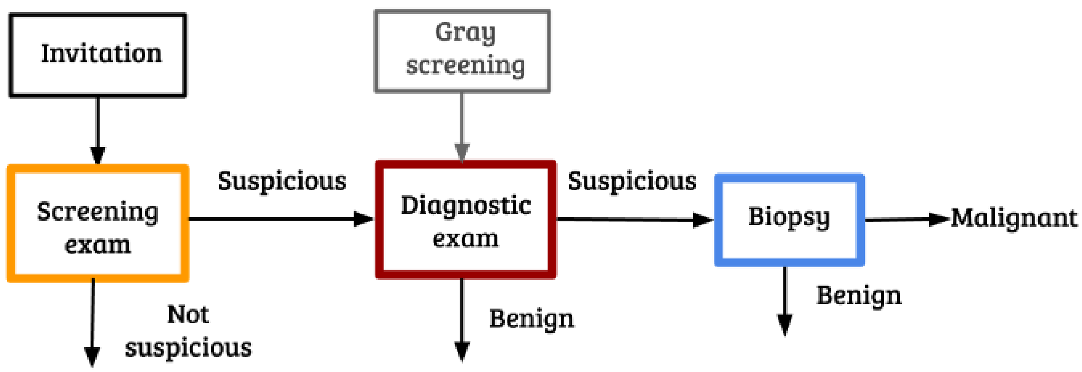


Figure 1. Overview of the breast cancer screening process the way in which it is commonly executed in Europe. In some countries (such as Germany) not all women who are invited participate in regular screening, but sometimes go opportunistically for a diagnostic mammogram. This is referred to as 'gray screening'. (image by author)

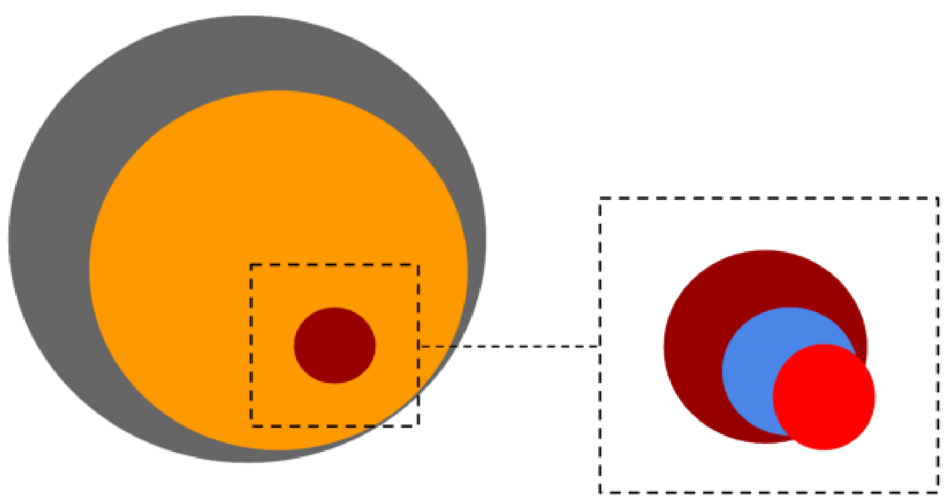


Figure 3. Illustration of different 'rates' in the screening process. The big gray blob represents all women in the age group which are screened, the orange circle all women that actually attend screening, the dark red circle all women that are recalled, the bright blue circle all cases that are biopsied and the bright red circle all cancers (image by author).

[1]:
```
import pandas as pd
import geopandas as gpd
import numpy as np
import os
import glob
import random
import osmnx as ox
import matplotlib.pyplot as plt
import folium
import libpysal as lps
import seaborn as sns
import mapclassify as mc
import esda
from difflib import SequenceMatcher
from pylab import *
import psycopg2
from pathlib import Path
import statsmodels.api as sm
from scipy import stats
from statsmodels.graphics.api import abline_plot
import pysal as ps
import csv
import altair as alt
from matplotlib import colors
from matplotlib.collections import LineCollection
from shapely.geometry import Point, Polygon
import osmnx as ox
from sqlalchemy import create_engine
pd.set_option('display.max_columns', 500)
engine = create_engine('postgresql://postgres@localhost:5432/david')
con = psycopg2.connect(database="david", user="postgres",host="localhost")
# Imports
from geoalchemy2 import Geometry, WKTElement
from sqlalchemy import *
import pandas as pd
import geopandas as gpd
geo_engine = create_engine('postgresql://postgres@localhost:5432/david')
import pandas as pd
import numpy as np
import os
import glob
import random
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import csv
from matplotlib.collections import LineCollection
from sqlalchemy import create_engine
from mgwr.gwr import GWR, MGWR
from mgwr.sel_bw import Sel_BW
from mgwr.utils import compare_surfaces, truncate_colormap
import multiprocessing as mp
from pysal.explore.pointpats import PointPattern, PoissonPointProcess,␣
↪as_window, G, F, J, K, L, Genv, Fenv, Jenv, Kenv, Lenv
import importlib

%matplotlib inline
```

2 Load data

[2]:
```
#Set working directory
mydir = Path(os.getcwd())
```

[3]:
```
sql = """select *  from geounits.communes_ch_2056 where "KANTONSNUM"  = 25;"""
communes = gpd.GeoDataFrame.from_postgis(sql, con = engine, geom_col='geom' )
sql = """select *   from geounits.lake_2056 """
lake = gpd.GeoDataFrame.from_postgis(sql, con = engine, geom_col='geom' )
```

[336]:
```
gdf_centre = gpd.read_file(data_folder/'BC_ScreeningCenters.geojson',driver =␣
↪'GeoJSON')
```

[44]:
```
data_folder = mydir / 'Data' #Set data folder
result_folder = mydir / 'Results' #Set data folder
file = data_folder / "giracs_input.csv" #Data source file containing the␣
↪screening data
df = pd.read_csv(file)
#Transform to geodataframe
geometry = [Point(xy) for xy in zip(df['GKODE'], df['GKODN'])]
# Coordinate reference system : WGS84
crs = {'init': 'epsg:2056'}
gdf = gpd.GeoDataFrame(df, crs=crs, geometry=geometry)
```

3 Data filtering

[45]:
```
#Get number of people in the dataframe
print('Number of people in the dataset: ',len(gdf.numerodossier.unique()))
```

```
patients =␣
↪gdf[['numerodossier','medecin','autremedecin','mammoanterieure','atf','mammo','rappel']].
↪groupby('numerodossier').sum(min_count = 1)
print('Number of people having done a breast cancer screening (mammography):␣
↪',len(patients[patients.mammo > 0]))
```

```
Number of people in the dataset:  131716
Number of people having done a breast cancer screening (mammography):  42648
```

3.0.1 Discard duplicates

[46]:
```
clean_dupli =gdf[(gdf.
↪duplicated(subset=['numerodossier','numeroinvitation'],keep=False))].
↪sort_values(['numerodossier','month_invit','day_invit','mammo','groupeage']).
↪drop_duplicates(subset = ['numerodossier','numeroinvitation'],keep = 'first')
```

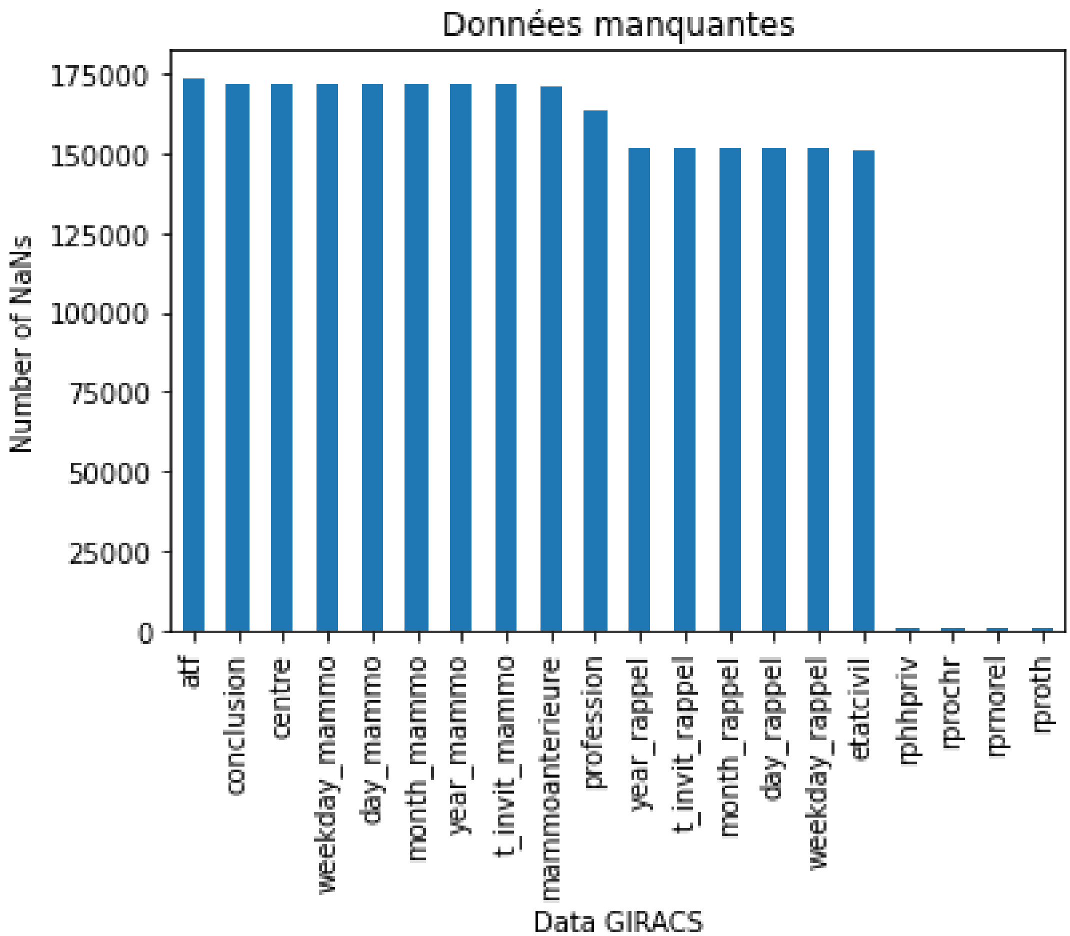
[47]:
```
gdf['_dummy'] = gdf['numerodossier'].astype(str) + gdf['numeroinvitation'].
↪astype(str)
clean_dupli['_dummy'] = clean_dupli['numerodossier'].astype(str) +␣
↪clean_dupli['numeroinvitation'].astype(str)
```

[48]:
```
gdf = gdf[(gdf._dummy.isin(clean_dupli._dummy))==False]
```

[49]:
```
gdf = pd.concat([gdf,clean_dupli])
```

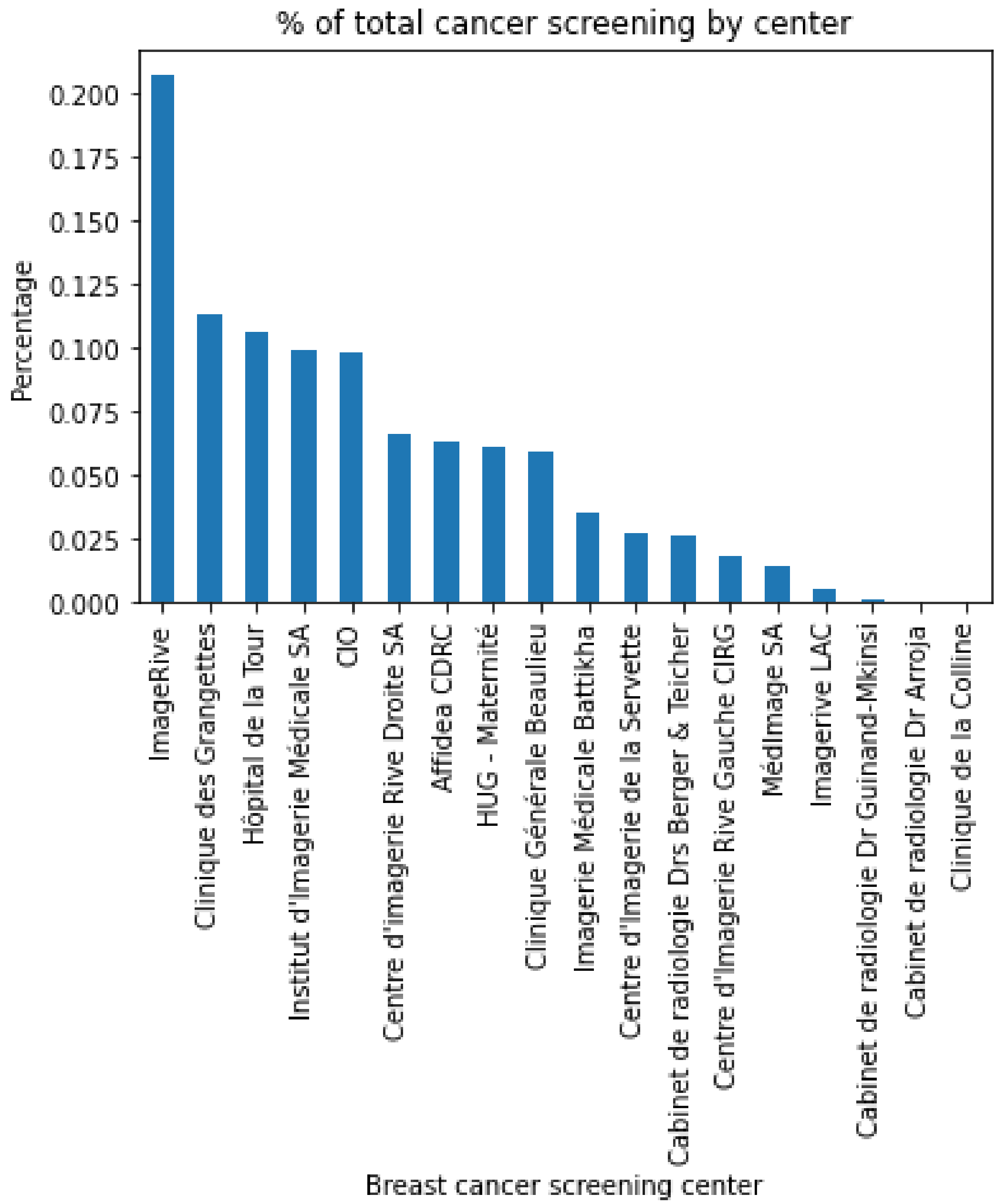
[50]:
```
#Plot the 20 variables having the most NAs
isna_ = gdf.isnull().sum().sort_values(ascending=False)
plt.plot()
plt.title('Données manquantes')
plt_1=isna_[:20].plot(kind='bar')
plt.ylabel('Number of NaNs')
plt.xlabel('Data GIRACS')
```

