Nonparametric Analysis of US Dairy Production and Consumption

Spatial Nonparametric

Teo Bucci[*] Filippo Cipriani[†] Gabriele Corbo[‡] Andrea Puricelli[§]

2023-02-17

Contents

1 Load libraries and data

```
library("readxl")
library(sf)
library(maps)
library(ggspatial)
library(tidyr)
library(raster)
library(sp)
library(npsp)
library(ggplot2)
```

```
source("05-Prepare-data-spatial.R",echo=FALSE)
```

2 Study of the Dairy prodction in 2007

Using log values of the population and the dairy sales to avoid scalability problems and setting up the coordinates for spatial analysis

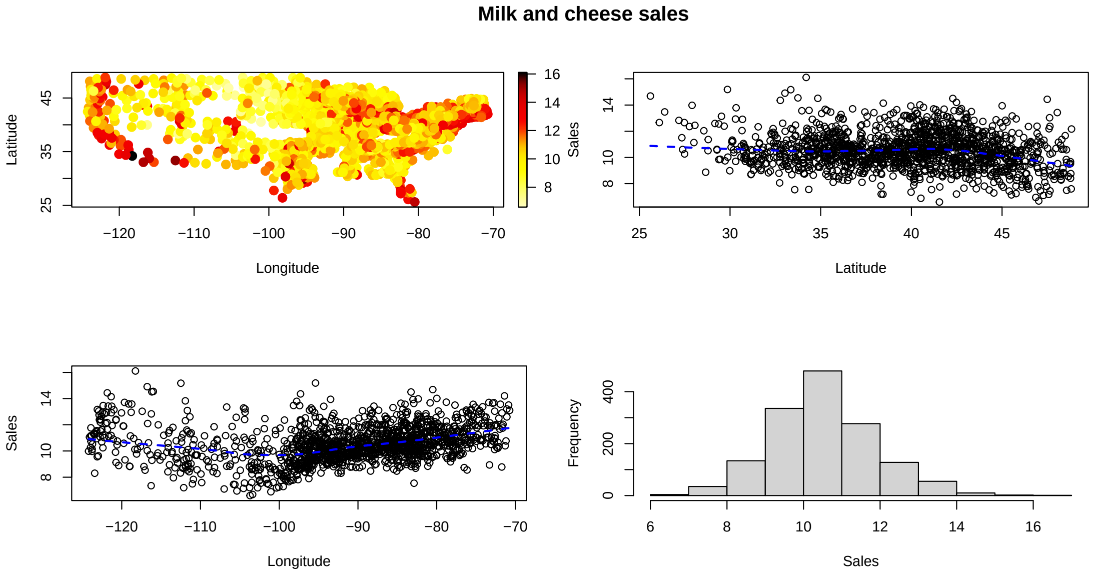
[*] teo.bucci@mail.polimi.it

[†] filippo.cipriani@mail.polimi.it

[‡] gabriele.corbo@mail.polimi.it

[§] andrea3.puricelli@mail.polimi.it

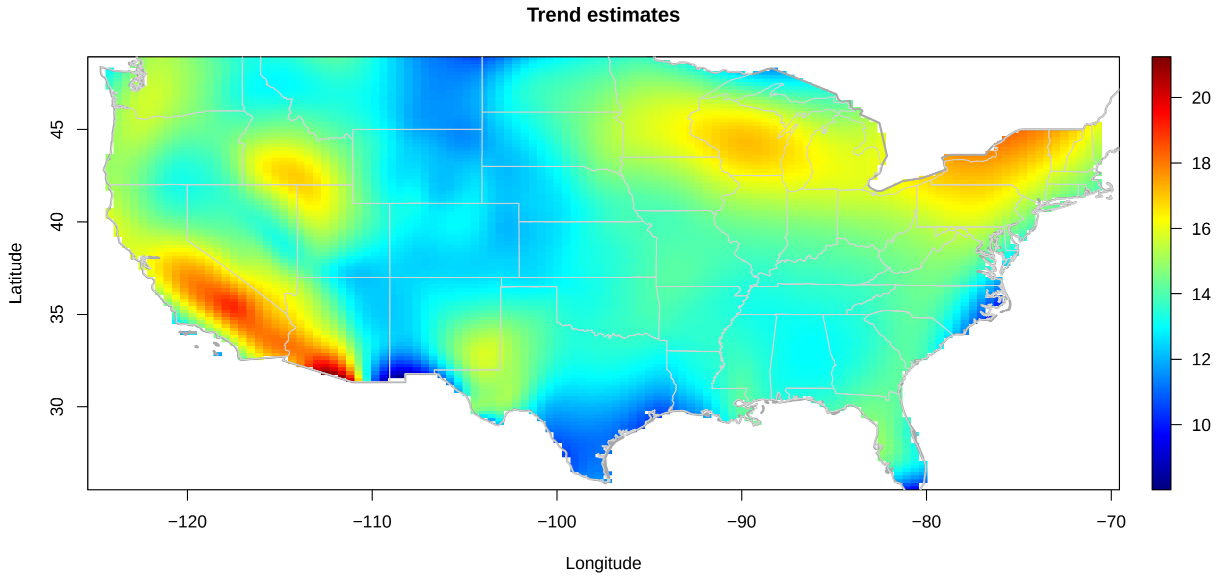
```r
dat1 <- all_sales_county_2007
dat1 <- dat1[-c(1,2,3)]
log_pop <- log(dat1$Population)
log_value <- log(dat1$Value)
