Spatial multivariate methods

in practice

Stéphane Dray

2022-02-28

Spatial data in R

Several packages allows to deal with spatial objects in R.

- The package `sp` provides classes to manage spatial objects with or without data
- Spatial weighting matrices can be defined and managed with `spdep`
- Spatial multivariate methods are implemented in `adespatial`
- Spatial objects can be used as arguments in `adegraphics` functions to create maps

```r
library(ade4)
library(adegraphics)
library(adespatial)
library(spdep)
```

`adespatial`

A package for spatial multivariate analysis interfacing `ade4` and `spdep`

More details in the vignette:

https://cran.r-project.org/web/packages/adespatial/vignettes/tutorial.html

or

```
vignette("tutorial", package = "adespatial")
```

Spatial data

- Classes are provide in R to deal with spatial data (raster, polygons, lines, points)
- We will use implementations in `sp` package but it will be replaced by `sf` in the future
- Import/export functions allows to interface with GIS software
- Spatial proximities are managed by `spdep` functionalities

Spatial weighting matrix

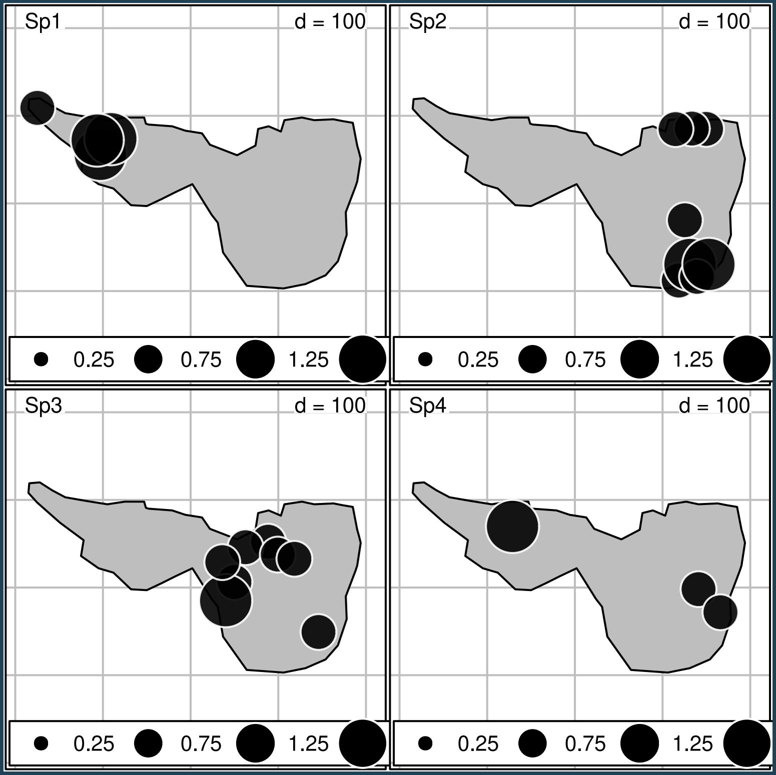
A two-step procedure:

- define a binary neighboring matrix (`nb` object)
- consider optional non-binary weights and standardization to define a spatial weighting matrix (`listw` object)

```
data(mafragh)
lw <- nb2listw(mafragh$nb)
s.label(mafragh$xy, plabel.cex = 0, Sp = mafragh$Spatial.contour,
    nb = mafragh$nb)
```

Spatial mapping

```
s.value(mafragh$xy, mafragh$flo[, 1:4], method = "size",
    symbol = "circle", Sp = mafragh$Spatial.contour)
```



Spatial autocorrelation

Compute Moran's index for the five first species:

```r
moran.randtest(mafragh$flo[, 1:5], lw)
```

```
## class: krandtest lightkrandtest
## Monte-Carlo tests
## Call: moran.randtest(x = mafragh$flo[, 1:5], listw = lw)
##
## Number of tests:   5
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test         Obs    Std.Obs   Alter Pvalue
## 1  Sp1  0.33419967  6.2600537 greater  0.001
## 2  Sp2  0.38260466  6.4139654 greater  0.001
## 3  Sp3  0.16729450  3.1375972 greater  0.005
## 4  Sp4 -0.03331651 -0.5118008 greater  0.935
## 5  Sp5  0.06837558  1.5826055 greater  0.081
```