[50]:
```
Text(0.5, 0, 'Data GIRACS')
```



[51]:
```
plt.plot()
plt.title('% of total cancer screening by center')
gdf.centre.value_counts(normalize=True).plot.bar()
plt.ylabel('Percentage')
plt.xlabel('Breast cancer screening center')
```

[51]: Text(0.5, 0, 'Breast cancer screening center')



We see that the ImageRive is far ahead of any other center, gathering more than 20% of all screenings. We could test:

- Is this correlated to the total population living at less than 1,2,5k ?
  - Probably not that much, I highly doubt that Imagerive LAC has very little population around or that Clinique de la Colline has zero.
  - Questions for Beatrice : Why would that be? Was that a known fact? Close to where people work? Incentives to go there (flyers, reputation of quality, recommendation from physician,convenience of getting an appointment,...)?

[53]:
```
patients.loc[patients.atf > 0, 'atf'] =1
```

[54]:
```
gdf['atf'] = gdf['atf'].astype(str)
gdf.loc[gdf.atf.isna(), 'atf'] = np.nan
gdf['atf'] = gdf['atf'].astype('category')
```

[55]:
```
gdf.loc[gdf.localité.str.contains('Meyrin'),'localité'] = 'Meyrin'
gdf.loc[gdf.localité.str.contains('Lancy'),'localité'] = 'Lancy'
```

3.1 Exclude age categories outside HUG guidelines

In 1999, population-based breast cancer screening was introduced in the canton of Geneva, an area with approximately 500 000 inhabitants. All individuals aged **50-74** are invited from a central screening centre to a biennial screening cycle. The programme was gradually expanded by successively inviting new birth cohorts, and was fully rolled-out in 2013. The different age cohorts included varied in size due to natural vartions in the population.

For analysis of participation patterns, we included all individuals with at least three complete screening rounds

[56]:
```
gdf.groupeage = gdf.groupeage.astype('category')
gdf['groupeage_cat'] = gdf.groupeage.cat.codes
```

[57]:
```
#Get age categories that are too young or too old...(according to HUG␣
↪guidelines)
# QUESTION : Why are these present at all?
age_cats = df.groupby('groupeage').size()
age_cats1000 = age_cats[age_cats > 1000]
age_cats =  pd.DataFrame(age_cats).reset_index()
age_cats.columns = ['groupe_age','n']
```

[58]:
```
df_final = gdf[(gdf.groupeage.isin(age_cats1000.index))]
```

[59]:
```
import six

def render_mpl_table(data, col_width=3.0, row_height=0.625, font_size=14,
                     header_color='#40466e', row_colors=['#f1f1f2', 'w'],␣
↪edge_color='w',
                     bbox=[0, 0, 1, 1], header_columns=0,
                     ax=None, **kwargs):
    if ax is None:
        size = (np.array(data.shape[::-1]) + np.array([0, 1])) * np.
↪array([col_width, row_height])
        fig, ax = plt.subplots(figsize=size)
        ax.axis('off')
```

```
    mpl_table = ax.table(cellText=data.values, bbox=bbox, colLabels=data.
↪columns, **kwargs)

    mpl_table.auto_set_font_size(False)
    mpl_table.set_fontsize(font_size)

    for k, cell in  six.iteritems(mpl_table._cells):
        cell.set_edgecolor(edge_color)
        if k[0] == 0 or k[1] < header_columns:
            cell.set_text_props(weight='bold', color='w')
            cell.set_facecolor(header_color)
        else:
            cell.set_facecolor(row_colors[k[0]%len(row_colors) ])
    return ax
```

[60]:
```
render_mpl_table(age_cats, header_columns=0, col_width=2.0)
```

[60]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9485340b80>

groupe_age	n
30-34	3
35-39	2
40-44	3
45-49	238
50-54	81778
55-59	50471
60-64	40320
65-69	51631
70-74	27112
75-79	500
80-84	105
85-89	18
90-94	1

3.2 Number of invitations by year

Women invited in 2019 are excluded since they might not have had the time to participate at the time of data extraction

[61]:
```
year_cats = df.groupby('year_invit').size()
year_cats =  pd.DataFrame(year_cats).reset_index()
year_cats.columns = ['year','n']
```

```
render_mpl_table(year_cats, header_columns=0, col_width=2.0)
```

[61]: <matplotlib.axes._subplots.AxesSubplot at 0x7f948507f1c0>

year	n
1999	4494
2000	5619
2001	2481
2002	3919
2003	2737
2004	3156
2005	2314
2006	3702
2007	2572
2008	3041
2009	1872
2010	2423
2011	1681
2012	2848
2013	14236
2014	29393
2015	33912
2016	30186
2017	32399
2018	32067
2019	37130

```
[62]: #Filter dataset for : Age categories, year of invitation and number of␣
      ↪invitation
      df_final = df_final[(df_final.year_invit < 2019)]
```

3.3 Number of invitations by woman

```
[69]: n_invit = df_final.groupby('numerodossier').numeroinvitation_seq.nunique()
      df_final = df_final.join(n_invit, on='numerodossier', rsuffix='_n')
```

3.4 Disparity between "numeroinvitation" and actual number of invitations recorded in the database

```
[70]: #Return a sequence that corresponds to the actual numeroinvitation ...without␣
      ↪the weird things we find in the original column
      df_final['dt_invit'] = pd.to_datetime([f'{y}-{m}-{d}' for y, m, d in␣
      ↪zip(df_final.year_invit, df_final.month_invit, df_final.day_invit)])
      df_final = df_final.sort_values(['numerodossier','dt_invit'])
      df_final['numeroinvitation_seq'] = df_final.groupby('numerodossier').
      ↪cumcount()+1
```

```
[71]: df_final.groupby('numeroinvitation_seq').size()
```

```
[71]: numeroinvitation_seq
      1    122782
      2     60815
      3     29976
      4       316
      5        15
      dtype: int64
```

3.4.1 # by numeroinvitation

```
[72]: women_by_n_invite = df_final[['numerodossier','numeroinvitation']].
      ↪drop_duplicates().groupby('numeroinvitation').count().reset_index()
      women_by_n_invite.columns = ['numeroinvitation','n']
      render_mpl_table(women_by_n_invite, header_columns=0, col_width=3.0)
```

```
[72]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9484a412b0>
```

numeroinvitation	n
0	11
1	42033
2	32754
3	27149
4	21739
5	17906
6	15792
7	14688
8	18908
9	14692
10	7642
11	518
12	64
13	7
14	1

3.4.2 # by actual number of invitations recorded in the database

[73]:
```
women_by_n_invite = df_final[['numerodossier','numeroinvitation_n']].
 ↪drop_duplicates().groupby('numeroinvitation_n').count().reset_index()
women_by_n_invite.columns = ['# invitations','n']
render_mpl_table(women_by_n_invite, header_columns=0, col_width=2.0)
```

[73]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94849611c0>

# invitations	n
1	61967
2	30839
3	29660
4	301
5	15

3.5 Women invited at least 3 times

[74]:
```
n_invit3 = n_invit[n_invit>2]
df_final_3invit = df_final[(df_final.numerodossier.isin(n_invit3.index))].
 ↪sort_values(['numerodossier','numeroinvitation'])
```

3.6 Time intervals between any two invitation (to the same woman)

[75]:
```
%%time
df_final['diff_years'] = df_final[['numerodossier','dt_invit']].
 ↪groupby('numerodossier').diff()['dt_invit']/np.timedelta64(1,'Y')
```

```
CPU times: user 46.3 s, sys: 608 ms, total: 46.9 s
Wall time: 47.3 s
```

[76]:
```
dossiers_bug = df_final[df_final.diff_years < 0].numerodossier.values
dossiers_timing_long = df_final[df_final.diff_years > 3].numerodossier.values
dossiers_timing_court = df_final[(df_final.diff_years < 1)&(df_final.diff_years␣
 ↪>= 0)].numerodossier.values
```

```
#
# df_final = df_final[(df_final.numerodossier.isin(dossiers_bug) == False)]
# df_final = df_final[(df_final.numerodossier.isin(dossiers_timing_long) ==␣
↪False)]
# df_final = df_final[(df_final.numerodossier.isin(dossiers_timing_court) ==␣
↪False)]
```

[77]:
```
print('Number of dossiers were the interval between any two invitations is␣
↪negative : {}'.format(dossiers_bug.shape[0]))

print('Number of dossiers were at least one invitation has been sent in an␣
↪interval of more than 3 years after the precedent : {}'.
↪format(dossiers_timing_long.shape[0]))

print('Number of dossiers were at least one invitation has been sent in␣
↪interval of more less than 1 years after the precedent : {}'.
↪format(dossiers_timing_court.shape[0]))
```

```
Number of dossiers were the interval between any two invitations is negative : 0
Number of dossiers were at least one invitation has been sent in an interval of
more than 3 years after the precedent : 2309
Number of dossiers were at least one invitation has been sent in interval of
more less than 1 years after the precedent : 758
```

I am unsure at this time if they should be discarded…It might not be a big problem

3.7 Participation changes between any two invitation (to the same woman)

[84]:
```
# df_final = df_final.
↪drop(['mammo_last_invite_x','mammo_last_invite_y','mammo_last_invite'],axis␣
↪= 1)
```

[79]:
```
df_final['mammo_last_invite'] = df_final[['numerodossier','mammo']].
↪groupby('numerodossier').diff()['mammo']
```

[80]:
```
participation_chg = pd.
↪DataFrame(df_final[['numerodossier','mammo_last_invite']].
↪groupby(['mammo_last_invite']).numerodossier.count()).reset_index()
participation_chg.columns = ['Participation change','n']
render_mpl_table(participation_chg, header_columns=0, col_width=3.0)
```

[80]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9484481f70>
```

Participation change	n
-1.0	6884.0
0.0	76842.0
1.0	7396.0

[81]:
```
participation_change = df_final[['numerodossier','mammo_last_invite']].dropna().
 ↪groupby(['numerodossier']).mammo_last_invite.nunique().reset_index()
participation_change.columns = ['numerodossier','participation_change']
df_final = df_final.merge(participation_change, on='numerodossier',how = 'left')
```

3.8 Final dataset

[82]:
```
df_final.loc[df_final.atf == 'nan','atf'] = np.nan
```

[83]:
```
df_final['atf'] = df_final['atf'].astype(float)
```

[84]:
```
df_final = df_final.reset_index(drop = True)
```

[85]:
```
print('Number of people without a conclusion (NULL) while having done a␣
 ↪screening : ',len(df_final[(df_final.conclusion.isnull()==True)&(df_final.
 ↪day_mammo.isnull()==False)]))
```

```
Number of people without a conclusion (NULL) while having done a screening :  17
```

[86]:
```
#Get number of people in the dataframe
print('Number of people in the dataset: ',len(df_final.numerodossier.unique()))
patients =␣
 ↪df_final[['numerodossier','medecin','autremedecin','mammoanterieure','atf','mammo','rappel'
 ↪groupby('numerodossier').sum(min_count = 1)
print('Number of people having done a breast cancer screening (mammography):␣
 ↪',len(patients[patients.mammo > 0]))
```

