The accessibility of mental health care services in Italy

A geographic analysis through measures of autocorrelation and spatial regression

Geospatial analysis and representation course. A.Y. 2022/23. Alessandra Pomella

Contents

Dependencies installation and loading

```
# Installing packages not yet installed
packages <- c("rgdal","spdep","ggplot2","readxl","tidyverse","dplyr","sf","sp","latticeExtra","RColorBre
installed_packages <- packages %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  install.packages(packages[!installed_packages])}

# Packages loading
invisible(lapply(packages, library, character.only = TRUE))
```

Data loading

1) Geographic data

Option 1: Italian geographic data divided by region, source:Eurostat

```
# Administrative boundaries: © EuroGeographics, © TurkStat.
# Source: European Commission - Eurostat/GISCO
nutsrg <- read_sf("https://raw.githubusercontent.com/eurostat/Nuts2json/master/pub/v2/2021/3857/20M/nuts
nutsrg <- nutsrg[grepl("^IT", nutsrg$id), ]
head(nutsrg)
#> Simple feature collection with 6 features and 2 fields
#> Geometry type: MULTIPOLYGON
#> Dimension:     XY
#> Bounding box:  xmin: 906585 ymin: 4706403 xmax: 1547017 ymax: 5955426
#> Geodetic CRS:  WGS 84
#> # A tibble: 6 x 3
#>   id    na                                                         geometry
#>   <chr> <chr>                                            <MULTIPOLYGON [°]>
#> 1 ITG2  Sardegna                        (((1074194 4800470, 1066193 4746996~
#> 2 ITH1  Provincia Autonoma di Bolzano/Bozen (((1389009 5889852, 1373409 5881265~
#> 3 ITH2  Provincia Autonoma di Trento    (((1239802 5731773, 1215401 5733334~
#> 4 ITH3  Veneto                          (((1391410 5846527, 1375009 5825059~
#> 5 ITH4  Friuli-Venezia Giulia           (((1527416 5715769, 1527816 5730992~
#> 6 ITH5  Emilia-Romagna                  (((1100995 5641999, 1122596 5628338~
st_crs(nutsrg) <- 3857
plot(nutsrg$geometry)
```

```
EU_adm <- readOGR("data/NUTS_RG_20M_2021_3035","NUTS_RG_20M_2021_3035")
#> OGR data source with driver: ESRI Shapefile
#> Source: "C:\Users\Alessandra\Desktop\LM_Magistrale\DATA_SCIENCE\ANNO2\geospatial\project\mental_heal
```

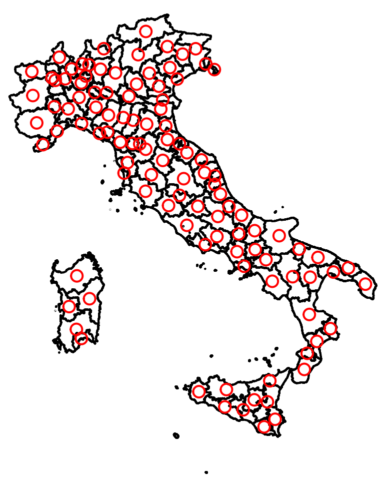
```
#> with 2010 features
#> It has 9 fields
italy_adm <- EU_adm[EU_adm$CNTR_CODE=='IT',]
```

Option 2: Italian geographic data divided by province, source:Istat The geographic data actually used will be those from Istat (option 2) since, in addition to the regional division, they also contain more information with a provincial degree of detail, which is important for analysis (more specific information, larger sample).

```
# Italian administrative boundaries updated to January 1, 2023.
# The dataset is published by Istat under Creative Commons - Attribution - 3.0 licence.
italy_provinces <- readOGR("data/Limiti01012023/ProvCM01012023","ProvCM01012023_WGS84")
#> OGR data source with driver: ESRI Shapefile
#> Source: "C:\Users\Alessandra\Desktop\LM_Magistrale\DATA_SCIENCE\ANNO2\geospatial\project\mental_heal
#> with 107 features
#> It has 12 fields
#> Integer64 fields read as strings:  COD_RIP COD_REG COD_PROV COD_CM COD_UTS
```

We calculate and display centroids for each province.

```
coords <- coordinates(italy_provinces)
plot(italy_provinces)
points(coords, col="red", cex=0.8)
```



In the following part, I proceed to integrate the **geographic data**, by province, with the respective data of **population** and presence and distribution of **Consultori** centers in the territory. “Consultori” centers are integrated social and health services with multidisciplinary expertise including free psychological care. They are here taken as a proxy for an accessible mental health care service.

2) Consultori data

In the following section I upload, clean, and aggregate by province the data on Consultori. The dataset below contains the list of healthcare facilities (public and accredited private) providing "family counseling" care services (Consultori familiari) - year 2022, made available by the Ministry of Health, under Italian Open Data Licence.

```
library("readxl")
library("tidyverse")

consultori <- read.csv('data/consultori_C_17_dataset_70_0_upFile.csv', sep = ";", header = TRUE)
consultori <- consultori %>% drop_na(Anno)
c <- consultori %>% group_by(Sigla.provincia) %>% count()
c <- c %>% rename('SIGLA'='Sigla.provincia','consultori'='n')
c[107,][1] = 'NA'
```

Merging geographic and Consultori data by province.

```
ITA_PRO <- merge(italy_provinces, c, by="SIGLA")
```