Multispati

Perform the analysis

```
pca_veg <- dudi.pca(mafragh$flo, scale = FALSE, scannf = FALSE)
ms_veg <- multispati(pca_veg, lw, scannf = FALSE)
```

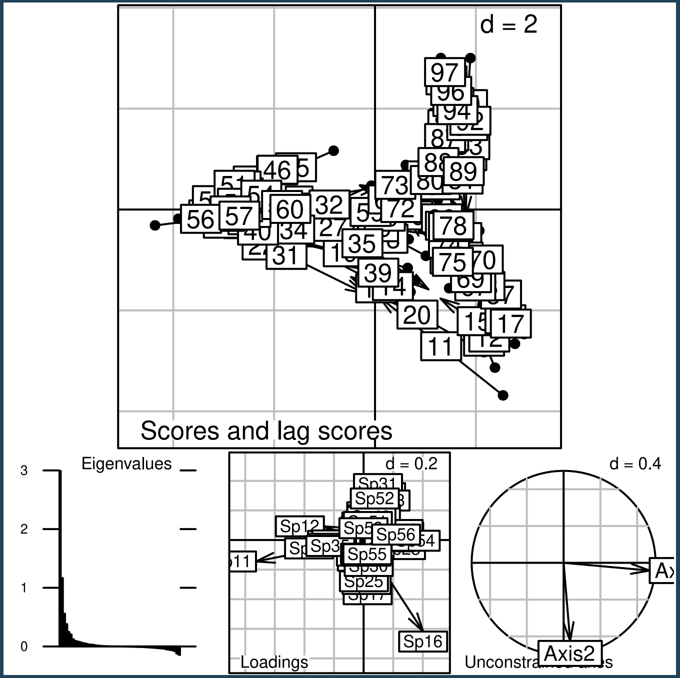
Have a look to the summary

```
summary(ms_veg)
```

```
##
## Multivariate Spatial Analysis
## Call: multispati(dudi = pca_veg, listw = lw, scannf = FALSE)
##
## Scores from the initial duality diagram:
##          var      cum     ratio     moran
## RS1 5.331174 5.331174 0.2834660 0.4947964
## RS2 1.972986 7.304159 0.3883725 0.4435555
##
## Multispati eigenvalues decomposition:
##          eig      var     moran
## CS1 2.992293 4.862003 0.6154445
## CS2 1.164390 1.885904 0.6174172
```

Plot the results

```
g1 <- plot(ms_veg)
```

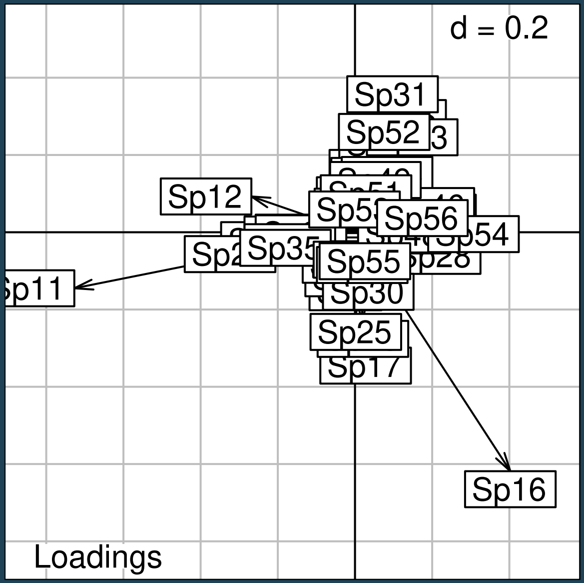



```
names(g1)
```

```
## [1] "row"       "eig"       "loadings" "Xax"
```