```
Number of people in the dataset:  122782
Number of people having done a breast cancer screening (mammography):  40300
```

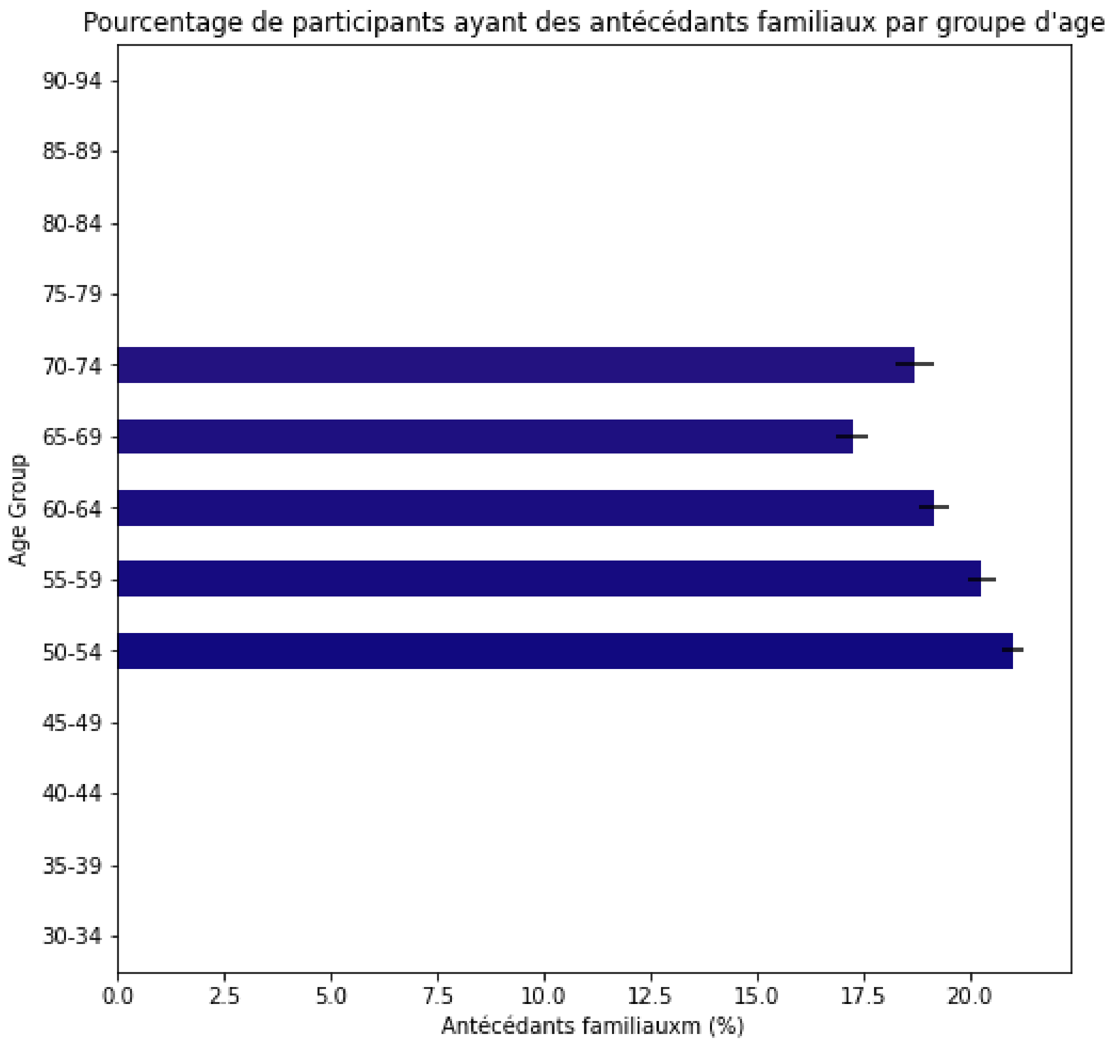
[87]:
```
patients = patients.merge(participation_change, on='numerodossier',how = 'left')
```

4 Descriptive analyses

[88]:
```
age_group = df_final.groupby('groupeage').sum()/df_final.groupby('groupeage').
↪count()
age_group['Age group'] = age_group.index
```

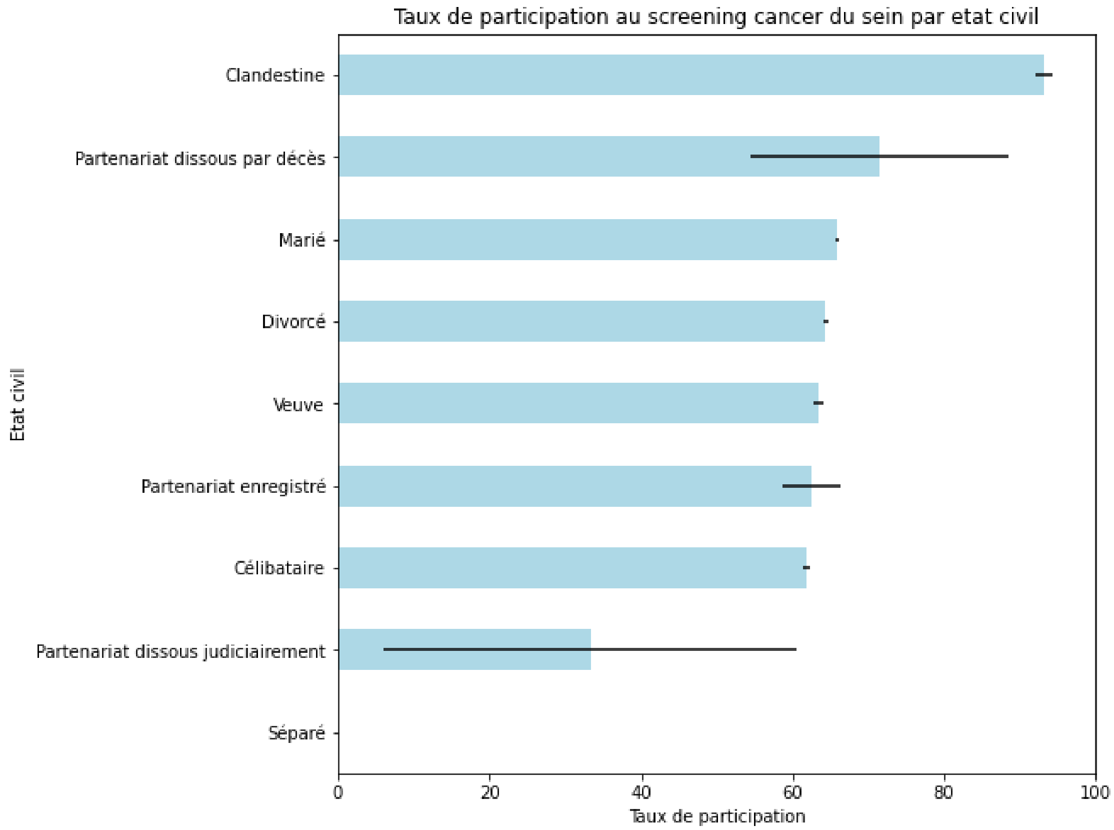
[89]:
```
c = df_final.groupby(['groupeage','atf'])['atf'].count()
test = (c / c.groupby(level=[0]).transform("sum")).unstack('atf').fillna(0)
yerr = ((test[1]*(1-test[1]))/(c.unstack('atf')[0]+c.unstack('atf')[1]))**(0.5)
test *=100
yerr *= 100
```

[90]:
```
f,ax = plt.subplots(figsize = (8,8))
my_colors = ['r', 'b']*5 # <-- this concatenates the list to itself 5 times.
my_colors = [(0.5,0.8,0.5), (1, 1, 1)]*5 # <-- make two custom RGBs and repeat/
↪alternate them over all the bar elements.
my_colors = [(x/59.0, x/114.0, 0.5) for x in range(len(test[1]))]
test[1].plot(kind = 'barh',xerr = yerr,color = my_colors,ax = ax)
ax.set_title("Pourcentage de participants ayant des antécédants familiaux par␣
↪groupe d'age")
ax.set_ylabel('Age Group')
ax.set_xlabel('Antécédants familiauxm (%)')
plt.savefig(result_folder/'atf_by_agegroup.png',bbox_inches='tight',␣
↪transparent=True,dpi = 400)
```



[91]:

```
c = df_final.groupby(['etatcivil','mammo'])['mammo'].count()
test = (c / c.groupby(level=[0]).transform("sum")).unstack('mammo').fillna(0)
yerr = ((test[1]*(1-test[1]))/(c.unstack('mammo')[0]+c.
 ↪unstack('mammo')[1]))**(0.5)
test *=100
yerr *= 100
f,ax = plt.subplots(figsize = (8,8))
test[1].sort_values().plot(kind = 'barh',xerr = yerr,xlim = (0,100),color =␣
 ↪['lightblue'],ax = ax)
ax.set_title('Taux de participation au screening cancer du sein par etat civil')
ax.set_ylabel('Etat civil')
ax.set_xlabel('Taux de participation')
plt.savefig(result_folder/'tx_participation_etatcivil.png',bbox_inches='tight',␣
 ↪transparent=True,dpi = 400)
```



[92]:
```
patients_mammo = patients.groupby('mammo').count().medecin
```

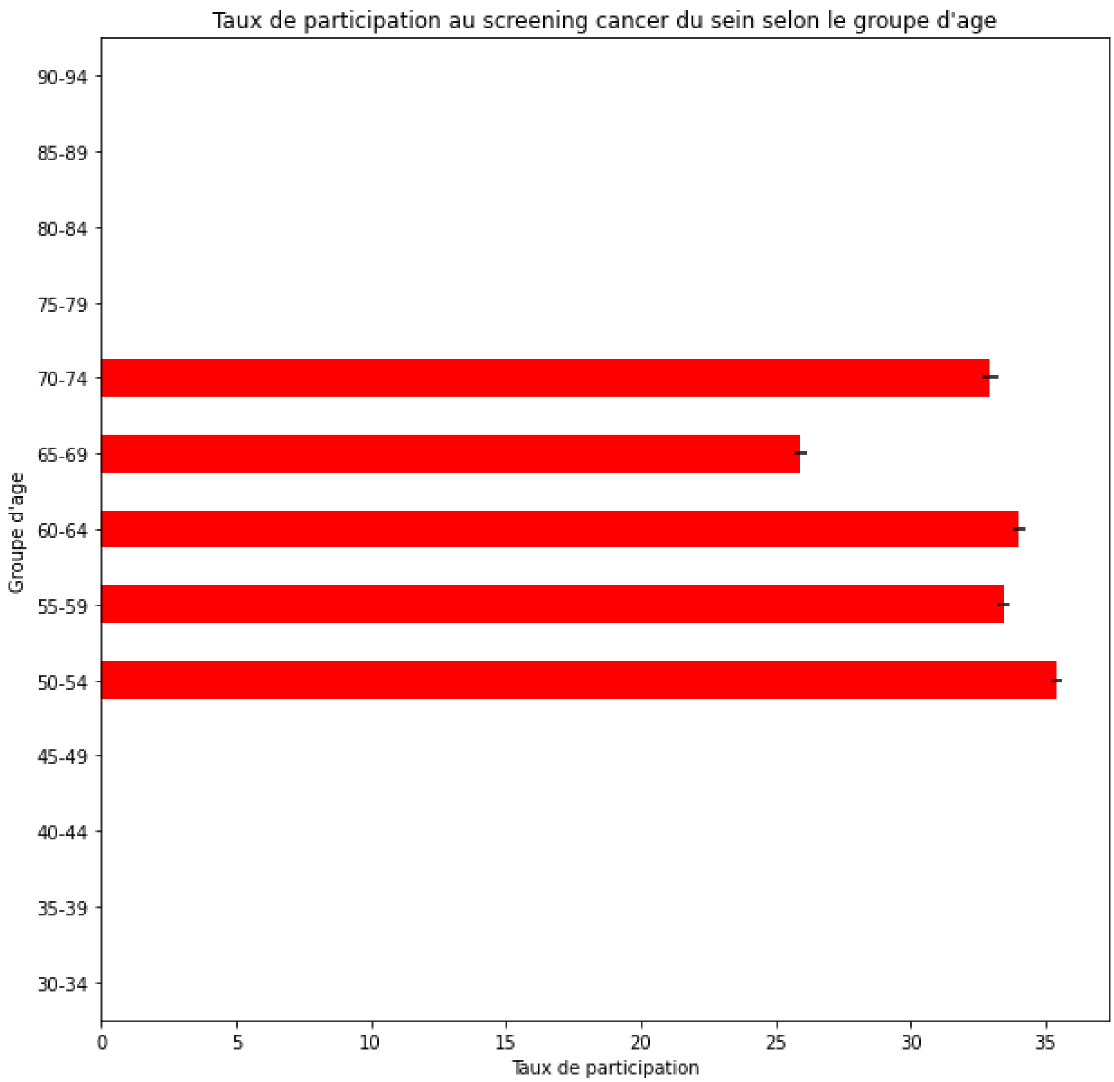
[93]:
```
patients_mammo['mammo_n'] = patients_mammo.index
```

[94]:
```
patients_mammo
```

[94]:
```
mammo
0                                                        82482
1                                                        19001
2                                                        12945
3                                                         8354
mammo_n    Int64Index([0, 1, 2, 3], dtype='int64', name='…
Name: medecin, dtype: object
```

[188]:
```
c = df_final.groupby(['groupeage','mammo'])['mammo'].count()
test = (c / c.groupby(level=[0]).transform("sum")).unstack('mammo').fillna(0)
yerr = ((test[1]*(1-test[1]))/(c.unstack('mammo')[0]+c.
 ↪unstack('mammo')[1]))**(0.5)
test *=100
yerr *= 100
```

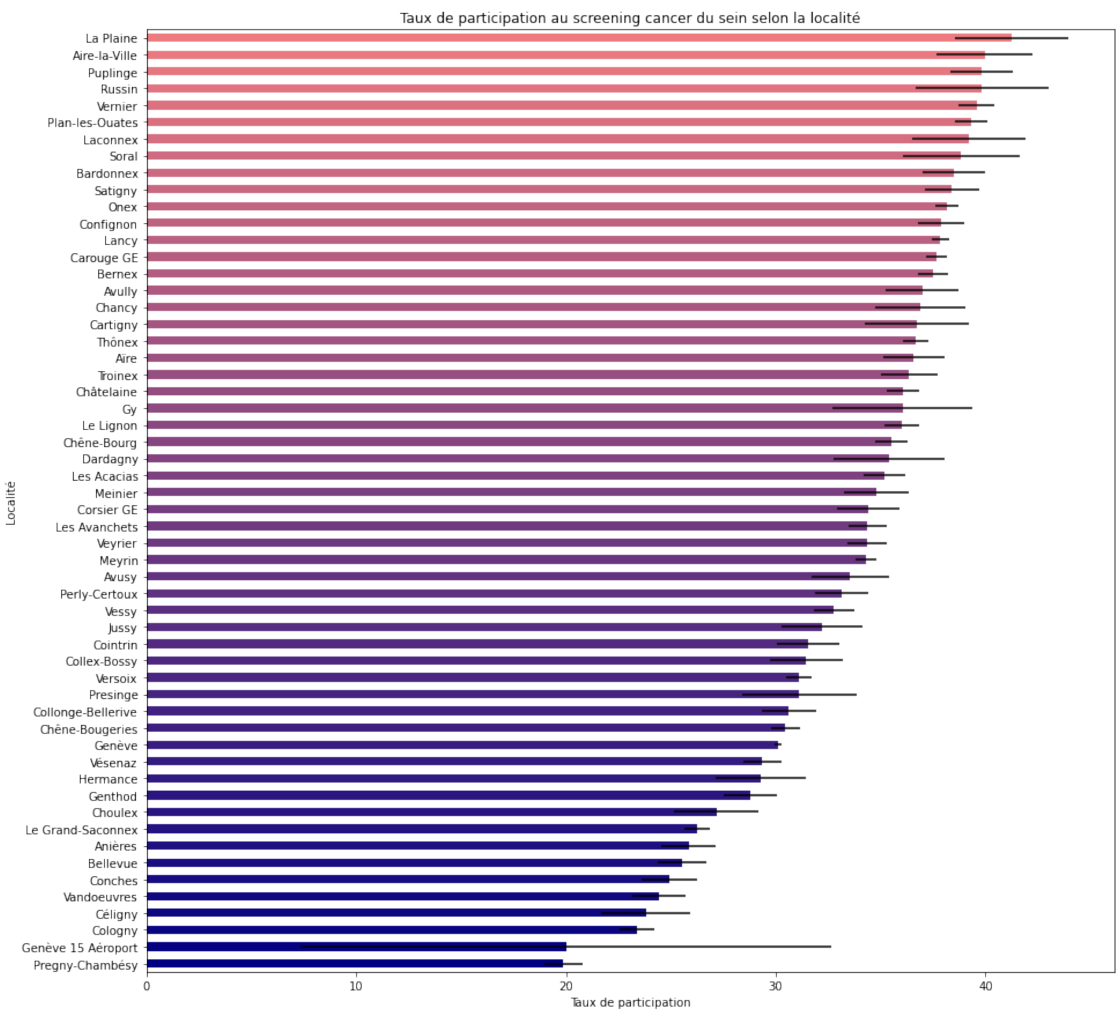
```
f,ax = plt.subplots(figsize = (10,10))
test[1].plot(kind = 'barh',xerr = yerr,color = ['red'],ax = ax)
ax.set_title('Taux de participation au screening cancer du sein selon le groupe␣
↪d\'age')
ax.set_ylabel('Groupe d\'age')
ax.set_xlabel('Taux de participation')
plt.savefig(result_folder/'tx_participation_age.png',bbox_inches='tight',␣
↪transparent=True,dpi = 400)
```



[96]:
```
c = df_final.groupby(['localité','mammo'])['mammo'].count()
test = (c / c.groupby(level=[0]).transform("sum")).unstack('mammo').fillna(0)
yerr = ((test[1]*(1-test[1]))/(c.unstack('mammo')[0]+c.
↪unstack('mammo')[1]))**(0.5)
```

