In [1]:
```
from requirements import *
import pandas as pd
import seaborn as sns
from matplotlib import pyplot
from collections import Counter
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import scipy.cluster.hierarchy as sch
from sklearn.preprocessing import normalize
from sklearn import preprocessing
import geopandas as gpd
import folium
from folium import plugins
from datetime import timedelta, date
import shapely
from shapely.geometry import Polygon, Point
from matplotlib import dates as dt
from shapely.wkt import loads
pd.set_option('display.max_columns', None)
sns.set(font_scale=1)
```


1. Understanding COVID 19 with spatial data

To understand where the pandemic has had more effect, we will plot in a map the number of cases that have been detected. We will use the dataset 'casos_per_municipi'. This data was last updated 12th December, but we can find it updated in the following link: https://analisi.transparenciacatalunya.cat/ca/Salut/Registre-de-casos-de-COVID-19-realitzats-a-Catalun/jj6z-iyrp

In [2]:
```
filename = 'data/timelapse/casos_per_municipi_612.csv'
sep=","
encoding="utf-8"
data_covid_mun = pd.read_csv(filename,sep=';')
data_covid_mun
```

Out[2]:

	TipusCasData	ComarcaCodi	ComarcaDescripcio	MunicipiCodi	MunicipiDescripcio	SexeCodi	S
0	05/12/2020	2.0	Alt Empordà	17047.0	Castelló d'Empúries	0	
1	05/12/2020	40.0	Vallès Occidental	8184.0	Rubí	1	
2	05/12/2020	24.0	Osona	8283.0	Tona	1	

	TipusCasData	ComarcaCodi	ComarcaDescripcio	MunicipiCodi	MunicipiDescripcio	SexeCodi	S
3	05/12/2020	36.0	Tarragonès	43153.0	Torredembarra	1	
4	05/12/2020	36.0	Tarragonès	43131.0	Roda de Berà	0	
...	...	...	...	...	...	...	
266104	26/02/2020	2.0	Alt Empordà	17066.0	Figueres	1	
266105	26/02/2020	19.0	Garrotxa	17114.0	Olot	0	
266106	26/02/2020	NaN	NaN	NaN	No classificat	0	
266107	26/02/2020	10.0	Baix Empordà	17118.0	Palamós	0	
266108	26/02/2020	NaN	NaN	NaN	No classificat	1	

266109 rows × 9 columns

Now we load the geolocation of the cities.

In [3]:

```
filename = 'data/timelapse/coordenades_municipis.xlsx'
sep=","
encoding="utf-8"
municipis = pd.read_excel(filename)
municipis = municipis[['Municipi','Longitud','Latitud']]
municipis['Localització'] = municipis['Latitud'].astype(str) + ',' +  municipis['Longit
municipis.head()
```

```
NumExpr defaulting to 8 threads.
```

Out[3]:

	Municipi	Longitud	Latitud	Localització
0	Alcover	1.170206	41.262632	41.262631863,1.170205538
1	Alió	1.306398	41.295126	41.295126206,1.306398395
2	Bràfim	1.341137	41.268738	41.268738017,1.341136631
3	Cabra del Camp	1.276657	41.395279	41.395278969,1.276657071
4	Figuerola del Camp	1.265317	41.371915	41.371915239,1.26531658

The goal is to keep a dataset with the diary updates of cases by town/city to plot it in the map in a timelapse. Therefore, we will add the localization of each town to the dataset.

In [4]:

```
diccionari_Municipi_loc = dict(zip(municipis['Municipi'],municipis['Localització']),col

data_covid_mun['Localització'] = data_covid_mun['MunicipiDescripcio'].apply(lambda x: d

longlat = data_covid_mun["Localització"].str.split(",", n = 1, expand = True)
data_covid_mun['Latitud'] = longlat[0]
data_covid_mun['Longitud'] = longlat[1]
data_covid_mun.head()
```

Out[4]:

	TipusCasData	ComarcaCodi	ComarcaDescripcio	MunicipiCodi	MunicipiDescripcio	SexeCodi	SexeDe
0	05/12/2020	2.0	Alt Empordà	17047.0	Castelló d'Empúries	0	

	TipusCasData	ComarcaCodi	ComarcaDescripcio	MunicipiCodi	MunicipiDescripcio	SexeCodi	SexeDe
1	05/12/2020	40.0	Vallès Occidental	8184.0	Rubí	1	
2	05/12/2020	24.0	Osona	8283.0	Tona	1	
3	05/12/2020	36.0	Tarragonès	43153.0	Torredembarra	1	
4	05/12/2020	36.0	Tarragonès	43131.0	Roda de Berà	0	

We are interested only in the localization, the date and the number of cases identified:

In [5]:
```
data = data_covid_mun[['TipusCasData','Longitud','Latitud','NumCasos','TipusCasDescripc
data.head()
```

Out[5]:

	TipusCasData	Longitud	Latitud	NumCasos	TipusCasDescripcio
0	05/12/2020	3.074791957	42.258576569	2.0	Sospitós
1	05/12/2020	2.031335058	41.493484672	7.0	Sospitós
2	05/12/2020	2.229175918	41.850021542	1.0	Sospitós
3	05/12/2020	1.396064236	41.146065786	2.0	Positiu PCR
4	05/12/2020	1.456926185	41.186001973	2.0	Sospitós

Now this is the dataset we need to build the timelapse. Before starting to build the map:

In [6]:
```
print("We keep the information of " + str(len(data.TipusCasData.unique()))
      + " days.")
print("There is a total of " + str(len(data))+ " rows containing the information: "  +
print("For every day, we have the number of cases and the type: " + str(data.TipusCasDe
```

```
We keep the information of 284 days.
There is a total of 266109 rows containing the information: ['TipusCasData' 'Longitud'
'Latitud' 'NumCasos' 'TipusCasDescripcio']
For every day, we have the number of cases and the type: ['Sospitós' 'Positiu PCR' 'Posi
tiu per ELISA' 'Epidemiològic'
 'PCR probable' 'Positiu per Test Ràpid'].
```

We will only keep the rows with "Positius" of any kind.