log_dat <- as.data.frame(cbind(log_pop,log_value,"x" = dat1$x, "y" = dat1$y))
coordinates(log_dat) <- c("y","x")

scattersplot(log_dat, main = 'Milk and cheese sales', xlab = 'Longitude',
             ylab = 'Latitude', zlab = 'Sales')
```



3 Trend estimates for log_value

Plotting the estimates of the model using a local linear estimator

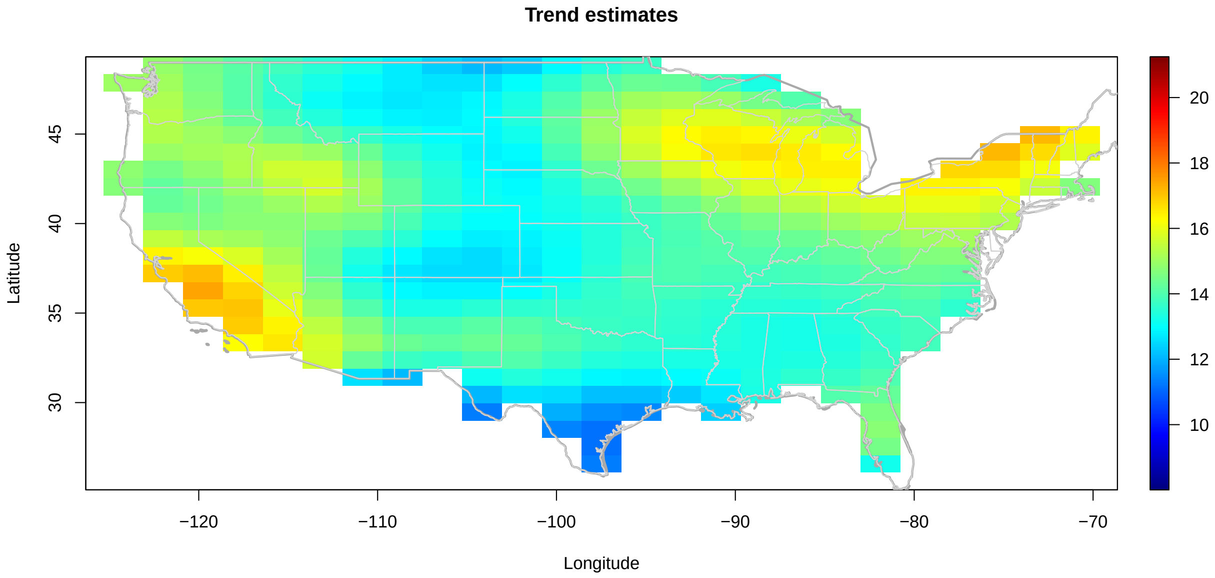
```r
attr_prec <- attributes(precipitation)
border <- attr_prec$border
interior <- attr_prec$interior
x <- coordinates(log_dat)
y <- log_dat$log_value
lp <- locpol(x, y, nbin = c(120, 120), h = diag(c(5, 5)))
lp <- mask(lp, window = border)
attr <- attributes(log_dat)
slim <- range(y)
col <- jet.colors(256)

simage(lp,  slim = slim, main = "Trend estimates",
       col = col, xlab = "Longitude", ylab = "Latitude", asp = 1)
plot(border, border = "darkgray", lwd = 2, add = TRUE)
plot(interior, border = "lightgray", lwd = 1, add = TRUE)
```



