

Advanced Systems II

1. OBJECTIUS

Introduir la creació de mapes amb R i concretament amb ggplot2. Els mapes són una eina de sistemes avançats de visualització. Ja hem vist alguns exemples d'ús, i ara crearem algun mapa senzill

2. PAQUETS A INSTAL·LAR (per part 3 i 4)

Hi ha molts paquets R disponibles a CRAN (la xarxa d'arxius completa R, que és el dipòsit principal dels paquets R). Per aquest exercici i el pròxim seminari, haureu d'instal·lar alguns paquets addicionals

Aquests són paquets que necessitareu (a més de ggplot2 i dplyr), però que probablement ja teniu. (No us molesteu a instal·lar-los si ja els teniu):

```
> install.packages (c("devtools", "stringr")) #cliqueu no si us demana restart R
```

També necessitareu, alguns paquets de mapes estàndard:

```
> install.packages (c ("maps", "mapdata", "ggmap"))
```

3. COM FEM MAPES

Normalment podem descompondre la creació d'un mapa en dos “problemes”: Utilitzar dades d'una certa font per dibuixar el mapa (background) i afegir al mapa informació de les metadades d'una altra font.

Tractarem majorment amb [classes sp](#) per dades espacials (per defecte) o directament [classes sf](#) (simple features).

Ara que tenim els paquets instal·lats, carreguem les llibreries que no tinguem carregades:

```
> library(ggplot2)
> library(ggmap)
> library(maps)
> library(mapdata)
```

- El paquet de mapes (maps) conté una gran quantitat de contorns de continents, països, estats i comtats que han estat en R durant molt de temps (pot ser doncs, que no estigui totalment actualitzat o que no estigui 100% correcte i això, ho haurem de tenir en compte). També conté esquemes i punts de ciutats, etc. Exemples: nz, State, world, etc.
- El paquet *mapdata* conté uns quants esquemes més, amb una resolució més alta.
- Tot i que el paquet “maps”, inclou una funció de traçat, optarem per utilitzar ggplot2 per representar els mapes.

Recordeu que **ggplot2** funciona en marcs de dades (*dataframes*). Per tant, necessitem passar les dades dels mapes a un format *dataframe* que **ggplot2** entengui.

ggplot2 proporciona la funció `map_data()`.

Penseu-ho com una funció que converteix una sèrie de punts al llarg d'un esquema en un marc de dades d'aquests punts.

Sintaxi: `map_data ("nom")` on "nom" és una cadena amb el nom d'un mapa al paquet de mapes o dades de mapes

Exemple:

```
> usa <- map_data("usa")
> str(usa)
'data.frame':  7243 obs. of  6 variables:
 $ long      : num  -101 -101 -101 -101 -101 ...
 $ lat       : num   29.7 29.7 29.7 29.6 29.6 ...
 $ group     : num    1 1 1 1 1 1 1 1 1 ...
 $ order     : int    1 2 3 4 5 6 7 8 9 10 ...
 $ region    : chr   "main" "main" "main" "main" ...
 $ subregion : chr    NA NA NA NA ...
>
```

L'estructura d'aquests *dataframes* són senzilles:

- **long** - és la longitud. A l'oest del meridià primer aquestes són negatives.
- **lat** - és la latitud.
- **order** - només mostra en quin ordre **ggplot** ha de "connectar els punts"
- **region & subregion** - indiquen quina regió o subregió envolta un conjunt de punts.
- **group** - és molt important. Les funcions de **ggplot2** poden adoptar un argument de grup que controla (entre altres coses) si els punts adjacents haurien d'estar connectats per línies. Si formen el mateix grup, es connecten, però si estan en grups diferents, no ho fan. Essencialment, tenir punts en diferents grups significa que **ggplot** "aixeca el bolígraf" quan va entre ells.

4. POLYGON MAPS

Quan tenim la longitud i latitud dels mapes, una manera senzilla de dibuixar mapes, com veurem en l'exercici 1, és utilitzant `geom_polygon()`. Aquesta geometria ens ajuda a dibuixar contorns per diferents regions.

`geom_polygon()` dibuixa línies entre punts i "les tanca" (és a dir, dibuixa una línia des de l'últim punt fins al primer punt).

EXERCICIS

1.- Anem a testear la comanda `geom_polygon()` per dibuixar mapes quan tenim dades espacials que fan ús de la longitud-latitud. Per fer ús de `ggplot`, recordeu que necessitem les dades, el mapeig (al nostre sistema de coordenades com a mínim) i la geometria. En aquest cas, el nostre sistema de coordenades és $x = \text{Long}$ i $y = \text{Lat}$. Per tant, com sempre, heu de fer un mapeig (usant l'estètica `aes`) per representar les dades en el vostre sistema de coordenades. A més, fent ús de `group` en el vostre mapeig, agrupeu les dades espacials de manera que es connectin els punts que han d'estar connectats per línies. Finalment, si afegiu l'argument `coord_quickmap()` us permet ajustar els eixos per assegurar-nos que la longitud i la latitud es representen a la mateixa escala.

(a) Seguint les indicacions que se us ha donat a l'enunciat, dibuixeu el mapa de USA

NOTA: Recordeu assignar primer a la variable de nom `usa`, un marc de dades (`dataframe`) de USA. Això podeu fer-ho gràcies a la funció `map_data` que heu vist més amunt:

(b) Utilitzeu el clàssic layer de llum fosca per a `ggplot2` adequat per als mapes: `theme_set(theme_bw())`

(c) Dibueixeu el mapa de Nova Zelanda (`nz`) anàlogament

(d) Ara en canvi feu ús de `fill` i `colour` que ja hem usat en altres geometries per omplir de blanc ("white") l'interior del mapa i posar els contorns en gris ("grey50")

2.- Fent ús de la funció, `map_data`, també podem obtenir un `dataframe` que ens indiqui les **fronteres estatals** del mapa 'usa':

```
> states <- map_data("state")
```

```
~/
> head(states)
   long    lat group order  region subregion
1 -87.46201 30.38968     1     1 alabama      <NA>
2 -87.48493 30.37249     1     2 alabama      <NA>
3 -87.52503 30.37249     1     3 alabama      <NA>
4 -87.53076 30.33239     1     4 alabama      <NA>
5 -87.57087 30.32665     1     5 alabama      <NA>
6 -87.58806 30.32665     1     6 alabama      <NA>
```

Pinteu de colors els estats (`atribut='region'`) i poseu les fronteres en blanc. Afegiu `guides(fill=FALSE)` per tal d'eliminar la llegenda.

NOTA: En algunes versions de R heu de posar `guides(fill="none")`