Loadings for variables

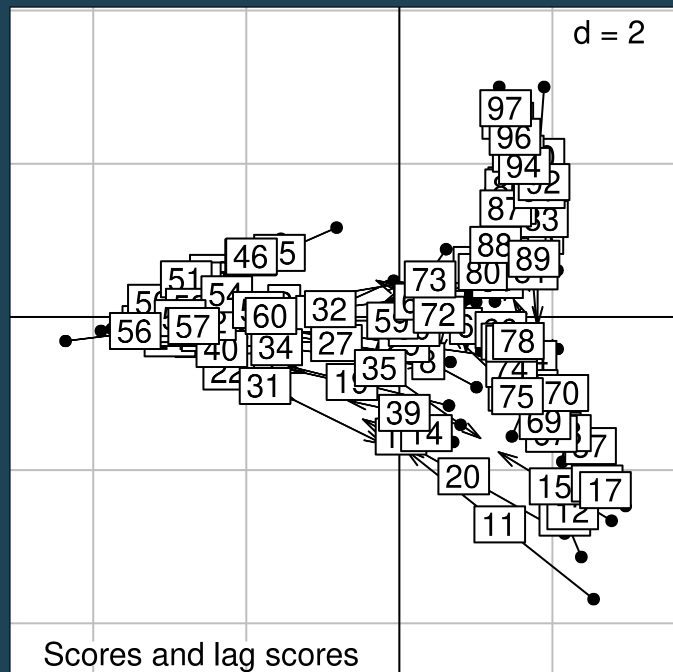
```
g1$loadings
```



$\mathbf{A}^*$: coefficients (loadings) for the variables of $\mathbf{X}$ (`ms_veg$c1`)

Scores for individuals and lagged scores

```
g1$row
```

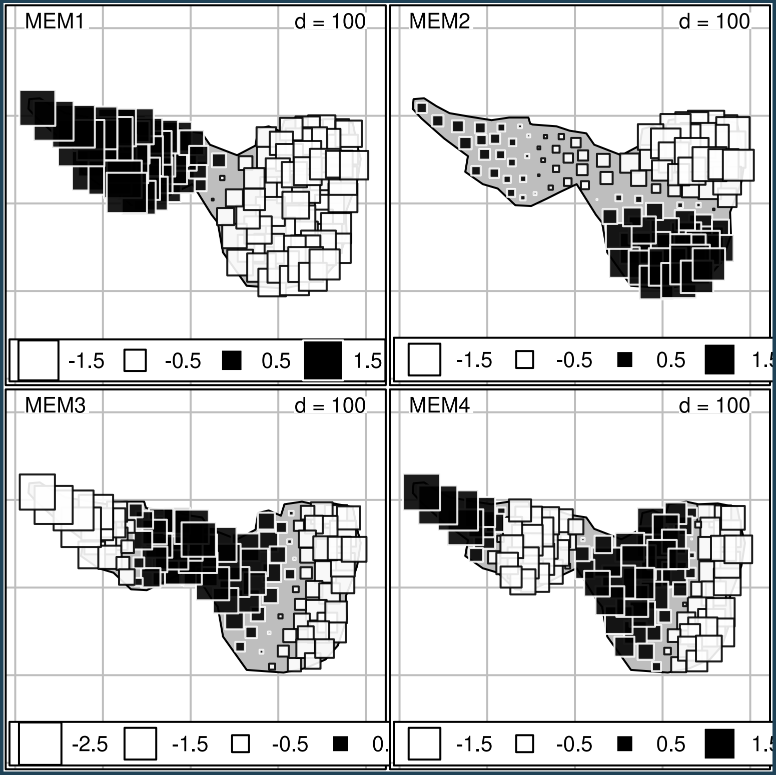


mem_sub

- $\mathbf{XQA}$: scores of individuals (`ms_veg$li`)
- $\mathbf{WXQA}$: lagged scores (`ms_veg$ls`)

Moran's Eigenvectors Maps

```r
me <- mem(lw)
s.value(mafragh$xy, me[, 1:4], Sp = mafragh$Spatial.contour)
```