3) Population data

The excel in use contains population data by province (sheet = 1) and the abbreviations (sigle) of the provinces (sheet = 2) through which to merge the datasets. I pre-process below population data by changing column names and nature of content (e.g., disambiguating the use of comma and period in the Italian versus international system), while also updating some rows (more details on this in the process). The data are obtained from the "HEALTH FOR ALL - ITALY" database system, provided by Istat. Health for all - Italy (henceforth HFA) is a territorial information system on health and healthcare, structured in such a way that it can be queried by the HFA software provided by the World Health Organization adapted to different national systems (from H4A website). It contains about 4000 indicators and enables the composition of downloadable datasets, maps, and analyses. The datasets I will use for population, and then for addiction, unemployment and hospitalization rates, are composed and downloaded from H4A and available in the data folder.

```
pop_prov <- read_excel('data/pop_prov.xlsx',sheet = 1)
prov_dic <- read_excel('data/pop_prov.xlsx',sheet = 2)
```

```
prov_dic <- prov_dic %>% select(Provincia_clean,Sigla) %>% rename(Provincia=Provincia_clean)
pop_prov <- janitor::row_to_names(pop_prov, 1, remove_rows_above = T)
```

```
pop_prov$`Ultimo disponibile`<- gsub(",", ".", pop_prov$`Ultimo disponibile`)
pop_prov[,5:6] <- sapply(pop_prov[,5:6],as.numeric)
pop_prov <- pop_prov %>% select(Provincia, `Ultimo disponibile`,Anno_ultimo)
```

```
pop_prov[pop_prov$Provincia=="Reggio nell'Emilia",][1] <- 'Reggio Emilia'
pop_prov[pop_prov$Provincia=="Reggio di Calabria",][1] <- 'Reggio Calabria'
pop_prov[pop_prov$Provincia=="Medio-Campidano",][1] <- 'Medio Campidano'
```

The integration below is due to the reform of the Sardinian provinces (Legge Regionale n.2 del 4 febbraio 2016), which involved mergers and redistributions (more details below, in the section on regression). In summary, data on the former (now suppressed) provinces are to be merged into either newly created province (South Sardinia) or added to already existing ones (Sassari, Nuoro), since the H4A database does not report the updated provincial division unlike the geographic and Consultori data.

```
pop_prov[pop_prov$Provincia=="Cagliari",][2] <- 421688
pop_prov[pop_prov$Provincia=="Sassari",][2] <- 476516
pop_prov[pop_prov$Provincia=="Nuoro",][2] <- 200376
pop_prov[pop_prov$Provincia=="Oristano",][2] <- 151655
pop_prov <- rbind(pop_prov,list("Provincia" = "Sud Sardegna","Ultimo disponibile" = 337178,"Anno_ultimo

new_row <- list("Provincia" = "Sud Sardegna", "Sigla"="SU")
prov_dic <- rbind(prov_dic,new_row)
prov_dic
#> # A tibble: 112 x 2
#>    Provincia Sigla
#>  * <chr>     <chr>
#>  1 Roma      RM
#>  2 Milano    MI
#>  3 Napoli    NA
#>  4 Torino    TO
#>  5 Palermo   PA
#>  6 Brescia   BS
#>  7 Bari      BA
#>  8 Catania   CT
#>  9 Bergamo   BG
#> 10 Salerno   SA
#> # ... with 102 more rows
```

```
pop_prov_sigle <- left_join(prov_dic,pop_prov,by=("Provincia"="Provincia")) %>% rename(SIGLA=Sigla,Popul
head(pop_prov_sigle)
#> # A tibble: 6 x 4
#>   Provincia SIGLA Population Anno_ultimo
#>   <chr>     <chr>      <dbl>       <dbl>
#> 1 Roma      RM       4224163        2021
#> 2 Milano    MI       3228222        2021
#> 3 Napoli    NA       2987561        2021
#> 4 Torino    TO       2213788        2021
#> 5 Palermo   PA       1208905        2021
#> 6 Brescia   BS       1254433        2021
```

Checking if there are any NAs left.

```
pop_prov_sigle[is.na(pop_prov_sigle$Population)==TRUE,]
#> # A tibble: 1 x 4
#>   Provincia SIGLA Population Anno_ultimo
#>   <chr>     <chr>      <dbl>       <dbl>
#> 1 Totale    <NA>          NA          NA
```

Merging geographic and Consultori data with population data, while checking for NAs.

```
ITA_PRO <- merge(ITA_PRO,pop_prov_sigle,by='SIGLA')
ITA_PRO@data[is.na(ITA_PRO$Population)==TRUE,]
#>  [1] SIGLA       COD_RIP     COD_REG     COD_PROV    COD_CM      COD_UTS
#>  [7] DEN_PROV    DEN_CM      DEN_UTS     TIPO_UTS    SHAPE_AREA  Shape_Ar_1
#> [13] consultori  Provincia   Population  Anno_ultimo
#> <0 rows> (or 0-length row.names)
```

```
ITA_PRO@data[is.na(ITA_PRO@data)==TRUE,]
#>  [1] SIGLA       COD_RIP     COD_REG     COD_PROV    COD_CM      COD_UTS
#>  [7] DEN_PROV    DEN_CM      DEN_UTS     TIPO_UTS    SHAPE_AREA  Shape_Ar_1
#> [13] consultori  Provincia   Population  Anno_ultimo
#> <0 rows> (or 0-length row.names)
```

I here add a new variable that encapsulates the ratio between resident population in the province and the number of Consultori ("pop/cons"), thus how many people a Consultorio alone serves, in the province in question. The variable "pc_disc" is then the discrete transformation of the previous one, for which I chose to use manual breaks following the information given by the distribution of the variable (checked with summary).

```
ITA_PRO@data <- ITA_PRO@data %>% mutate("pop/cons"=Population%/%consultori)

summary(ITA_PRO@data$`pop/cons`)
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>    9148   19602   28684   30885   40723   96048

ITA_PRO@data$pc_disc <- cut(ITA_PRO@data$`pop/cons`, breaks = c(9000, 10000,20000,30000, 40000, 50000,
```