```python
test *=100
yerr *= 100

f,ax = plt.subplots(figsize = (15,15))
my_colors = ['r', 'b']*5 # <-- this concatenates the list to itself 5 times.
my_colors = [(0.5,0.8,0.5), (1, 1, 1)]*5 # <-- make two custom RGBs and repeat/alternate them over all the bar elements.
my_colors = [(x/59.0, x/114.0, 0.5) for x in range(len(test[1]))]
test[1].sort_values().plot(kind = 'barh',xerr = yerr,color = my_colors,ax = ax)
ax.set_title('Taux de participation au screening cancer du sein selon la localité')
ax.set_ylabel('Localité')
ax.set_xlabel('Taux de participation')
plt.savefig(result_folder/'tx_participation_locality.png',bbox_inches='tight',transparent=True,dpi = 400)
```



Get net income and gini index data from the federal office

http://www.estv2.admin.ch/f/dokumentation/zahlen_fakten/karten/dbst/2015/grafiken_2015.php

- Revenus nets

Le revenu net correspond à une valeur statistique déterminée par le revenu imposable auquel sont rajoutées les déductions fiscales pour enfants ou personnes nécessiteuses à charge, pour primes d'assurances et intérêts de capitaux d'épargne et pour double activité des conjoints.

- Revenu équivalent net

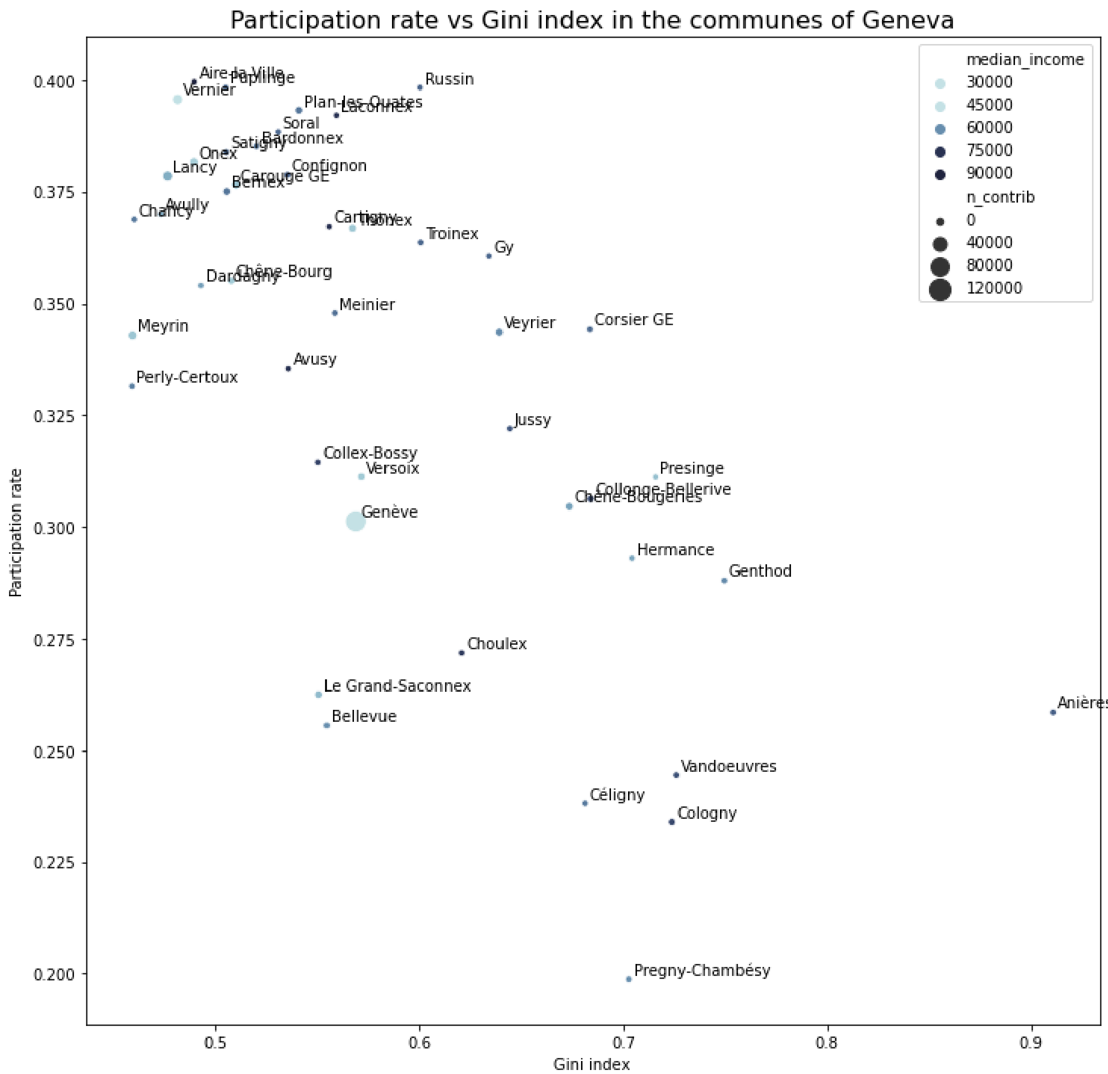
Pour pouvoir comparer le bien-être matériel des ménages de tailles différentes, le revenu net de chaque ménage est divisé par un facteur d'équivalence. Ce rapport entre le revenu net et le facteur d'équivalence constitue le revenu équivalent net. Un facteur d'équivalence de 1 est considéré pour les personnes seules et de 1.5 pour les couples de personnes mariées; à ces nombres est encore ajouté un montant de 0.3 par enfant ou par personne nécessiteuse à charge du contribuable. Par exemple, le revenu équivalent net d'un ménage de personnes mariées avec deux enfants à charge est égal au revenu net du ménage divisé par 2.1 (1.5+0.3+0.3).

[97]:
```
df_income_gini = pd.read_excel('/Users/david/Dropbox/PhD/Data/Databases/OFS/
↪Revenus/revenus_nets.xls',sheet_name = 'prepared_data')
df_income_gini = df_income_gini[df_income_gini.canton == 'Genève']
```

[98]:
```
c = df_final.groupby(['localité','mammo'])['mammo'].count()
test = (c / c.groupby(level=[0]).transform("sum")).unstack('mammo').fillna(0)
test.columns = ['no','Participation rate']
```

[99]:
```
df_income_gini = df_income_gini.set_index('commune').join(test)
df_income_gini['commune'] = df_income_gini.index
```

[100]:
```
cmap = sns.cubehelix_palette(rot=-.2, as_cmap=True)
f,ax = plt.subplots(figsize = (12,12))
sns.scatterplot(x="gini", y="Participation rate",markers = True,
                     hue="median_income", size="n_contrib",
                     palette=cmap, sizes=(20, 200),
                     data=df_income_gini,ax = ax)
for x,y,label in zip(df_income_gini.gini,df_income_gini['Participation␣
↪rate'],df_income_gini.commune):
    ax.annotate(label, xy=(x, y), xytext=(3, 3), textcoords="offset points")
ax.set_xlabel('Gini index')
ax.set_title('Participation rate vs Gini index in the communes of Geneva',␣
↪fontsize = 16)
f.savefig(result_folder/'gini_participation_communes.png',bbox_inches='tight',␣
↪transparent=True,dpi = 400)
```



[101]:
```python
df_desc_screen = pd.DataFrame()
df_desc_screen['Invited 1, n'] = df_final[df_final.numeroinvitation == 1].groupby('year_invit').numerodossier.count()
df_desc_screen['Round 1'] = df_final[(df_final.numeroinvitation == 1)&(df_final.mammo == 1)].groupby('year_invit').numerodossier.count().astype(str)+' ('+(df_final[(df_final.numeroinvitation == 1)&(df_final.mammo == 1)].groupby('year_invit').numerodossier.count()*100/df_desc_screen['Invited 1, n']).round(1).astype(str)+ '%)'
##
df_desc_screen['Invited 2, n'] = df_final[df_final.numeroinvitation == 2].groupby('year_invit').numerodossier.count()
```

```
df_desc_screen['Round 2'] = df_final[(df_final.numeroinvitation == 2)&(df_final.
↪mammo == 1)].groupby('year_invit').numerodossier.count().astype(str)+'␣
↪('+(df_final[(df_final.numeroinvitation == 2)&(df_final.mammo == 1)].
↪groupby('year_invit').numerodossier.count()*100/df_desc_screen['Invited 2,␣
↪n']).round(1).astype(str)+ '%)'
##
df_desc_screen['Invited 3, n'] = df_final[df_final.numeroinvitation == 3].
↪groupby('year_invit').numerodossier.count()
df_desc_screen['Round 3'] = df_final[(df_final.numeroinvitation == 3)&(df_final.
↪mammo == 1)].groupby('year_invit').numerodossier.count().astype(str)+'␣
↪('+(df_final[(df_final.numeroinvitation == 3)&(df_final.mammo == 1)].
↪groupby('year_invit').numerodossier.count()*100/df_desc_screen['Invited 3,␣
↪n']).round(1).astype(str)+ '%)'
```

[102]:
```
render_mpl_table(df_desc_screen.reset_index(), header_columns=0, col_width=3.0)
```

[102]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94830c3dc0>

year_invit	Invited 1, n	Round 1	Invited 2, n	Round 2	Invited 3, n	Round 3
1999	4460	4 (0.1%)	nan	nan	nan	nan
2000	5421	3 (0.1%)	5.0	nan	nan	nan
2001	285	1 (0.4%)	2154.0	10 (0.5%)	31.0	1 (3.2%)
2002	362	7 (1.9%)	3371.0	83 (2.5%)	150.0	6 (4.0%)
2003	338	35 (10.4%)	600.0	125 (20.8%)	1768.0	311 (17.6%)
2004	306	21 (6.9%)	286.0	68 (23.8%)	2446.0	335 (13.7%)
2005	325	39 (12.0%)	250.0	25 (10.0%)	444.0	118 (26.6%)
2006	476	33 (6.9%)	415.0	52 (12.5%)	442.0	95 (21.5%)
2007	555	25 (4.5%)	428.0	24 (5.6%)	213.0	18 (8.5%)
2008	660	33 (5.0%)	402.0	28 (7.0%)	349.0	34 (9.7%)
2009	401	30 (7.5%)	290.0	44 (15.2%)	213.0	30 (14.1%)
2010	678	46 (6.8%)	341.0	31 (9.1%)	258.0	36 (14.0%)
2011	507	39 (7.7%)	178.0	33 (18.5%)	180.0	26 (14.4%)
2012	689	133 (19.3%)	512.0	119 (23.2%)	319.0	94 (29.5%)
2013	2577	1094 (42.5%)	2015.0	912 (45.3%)	1467.0	747 (50.9%)
2014	5236	1754 (33.5%)	3870.0	1397 (36.1%)	3529.0	1295 (36.7%)
2015	4885	1793 (36.7%)	4767.0	1722 (36.1%)	4051.0	1547 (38.2%)
2016	4814	1837 (38.2%)	4168.0	1688 (40.5%)	3297.0	1381 (41.9%)
2017	4385	1695 (38.7%)	4271.0	1718 (40.2%)	4085.0	1540 (37.7%)
2018	4673	1848 (39.5%)	4431.0	1848 (41.7%)	3907.0	1712 (43.8%)

[103]:
```
df_desc_screen = pd.DataFrame()
df_desc_screen['Invited 1, n'] = df_final[df_final.numeroinvitation_seq == 1].
↪groupby('year_invit').numerodossier.count()
```