In [7]:
```
data = data[data.TipusCasDescripcio.str.contains('Positiu')]
print("There is a total of " + str(len(data))+ " rows.")
print("For every day, we have the number of cases and the type: " + str(data.TipusCasDe
```

```
There is a total of 73223 rows.
For every day, we have the number of cases and the type: ['Positiu PCR' 'Positiu per ELI
SA' 'Positiu per Test Ràpid'].
```

We sort the data to have first the oldest detected positives.

In [8]:
```
data = data.dropna()
data = data.reindex(index=data.index[::-1])
data.head()
```

Out[8]:

	TipusCasData	Longitud	Latitud	NumCasos	TipusCasDescripcio
266083	28/02/2020	2.824865628	41.983155443	1.0	Positiu PCR

	TipusCasData	Longitud	Latitud	NumCasos	TipusCasDescripcio
266061	02/03/2020	2.177241106	41.382405393	1.0	Positiu PCR
266056	02/03/2020	2.177241106	41.382405393	1.0	Positiu PCR
266038	04/03/2020	2.177241106	41.382405393	1.0	Positiu PCR
266025	05/03/2020	2.187770377	41.485795664	1.0	Positiu PCR

In [9]:
```
data.to_csv('data/generated_data_towns.csv')
```

2. Timelapse generation

To plot a timelapse, I will use the Folium library that contains a HeatMapWithTime function. To suit the requirements of this function, we will need a list with the structure ['Latitude','Longitude','Heat'], where Heat requires a number between 0 and 1. As our Heat is the number Covid-19 positives, we will normalize the column "NumCasos" using the formula below. Adjusting the alpha we will be able

$$f(x) = \frac{a \log(1+x)}{1 + a \log(1+x)}$$

to control the heat of the plotted points:

In [10]:
```
##creates the list with [Lat,Lon,Heat] for every day
def heat_generation(data):
    max_value = data.NumCasos.max()
    alpha = 1000
    heat_list = [
        [
            [
                row.Latitud,
                row.Longitud,
                alpha*np.log(1 + row.NumCasos / max_value)
                / (1 + alpha * np.log(1 + row.NumCasos / max_value)),
            ]
            #
            for index, row in data[data.TipusCasData == i].iterrows()
        ]
        #for each day
        for i in data.TipusCasData.unique()
    ]
    return heat_list
```

In [11]:
```
##plot the timelapse
def plot(heat_list,location,zoom,rati,data):
    mapa = folium.Map(location = location,#[41.723864,1.901103],
                      zoom_start=zoom,#7.5,
                      ratio = rati,# 40,
                      tiles = "cartodbpositron")

    heatmap = plugins.HeatMapWithTime(
        heat_list,
        index=list(data.TipusCasData.unique()),
        name="Timelapse",
```

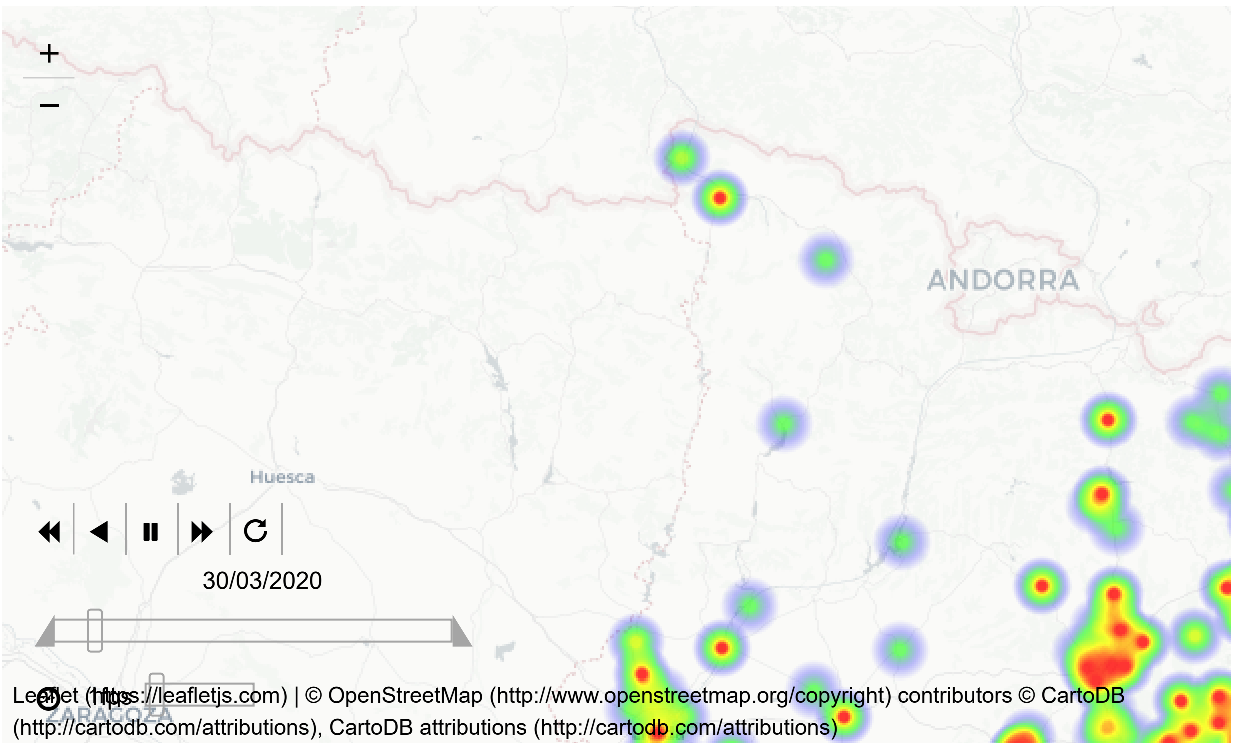
```
        auto_play=True,
        max_opacity=0.8
    )
    heatmap.add_to(mapa)
    # Layercontrols lets you change visuals in the html page
    ctrl = folium.LayerControl()
    ctrl.add_to(mapa)
    return mapa
```

In [12]:
```
heat_list = heat_generation(data)
mapa_catalunya = plot(heat_list,
                     location = [41.723864,1.901103],
                     zoom = 7.5,
                      rati = 40,
                      data = data
                     )
```

Now in the 'heat_list' variable we have the localization of the cases by day and the incidence of positive cases. Finally we can plot the timelapse:

In [13]:
```
mapa_catalunya
```

Out[13]:



In [14]:
```
mapa_catalunya.save('data/output_maps/general_map.html')
```

What this visualization tells us is:

- The pandemic started in Barcelona and extended fastly through the metropolitan area.
- Lleida and Igualada became early a focus of Covid 19.
- As could be expected, there is more incidence of the virus in the most populated areas.

