sdcSpatial: Privacy protected density maps

Edwin de Jonge @edwindjonge

Statistics Netherlands Research & Development
@edwindjonge

useR! 2019, July 11 2019

sdcSpatial: Privacy protected maps

sdcSpatial: Privacy protected maps

Takeout message: `sdcSpatial` has methods for:

- **Creating** a **raster** map: `sdc_raster` for pop density, value density and mean density, using the excellent `raster` (Hijmans 2019).
- **Finding out** which locations are **sensitive**: `plot_sensitive`, `is_sensitive`.
- Adjusting raster map for **protecting data**: `protect_smooth`, `protect_quadtree`.
- **Removing sensitive** locations.

Who am I and why sdcSpatial?

- Statistical consultant, Data Scientist @cbs.nl / Statistics NL
- Statistics Netherlands is producer main official statistics in the Netherlands:
  - Stats on Demographics, economy (GDP), education, environment, agriculture, Finance etc.
  - Part of the European Statistical System, ESS.

Motivation for `sdcSpatial`

- ESS has European Code of Statistical Practice (predates GDPR, European law on Data Protection):
  no individual information may be revealed.

Sdc in `sdcSpatial`?

SDC = "Statistical Disclosure Control"

Collection of statistical methods to:

- Check if data is safe to be published
- Protect data by slightly altering (aggregated) data
  - adding noise
  - shifting mass
- Most SDC methods operate on records.
- **`sdcSpatial` works upon locations.**

Data

```r
data(dwellings, package="sdcSpatial")
nrow(dwellings)
```

```
## [1] 90603
```

```r
head(dwellings) # consumption/unemployed are simulated!
```

```
##        x      y consumption unemployed
## 1 149712 470104    2049.926      FALSE
## 2 149639 469906    1814.938      FALSE
## 3 149631 469888    2074.882      FALSE
## 4 149788 469831    1927.989      FALSE
## 5 149773 469834    2164.969      FALSE
## 6 149688 469898    1987.958      FALSE
```

Let's create a `sdc_raster`

Creation:

```
library(sdcSpatial)
unemployed <- sdc_raster( dwellings[c("x", "y")] # realistic locations
                        , dwellings$unemployed # simulated data!
                        , r = 500 # raster resolution of 500m
                        , min_count = 10 # min support
                        )
```

What has been created?

```
print(unemployed)
```

```
## logical sdc_raster object:
##    resolution: 500 500 ,  max_risk: 0.95 , min_count: 10
##    mean sensitivity score [0,1]:  0.4249471
```

42% of the data on this map is sensitive!

What is sensitivity?

Binary score (`logical`) per raster cell indicating if it's unsafe to publish.

Calculated:

a) Per location (x_i, y_i) (raster cell)

b) Using risk function `disclosure_risk` $r(x, y) \in [0, 1]$. How accurate can an attacker estimate the value of an individual?
If $r(x_i, y_i) > \texttt{max\_risk}$ then (x_i, y_i) is sensitive.

c) Using a minimum number of observations.
If $\text{count}_i < \texttt{min\_count}$, then (x_i, y_i) is sensitive.

Disclosure risks

External (`numeric`)

$$r(x,y) = \max \frac{v_i}{\sum_{i \in (x,y)} v_i} \text{with } v_i \in \mathbb{R}$$

Discrete (`logical`)

$$r(x,y) = \frac{1}{n} \sum_{i \in (x,y)} v_i \text{ with } v_i \in \{0,1\}$$

Type of raster density maps:

(Stored in `unemployed$value`):

Density can be area-based:

- **number of people** per square (`$count`): population density.
- **(total) value** per square (`$sum`): number of unemployed per square.