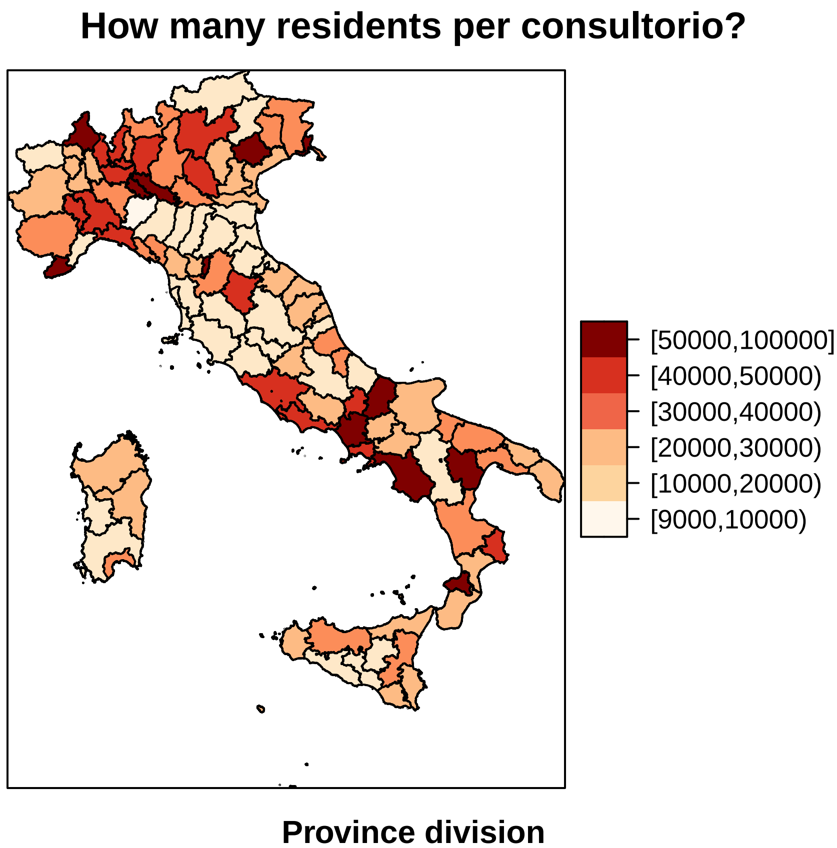
Inspecting the resulting dataframe.

```
View(ITA_PRO@data)
```

Choropleth map

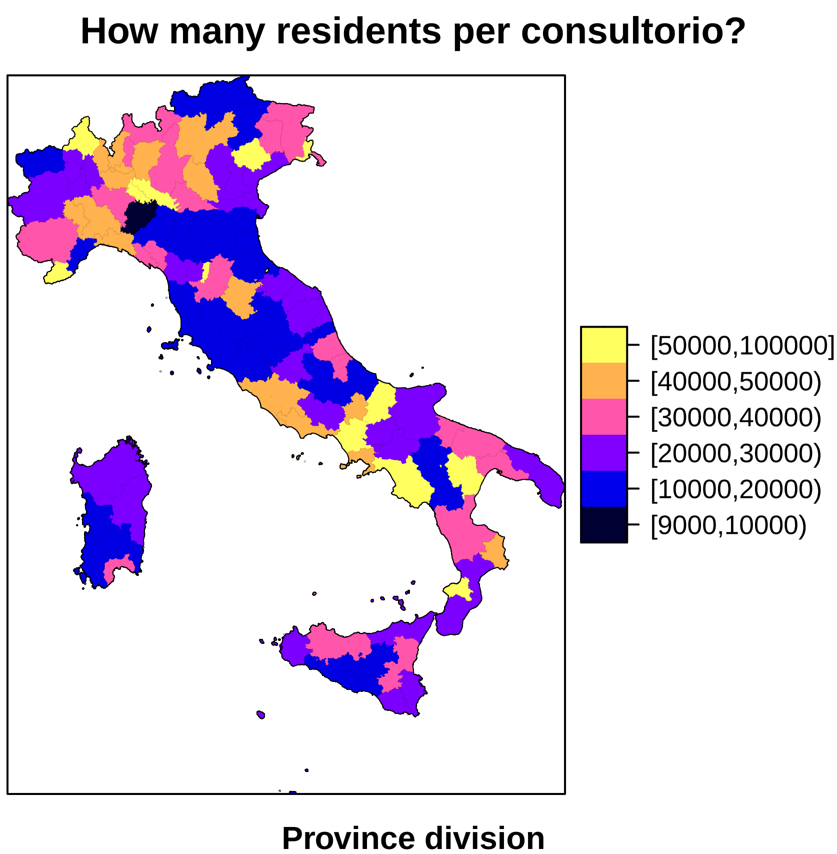
Areas with greater color intensity correspond to areas with less coverage by Consultori services (the more the province tends toward red, the fewer counseling services available; the choice was dictated by the fact that the association "few counseling services" -> "more problematic/alarming situation" could be intuitively intelligible).

```
library(RColorBrewer)
my.palette <- brewer.pal(n = 9, name = "OrRd")
spplot(ITA_PRO,zcol="pc_disc",main="How many residents per consultorio?",sub="Province division", full=
       col.regions = my.palette, sp.layout = list(
        list("sp.polygons", ITA_PRO, first = TRUE, fill = "grey")))
```



Below, I re-plot the same choropleth map yet colored with a different palette, to highlight certain areas which in this way contrast more.

```
spplot(ITA_PRO, "pc_disc", main = "How many residents per consultorio?", sub = "Province division",
    col = "transparent", sp.layout = list(
        list("sp.polygons", ITA_PRO, first = TRUE, fill = "grey")))
```