```
df_desc_screen['Round 1'] = df_final[(df_final.numeroinvitation_seq ==␣
↪1)&(df_final.mammo == 1)].groupby('year_invit').numerodossier.count().
↪astype(str)+' ('+(df_final[(df_final.numeroinvitation_seq == 1)&(df_final.
↪mammo == 1)].groupby('year_invit').numerodossier.count()*100/
↪df_desc_screen['Invited 1, n']).round(1).astype(str)+ '%)'
##
df_desc_screen['Invited 2, n'] = df_final[df_final.numeroinvitation_seq == 2].
↪groupby('year_invit').numerodossier.count()
df_desc_screen['Round 2'] = df_final[(df_final.numeroinvitation_seq ==␣
↪2)&(df_final.mammo == 1)].groupby('year_invit').numerodossier.count().
↪astype(str)+' ('+(df_final[(df_final.numeroinvitation_seq == 2)&(df_final.
↪mammo == 1)].groupby('year_invit').numerodossier.count()*100/
↪df_desc_screen['Invited 2, n']).round(1).astype(str)+ '%)'
##
df_desc_screen['Invited 3, n'] = df_final[df_final.numeroinvitation_seq == 3].
↪groupby('year_invit').numerodossier.count()
df_desc_screen['Round 3'] = df_final[(df_final.numeroinvitation_seq ==␣
↪3)&(df_final.mammo == 1)].groupby('year_invit').numerodossier.count().
↪astype(str)+' ('+(df_final[(df_final.numeroinvitation_seq == 3)&(df_final.
↪mammo == 1)].groupby('year_invit').numerodossier.count()*100/
↪df_desc_screen['Invited 3, n']).round(1).astype(str)+ '%)'
```

[104]:
```
render_mpl_table(df_desc_screen.reset_index(), header_columns=0, col_width=3.0)
```

[104]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9482d1c2b0>

year_invit	Invited 1, n	Round 1	Invited 2, n	Round 2	Invited 3, n	Round 3
1999	4461	4 (0.1%)	nan	nan	nan	nan
2000	5426	3 (0.1%)	nan	nan	nan	nan
2001	2470	12 (0.5%)	nan	nan	nan	nan
2002	3883	96 (2.5%)	nan	nan	nan	nan
2003	2735	474 (17.3%)	nan	nan	nan	nan
2004	3153	442 (14.0%)	nan	nan	nan	nan
2005	2290	378 (16.5%)	nan	nan	nan	nan
2006	3681	524 (14.2%)	nan	nan	nan	nan
2007	2559	178 (7.0%)	nan	nan	nan	nan
2008	3026	226 (7.5%)	nan	nan	nan	nan
2009	1831	232 (12.7%)	nan	nan	nan	nan
2010	2371	278 (11.7%)	nan	nan	nan	nan
2011	1624	265 (16.3%)	nan	nan	nan	nan
2012	2755	741 (26.9%)	nan	nan	nan	nan
2013	14112	6654 (47.2%)	nan	nan	nan	nan
2014	29250	10442 (35.7%)	nan	nan	nan	nan
2015	23011	7365 (32.0%)	10765.0	4983 (46.3%)	12.0	nan
2016	4935	1890 (38.3%)	25002.0	9391 (37.6%)	194.0	40 (20.6%)
2017	4471	1743 (39.0%)	19930.0	7363 (36.9%)	7922.0	3408 (43.0%)
2018	4738	1886 (39.8%)	5118.0	2260 (44.2%)	21848.0	8622 (39.5%)

4.1 Deprivation index

[105]:
```
query = """select nbid,"LOCALITY"␣
↪locality,ciqmd,ptot,pm,pf,p0004,p0509,p1014,p1519,p2024,p2529,p3034,p3539,p4044,p4549,p5054
↪rad3sec,rad3tert,rprprot,rprcath,rprochr,rprjew,rprmusl,rproth,rprnorel,rpnch,
rpnoce
rpncam,
rpncas,
rpnceu,rpneeu,rpneceu,rpnfe,rpnme,rpnnaf,rpnnam,rpnneu,rpnnweu,rpnsam,rpnsas,rpnseas,rpnseeu,r
,rphhpriv,rhhcoll,rhhp1p,rhhp2p,rhhp3p,rhhp4p,rhhp5p,rhhp6mp,rpfnone,rpfobl,rpfgen,rpfprof,rpf
↪,rpfbac,rpfmas,rpfphd,rad,radf,radune,rado,radunef,radslib,dmdrent,b.geom␣
↪geometry from data_raw.microgis_data_gva a, data_raw.microgis_geo_gva b␣
↪where a.nbid = b."NBID" and b.geom is not null ;"""
microgis_data = gpd.GeoDataFrame.from_postgis(query,con = geo_engine,geom_col =␣
↪'geometry')
microgis_data = microgis_data.dropna()
microgis_data.crs = 'epsg:2056'
```

[106]:
```
microgis_data.to_csv('./microgis_data_depriv.csv',index = False)
```

[107]:
```
microgis_data['tertiary_education'] =␣
↪microgis_data[['rpfbac','rpfmas','rpfphd']].sum(axis = 1)
microgis_data['rpforeign'] = 100 - microgis_data['rpnch']
```

[108]:
```
microgis_data = microgis_data[microgis_data.nbid.isin(df.nbid.unique())]
```

[109]:
```
microgis_data = microgis_data[microgis_data.ptot > 5]
```

[110]:
```
microgis_data[['rpforeign','ciqmd','rado','radune','tertiary_education','dmdrent','rad3prim','
```

[110]:
```
      rpforeign  ciqmd   rado  radune  tertiary_education  dmdrent  rad3prim  \
0         27.03   54.6  94.35    5.65               20.36   1134.0      0.00
1         42.98   33.4  91.79    8.21               26.09   1248.0      0.00
2         44.34   39.4  88.57   11.43               43.41   1156.0      0.00
5         42.13   34.1  91.81    8.19               20.32   1142.0      0.00
6         64.66   52.7  91.20    8.80               51.55   1302.0      0.00
…           …    …     …      …                   …        …        …
2824      19.44   75.7  95.03    4.97               21.07   2209.0      3.01
2825      16.62   32.5  97.46    2.54               40.82   2847.0      3.13
2827      14.32   74.4  92.57    7.43               27.83   1345.0      0.00
2828      32.86  100.3  98.29    1.71               13.54   1702.0      0.82
2829      39.58   34.8  80.96   19.04               25.80    891.0      0.00

      rad3tert
0        71.49
```

```
1          90.25
2          88.99
5          79.80
6          97.95
…          …
2824       94.20
2825       86.09
2827       96.68
2828       87.75
2829       90.36

[1927 rows x 8 columns]
```

[111]:

```
from sklearn.preprocessing import StandardScaler
x =␣
↪microgis_data[['rpforeign','ciqmd','rad','tertiary_education','dmdrent','rad3tert']].
↪values
x = StandardScaler().fit_transform(x)
```

[112]:

```
from sklearn.decomposition import PCA
pca_depriv = PCA(n_components=3)
principalComponents_depriv = pca_depriv.fit_transform(x)
```

[113]:

```
principal_depriv_Df = pd.DataFrame(data = principalComponents_depriv
             , columns = ['principal component 1', 'principal component␣
↪2','principal component 3'])
```

[114]:

```
print('Explained variation per principal component: {}'.format(pca_depriv.
↪explained_variance_ratio_))
```

```
Explained variation per principal component: [0.29991751 0.24073175 0.16119664]
```

[115]:

```
pd.DataFrame(pca_depriv.
↪components_,columns=microgis_data[['rpforeign','ciqmd','rado','tertiary_education','dmdrent
↪columns,index = ['PC-1','PC-2','PC-3'])
```

[115]:

	rpforeign	ciqmd	rado	tertiary_education	dmdrent	rad3tert
PC-1	-0.148354	0.555674	-0.207914	0.502269	0.560607	0.243795
PC-2	-0.613532	0.080575	-0.448087	-0.429702	0.172057	-0.449507
PC-3	0.191399	0.314018	0.595961	-0.037958	0.279590	-0.655727

[116]:

```
microgis_data['deprivation_pca'] = principalComponents_depriv.T[0]
```

[117]:

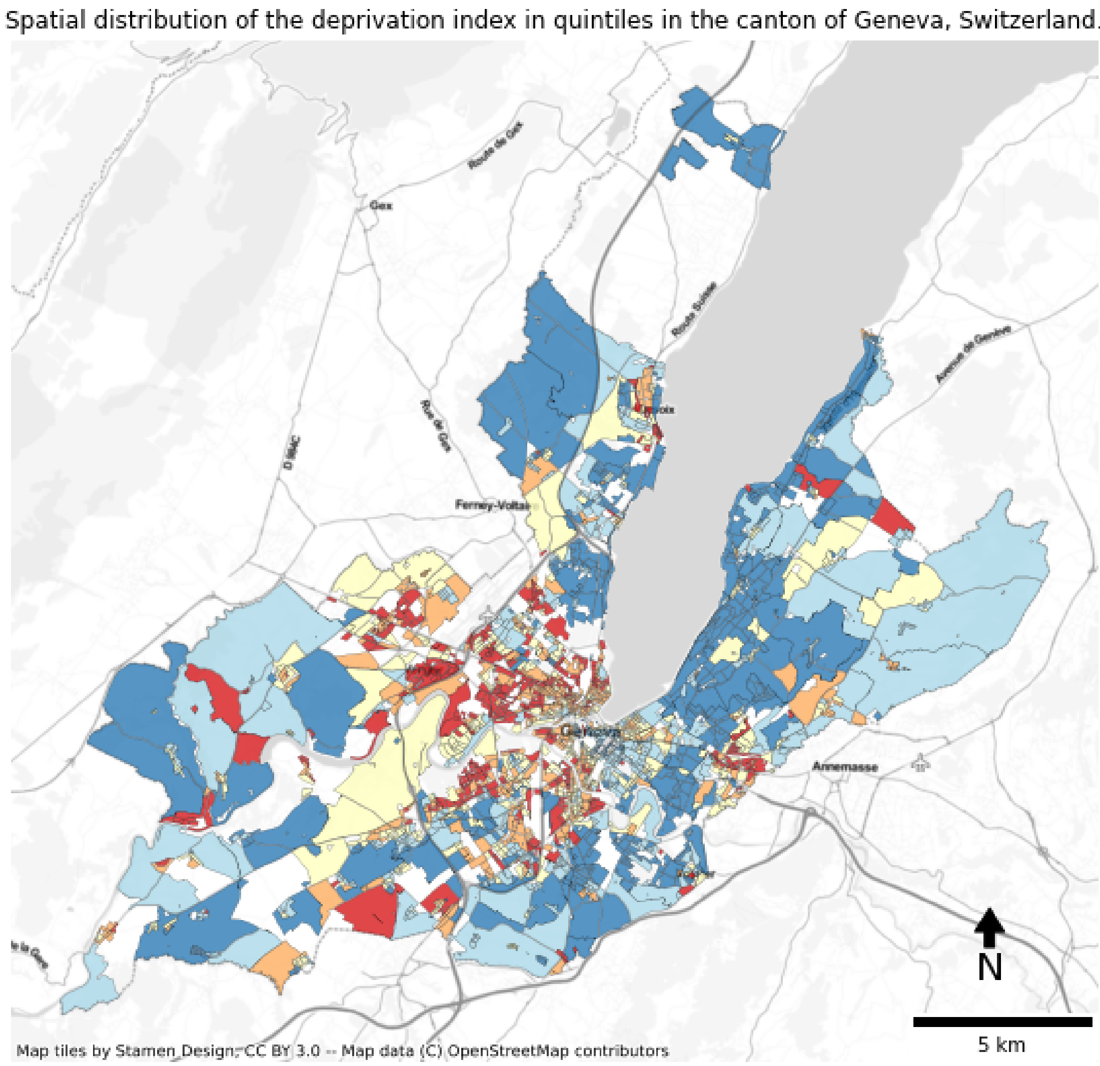
```
microgis_data['deprivation_pca_q5'] = pd.
↪qcut(microgis_data['deprivation_pca'],5, labels = False)
```

[118]:
```
dict_cl = {np.nan:'#bababa',0:'#d7191c',
1:'#fdae61',
2:'#ffffbf',
3:'#abd9e9',
4:'#2c7bb6'}
```

[154]:
```
microgis_data['depriv_color'] = microgis_data['deprivation_pca_q5'].map(dict_cl)
```

[156]:
```
import contextily as ctx
from matplotlib_scalebar.scalebar import ScaleBar