4 Bandwidth selection

Tuning of the bandwidth using CV

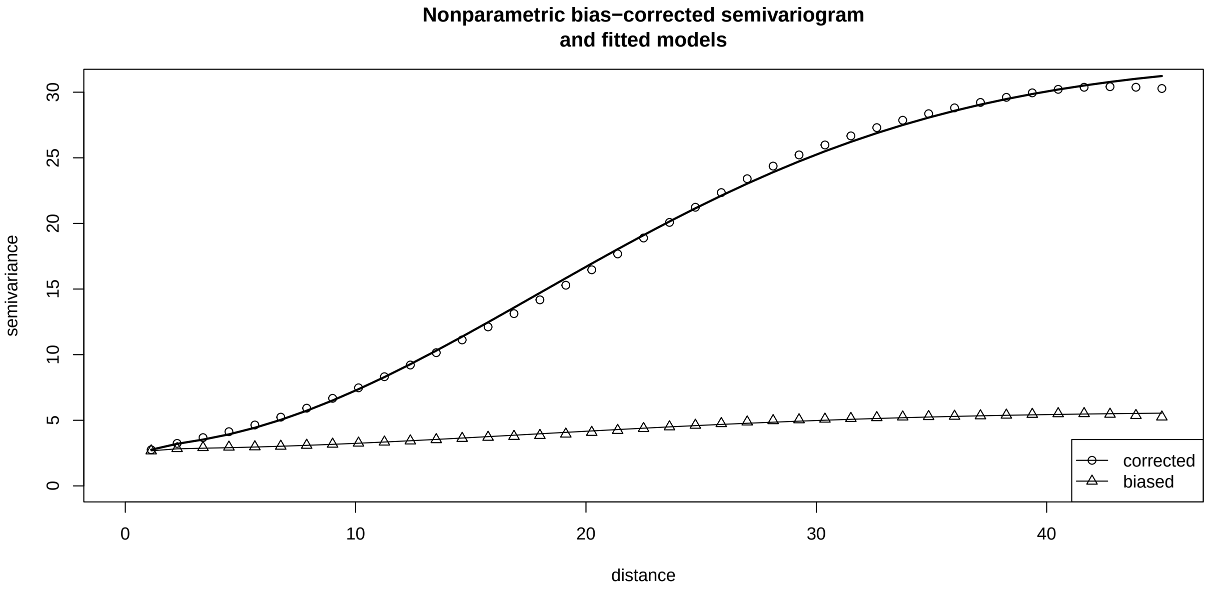
```
bin <- binning(x, y, window = border)
lp0.h <- h.cv(bin)$h
lp0 <- locpol(bin, h = lp0.h, hat.bin = TRUE)
lp0 <- mask(lp0, window = border)

simage(lp0, slim = slim, main = "Trend estimates",
       col = col, xlab = "Longitude", ylab = "Latitude", asp = 1)
plot(border, border = "darkgray", lwd = 2, add = TRUE)
plot(interior, border = "lightgray", lwd = 1, add = TRUE)
```