3. Is there a correlation between the number of cases detected and the mean income of a zone?

To detect if there is a relation between the mean income of a place and the incidence of the virus, we will add a new dataset that contains the number of positives by ABS (Àrees Bàsiques de Salut), that we can find updated in this open data portal: https://analisi.transparenciacatalunya.cat/ca/Salut/Registre-de-casos-de-COVID-19-realitzats-a-Catalun/xuwf-dxjd.

Each ABS has a socioeconomic index that measures the richness or poverty of a zone. As higher the ISC, harder the socioeconomic situation of the territory. We will also load the information of the population of each ABS, with which we will compute the percentage of people afected by the virus in each zone.

So we will plot this information in a map and see if we can identify a correlation.

3.1 Data collection

First we load the cases by ABS, the content of the dataset we will use:

1. Date
2. ABS Code
3. ABS Description
4. Number of cases

In [15]:

```
##LOAD THE CASES BY ABS
filename = 'data/income/casos_per_ABS_612.csv'
# Read the data as a dataframe
data_covid = pd.read_csv(filename,sep=';')
data_covid.head(2)
```

Out[15]:

	TipusCasData	RegioSanitariaCodi	RegioSanitariaDescripcio	SectorSanitariCodi	SectorSanitariDescripci
0	26/2/2020	7,802	Metropolità Nord	7842	Vallès Orient
1	26/2/2020	7,803	Barcelona Ciutat	7850	Barcelona Sarrià-Sar Gerva

For the last timelapses, we will make a copy of this data:

In [16]:

```
data_timelapse = data_covid.copy()
data_timelapse = data_timelapse.rename(columns={"ABSCodi": "CODIABS"})
data_timelapse = data_timelapse[data_timelapse.TipusCasDescripcio.str.contains('Positiu
```

In [17]:

```
data_covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 302872 entries, 0 to 302871
Data columns (total 11 columns):
 #   Column                    Non-Null Count   Dtype 
---  ------                    --------------   ----- 
 0   TipusCasData              302872 non-null  object
 1   RegioSanitariaCodi        302872 non-null  object
 2   RegioSanitariaDescripcio  302872 non-null  object
 3   SectorSanitariCodi        302872 non-null  int64 
 4   SectorSanitariDescripcio  302872 non-null  object
```

```
 5   ABSCodi                  302872 non-null  int64
 6   ABSDescripcio            302872 non-null  object
 7   SexeCodi                 302872 non-null  int64
 8   SexeDescripcio           302872 non-null  object
 9   TipusCasDescripcio       302872 non-null  object
 10  NumCasos                 302872 non-null  int64
dtypes: int64(4), object(7)
memory usage: 25.4+ MB
```

We will filter the dataset by only the "Positiu" the positive cases and then we will group by ABS. In this new dataframe we will have all the ABS with the number of positive cases detected.

In [18]:
```python
data_covid = data_covid[data_covid.TipusCasDescripcio.str.contains('Positiu')]
data_covid = pd.DataFrame(data_covid.groupby(['ABSDescripcio','ABSCodi'],sort=False)['N
data_covid = data_covid.reset_index()
data_covid.head()
```

Out[18]:

	ABSDescripcio	ABSCodi	NumCasos
0	Girona 1	124	907
1	Barcelona 10-C	73	851
2	Barcelona 9 H	395	1044
3	Barcelona 8 J	385	1308
4	Sant Cugat del Vallès 2	389	1037

In [19]:
```python
data_covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 373 entries, 0 to 372
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   ABSDescripcio  373 non-null    object
 1   ABSCodi        373 non-null    int64
 2   NumCasos       373 non-null    int64
dtypes: int64(2), object(1)
memory usage: 8.9+ KB
```

In [20]:
```python
data_covid.NumCasos.sum()
```

Out[20]:
```
327639
```

We will also load the population of each ABS to have a better knowledge of the real impact in the community and add it to the data_covid dataframe.

In [21]:
```python
filename = 'data/income/poblacio_per_ABS.xlsx'
populationABS = pd.read_excel(filename)
populationABS.head()
```

Out[21]:

	ABSDESC	CODIABS	POBLACIÓ
0	ALT URGELL-SUD	169	3813
1	ALTA RIBAGORÇA	180	3452
2	ARAN	7	9268
3	LA CERDANYA	103	16345

	ABSDESC	CODIABS	POBLACIÓ
4	LA POBLA DE SEGUR	179	4236

In [22]:
```
data_covid = data_covid.rename(columns={"ABSCodi": "CODIABS"})
data_covid = pd.merge(populationABS, data_covid, on="CODIABS")
data_covid.head()
```

Out[22]:

	ABSDESC	CODIABS	POBLACIÓ	ABSDescripcio	NumCasos
0	ALT URGELL-SUD	169	3813	Alt Urgell-sud	91
1	ALTA RIBAGORÇA	180	3452	Alta Ribagorça	76
2	ARAN	7	9268	Aran	397
3	LA CERDANYA	103	16345	La Cerdanya	866
4	LA POBLA DE SEGUR	179	4236	La Pobla de Segur	165

Now we will load the ISC index and the geodata information. In the 'indicador_socioeco' file there is a lot of information but we will only take the ISC index given to each ABS.

