Overview of the spnet package

Emmanuel Rousseaux, Gilbert Ritschard

2014.07.22

Contents

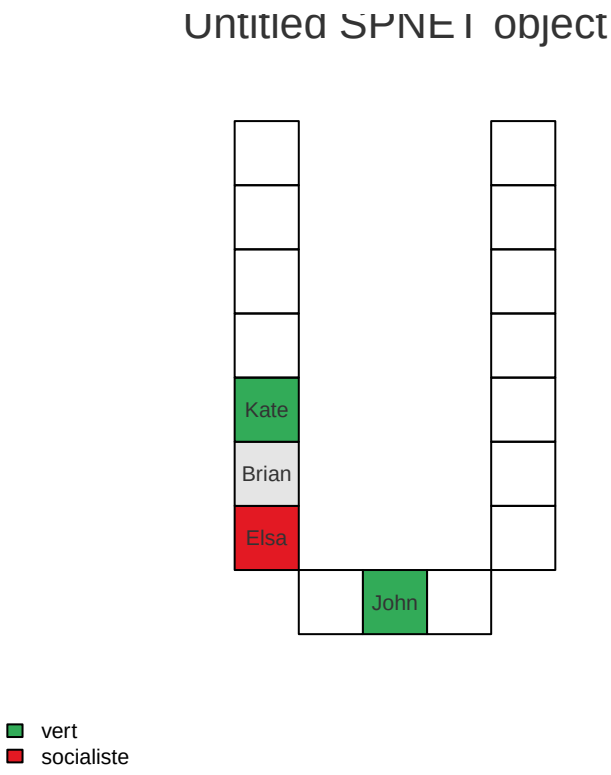
Introduction

The `spnet` package offers methods for dealing with spacial social networks. It allows to plot networks for which actors have a specific location on a map (participants in a political debate, cities, etc.). `SpatialPolygons` objects from the `sp` package are supported.

Main functionalities / Gallery

The `spnet.example.basic` function provides a working example involving basic functionnalities of the `spnet` package.

```
net1 <- spnet.example.basic()
plot(net1)
```



Usage

Create a SpatialNetwork object

Setting labels

Setting colors

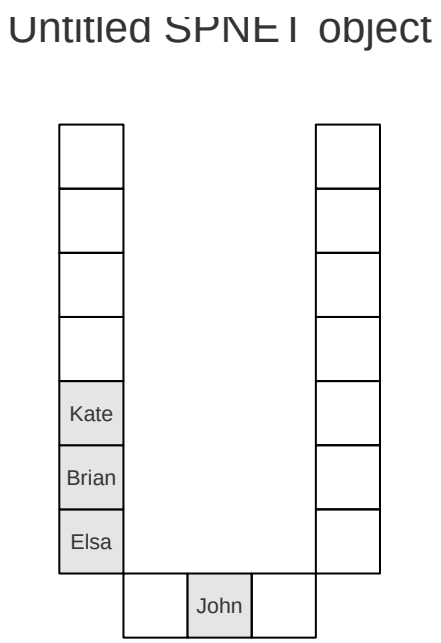
To set colors you basically need:

- A categorical variable affecting each node to a class

- a legend of color specifying the color to use for each class

Here is a practical example. First, we create an basic `spnet` object containing a map.

```
net1 <- spnet.example.basic.map()
plot(net1)
```



This example contains the following data:

```
data.frame(net1)
```

```
##    NODE POSITION
## 1  John        2
## 2  Elsa        4
## 3 Brian        6
## 4  Kate        8
```

We add a categorical variable affecting each node to a class:

```
net1$parti <- c('vert', 'socialiste', 'autre', 'vert')
```

Data are now:

```
data.frame(net1)
```

```
##    NODE POSITION      parti
## 1  John        2       vert
## 2  Elsa        4 socialiste
## 3 Brian        6      autre
## 4  Kate        8       vert
```

Then we specify we want to use the variable `parti` to colorize the map:

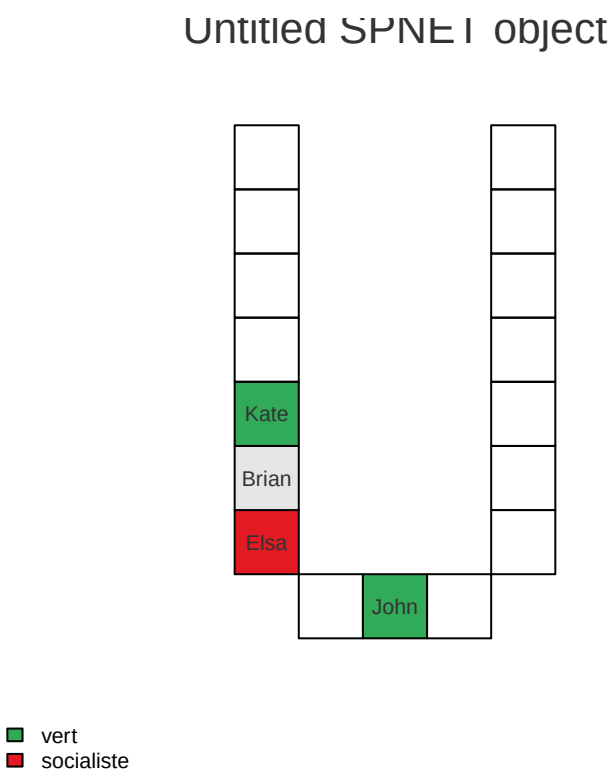
```
spnet.color.variable(net1) <- "parti"
```

Finally we specify the colors to use:

```
spnet.color.legend(net1) <- c('vert' = "#32AB58", 'socialiste' = "#E31923")
```

Now the `plot` function is able to colorize the graphic:

```
plot(net1)
```



Dealing with a quantitative covariate: rendering individual barplots

We may need to render a quantitative attribute related to each node of the network. To that purpose we provide a simple barplot tool. This section details how to use it. We start with a fresh `spnet` object and equipe it with a map.

```
ex.bp <- spnet.example.basic.map()
```

A fresh `spnet` object contains the following default barplot settings:

```
spnet.barplot.list(ex.bp)
```

```
## $variable
## [1] ""
##
## $bound.lower
## [1] -0.5 -0.5
##
## $bound.upper
## [1]  0.5 -0.5
##
## $fgcolor
## [1] "#666666"
##
## $bgcolor
## [1] "#eeeeee"
##
## $width
## [1] 8
```

The first point is to

```
ex.bp$content <- c(0.1,0.3,0.5,0.9)
ex.bp
```

```
## This is a valid 'SpatialNetwork' object.
##
## - Data: (first rows)
##
##    NODE POSITION content
## 1  John        2     0.1
## 2  Elsa        4     0.3
## 3 Brian        6     0.5
## 4  Kate        8     0.9
##
## - Map:
##     Length: 17
##
## - Plotting options:
```

```
spnet.barplot.variable(ex.bp) <- "content"
spnet.barplot.list(ex.bp)
```

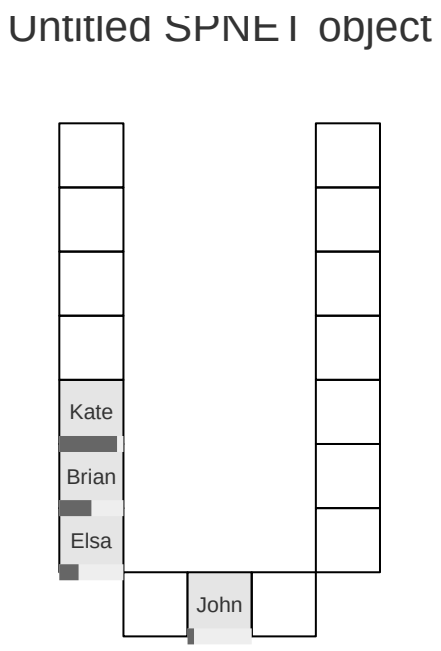
```
## $variable
## [1] "content"
##
## $bound.lower
## [1] -0.5 -0.5
##
## $bound.upper
## [1]  0.5 -0.5
##
## $fgcolor
## [1] "#666666"
```

```
##
## $bgcolor
## [1] "#eeeeee"
##
## $width
## [1] 8
```