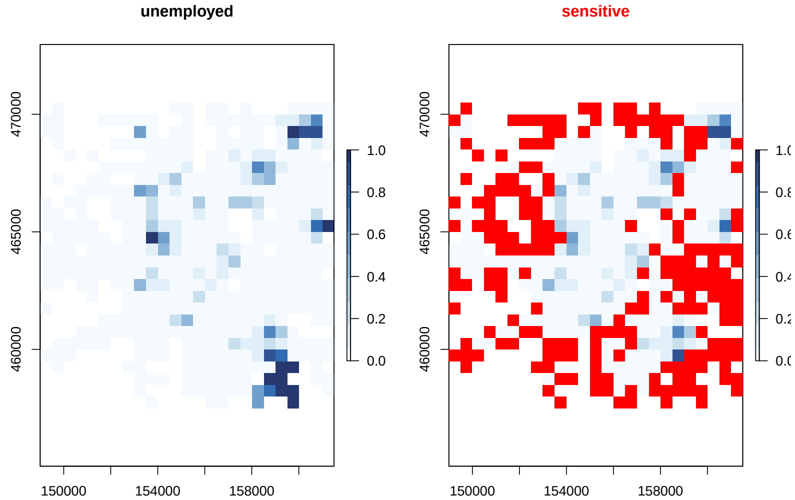
Or density can population-based:

- **Mean value** per square (`$mean`): unemployment rate per square.

Note: All density types are valid, but (total) value per square strongly interacts with population density.
(e.g. https://xkcd.com/1138).

Plotting a `sdc_raster`

```r
plot(unemployed, "mean")
```



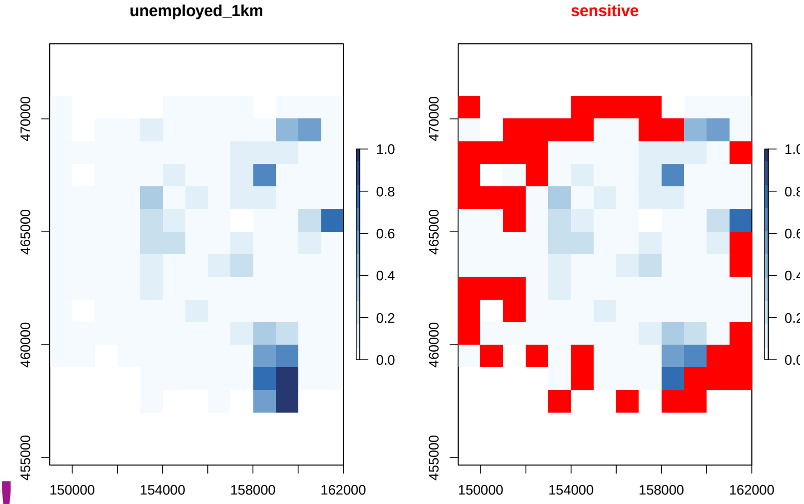
How to reduce sensitivity?

Options:

a) Use a coarser raster: `sdc_raster`.
b) Apply spatial smoothing: **`protect_smooth`** (Wolf and Jonge 2018; Jonge and Wolf 2016).
c) Aggregate sensitive cells hierarchically with a quad tree until not sensitive: **`protect_quadtree`** (Suñé et al. 2017).
d) Remove sensitive locations: **`remove_sensitive`**.

Option: coarser raster

```r
unemployed_1km <- sdc_raster( dwellings[c("x", "y")]
                            , dwellings$unemployed, r =1e3) # 1km!
plot(unemployed_1km, "mean")
```



Option: Coarsening

Pros

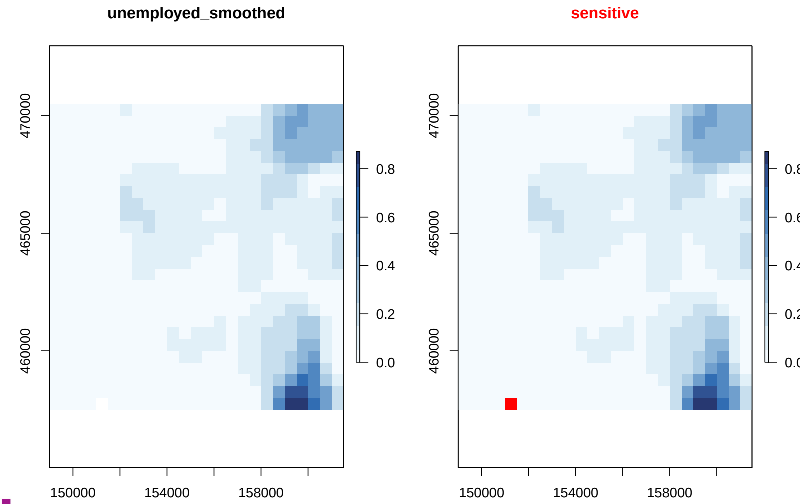
- Simple and easy explainable

Cons

- Detailed spatial patterns are removed
- visually unattractive: “Blocky”