In [23]:
```
#LOAD THE ISC INDEX
filename = 'data/income/indicador_socioeco.xlsx'
dataind = pd.read_excel(filename)
dataind.head()
```

Out[23]:

	idabs	ABS	Població exempta de copagament farmacèutic	Població amb rendes inferiors a 18.000 euros	Població amb rendes superiors a 100.000 euros	Població amb ocupacions manuals	Població amb nivell d'instrucció insuficient	Taxa de mortalitat prematura	H
0	1	AGRAMUNT	0.748476	1.121456	0.371568	1.392562	0.985131	0.835667	
1	2	ALCARRÀS	0.722859	1.211158	0.251877	1.351089	1.283358	0.996216	
2	3	ALFARRÀS/ALMENAR	1.181349	1.154163	0.222573	1.302565	1.211848	1.228782	
3	4	AMPOSTA	1.313790	1.188989	0.249440	1.357604	1.374200	1.067162	
4	5	ANGLÈS	0.943633	1.063016	0.340070	1.240794	1.196476	0.770039	

In [24]:
```
dataind = dataind[['idabs','ISC reescalat']]
dataind.head()
```

Out[24]:

	idabs	ISC reescalat
0	1	44.339777
1	2	49.502202
2	3	52.124912
3	4	50.919598

	idabs	ISC reescalat
4	5	44.738977

Now we load the geolocalization of each ABS with a shapefile.

In [25]:
```
shapefile = gpd.read_file("data/income/ABS_2018/ABS_2018.shp",crs="epsg:4326")
shapefile.head()
```

```
Failed to auto identify EPSG: 7
```

Out[25]:

	CODIABS	NOMABS	CODISS	NOMSS	CODIRS	NOMRS	CODIAGA	NOMAGA	ÀREA	g
0	379	Lleida - 7	6156	Lleida	61	Regió Sanitària Lleida	05	Lleida	24324352.29	F ((29 461 29
1	138	Lleida - 4	6156	Lleida	61	Regió Sanitària Lleida	05	Lleida	15710735.26	F ((30 461 30
2	248	Terrassa - B	7843	Vallès Occidental Oest	78	Regió Sanitària Barcelona	66	Vallès Occidental Oest	7055788.22	F ((42 460 42
3	345	Sabadell - 4B	7844	Vallès Occidental Est	78	Regió Sanitària Barcelona	35	Vallès Occidental Est	3087174.84	F ((42 459 42
4	191	Sabadell - 1B	7844	Vallès Occidental Est	78	Regió Sanitària Barcelona	35	Vallès Occidental Est	1570879.25	F ((42 459 42

Joining the information

We add to the shapefile the ISC index:

In [26]:
```
#idabs -> ISC
diccionari_ABS_codi = dict(zip(dataind['idabs'],dataind['ISC reescalat']),columns = (['
```

In [27]:
```
shapefile['CODIABS'] =  shapefile['CODIABS'].astype(int)
shapefile['ISC'] = shapefile['CODIABS'].apply(lambda x: diccionari_ABS_codi.get(x))
```

In [28]:
```
shapefile.head(2)
```

Out[28]:

	CODIABS	NOMABS	CODISS	NOMSS	CODIRS	NOMRS	CODIAGA	NOMAGA	ÀREA	geor

	CODIABS	NOMABS	CODISS	NOMSS	CODIRS	NOMRS	CODIAGA	NOMAGA	ÀREA	geor
0	379	Lleida - 7	6156	Lleida	61	Regió Sanitària Lleida	05	Lleida	24324352.29	POLY ((29332 461371 29333
1	138	Lleida - 4	6156	Lleida	61	Regió Sanitària Lleida	05	Lleida	15710735.26	POLY ((30142 461552 30142

We merge the data_covid and shapefile:

In [29]:
```
#merge with the shapefile
sh = pd.merge(shapefile, data_covid, on="CODIABS")
data = sh[['geometry','CODIABS','ABSDescripcio','NumCasos','ISC','POBLACIÓ']]
data.head()
```

Out[29]:

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ
0	POLYGON ((293325.537 4613716.748, 293337.234 4...	379	Lleida 7 Onze de Setembre	1215	28.885710	17381
1	POLYGON ((301421.259 4615521.650, 301428.290 4...	138	Lleida 4 Balàfia-Pardinyes	1498	51.264958	27056
2	POLYGON ((420649.529 4604118.364, 420712.209 4...	248	Terrassa B	1415	70.236305	27907
3	POLYGON ((424041.189 4599915.081, 424033.228 4...	345	Sabadell 4 B	973	53.888378	24233
4	POLYGON ((427125.633 4598252.103, 426969.915 4...	191	Sabadell 1B	845	34.717657	21175

We will now compute a 'ratio' column between the cases and the population:

In [30]:
```
data['ratio'] = (data['NumCasos'] / data['POBLACIÓ'])*100
data.head()
```

Out[30]:

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ	ratio
0	POLYGON ((293325.537 4613716.748, 293337.234 4...	379	Lleida 7 Onze de Setembre	1215	28.885710	17381	6.990392
1	POLYGON ((301421.259 4615521.650, 301428.290 4...	138	Lleida 4 Balàfia-Pardinyes	1498	51.264958	27056	5.536665
2	POLYGON ((420649.529 4604118.364, 420712.209 4...	248	Terrassa B	1415	70.236305	27907	5.070412

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ	ratio
3	POLYGON ((424041.189 4599915.081, 424033.228 4...	345	Sabadell 4 B	973	53.888378	24233	4.015186
4	POLYGON ((427125.633 4598252.103, 426969.915 4...	191	Sabadell 1B	845	34.717657	21175	3.990555

Finally we load the income data for cities with populationh higher than 200.000. To join it with the ABS information we will also need the Postal Code information of each ABS.