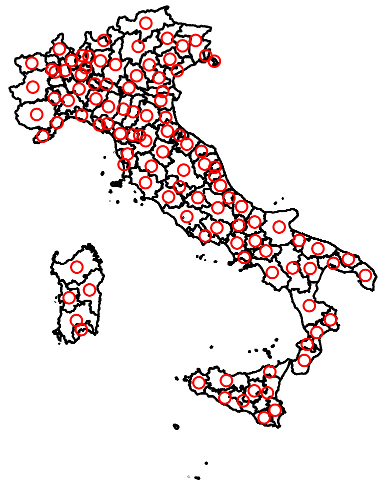


Spatial autocorrelation analysis

Referring to the distances amongst centroids, we need to define the neighbourhood relationships among spatial units to determine who is close to whom, and what may be the mutual influences resulting from spatial proximity (set neighbourhood criterion, compute weights matrices, compute autocorrelation index). I apply the k-nearest neighbours, the critical cut-off neighbourhood, the contiguity-based neighbourhood definition of proximity, to then pick the critical cut-off one and compute spatial weights matrices. Then, I proceed to calculate the Moran's Index of autocorrelation for the variable "population per consultorio", and test its statistical significance. The last part of the section is dedicated to estimating any local autocorrelation patterns of the same variable.

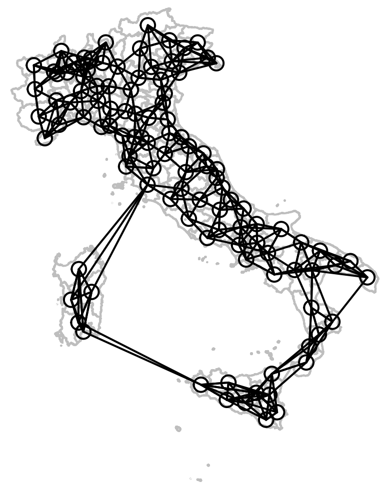
1) Define the proximity/neighbourhood relationships among spatial units Centroids

```
coords <- coordinates(italy_provinces)
plot(italy_provinces)
points(coords, col="red", cex=0.8)
```



k-Nearest neighbours

```
#knn1IT <- knn2nb(knearneigh(coords,k=5,longlat=TRUE))
knn1IT <- knn2nb(knearneigh(coords,k=5))
plot(italy_provinces, border="grey")
plot(knn1IT, coords, add=TRUE)
```



Critical cut-off

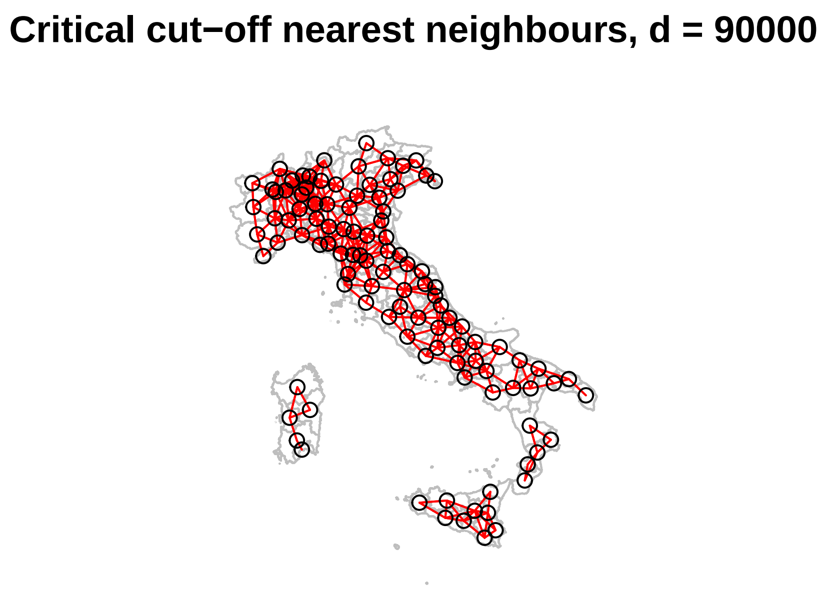
```
# knn1IT <- knn2nb(knearneigh(coords,k=1,longlat=T))
knn1IT <- knn2nb(knearneigh(coords,k=1))

# all.linkedT <- max(unlist(nbdists(knn1IT, coords,longlat=T)))
all.linkedT <- max(unlist(nbdists(knn1IT, coords)))
all.linkedT
#> [1] 75028.63
```

We can try different neighbourhood definitions for different values of the cut-off distance (> 75028.63, which resulted to be the minimum interprovincial distance for each province to have at least one neighbor)

```
dnb75029 <- dnearneigh(coords, 0, 75029); dnb75029
#> Neighbour list object:
#> Number of regions: 107
#> Number of nonzero links: 472
#> Percentage nonzero weights: 4.122631
#> Average number of links: 4.411215
dnb76000 <- dnearneigh(coords, 0, 76000); dnb76000
#> Neighbour list object:
#> Number of regions: 107
#> Number of nonzero links: 488
#> Percentage nonzero weights: 4.262381
#> Average number of links: 4.560748
dnb90000 <- dnearneigh(coords, 0, 90000); dnb90000
```

```
#> Neighbour list object:
#> Number of regions: 107
#> Number of nonzero links: 684
#> Percentage nonzero weights: 5.974321
#> Average number of links: 6.392523
```

```
plot(italy_provinces, border="grey",xlab="",ylab="",xlim=NULL)
title(main="Critical cut-off nearest neighbours, d = 90000")
plot(dnb90000, coords, add=TRUE, col="red")
```



Contiguity-based

```
contnb_q <- poly2nb(italy_provinces, queen=T)
contnb_q
#> Neighbour list object:
#> Number of regions: 107
#> Number of nonzero links: 476
#> Percentage nonzero weights: 4.157568
#> Average number of links: 4.448598
plot(italy_provinces, border="grey")
plot(contnb_q, coords, add=TRUE)
```

2) Defining spatial weights and weights matrices

```
dnb75029.listw <- nb2listw(dnb75029,style="W", zero.policy = TRUE)
dnb76000.listw <- nb2listw(dnb76000,style="W", zero.policy = TRUE)
dnb90000.listw <- nb2listw(dnb90000,style="W", zero.policy = TRUE)
```

Global autocorrelation

Moran's I test of spatial autocorrelation.