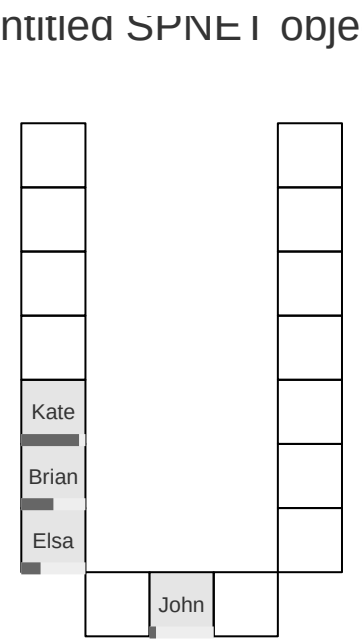
```
ex.bp
```

```
## This is a valid 'SpatialNetwork' object.
##
## - Data: (first rows)
##
##    NODE POSITION content
## 1  John        2     0.1
## 2  Elsa        4     0.3
## 3 Brian        6     0.5
## 4  Kate        8     0.9
##
## - Map:
##     Length: 17
##
## - Plotting options:
##     Variable used to draw barplots: 'content'
```

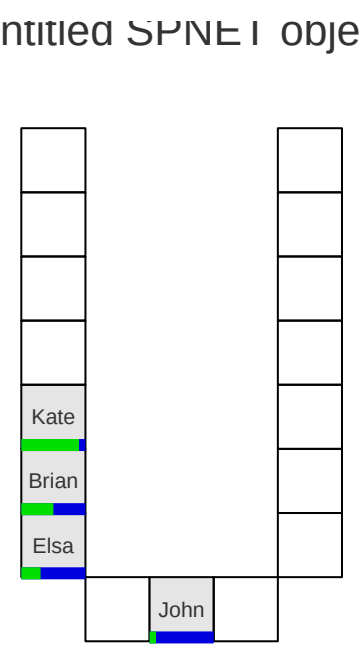
```
plot(ex.bp)
```




```
spnet.barplot.bound.lower(ex.bp) <- c(-0.5,-0.44)
spnet.barplot.bound.upper(ex.bp) <- c(0.5,-0.44)
spnet.barplot.width(ex.bp) <- 6
plot(ex.bp)
```




```
spnet.barplot.fgcolor(ex.bp) <- "#00dd00"
spnet.barplot.bgcolor(ex.bp) <- "#0000dd"
plot(ex.bp)
```



Maps

SpatialPolygons maps

Rooms

The easiest way to create a room to represent a debate is with the `room.create.grid` function. Here is an example of use:

```
col <- 5
row <- 6
m <- matrix(rep(-1, col*row), nrow = row)
m[1,2:4] <- 0
m[3,c(1,5)] <- 0
m[4,c(1,5)] <- 0
m[5,c(1,5)] <- 0
m[6,c(1,5)] <- 0
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   -1    0    0    0   -1
## [2,]   -1   -1   -1   -1   -1
## [3,]    0   -1   -1   -1    0
## [4,]    0   -1   -1   -1    0
## [5,]    0   -1   -1   -1    0
## [6,]    0   -1   -1   -1    0
```

```
room1 <- room.create.grid(m, seat.width=2, seat.height=1)
spnet.map.plot.position(room1)
```

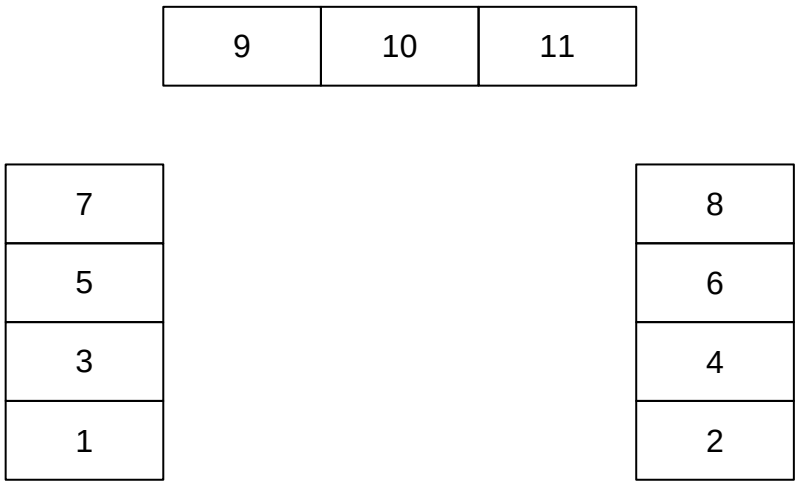


Figure 1: A simple room with table in invered 'U' form