In [31]:
```
filename = 'data/income/renda.xlsx'
# Read the data as a dataframe
renda = pd.read_excel(filename)
renda.head(2)
```

Out[31]:

	Barris	Població	Renda disponible	Codi Postal
0	Casagemes	Badalona	27345	08911
1	Badalona	Badalona	25097	08912

In [32]:
```
filename = 'data/income/ABSCodiPostal.xlsx'
codiPos = pd.read_excel(filename)
codiPos['CodiPostal'] = '0'+ codiPos['CodiPostal'].astype(str)
codiPos.head(2)
```

Out[32]:

	Població	Nom ABS	CodiABS	CodiPostal
0	Badalona	Badalona 1	272	08911
1	Badalona	Badalona 10	281	08917

We join this two dataframes by Codi Postal (it will be None for large cities):

In [33]:
```
renda = renda.rename(columns={"Codi Postal": "CodiPostal"})
renda.CodiPostal = renda.CodiPostal.astype(int)
codiPos.CodiPostal =codiPos.CodiPostal.astype(int)
```

In [34]:
```
income_data = pd.merge(renda, codiPos, on="CodiPostal")
income_data = income_data[['CodiABS','Renda disponible']]
income_data.head(4)
```

Out[34]:

	CodiABS	Renda disponible
0	272	27345
1	273	27345
2	274	27345
3	275	18682

We add the income by ABS code to the data dataframe.

In [35]:
```
diccionari_ABS_RENDA = dict(zip(income_data['CodiABS'],income_data['Renda disponible']))
```

```
data['Income'] = data['CODIABS'].apply(lambda x: diccionari_ABS_RENDA.get(x))
```

In [36]:
```
data.head()
```

Out[36]:

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ	ratio	Income
0	POLYGON ((293325.537 4613716.748, 293337.234 4...	379	Lleida 7 Onze de Setembre	1215	28.885710	17381	6.990392	NaN
1	POLYGON ((301421.259 4615521.650, 301428.290 4...	138	Lleida 4 Balàfia-Pardinyes	1498	51.264958	27056	5.536665	NaN
2	POLYGON ((420649.529 4604118.364, 420712.209 4...	248	Terrassa B	1415	70.236305	27907	5.070412	20835.0
3	POLYGON ((424041.189 4599915.081, 424033.228 4...	345	Sabadell 4 B	973	53.888378	24233	4.015186	21603.0
4	POLYGON ((427125.633 4598252.103, 426969.915 4...	191	Sabadell 1B	845	34.717657	21175	3.990555	22625.0

In 'data' we will have:

1. geometry: ABS Location
2. CODIABS
3. ABSDescripcio
4. NumCasos
5. ISC
6. Ratio infected people
7. Income (cities with population > 200.000)

In [37]:
```
data = data.to_crs(epsg=4326)
```

This is the data we will use for setting the map:

In [38]:
```
data.head()
```

Out[38]:

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ	ratio	Income
0	POLYGON ((0.51817 41.64841, 0.51831 41.64851, ...	379	Lleida 7 Onze de Setembre	1215	28.885710	17381	6.990392	NaN
1	POLYGON ((0.61470 41.66671, 0.61479 41.66677, ...	138	Lleida 4 Balàfia-Pardinyes	1498	51.264958	27056	5.536665	NaN

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ	ratio	Income
2	POLYGON ((2.04803 41.58482, 2.04878 41.58497, ...	248	Terrassa B	1415	70.236305	27907	5.070412	20835.0
3	POLYGON ((2.08925 41.54729, 2.08915 41.54732, ...	345	Sabadell 4 B	973	53.888378	24233	4.015186	21603.0
4	POLYGON ((2.12643 41.53260, 2.12455 41.53348, ...	191	Sabadell 1B	845	34.717657	21175	3.990555	22625.0

In [39]:
```
data.to_file('data/income/generated_geodataISC.geojson')
```

```
Normalized/laundered field name: 'ABSDescripcio' to 'ABSDescrip'
```

3.2 Building the map

We finally have the data sorted in the same dataframe.

In [40]:
```
######### IF ANYTHING GOES WRONG, OPEN THE geodataISC.geojson
#filename = 'data/income/genereated_geodataISC.geojson'
#data = gpd.read_file(filename)
#data
```

For the cartoframes Map, we will color the ABS zones by the ISC index and add a circle as big as the number of positive cases detected in the ABS. For that, we will compute the centroids of each ABS in which we will plot the circle.

In [41]:
```
data = data.to_crs(epsg=4326)
```

In [42]:
```
data['centroids'] = data['geometry'].centroid
data.head()
```

Out[42]:

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ	ratio	Income	centroids
0	POLYGON ((0.51817 41.64841, 0.51831 41.64851, ...	379	Lleida 7 Onze de Setembre	1215	28.885710	17381	6.990392	NaN	POINT (0.58302 41.64366)
1	POLYGON ((0.61470 41.66671, 0.61479 41.66677, ...	138	Lleida 4 Balàfia-Pardinyes	1498	51.264958	27056	5.536665	NaN	POINT (0.63546 41.64773)

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ	ratio	Income	centroids
2	POLYGON ((2.04803 41.58482, 2.04878 41.58497, ...	248	Terrassa B	1415	70.236305	27907	5.070412	20835.0	POINT (2.04961 41.56712)
3	POLYGON ((2.08925 41.54729, 2.08915 41.54732, ...	345	Sabadell 4 B	973	53.888378	24233	4.015186	21603.0	POINT (2.07910 41.55625)
4	POLYGON ((2.12643 41.53260, 2.12455 41.53348, ...	191	Sabadell 1B	845	34.717657	21175	3.990555	22625.0	POINT (2.11935 41.53819)

3.2.1 Map by ISC and ratio of infection

In [43]:
```
data.ratio.describe()
```

Out[43]:
```
count    371.000000
mean       4.225076
std        1.290735
min        1.513750
25%        3.369896
50%        4.051877
75%        4.937168
max        9.016510
Name: ratio, dtype: float64
```

In [44]:
```
data.ISC.describe()
```

Out[44]:
```
count    365.000000
mean      41.540767
std       15.036885
min        0.000000
25%       33.804476
50%       41.999450
75%       49.672796
max      100.000000
Name: ISC, dtype: float64
```

We plot the map with the values I think describe better the situation: I am taking the quartiles of the ISC and the ratio computed before. As darker the zone, more complex socioeconomic situation.