ax = microgis_data.plot(alpha = 0.8,figsize = (10,10),color = microgis_data.
 ↪depriv_color , linewidth = 0.2, edgecolor = 'k')
ax.set_title('Spatial distribution of the deprivation index in quintiles in the␣
 ↪canton of Geneva, Switzerland.')
ctx.add_basemap(ax, url=ctx.providers.Stamen.TonerLite,crs = 'EPSG:2056')
# add scale bar
scalebar = ScaleBar(1, units="m", location="lower right")
ax.add_artist(scalebar)
ax.set_axis_off()
x, y, arrow_length = 0.9, 0.15, 0.06
ax.annotate('N', xy=(x, y), xytext=(x, y-arrow_length),
            arrowprops=dict(facecolor='black', width=5, headwidth=15),
            ha='center', va='center', fontsize=20,
            xycoords=ax.transAxes)
plt.savefig(result_folder/'Deprivation_cantongva.png',dpi = 800)
```



[158]:
```
microgis_data[microgis_data.locality == 'Genève'].depriv_color.nunique()
```

[158]: 5

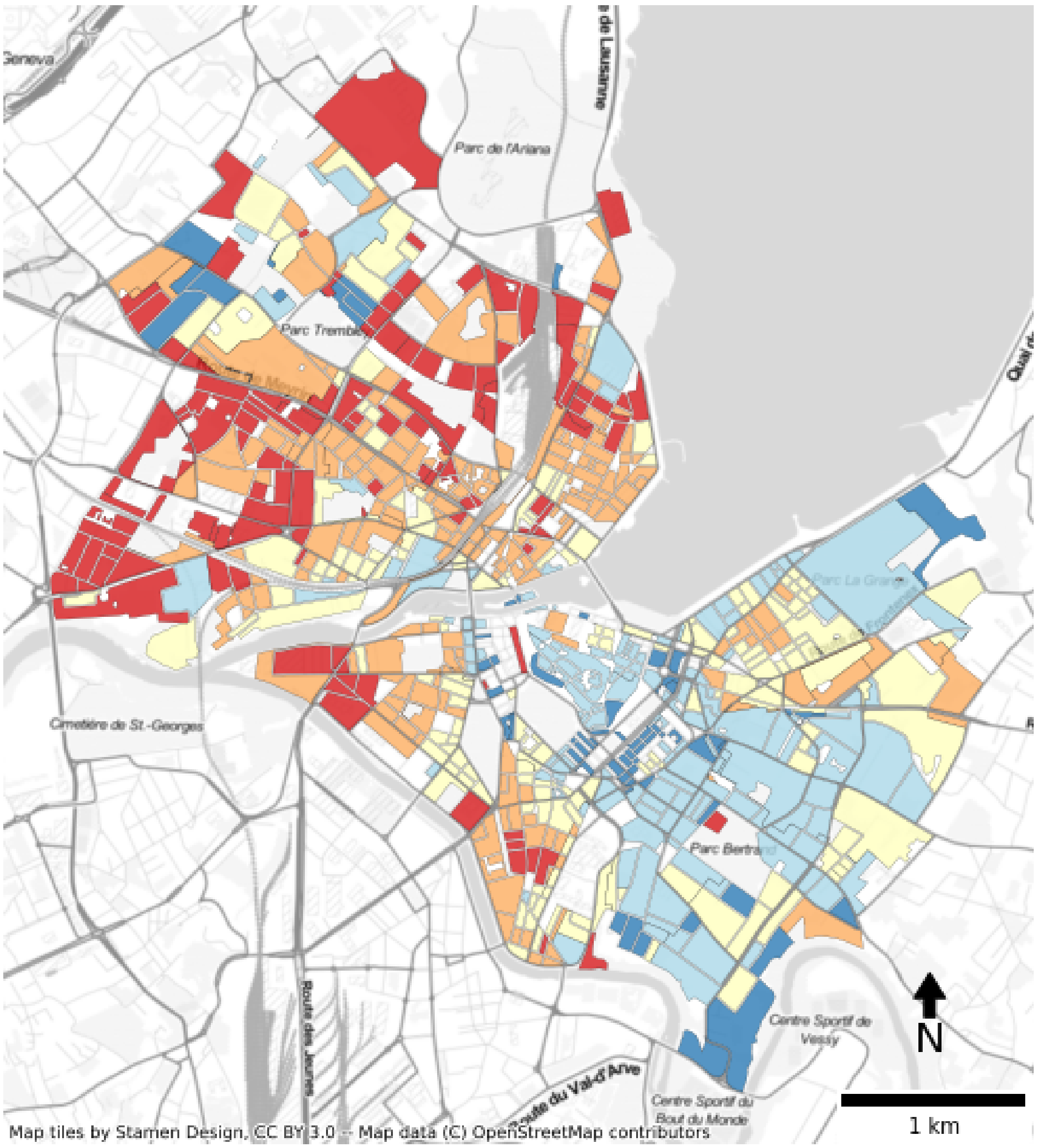
[159]:
```
microgis_gva = microgis_data[microgis_data.locality == 'Genève']
```

[169]:
```
import contextily as ctx
from matplotlib_scalebar.scalebar import ScaleBar

ax = microgis_gva.plot(alpha = 0.8,figsize = (10,10),color = microgis_gva.depriv_color, linewidth = 0.2, edgecolor = 'k')
ax.set_title('Spatial distribution of the deprivation index in quintiles in Geneva, Switzerland.')
ctx.add_basemap(ax, url=ctx.providers.Stamen.TonerLite,crs = 'EPSG:2056')
```

```
# add scale bar
scalebar = ScaleBar(1, units="m", location="lower right")
ax.add_artist(scalebar)
ax.set_axis_off()
x, y, arrow_length = 0.9, 0.15, 0.06
ax.annotate('N', xy=(x, y), xytext=(x, y-arrow_length),
            arrowprops=dict(facecolor='black', width=5, headwidth=15),
            ha='center', va='center', fontsize=20,
            xycoords=ax.transAxes)
plt.savefig(result_folder/'Deprivation_gva.png',dpi = 800)
```

Spatial distribution of the deprivation index in quintiles in Geneva, Switzerland.



[120]:
```
import math
c = gdf.groupby(['nbid','mammo'])['mammo'].count()
test = (c / c.groupby(level=[0]).transform("sum")).unstack('nbid').fillna(0).
↪T*100
test.columns = ['no','mammo_rate']
test['n'] = gdf.nbid.value_counts()
test['SEp'] = (test.mammo_rate*(100-test.mammo_rate))/test.n
test['SEp'] = test['SEp'].apply(lambda x: math.sqrt(x))
```

[121]:
```
test.loc[test.SEp> 15,'mammo_rate'] = np.nan
```

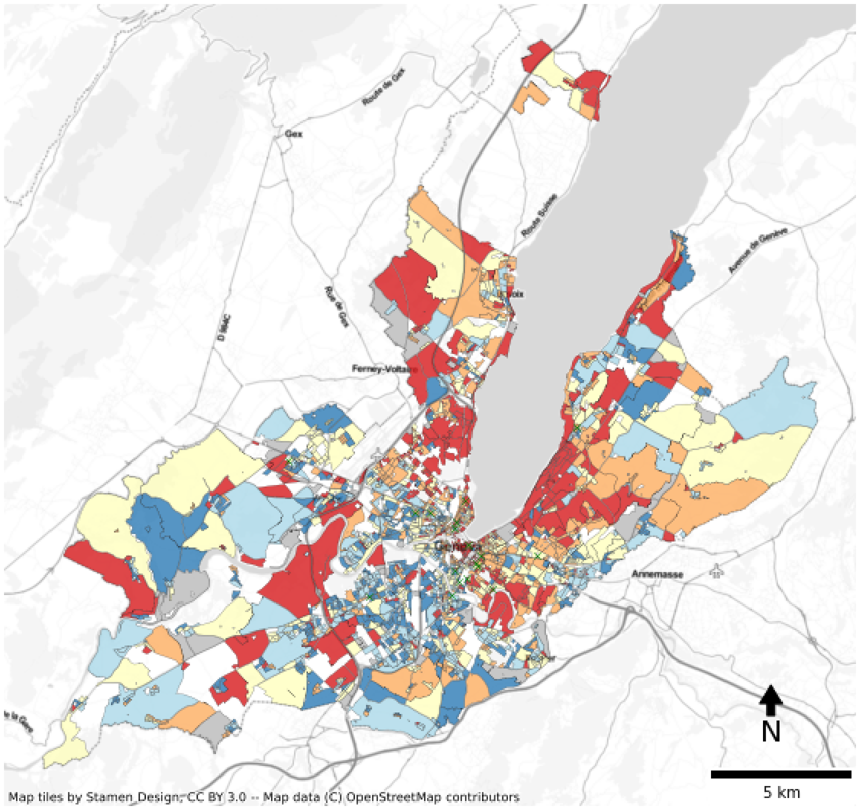
[122]:
```
microgis_data = pd.merge(microgis_data,test[['mammo_rate','SEp']],left_on =␣
↪'nbid',right_index = True, how = 'left')
microgis_data['mammo_rate_q5'] = pd.qcut(microgis_data['mammo_rate'],5, labels␣
↪= False)
```

[339]:
```
gdf_centre = gdf_centre.to_crs(epsg = 2056)
```

[343]:
```
ax = microgis_data.plot(alpha = 0.8,figsize = (10,10),color =␣
↪microgis_data['mammo_rate_q5'].map(dict_cl), legend = True, linewidth = 0.2,␣
↪edgecolor = 'k')
gdf_centre.plot(ax = ax,markersize = 20, color = 'green',marker = 'x',linewidth␣
↪= 0.5)
ax.set_title('Spatial distribution of the mammography screening participation␣
↪rate in quintiles in the canton of Geneva, Switzerland.')
ctx.add_basemap(ax, url=ctx.providers.Stamen.TonerLite,crs = 'EPSG:2056')
ax.set_axis_off()
# add scale bar
scalebar = ScaleBar(1, units="m", location="lower right")
ax.add_artist(scalebar)
ax.set_axis_off()

x, y, arrow_length = 0.9, 0.15, 0.06
ax.annotate('N', xy=(x, y), xytext=(x, y-arrow_length),
            arrowprops=dict(facecolor='black', width=5, headwidth=15),
            ha='center', va='center', fontsize=20,
            xycoords=ax.transAxes)
plt.savefig(result_folder/'Mammo_rate_cantongva.png',dpi = 800)
```

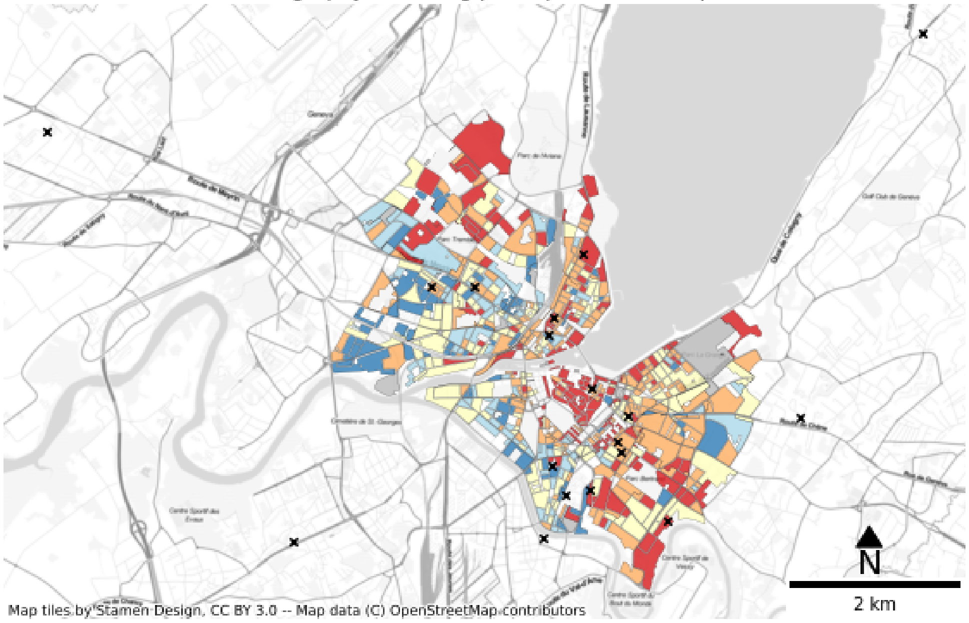
Spatial distribution of the mammography screening participation rate in quintiles in the canton of Geneva, Switzerland.



[346]:

```
ax = microgis_gva.plot(alpha = 0.8,figsize = (10,10),color =␣
↪microgis_gva['mammo_rate_q5'].map(dict_cl), legend = True, linewidth = 0.2,␣
↪edgecolor = 'k')
gdf_centre.plot(ax = ax,markersize = 20, color = 'black',marker =␣
↪'x',edgecolor='black')
ax.set_title('Spatial distribution of the mammography screening participation␣
↪rate in quintiles in Geneva, Switzerland.')
ctx.add_basemap(ax, url=ctx.providers.Stamen.TonerLite,crs = 'EPSG:2056')
ax.set_axis_off()
# add scale bar
scalebar = ScaleBar(1, units="m", location="lower right")
ax.add_artist(scalebar)
ax.set_axis_off()
x, y, arrow_length = 0.9, 0.15, 0.06
ax.annotate('N', xy=(x, y), xytext=(x, y-arrow_length),
            arrowprops=dict(facecolor='black', width=5, headwidth=15),
            ha='center', va='center', fontsize=20,
            xycoords=ax.transAxes)
plt.savefig(result_folder/'Mammo_rate_gva.png',dpi = 800)
```

Spatial distribution of the mammography screening participation rate in quintiles in Geneva, Switzerland.



5 Determinants of participation, first participation, re-participation

5.1 1. Determinants of participation (any participation)

[87]:
```
df_participation =␣
↪df[['groupeage','numeroinvitation','numerodepistage','etatcivil','rappel',␣
↪'year_invit', 'month_invit', 'day_invit',
       'localité','center_density','center_nearest','ptot',
       'rad3prim', 'rad3sec', 'rad3tert', 'rprprot', 'rprcath', 'rprochr',
       'rprjew', 'rprmusl', 'rproth', 'rprnorel', 'rpnch', 'rphhpriv',
       'rpfnone', 'rpfobl', 'rpfgen', 'rpfprof', 'rpfmat', 'rpfprsf',
       'rpfprss', 'rpfbac', 'rpfmas', 'rpfphd', 'rad', 'radf', 'radunef',
       'radslib', 'dmdrent','mammo']]
# df_participation =␣
↪df[['numerodossier','groupeage','numeroinvitation','numerodepistage','etatcivil','rappel',␣
↪'year_invit', 'month_invit',␣
↪'day_invit','localité','center_density','center_nearest','mammo']]
```

[88]:
```
df_participation['age_cat'] = pd.factorize(df_participation['groupeage'],␣
↪sort=True)[0] + 1
```

[89]:
```
df_participation.loc[df_participation.localité.str.
↪contains('Meyrin'),'localité'] = 'Meyrin'
```

```
df_participation.loc[df_participation.localité.str.
 ↪contains('Grand-Lancy'),'localité'] = 'Grand-Lancy'
df_participation.loc[df_participation.localité.str.
 ↪contains('Petit-Lancy'),'localité'] = 'Petit-Lancy'
```

[90]:
```
df_participation = pd.concat([df_participation,pd.get_dummies(df_participation.
 ↪etatcivil),pd.get_dummies(df_participation.localité)],axis = 1).
 ↪drop(['groupeage','etatcivil','localité'],axis = 1)
# df_participation = pd.concat([df_participation,pd.
 ↪get_dummies(df_participation.etatcivil),pd.get_dummies(df_participation.
 ↪weekday_invit)],axis = 1).
 ↪drop(['groupeage','etatcivil','weekday_invit'],axis = 1)
```

[91]:
```
import statsmodels.api as sm
from scipy import stats
from statsmodels.graphics.api import abline_plot
from sklearn.linear_model import LogisticRegression
import statsmodels.api as sm
```

[92]:
```
df_participation = df_participation.dropna()
```

[93]:
```
df_participation = df_participation.drop(df_participation.
 ↪std()[(df_participation.std() < 0.01)].index, axis=1)
```

[94]:
```
df_participation.shape
```

[94]:
```
(251858, 99)
```

[95]:
```
X = df_participation.drop(['mammo'],axis = 1)
y = df_participation.mammo
```

[96]:
```
logit_model=sm.Logit(y,X)
result=logit_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.216528
         Iterations 11
```

[ ]:
```
result.summary2()
```

[350]:
```
data_final_vars=df_participation.columns.values.tolist()
# y=['mammo']
# X=[i for i in data_final_vars if i not in y]
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
rfe = RFE(logreg, 20)
```

```python
rfe = rfe.fit(X, y.values)
print(rfe.support_)
print(rfe.ranking_)
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)

[ True  True  True False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
  True  True  True  True  True  True  True False False False False False
 False False  True False False False False False False False  True False
  True  True False False False False False  True False False False False
  True  True False False False False False False False False False False
  True  True False  True False False False False False False False False
 False False]
[ 1  1  1 76 29 64 38 75 79 70 74 67 73 58 59 57 43 61 56 63 25 51 55 53
 49 54 52 50 46 48 33 65 60 72 66 78  1  1  1  1  1  1  1 10  8 13 14 45
  5  2  1 68 21 36 77 39 69 15  1 27  1  1 37 42 19 40 20  1 12 47 28 18
  1  1 26 11 30 24 34 41 17 23  4 62  1  1  7  1 32 71 31  9  3 44 16 35
 22  6]
```

[370]:
```
cols = ['numeroinvitation','numerodepistage','rappel','age_cat','Clandestine',␣
↪'Célibataire', 'Divorcé', 'Marié','Partenariat enregistré',␣
↪'Veuve','Cointrin','Collonge-Bellerive','Cologny','Genthod','La␣
↪Plaine','Laconnex','Pregny-Chambésy','Presinge','Russin']
```

[371]:
```
X = X[cols]
```

[373]:
```
logit_model=sm.Logit(y,X)
result=logit_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.223054
         Iterations 8
```

[377]:
```
result.summary2()
```

[377]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                           Results: Logit
=======================================================================
Model:                Logit             Pseudo R-squared:    0.641
Dependent Variable:   mammo             AIC:                 112393.8713
Date:                 2020-02-18 19:24  BIC:                 112592.1671
No. Observations:     251858            Log-Likelihood:      -56178.
Df Model:             18                LL-Null:             -1.5660e+05
Df Residuals:         251839            LLR p-value:         0.0000
Converged:            1.0000            Scale:               1.0000
No. Iterations:       8.0000
-----------------------------------------------------------------------
                        Coef.  Std.Err.     z     P>|z|   [0.025  0.975]
-----------------------------------------------------------------------
numeroinvitation       -0.1829   0.0046  -39.5187 0.0000 -0.1919 -0.1738
numerodepistage         1.4615   0.0073  199.6055 0.0000  1.4472  1.4759
rappel                 -0.7138   0.0161  -44.2814 0.0000 -0.7454 -0.6822
age_cat                -0.5888   0.0039 -150.0538 0.0000 -0.5965 -0.5811
Clandestine             4.4024   0.1643   26.7884 0.0000  4.0803  4.7245
Célibataire             2.7931   0.0288   97.0358 0.0000  2.7367  2.8496
Divorcé                 2.9073   0.0243  119.5472 0.0000  2.8597  2.9550
Marié                   2.9299   0.0186  157.3112 0.0000  2.8934  2.9664
Partenariat enregistré  2.7978   0.2037   13.7367 0.0000  2.3986  3.1970
Veuve                   3.2008   0.0432   74.0781 0.0000  3.1161  3.2855
Cointrin               -0.2162   0.1114   -1.9413 0.0522 -0.4344  0.0021
Collonge-Bellerive     -0.2603   0.0983   -2.6486 0.0081 -0.4529 -0.0677
Cologny                -0.2135   0.0767   -2.7846 0.0054 -0.3638 -0.0632
Genthod                -0.3789   0.1055   -3.5921 0.0003 -0.5857 -0.1722
La Plaine               0.1144   0.1729    0.6616 0.5082 -0.2245  0.4532
Laconnex               -0.1857   0.1834   -1.0127 0.3112 -0.5452  0.1737
```

```
Pregny-Chambésy         -0.4927    0.0903    -5.4581 0.0000 -0.6696 -0.3158
Presinge                -0.2560    0.2047    -1.2507 0.2110 -0.6573  0.1452
Russin                   0.1968    0.2139     0.9201 0.3575 -0.2225  0.6161
========================================================================

"""
```

[378]:
```
cols = ['numeroinvitation','numerodepistage','rappel','age_cat','Clandestine',␣
 ↪'Célibataire', 'Divorcé', 'Marié','Partenariat enregistré',␣
 ↪'Veuve','Collonge-Bellerive','Cologny','Genthod','Pregny-Chambésy']
```

[379]:
```
X = X[cols]
```

[380]:
```
logit_model=sm.Logit(y,X)
result=logit_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.223069
         Iterations 8
```

[381]:
```
result.summary2()
```

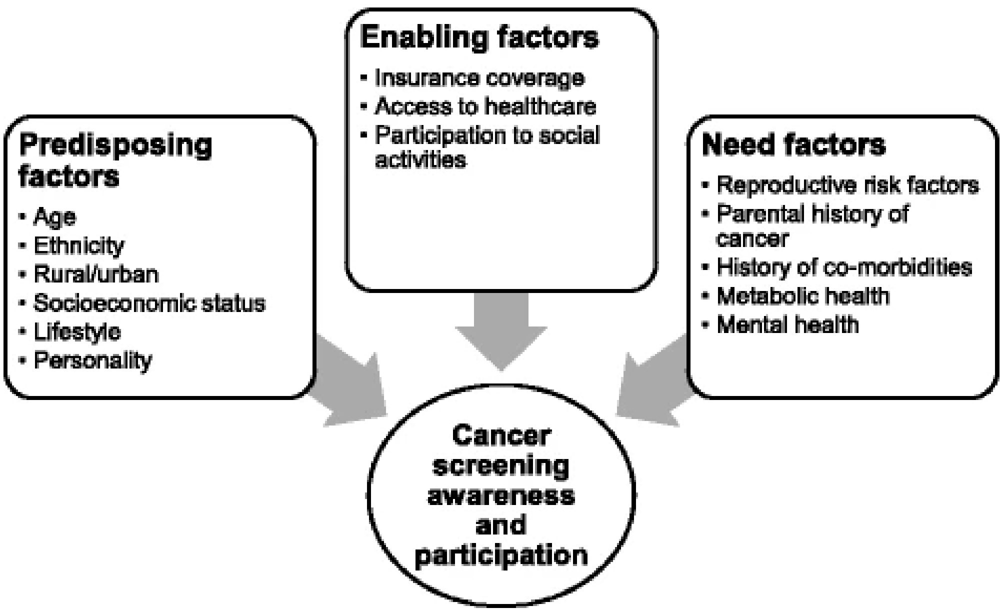
[381]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                            Results: Logit
========================================================================
Model:                  Logit              Pseudo R-squared:   0.641
Dependent Variable:     mammo              AIC:                112391.5520
Date:                   2020-02-18 19:27   BIC:                112537.6647
No. Observations:       251858             Log-Likelihood:     -56182.
Df Model:               13                 LL-Null:            -1.5660e+05
Df Residuals:           251844             LLR p-value:        0.0000
Converged:              1.0000             Scale:              1.0000
No. Iterations:         8.0000
------------------------------------------------------------------------
                          Coef.  Std.Err.     z     P>|z|   [0.025  0.975]
------------------------------------------------------------------------
numeroinvitation        -0.1829    0.0046   -39.5242 0.0000 -0.1919 -0.1738
numerodepistage          1.4615    0.0073   199.6125 0.0000  1.4472  1.4759
rappel                  -0.7139    0.0161   -44.2895 0.0000 -0.7455 -0.6823
age_cat                 -0.5890    0.0039 -150.1682 0.0000 -0.5967 -0.5813
Clandestine              4.4030    0.1643    26.7918 0.0000  4.0809  4.7251
Célibataire              2.7932    0.0288    97.0465 0.0000  2.7368  2.8496
Divorcé                  2.9076    0.0243   119.5700 0.0000  2.8600  2.9553
Marié                    2.9292    0.0186   157.3240 0.0000  2.8927  2.9657
Partenariat enregistré   2.8019    0.2035    13.7677 0.0000  2.4030  3.2008
Veuve                    3.2013    0.0432    74.0887 0.0000  3.1166  3.2859
```

```
Collonge-Bellerive      -0.2589    0.0983    -2.6352 0.0084 -0.4515 -0.0663
Cologny                 -0.2122    0.0767    -2.7676 0.0056 -0.3624 -0.0619
Genthod                 -0.3775    0.1055    -3.5789 0.0003 -0.5843 -0.1708
Pregny-Chambésy         -0.4913    0.0903    -5.4434 0.0000 -0.6682 -0.3144
==========================================================================