Option: `protect_smooth`

```
unemployed_smoothed <- protect_smooth(unemployed, bw = 1500)
plot(unemployed_smoothed, "mean")
```



Option: `protect_smooth`

Pro's

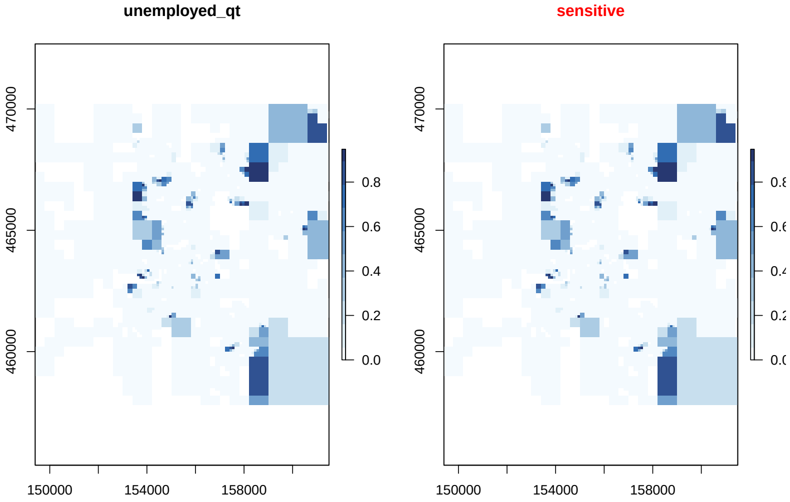
- Often enhances spatial pattern visualization, removing spatial noise.
- Makes it a density map and used as source for e.g. contour map.

Con's

- Does not remove all sensitive values (depends on bandwidth `bw`)
- A fixed band width is used for all locations: may remove detailed patterns...
  spatial processes often have location dependent band widths. (= future work)

Option: `protect_quadtree`

```r
unemployed_100m <- sdc_raster( dwellings[c("x","y")], dwellings$unemployed
                              , r = 100) # use a finer raster
unemployed_qt <- protect_quadtree(unemployed_100m)
plot(unemployed_qt)
```



Option: `protect_quadtree`

Pro

- Adapts to data density
- Adjusts until no sensitive data is left.

Cons

- Visually: “Blocky” / “Mondrian-like” result.

Publish: visual interpolation

In 5 lines we create a visual attractive map that is safe:

```
unemployed <- sdc_raster(dwellings[c("x","y")], dwellings$unemployed, r=500)
unemployed_smoothed <- protect_smooth(unemployed, bw = 1500)
unemployed_safe <- remove_sensitive(unemployed_smoothed)
mean_unemployed <- mean(unemployed_safe)
raster::filledContour(mean_unemployed, main="Unemployment rate")
```

The end

Thank you for your attention!

Questions?

Curious?

```r
install.packages("sdcSpatial")
```

Feedback and suggestions?

https://github.com/edwindj/sdcSpatial/issues

useR! 2019

References

Hijmans, Robert J. 2019. *Raster: Geographic Data Analysis and Modeling*. https://CRAN.R-project.org/package=raster.

Jonge, Edwin de, and Peter-Paul de Wolf. 2016. “Spatial Smoothing and Statistical Disclosure Control.” In *Privacy in Statistical Databases*, edited by Josep Domingo-Ferrer and Mirjana Pejić-Bach, 107–17. Springer.

Suñé, E., C. Rovira, D. Ibáñez, and M. Farré. 2017. “Statistical Disclosure Control on Visualising Geocoded Population Data Using Quadtrees.” http://nt17.pg2.at/data/x_abstracts/x_abstract_286.docx.

Wolf, Peter-Paul de, and Edwin de Jonge. 2018. “Spatial Smoothing and Statistical Disclosure Control.” In *Privacy in Statistical Databases - PSD 2018*, edited by Josep Domingo-Ferrer and Francisco Montes Suay. Springer.