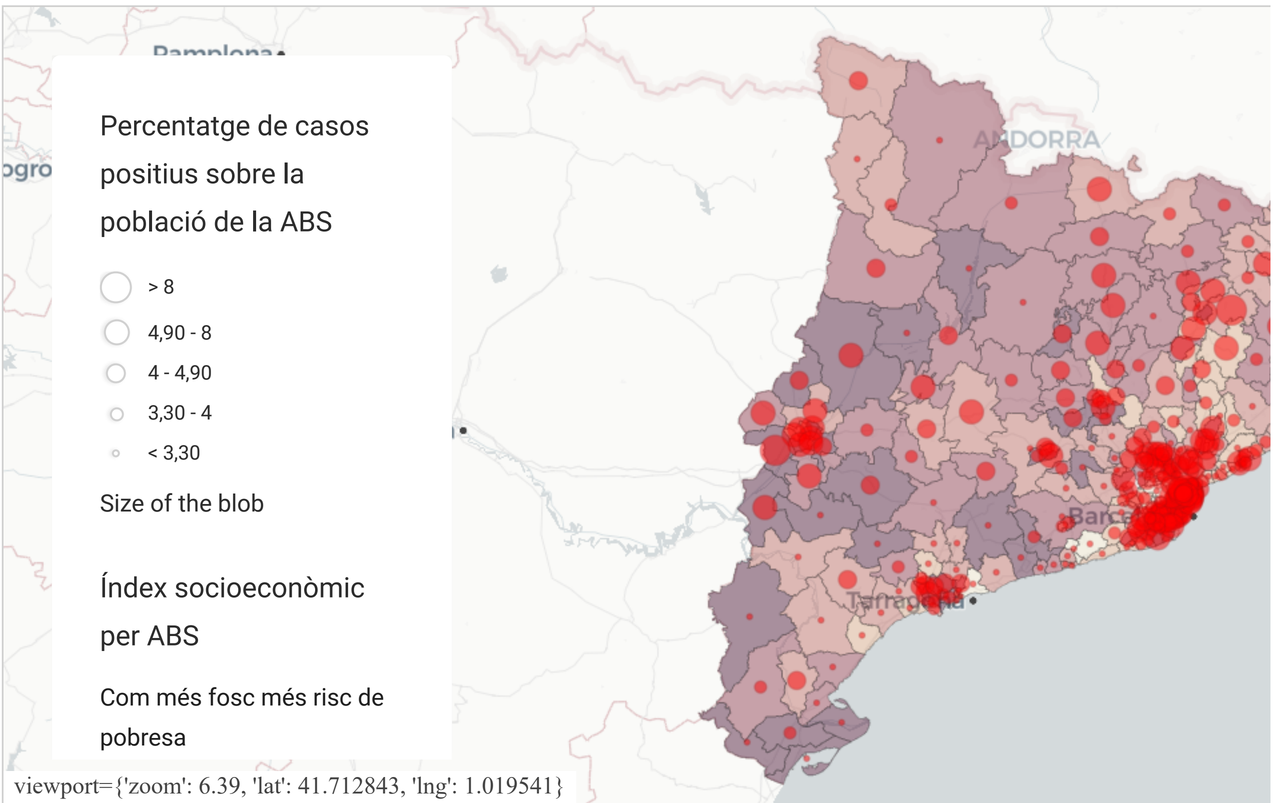
In [45]:
```
bks_ISC = [15,33.8,42,49]
bks_casos = [3.3,4,4.9,8]
ISC_MAP = Map([Layer(data,
           style = color_bins_style('ISC',
                                    breaks = bks_ISC,
                                    palette = 'BrwnYl',opacity = 0.5),
           legends = color_bins_legend(title='Índex socioeconòmic per ABS',
```

```
                                        description='Com més fosc més risc de pobresa',
                                     footer =''),
          encode_data=False
         ),
     Layer(data,
          #size_continuous_style('ratio', size_range= bks_casos),
          geom_col='centroids',
          legends=size_bins_legend(title='Percentatge de casos positius sobre la pobla
                                   footer='Size of the blob'),
          style=size_bins_style('ratio',
                                breaks = bks_casos,
                                color ='red',
                                opacity = 0.5),
          encode_data=False
         )
    ],
    show_info =True,
    size = (900,400),
    viewport={'zoom': 6.39, 'lat': 41.712843, 'lng': 1.019541}
)
```

In [46]:

```
ISC_MAP
```

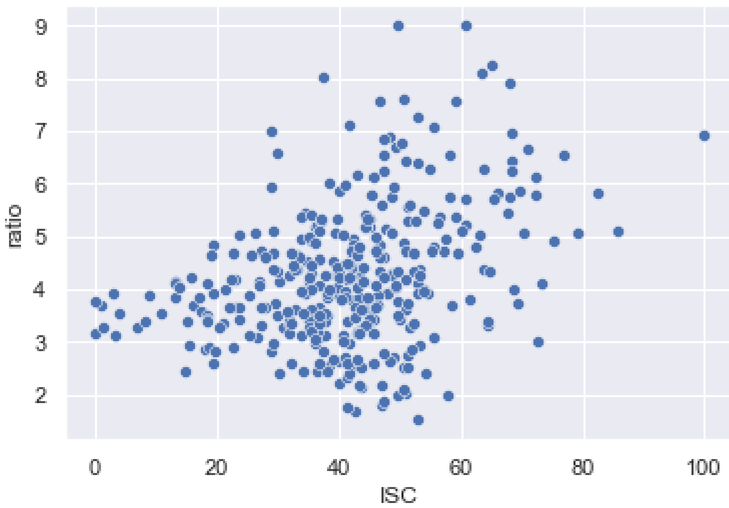
Out[46]:



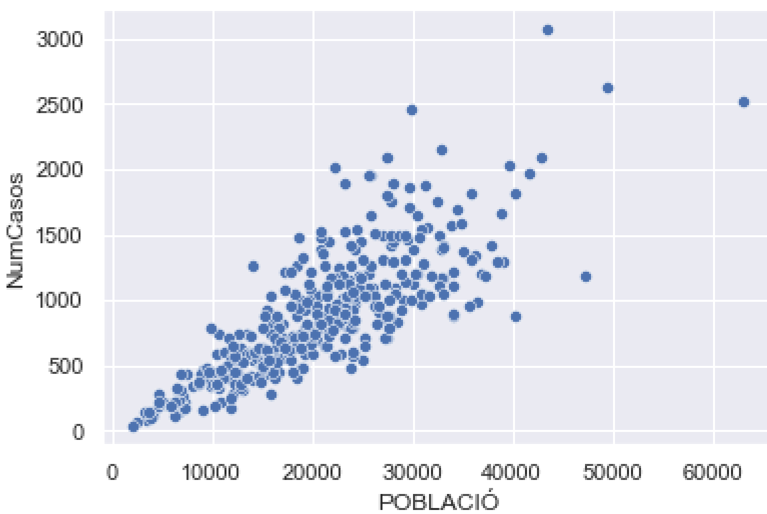
We can see a clear correlation between the ISC and the ratio of detected cases: on the darker zones the ratio number is bigger. As poorer the region, more cases detected. In the scatterplots below, we can see that for ISC between 0-20, the ratio of infected people is significantly lower.

In [47]:

```
sns.scatterplot(data.ISC,data.ratio)
plt.show()
```



In [72]:
```
sns.scatterplot(data.POBLACIÓ,data.NumCasos)
plt.show()
```



But a really correlated variable is the population size: the virus spreads easier if a lot of people live together.

In [49]:
```
corr =data.corr().style.background_gradient(cmap='coolwarm').set_precision(4)
corr
```

Out[49]:

	CODIABS	NumCasos	ISC	POBLACIÓ	ratio	Income
CODIABS	1.0000	-0.0840	0.1218	-0.0565	-0.0845	-0.2660
NumCasos	-0.0840	1.0000	0.0350	0.8122	0.6063	-0.1016
ISC	0.1218	0.0350	1.0000	-0.2230	0.3555	-0.6448
POBLACIÓ	-0.0565	0.8122	-0.2230	1.0000	0.0876	0.2007
ratio	-0.0845	0.6063	0.3555	0.0876	1.0000	-0.5134
Income	-0.2660	-0.1016	-0.6448	0.2007	-0.5134	1.0000

3.2.2 Map by income and ratio of infection (large cities)

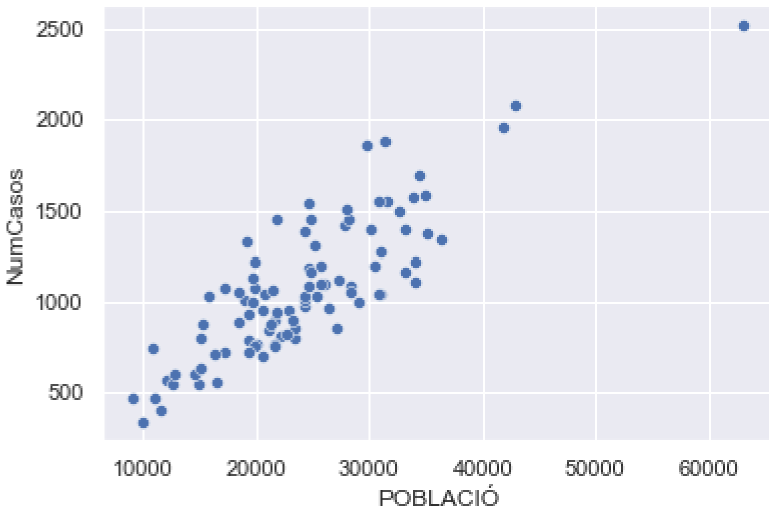
We now filter the data with the cities we have the information of income.

In [50]:
```
data_income = data.dropna()
data_income.head()
```

Out[50]:

	geometry	CODIABS	ABSDescripcio	NumCasos	ISC	POBLACIÓ	ratio	Income	centroids
2	POLYGON ((2.04803 41.58482, 2.04878 41.58497, ...	248	Terrassa B	1415	70.236305	27907	5.070412	20835.0	POINT (2.04961 41.56712)
3	POLYGON ((2.08925 41.54729, 2.08915 41.54732, ...	345	Sabadell 4 B	973	53.888378	24233	4.015186	21603.0	POINT (2.07910 41.55625)
4	POLYGON ((2.12643 41.53260, 2.12455 41.53348, ...	191	Sabadell 1B	845	34.717657	21175	3.990555	22625.0	POINT (2.11935 41.53819)
21	POLYGON ((2.13322 41.35863, 2.13322 41.35876, ...	383	Barcelona 3 H	1007	44.840758	18936	5.317913	18520.0	POINT (2.13964 41.36290)
24	POLYGON ((2.17814 41.37294, 2.17763 41.37236, ...	32	Barcelona 3-A	1190	46.571176	24697	4.818399	18520.0	POINT (2.17022 41.36313)

In [68]:
```
sns.scatterplot(data_income.POBLACIÓ,data_income.NumCasos)
plt.show()
```



In [51]:

```
data_income.Income.describe()
```

Out[51]:

```
count       94.000000
mean     21622.329787
std       6206.864558
min      12045.000000
25%      18458.000000
50%      19300.000000
75%      25996.250000
max      40022.000000
Name: Income, dtype: float64
```

In [52]:

```
data_income.ratio.describe()
```

Out[52]:

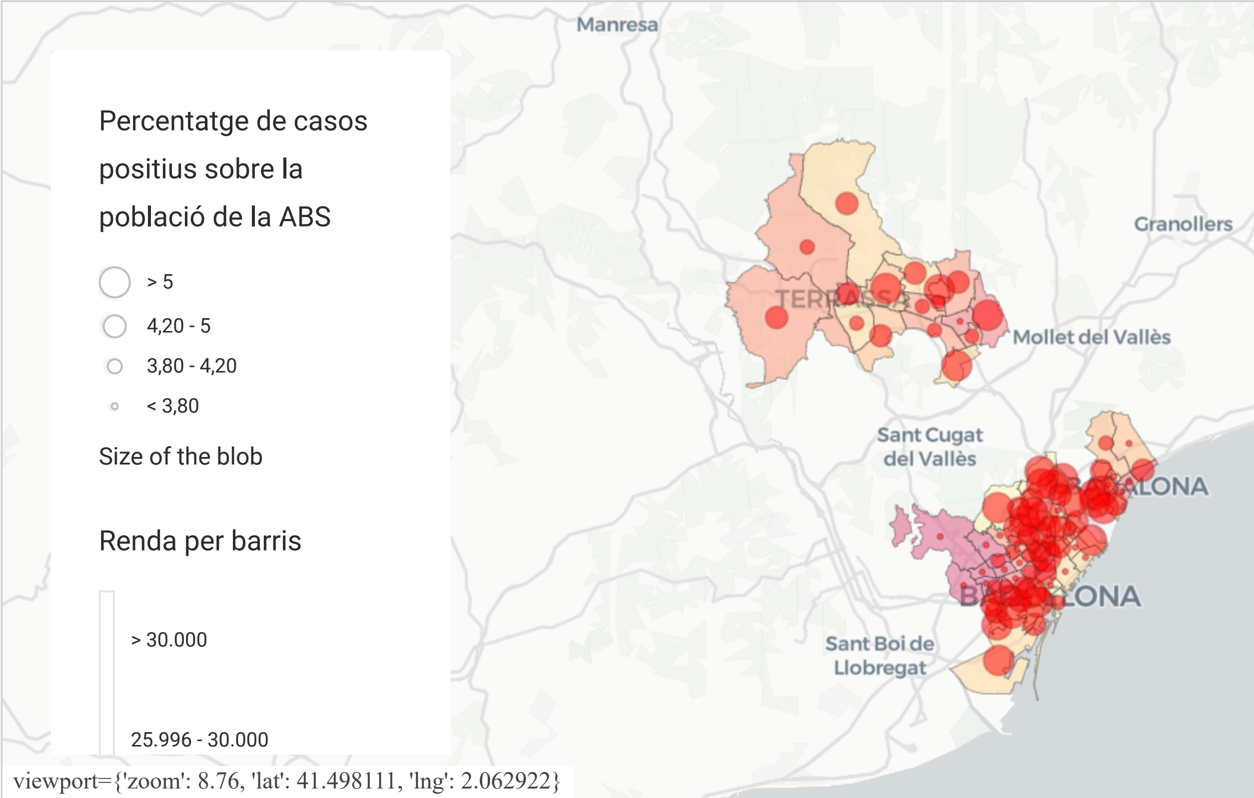
```
count    94.000000
mean      4.520052
std       0.908220
min       3.169795
25%       3.829954
50%       4.268621
75%       5.019949
max       6.947290
Name: ratio, dtype: float64
```

In [53]:

```
bks_renda = [18500,19400,21000,25996,30000]
#quartiles!
bks_casos =  bks_casos = [3.8,4.2,5]
Map([Layer(data_income,
           style = color_bins_style('Income',
                                    breaks = bks_renda,
                                    palette = 'pinkyl',opacity = 0.5),
           legends = color_bins_legend(title='Renda per barris',
                                       footer ='')
          ),
     Layer(data_income,
          geom_col='centroids',
          legends=size_bins_legend(title='Percentatge de casos positius sobre la poblac
                                   footer='Size of the blob'),
          style=size_bins_style('ratio',
                                breaks = bks_casos,
                                color ='red',
                                opacity = 0.5)
         )
    ],
```

```
    show_info =True,
    size = (900,400),
    viewport={'zoom': 8.76, 'lat':  41.498111, 'lng': 2.062922}
   )
```

Out[53]:

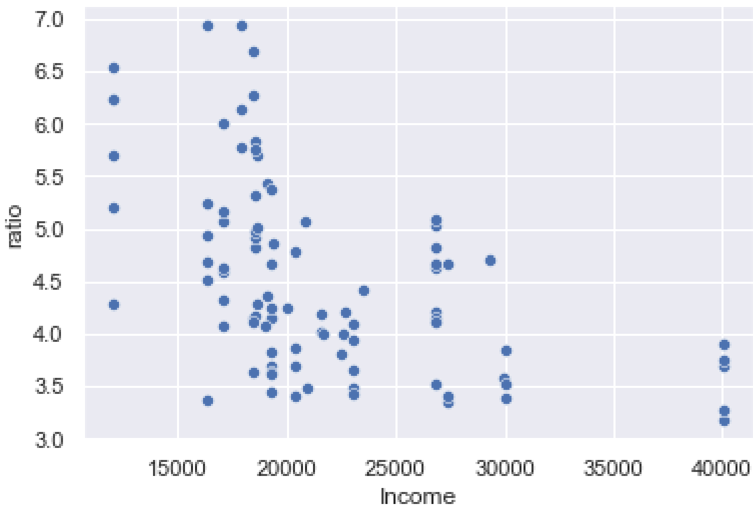


We can see that in the places with a higher income, the ratio of infected people is clearly lower. Specially in Barcelona, where Sarrià and Sant Gervasi present a completely different situation than Horta-Guinardó or Badalona.

In the scatterplot below, we can see that ABS with income over 20.000€ have a maximum if a 5% of infection, and with income over 30.000€ the infection rate decreases to a 4%. With incomes lower than 20.000€, we see that the infection rate can be up to a 7%.

In [54]:

```
sns.scatterplot(data_income.Income,data_income.ratio)
plt.show()
```



In [55]:
```
perc = (data.NumCasos.sum() /data.POBLACIÓ.sum())*100
print("Població afectada per la COVID19 a 6 de Desembre: "+ '%.2f'%(perc) + '%.')
```

```
Població afectada per la COVID19 a 6 de Desembre: 4.28%.
```