I try computing and testing Moran's index with the 3 cut-off distances and (a) with and (b) without randomization, and with (c) Monte-Carlo simulation.

a) Without randomisation

```
moran.test(ITA_PRO$`pop/cons`, dnb75029.listw, randomisation=FALSE, zero.policy = TRUE, na.action = na.
#>
#>  Moran I test under normality
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb75029.listw
#>
#> Moran I statistic standard deviate = 1.3968, p-value = 0.08123
```

```
#> alternative hypothesis: greater
#> sample estimates:
#> Moran I statistic       Expectation          Variance
#>       0.092887766      -0.009433962       0.005365943
moran.test(ITA_PRO$`pop/cons`, dnb76000.listw, randomisation=FALSE, zero.policy = TRUE, na.action = na.e
#>
#>  Moran I test under normality
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb76000.listw
#>
#> Moran I statistic standard deviate = 1.3919, p-value = 0.08198
#> alternative hypothesis: greater
#> sample estimates:
#> Moran I statistic       Expectation          Variance
#>       0.090577900      -0.009433962       0.005162983
moran.test(ITA_PRO$`pop/cons`, dnb90000.listw, randomisation=FALSE, zero.policy = TRUE, na.action = na.e
#>
#>  Moran I test under normality
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb90000.listw
#>
#> Moran I statistic standard deviate = 2.0185, p-value = 0.02177
#> alternative hypothesis: greater
#> sample estimates:
#> Moran I statistic       Expectation          Variance
#>       0.112562157      -0.009433962       0.003652925
```

b) With randomisation

```
moran.test(ITA_PRO$`pop/cons`, dnb75029.listw, randomisation=TRUE, zero.policy = TRUE, na.action = na.ex
#>
#>  Moran I test under randomisation
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb75029.listw
#>
#> Moran I statistic standard deviate = 1.4126, p-value = 0.07888
#> alternative hypothesis: greater
#> sample estimates:
#> Moran I statistic       Expectation          Variance
#>       0.092887766      -0.009433962       0.005246715
moran.test(ITA_PRO$`pop/cons`, dnb76000.listw, randomisation=TRUE, zero.policy = TRUE, na.action = na.ex
#>
#>  Moran I test under randomisation
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb76000.listw
#>
#> Moran I statistic standard deviate = 1.4076, p-value = 0.07962
#> alternative hypothesis: greater
#> sample estimates:
```

```
#> Moran I statistic       Expectation          Variance
#>       0.090577900      -0.009433962       0.005048266
moran.test(ITA_PRO$`pop/cons`, dnb90000.listw, randomisation=TRUE, zero.policy = TRUE, na.action = na.e
#>
#>  Moran I test under randomisation
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb90000.listw
#>
#> Moran I statistic standard deviate = 2.0413, p-value = 0.02061
#> alternative hypothesis: greater
#> sample estimates:
#> Moran I statistic       Expectation          Variance
#>       0.112562157      -0.009433962       0.003571804
```

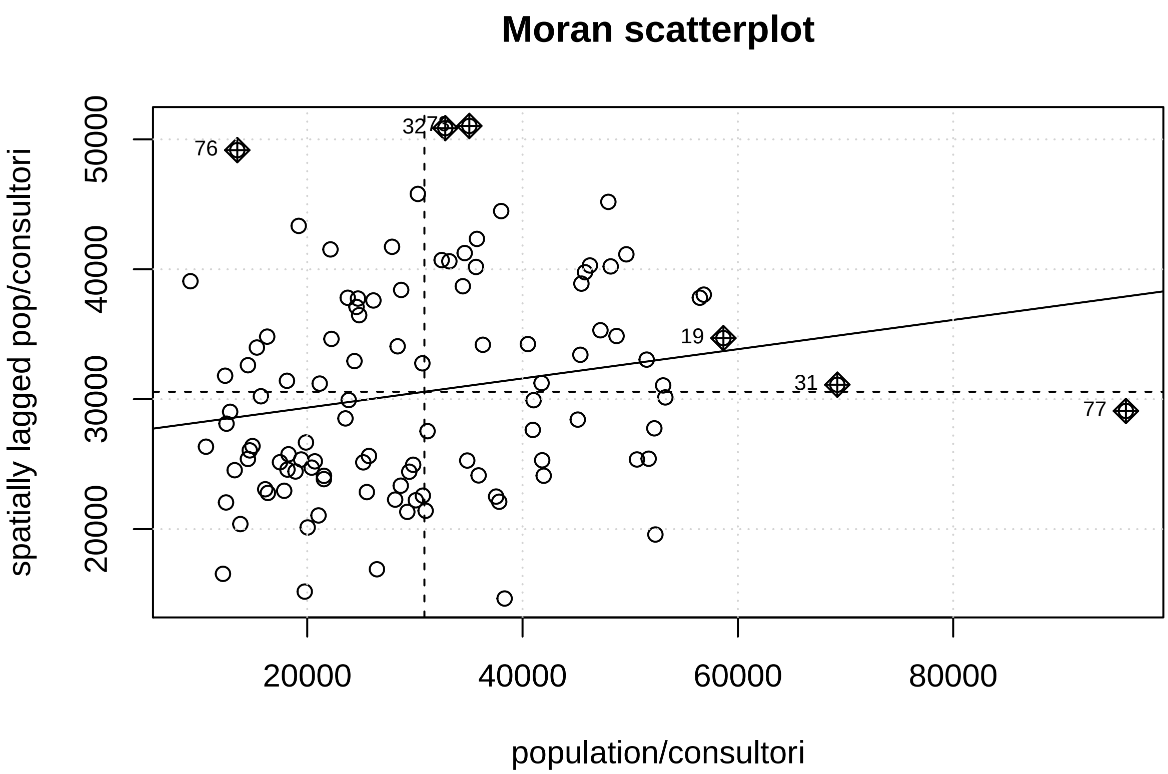
c) With Monte-Carlo simulation

```
moran.mc(ITA_PRO$`pop/cons`, dnb75029.listw, nsim=999,  zero.policy = TRUE, na.action = na.exclude)
#>
#>  Monte-Carlo simulation of Moran I
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb75029.listw
#> number of simulations + 1: 1000
#>
#> statistic = 0.092888, observed rank = 935, p-value = 0.065
#> alternative hypothesis: greater
moran.mc(ITA_PRO$`pop/cons`, dnb76000.listw, nsim=999,  zero.policy = TRUE, na.action = na.exclude)
#>
#>  Monte-Carlo simulation of Moran I
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb76000.listw
#> number of simulations + 1: 1000
#>
#> statistic = 0.090578, observed rank = 913, p-value = 0.087
#> alternative hypothesis: greater
moran.mc(ITA_PRO$`pop/cons`, dnb90000.listw, nsim=999,  zero.policy = TRUE, na.action = na.exclude)
#>
#>  Monte-Carlo simulation of Moran I
#>
#> data:  ITA_PRO$`pop/cons`
#> weights: dnb90000.listw
#> number of simulations + 1: 1000
#>
#> statistic = 0.11256, observed rank = 971, p-value = 0.029
#> alternative hypothesis: greater
```