5 Variogram estimation

Plotting the estimates of the bias-corrected semivariogram

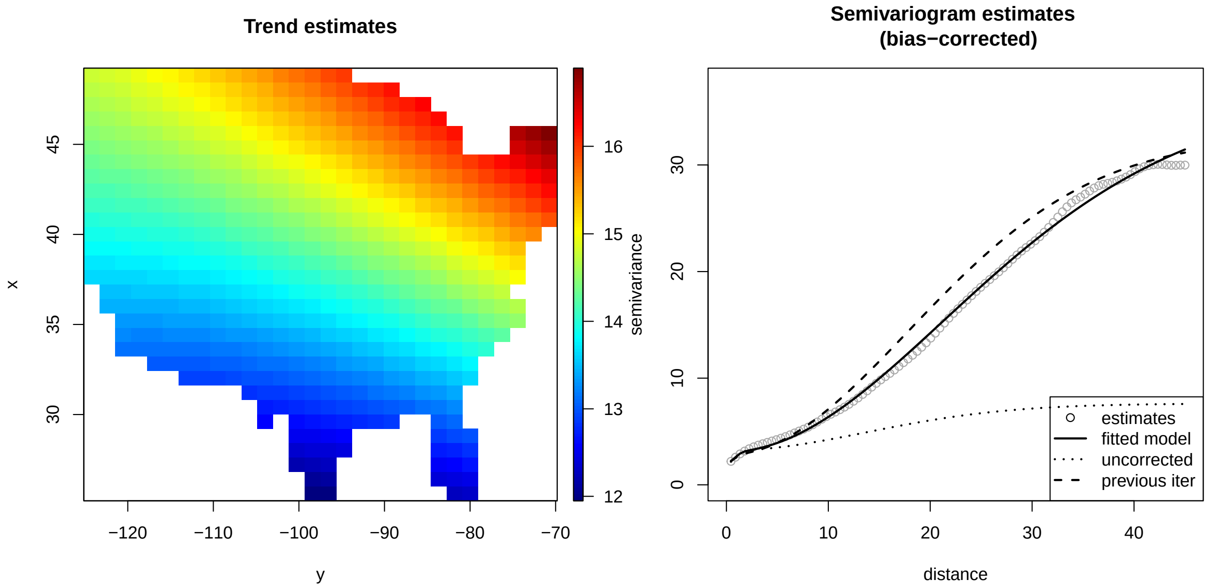
```
lp0 <- locpol(bin, h = lp0.h, hat.bin = TRUE)
svar.bin <- svariso(x, residuals(lp0), nlags = 40, maxlag = 45)
svar.h <- h.cv(svar.bin)$h
svar.np <- np.svar(svar.bin, h = svar.h)
svar.np2 <- np.svariso.corr(lp0, nlags = 40, maxlag = 45,
                            h = svar.h, plot = FALSE)
svm0 <- fitsvar.sb.iso(svar.np, dk = 0)
svm1 <- fitsvar.sb.iso(svar.np2, dk = 0)

plot(svm1, main = "Nonparametric bias-corrected semivariogram\nand fitted models",
     legend = FALSE, xlim = c(0,max(coords(svar.np2))),
     ylim = c(0,max(svar.np2$biny, na.rm = TRUE)))
plot(svm0, lwd = c(1, 1), add = TRUE)
plot(svar.np, type = "p", pch = 2, add = TRUE)
# abline(h = c(svm1$nugget, svm1$sill), lty = 3)
# abline(v = 0, lty = 3)
legend("bottomright", legend = c("corrected", 'biased'),
       lty = c(1, 1), pch = c(1, 2), lwd = c(1, 1))
```