Variable selection

```
mem_sub <- mem.select(mafragh$flo, lw)
```

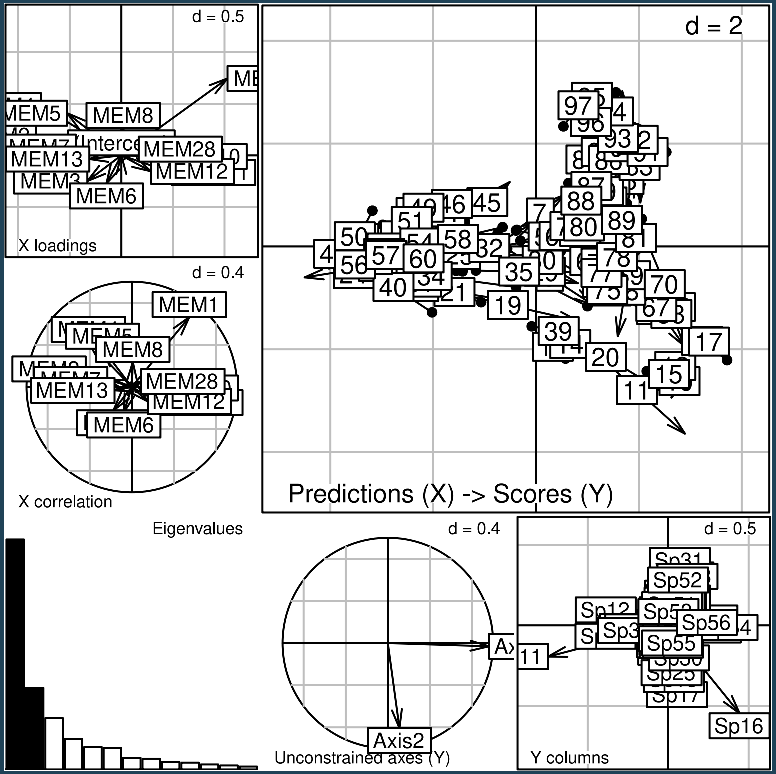
```
## Procedure stopped (alpha criteria): pvalue for variable 14 is 0.051000 (>
```

```
mem_sub$summary
```

```
##    variables order         R2      R2Cum  AdjR2Cum pvalue
## 1       MEM1     1 0.09111155 0.09111155 0.0815443  0.001
## 2       MEM4     4 0.05227241 0.14338396 0.1251581  0.001
## 3       MEM2     2 0.05057350 0.19395745 0.1679561  0.001
## 4       MEM5     5 0.03852474 0.23248219 0.1991119  0.001
## 5      MEM11    11 0.02368647 0.25616866 0.2152988  0.005
## 6       MEM3     3 0.02339942 0.27956808 0.2315393  0.003
## 7       MEM7     7 0.02291725 0.30248533 0.2476246  0.002
## 8      MEM10    10 0.02175612 0.32424145 0.2628089  0.003
## 9       MEM6     6 0.01692771 0.34116916 0.2730142  0.014
## 10      MEM8     8 0.01560389 0.35677306 0.2819792  0.016
## 11     MEM13    13 0.01530752 0.37208057 0.2908204  0.026
## 12     MEM12    12 0.01412745 0.38620803 0.2985235  0.029
## 13     MEM28    28 0.01390115 0.40010918 0.3061504  0.038
```

Spatial RDA

```r
rda_spat <- pcaiv(pca_veg, mem_sub$MEM.select, scannf = FALSE)
plot(rda_spat)
```



Variation partitioning

```
varipart(pca_veg$tab, mem_sub$MEM.select, mafragh$env)
```

```
## Variation Partitioning
## class: varipart list
##
## Test of fractions:
## class: krandtest lightkrandtest
## Monte-Carlo tests
## Call: varipart(Y = pca_veg$tab, X = mem_sub$MEM.select, W = mafragh$env)
##
## Number of tests:   3
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test       Obs   Std.Obs   Alter Pvalue
## 1   ab 0.4001092 16.692070 greater  0.001
## 2   bc 0.2366554  7.976584 greater  0.001
## 3  abc 0.4851531 11.827304 greater  0.001
##
##
## Individual fractions:
##          a          b          c          d
## 0.24849774 0.15161144 0.08504395 0.51484688
```

Your turn

1. Create a Rmd or a R file
2. Load the `irishdata` data set from `ade4`
3. See `?irishdata` for details
4. Perform a PCA on the data table `irishdata$tab`
5. Define a spatial weighting matrix from `irishdata$Spatial` using `poly2nb`
6. Compute Moran's index of autocorrelation for the PCA scores
7. Perform MULTISPATI analysis
8. Interpret