Results are comparable across the three methods: only with wider neighborhood definition (90000m cut-off distance) the index is significant (p.value < 0.05), although still quantitatively low (e.g. 0.11256, with Monte Carlo simulation and 90000 of cut-off distance). The global autocorrelation value of the population-per-consultorio variable does not seem pronounced.

Local autocorrelation

```
dnb9 <- dnearneigh(coordinates(ITA_PRO), 0, 90000)

dnb9.listw <- nb2listw(dnb9,style="W",zero.policy=TRUE)
mplot <- moran.plot(ITA_PRO$`pop/cons`, listw=dnb9.listw,  zero.policy=TRUE, main="Moran scatterplot", 
grid()
```



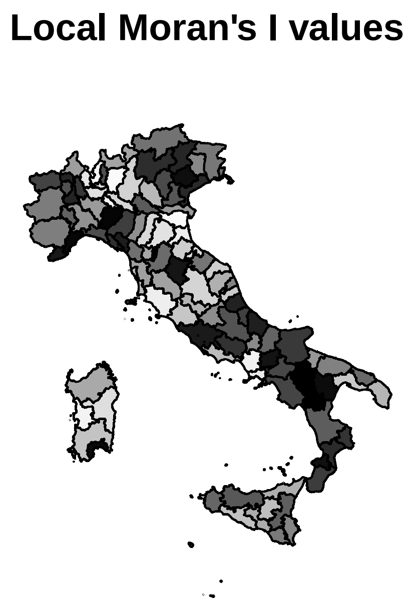

```
hotspot <- as.numeric(row.names(as.data.frame(summary(mplot))))
#> Potentially influential observations of
#>   lm(formula = wx ~ x) :
#>
#>    dfb.1_ dfb.x dffit   cov.r   cook.d hat
#> 19 -0.02   0.02  0.03    1.06_*  0.00   0.04
#> 31  0.10  -0.13 -0.14    1.09_*  0.01   0.07_*
#> 32  0.08   0.03  0.26    0.91_*  0.03   0.01
#> 73  0.04   0.07  0.26    0.91_*  0.03   0.01
#> 76  0.40  -0.31  0.41    0.91_*  0.08   0.02
#> 77  0.48  -0.59 -0.61_*  1.23_*  0.18   0.19_*
```

```
ITA_PRO@data$wx <- lag.listw(dnb9.listw, ITA_PRO$`pop/cons`)
```

```r
ITA_PRO@data$quadrant <- rep("None", length(ITA_PRO$`pop/cons`))
for(i in 1:length(hotspot))  {
  if (ITA_PRO$`pop/cons`[hotspot[i]]>mean(ITA_PRO$`pop/cons`) & ITA_PRO@data$wx[hotspot[i]]> mean(ITA_PI
        ITA_PRO@data$quadrant[hotspot[i]] <- "HH"
  if (ITA_PRO$`pop/cons`[hotspot[i]]>mean(ITA_PRO$`pop/cons`) & ITA_PRO@data$wx[hotspot[i]]< mean(ITA_PI
        ITA_PRO@data$quadrant[hotspot[i]] <- "HL"
  if (ITA_PRO$`pop/cons`[hotspot[i]]<mean(ITA_PRO$`pop/cons`) & ITA_PRO@data$wx[hotspot[i]]<mean(ITA_PR(
        ITA_PRO@data$quadrant[hotspot[i]] <- "LL"
  if (ITA_PRO$`pop/cons`[hotspot[i]]<mean(ITA_PRO$`pop/cons`) & ITA_PRO@data$wx[hotspot[i]]>mean(ITA_PR(
        ITA_PRO@data$quadrant[hotspot[i]] <- "LH"
  }
table(ITA_PRO@data$quadrant)
#>
#>   HH   HL   LH None
#>    4    1    1  101
```

```r
ITA_PRO@data$colours[ITA_PRO@data$quadrant=="None"] <- "white"
ITA_PRO@data$colours[ITA_PRO@data$quadrant=="HH"] <- "black"
ITA_PRO@data$colours[ITA_PRO@data$quadrant=="LL"] <- gray(0.9)
ITA_PRO@data$colours[ITA_PRO@data$quadrant=="LH"] <- gray(0.4)
ITA_PRO@data$colours[ITA_PRO@data$quadrant=="HL"] <- gray(0.7)
plot(ITA_PRO, col=ITA_PRO@data$colours)
legend(x=-10, y=73, legend=c("None", "Low-Low", "High-Low", "Low-High", "High-High"),
                      fill=c("white", gray(0.9), gray(0.7), gray(0.4),
                      "black"), bty="n", cex=0.8)
title(main="Regions with influence")
```


```r
lmI <- localmoran(ITA_PRO$`pop/cons`, dnb9.listw)

brks <- sort(as.numeric(lmI[,1]))
colours <- grey((0:length(lmI[,1]))/length(lmI[,1]))
plot(ITA_PRO, col=colours[findInterval(lmI[,1], brks, all.inside=TRUE)])
title(main="Local Moran's I values")
```




```r
pval <- as.numeric(lmI[,5])
ITA_PRO@data$colpval[pval>0.05] <- "white"
ITA_PRO@data$colpval[pval<=0.05 & pval>0.01] <- gray(0.9)
ITA_PRO@data$colpval[pval<=0.01 & pval>0.001] <- gray(0.7)
ITA_PRO@data$colpval[pval<=0.001 & pval>0.0001] <- gray(0.4)
ITA_PRO@data$colpval[pval<=0.0001] <- "black"
```

```r
plot(ITA_PRO, col=ITA_PRO@data$colpval)
legend(x=-10, y=73, legend=c("Not significant",
       "p-value = 0.05", "p-value = 0.01", "p-value = 0.001",
       "p-value = 0.0001"), fill=c("white", gray(0.9), gray(0.7),
       gray(0.4), "black"), bty="n", cex=0.8)
title(main="Local Moran's I significance map")
```