"""
```

5.2 2. Determinants of first participation

Fig. 1



Source : https://bmccancer.biomedcentral.com/articles/10.1186/s12885-018-4125-z - Known determinants of mammography screening participation: - Age (groupeage) - Civil status (etatcivil) - Proximity and density of BC screening facility (center_density & center_nearest) - Area-level deprivation

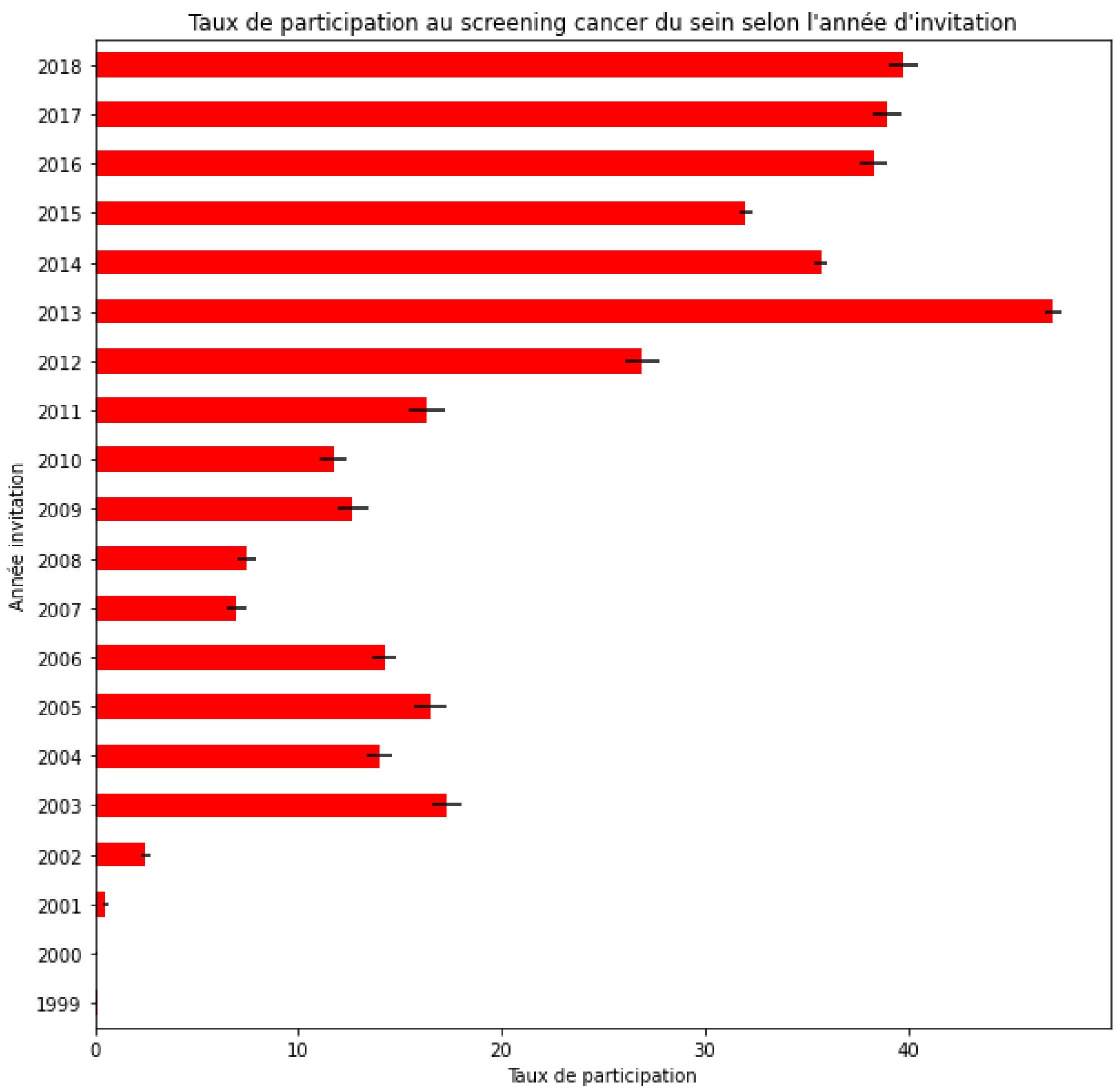
- Incomplete variables (many variables have only a value when the woman participated in the mammography):
  - mammoanterieure
  - Known risk factors (atf)

[269]:
```
df_1st = df_final[df_final.numeroinvitation_seq == 1]
```

[270]:
```
df_1st = pd.
↪merge(microgis_data[['nbid','deprivation_pca_q5','mammo_rate_q5']],df_1st,␣
↪how = 'right',on = 'nbid')
```

[271]:
```
c = df_1st.groupby(['year_invit','mammo'])['mammo'].count()
test = (c / c.groupby(level=[0]).transform("sum")).unstack('mammo').fillna(0)
yerr = ((test[1]*(1-test[1]))/(c.unstack('mammo')[0]+c.
 ↪unstack('mammo')[1]))**(0.5)
test *=100
yerr *= 100

f,ax = plt.subplots(figsize = (10,10))
test[1].plot(kind = 'barh',xerr = yerr,color = ['red'],ax = ax)
ax.set_title('Taux de participation au screening cancer du sein selon l\'année␣
 ↪d\'invitation')
ax.set_ylabel('Année invitation')
ax.set_xlabel('Taux de participation')
plt.savefig(result_folder/'tx_participation_yearinvit.png',bbox_inches='tight',␣
 ↪transparent=True,dpi = 400)
```



We see that the crude participation rate has increased a lot over years…it could suggest that communication campaigns about BC screening have worked and the screening got traction overtime.

It also means that the period 1999-2012 is hardly comparable with 2013-2018 and I would suggest to separate them. It would also benefit analyses by reducing computation time.

[272]:
```
df_1st_1318 = df_1st[df_1st.year_invit > 2012]
```

[274]:
```
df_1st_1318['age_cat'] = pd.factorize(df_1st_1318['groupeage'], sort=True)[0] +␣
 ↪1
df_1st_1318['etatcivil_cat'] = pd.factorize(df_1st_1318['etatcivil'],␣
 ↪sort=True)[0] + 1
```

[275]:
```
df_1st_1318 = pd.concat([df_1st_1318,pd.get_dummies(df_1st_1318.etatcivil),pd.
 ↪get_dummies(df_1st_1318.localité)],axis = 1).
 ↪drop(['groupeage','etatcivil','localité'],axis = 1)
```

[276]:
```
cols = ['age_cat','Clandestine', 'Célibataire', 'Divorcé', 'Marié','Partenariat␣
 ↪enregistré',␣
 ↪'Veuve','center_nearest','center_density','deprivation_pca_q5','mammo_rate_q5']
```

[277]:
```
cols =␣
 ↪['age_cat','etatcivil_cat','center_nearest','center_density','deprivation_pca_q5','mammo_ra
```

[279]:
```
df_1st_1318 = df_1st_1318.dropna(subset = cols)
```

[289]:
```
X = df_1st_1318[cols]
y = df_1st_1318.mammo.reset_index(drop = True)
```

[290]:
```
# X = X.values
X = pd.DataFrame(StandardScaler().fit_transform(X),columns = cols)
```

[292]:
```
logit_model=sm.Logit(y,X)
result=logit_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.485273
         Iterations 6
```

[293]:
```
result.summary2()
```

[293]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
==================================================================
```

```
Model:                  Logit            Pseudo R-squared: 0.264
Dependent Variable:     mammo            AIC:              77550.8104
Date:                   2020-07-10 13:51 BIC:              77606.5410
No. Observations:       79892            Log-Likelihood:   -38769.
Df Model:               5                LL-Null:          -52711.
Df Residuals:           79886            LLR p-value:      0.0000
Converged:              1.0000           Scale:            1.0000
No. Iterations:         6.0000
-------------------------------------------------------------------
                     Coef.  Std.Err.    z     P>|z|   [0.025  0.975]
-------------------------------------------------------------------
age_cat              0.0266   0.0092   2.8997 0.0037  0.0086  0.0445
etatcivil_cat        1.5869   0.0107 148.4609 0.0000  1.5660  1.6079
center_nearest      -0.0037   0.0166  -0.2239 0.8228 -0.0363  0.0289
center_density       0.0244   0.0165   1.4788 0.1392 -0.0079  0.0567
deprivation_pca_q5 -0.0360   0.0102  -3.5138 0.0004 -0.0561 -0.0159
mammo_rate_q5        0.2189   0.0101  21.6363 0.0000  0.1991  0.2387
===================================================================

"""
```

[294]:

```python
result.pred_table()
```

[294]:

```
array([[37882., 12325.],
       [ 2705., 26980.]])
```

[295]:

```python
mfx = result.get_margeff()
print(mfx.summary())
```

```
        Logit Marginal Effects
=====================================
Dep. Variable:                  mammo
Method:                          dydx
At:                           overall
================================================================================
======
                       dy/dx    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
------
age_cat               0.0041      0.001      2.900      0.004       0.001
0.007
etatcivil_cat         0.2476      0.001    490.513      0.000       0.247
0.249
center_nearest       -0.0006      0.003     -0.224      0.823      -0.006
0.005
center_density        0.0038      0.003      1.479      0.139      -0.001
```

```
0.009
deprivation_pca_q5    -0.0056      0.002     -3.514      0.000      -0.009
-0.002
mammo_rate_q5          0.0342      0.002     21.784      0.000       0.031
0.037
================================================================================
======
```

6 Join count

6.1 Global

[303]:
```python
from esda.join_counts import Join_Counts
from pysal.explore.pointpats import PointPattern

from pysal.lib.weights.weights import W
# import pysal.lib.cg
from scipy.spatial import cKDTree
from pysal.lib.weights.distance import get_points_array
```

[310]:
```python
df_1st_1318 = gpd.GeoDataFrame(df_1st_1318,geometry = df_1st_1318['geometry'])
```

[324]:
```python
import libpysal as lps
def get_KNNW(df,nn,lon,lat):
    """One liner to get fast knn weights calculation.
    `get_points_array` function: This function extracts the coordinates of all␣
 ↪vertices
    for a variety of geometry packages in Python and returns a `numpy` array.

    Then, we must build the `KDTree` using `scipy`. For nearly any application,␣
 ↪the `cKDTree` will be faster.
    `KDTree` is an implementation of the datastructure in pure Python, whereas␣
 ↪the `cKDTree` is
    an implementation in Cython."""
    if not df.empty:
        nodes = lps.cg.KDTree(np.array(df[[lon,lat]]))
        weight = lps.weights.KNN(nodes,k = nn)
    #     weight.transform = transform
        return df,weight
    else:
        return None,None
df_knn8, weight = get_KNNW(df_1st_1318,8,'E_shifted','N_shifted')
```

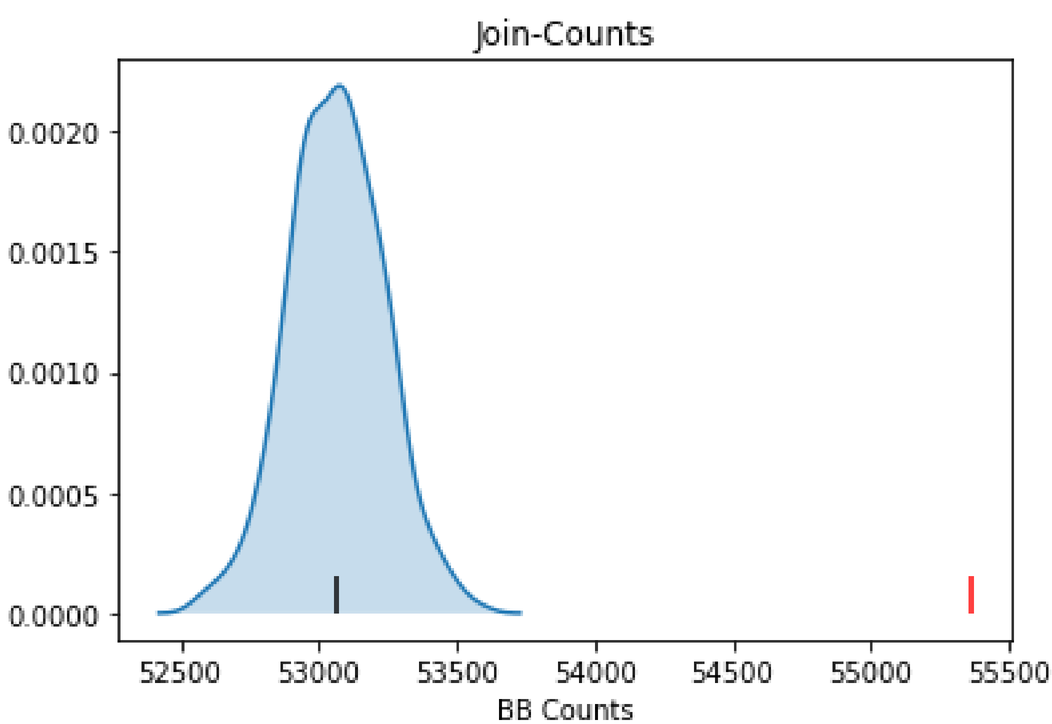
```
/Users/david/opt/anaconda3/envs/spatial/lib/python3.8/site-
packages/libpysal/weights/weights.py:172: UserWarning: The weights matrix is not
fully connected:
 There are 2383 disconnected components.
```

```
warnings.warn(message)
```

[329]:
```
def JoinCount(db,col,knn,w):
    xlabel = "Binary Join Count - {} - {}".format(col,str(knn))
    y = db[col]
    np.random.seed(12345)
    w.transform = 'b'
    jc = esda.Join_Counts(y,w)
    print(col,knn,jc.p_sim_bb,jc.p_sim_bw,jc.mean_bb,jc.mean_bw,sep = ',')
    ##
    sns.kdeplot(jc.sim_bb, shade=True)
    plt.vlines(jc.bb, 0, 0.00015, color='r')
    plt.vlines(jc.mean_bb, 0,0.00015)
    plt.xlabel('BB Counts')
    plt.title('Join-Counts')
    filename = xlabel +'.pdf'
#     plt.savefig(globalautocorr_spatial_result_folder/filename,dpi = 800,␣
↪bbox_inches = 'tight')
    return jc, plt.show()
```

[330]:
```
print('Variable','KNN','p_sim_bb','p_sim_bw','mean_bb','mean_bw',sep = ',')
jc_knn8 = JoinCount(df_knn8,'mammo',8,weight)
```

```
Variable,Distance,p_sim_bb,p_sim_bw,mean_bb,mean_bw
mammo,8,0.001,1.0,53064,179527
```



[333]:
```
df_knn8['no_mammo'] = 1-df_knn8['mammo']
```

[334]:
```
df_knn8[['mammo','no_mammo','geometry']].to_file(result_folder/'df_1st1318.
↪geojson',driver = 'GeoJSON')
```