6 Automatic modelling

`np.fitgeo()` repeats the same procedure all in one step

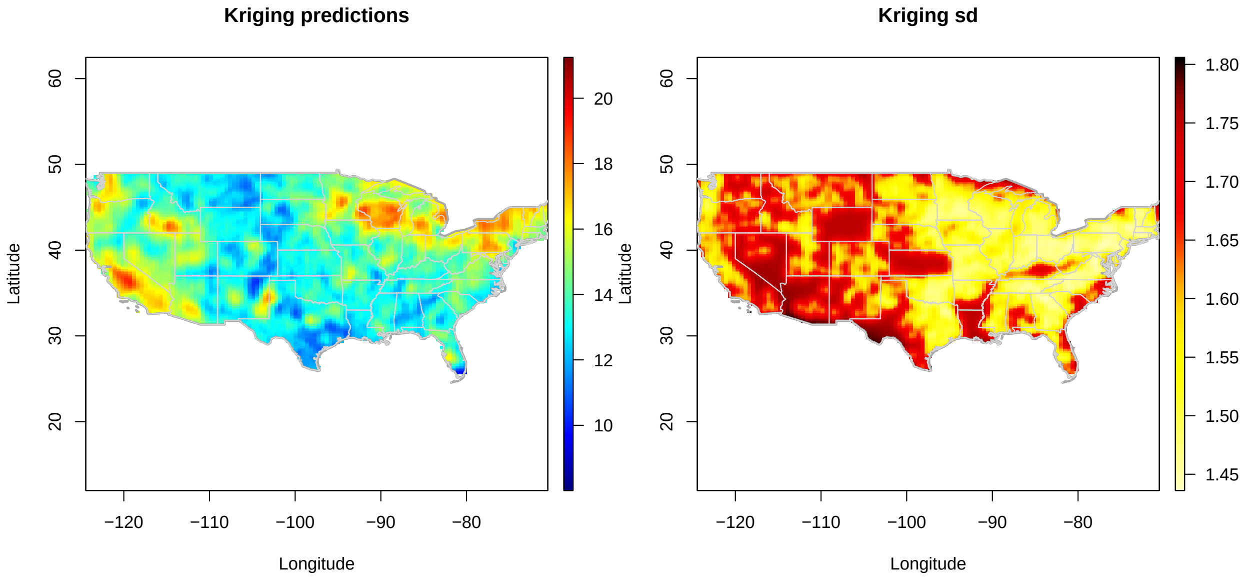
```
geomod <- np.fitgeo(x, y, nbin = c(30, 30), maxlag = 45,
                    svm.resid = TRUE, window = border)

plot(geomod)
```



7 Kriging

Using the model estimates and the semivariogram we can use a kriging prediction method

```
krig.grid <- np.kriging(geomod, ngrid = c(120, 120))
# Plot kriging predictions and kriging standard deviations
old.par <- par(mfrow = c(1,2), omd = c(0.0, 0.98, 0.0, 1))
krig.grid2 <- krig.grid
simage(krig.grid2, 'kpred', main = 'Kriging predictions', slim = slim,
       xlab = "Longitude", ylab = "Latitude" , col = col, asp = 1)
plot(border, border = "darkgray", lwd = 2, add = TRUE)
plot(interior, border = "lightgray", lwd = 1, add = TRUE)
simage(krig.grid2, 'ksd', main = 'Kriging sd',
       xlab = "Longitude", ylab = "Latitude" , col = hot.colors(256), asp = 1)
plot(border, border = "darkgray", lwd = 2, add = TRUE)
plot(interior, border = "lightgray", lwd = 1, add = TRUE)
```


Kriging predictions
Kriging sd
Latitude
Longitude
60
50
40
30
20
−120
−110
−100
−90
−80
20
18
16
14
12
10
1.80
1.75
1.70
1.65
1.60
1.55
1.50
1.45