Local Moran's I significance map

Very few regions seem to have an influence (coefficients dimension), and the significance analysis (p-values) does not show many positive results. A result that appears to be confirmed by the distribution of points in the Moran scatterplot and by the subsequent analysis, where points located in quadrants identifying local patterns of strong positive or negative spatial autocorrelation are few.

In summary, no relevant (index quantification) and significant (index testing) autocorrelation, neither global nor local, appears to exist for the variable under examination, with few exceptions.

Multivariate spatial regression model

For the regression part, I will use a dataset (again) taken from the ISTAT HFA database system. Here, as in the previous parts, a similar pre-processing procedure has to be done (e.g., regarding the reorganization of Sardinian provinces, data on the former provinces of Carbonia-Iglesias and Medio Campidano are to be merged into the newly created province of South Sardinia, while information on the former provinces of Olbia-Tempio and Ogliastra are to be merged into the province of Sassari and Nuoro, respectively*), since the H4A database does not report the updated provincial division, unlike the geographic and Consultori data.

Regarding the definition of the regression model and the choice of the dependent and independent variables, I use here the "mortality rate from mental disorders" as Y-varable, and the unemployment rate, the hospitalization rate and the resident/consultori ratio as Xs-variables, to relate and investigate the correlation between mortality rate from mental disorders and some variables related to health services (population-to-consultorio ratio, hospitalization rate), controlling for a proxy of socioeconomic conditions (unemployment rate). The definition of the model from an interpretative point of view, has several limitations, but it can be used as a starting point for other developments (eg using different variables with different degree of approximation, refining the regression model).

```
health_prov <- read_excel('data/regression_data.xlsx', sheet = 2)
health_prov <- janitor::row_to_names(health_prov, 1, remove_rows_above = T)
health_prov <- health_prov %>% rename(dp_rate=`Tasso mortalità disturbi psichici M+F`,
                                      suicide_rate=`Tasso mortalità sucidio,autolesione M+F`,
                                      unempl_rate = `Tasso disoccupazione 15+ M+F`,
                                      ospedalizz = `Tasso ospedalizzazione`)

health_prov$dp_rate <- gsub(",", ".", health_prov$dp_rate)
health_prov$suicide_rate <- gsub(",", ".", health_prov$suicide_rate)
health_prov$unempl_rate <- gsub(",", ".", health_prov$unempl_rate)
health_prov$ospedalizz <- gsub(",", ".", health_prov$ospedalizz)
health_prov[,4:7] <- sapply(health_prov[,4:7],as.numeric)

health_prov <- health_prov %>% select(Provincia, dp_rate, suicide_rate, unempl_rate, ospedalizz)
health_prov
#> # A tibble: 110 x 5
#>    Provincia            dp_rate suicide_rate unempl_rate ospedalizz
#>    <chr>                  <dbl>        <dbl>       <dbl>      <dbl>
#>  1 Torino                  5.48         0.74        8.26       80.4
#>  2 Vercelli                7.47         0.88        8.23       70.3
#>  3 Biella                  5.15         1.43        6.02       79.0
#>  4 Verbano-Cusio-Ossola    4.21         0.83        5.81       97.4
#>  5 Novara                  2.71         0.55        7.68       95.9
#>  6 Cuneo                   6.66         0.99        4.62       78.5
#>  7 Asti                    6.82         0.75        7.45       64.9
#>  8 Alessandria             6.3          0.6         6.87       94.4
#>  9 Aosta                   9.97         1.28        7.26       98.6
#> 10 Varese                  4.49         0.61        6.56       73.2
#> # ... with 100 more rows
```

```
health_prov[health_prov$Provincia=="Reggio nell'Emilia",][1] <- 'Reggio Emilia'
health_prov[health_prov$Provincia=="Reggio di Calabria",][1] <- 'Reggio Calabria'
health_prov[health_prov$Provincia=="Medio-Campidano",][1] <- 'Medio Campidano'
```

*For OT and Ogliastra values, ithe process of merging data into Sassari and Nuoro values would require some reasoning around weights and modes, omitted here for reasons of time and space. For SU values, I simply take the mean of the Carbonia and Medio Campidano ex provinces.

```
# for SU values, I simply take the mean of the Carbonia and Medio Campidano ex provinces
health_prov <- rbind(health_prov,list("Provincia" = "Sud Sardegna", "dp_rate" = (5.25+5.29)/2 ,"suicide_
```

```
health_prov <- left_join(prov_dic,health_prov,by=("Provincia"="Provincia")) %>% rename(SIGLA=Sigla)
```

Merging the resulting dataset to integrate the obtained variables with previous ones (while also checking for NAs).

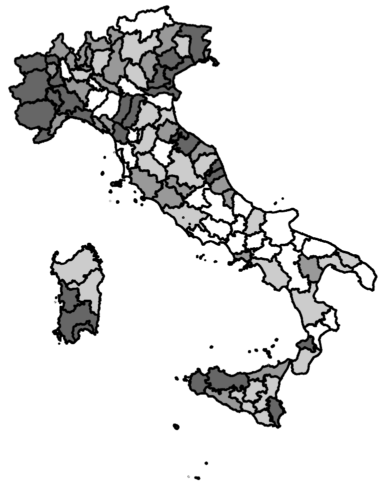
```
ITA_PRO <- merge(ITA_PRO,health_prov,by='SIGLA')
ITA_PRO@data[is.na(ITA_PRO$Population)==TRUE,]
#>  [1] SIGLA        COD_RIP      COD_REG      COD_PROV     COD_CM
#>  [6] COD_UTS      DEN_PROV     DEN_CM       DEN_UTS      TIPO_UTS
#> [11] SHAPE_AREA   Shape_Ar_1   consultori   Provincia.x  Population
#> [16] Anno_ultimo  pop/cons     pc_disc      wx           quadrant
```

```
#> [21] colours      colpval      Provincia.y  dp_rate      suicide_rate
#> [26] unempl_rate  ospedalizz
#> <0 rows> (or 0-length row.names)
```

The Moran's I test of spatial autocorrelation in OLS residuals

```
LinearSolow <- lm(dp_rate ~ unempl_rate + ospedalizz + `pop/cons`, ITA_PRO)
summary(LinearSolow)
#>
#> Call:
#> lm(formula = dp_rate ~ unempl_rate + ospedalizz + `pop/cons`,
#>     data = ITA_PRO)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -2.1873 -0.8900 -0.2725  0.4979  4.5902
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  6.528e+00  8.020e-01   8.139 9.64e-13 ***
#> unempl_rate -1.143e-01  2.563e-02  -4.461 2.09e-05 ***
#> ospedalizz   2.835e-04  6.485e-03   0.044  0.96522
#> `pop/cons`  -2.797e-05  8.566e-06  -3.265  0.00149 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 1.266 on 103 degrees of freedom
#> Multiple R-squared:  0.2504, Adjusted R-squared:  0.2285
#> F-statistic: 11.47 on 3 and 103 DF,  p-value: 1.503e-06
```

```
studres <- rstudent(LinearSolow)
resdistr <- quantile(studres)
colours <- grey((length(resdistr):2)/length(resdistr))
plot(ITA_PRO, col=colours[findInterval(studres, resdistr, all.inside=TRUE)])
```



```
lm.morantest(LinearSolow,dnb75029.listw,resfun=rstudent)
#>
#>  Global Moran I for regression residuals
#>
#> data:
#> model: lm(formula = dp_rate ~ unempl_rate + ospedalizz + `pop/cons`,
#> data = ITA_PRO)
#> weights: dnb75029.listw
#>
#> Moran I statistic standard deviate = 4.2849, p-value = 9.14e-06
#> alternative hypothesis: greater
#> sample estimates:
#> Observed Moran I      Expectation         Variance
#>      0.291735030     -0.017632670      0.005212687
lm.morantest(LinearSolow,dnb70600.listw,resfun=rstudent)
#> Error in lm.morantest(LinearSolow, dnb70600.listw, resfun = rstudent): oggetto "dnb70600.listw" non
lm.morantest(LinearSolow,dnb90000.listw,resfun=rstudent)
#>
#>  Global Moran I for regression residuals
#>
#> data:
#> model: lm(formula = dp_rate ~ unempl_rate + ospedalizz + `pop/cons`,
#> data = ITA_PRO)
#> weights: dnb90000.listw
#>
#> Moran I statistic standard deviate = 4.2846, p-value = 9.154e-06
```

```
#> alternative hypothesis: greater
#> sample estimates:
#> Observed Moran I      Expectation         Variance
#>      0.234786725     -0.017902477      0.003478204
```

The coefficients of unempl_rate and pop/cons appear to be significantly different from zero, and the Moran I for the regression residuals, whatever the matrix of weights according to different cut-off distances, turns out to be between 0.23 and 0.29 with p-values indicating significance. In order to develop interpretations and draw meaningful conclusions, a closer understanding of the composition of the indices and variables, as well as a greater understanding of interpretative social models of the phenomena under consideration, would be necessary.

In this regard, it would be interesting and useful to deepen the interpretation and develop an analysis of the different possible spatial regression models in case of violation of the assumption of non-autocorrelation of the variables, which could not be developed here for reasons of time and space.