Introduction to the North Carolina SIDS data set (revised)

Roger Bivand

July 31, 2010

1 Introduction

This data set was presented first in Symons et al. (1983), analysed with reference to the spatial nature of the data in Cressie and Read (1985), expanded in Cressie and Chan (1989), and used in detail in Cressie (1991). It is for the 100 counties of North Carolina, and includes counts of numbers of live births (also non-white live births) and numbers of sudden infant deaths, for the 1974–1978 and 1979–1984 periods. In Cressie and Read (1985), a listing of county neighbours based on shared boundaries (contiguity) is given, and in Cressie and Chan (1989), and in Cressie (1991, pp. 386–389), a different listing based on the criterion of distance between county seats, with a cutoff at 30 miles. The county seat location coordinates are given in miles in a local (unknown) coordinate reference system. The data are also used to exemplify a range of functions in the `S-PLUS` spatial statistics module user's manual (Kaluzny et al., 1996).

2 Getting the data into R

We will be using the **spdep** package, here version: spdep, version 0.5-17, 2010-07-31, the pkgsp package and the **maptools** package. The data from the sources refered to above is documented in the help page for the `nc.sids` data set in **spdep**. The actual data, included in a shapefile of the county boundaries for North Carolina has been made available in the **maptools** package[1]. These data are known to be geographical coordinates (longitude-latitude in decimal degrees) and are assumed to use the NAD27 datum.

```
> library(sp)
> library(maptools)
> library(spdep)
> nc_file <- system.file("etc/shapes/sids.shp", package = "spdep")[1]
> llCRS <- CRS("+proj=longlat +datum=NAD27")
> nc <- readShapeSpatial(nc_file, ID = "FIPSNO", proj4string = llCRS)
```

The shapefile format presupposes that you have three files with extensions `*.shp`, `*.shx`, and `*.dbf`, where the first contains the geometry data, the second the spatial index, and the third the attribute data. They are required to have the same name apart from the extension, and are read here using `readShapeSpatial()` into the `SpatialPolygonsDataFrame` object `nc`; the class is defined in **sp**. The centroids

[1]These data are taken with permission from: `http://sal.agecon.uiuc.edu/datasets/sids.zip`.

of the largest polygon in each county are available using the `coordinates` method from **sp** as a two-column matrix, and can be used to place labels:

```
> plot(nc, axes = TRUE)
> text(coordinates(nc), label = nc$FIPSNO, cex = 0.5)
```

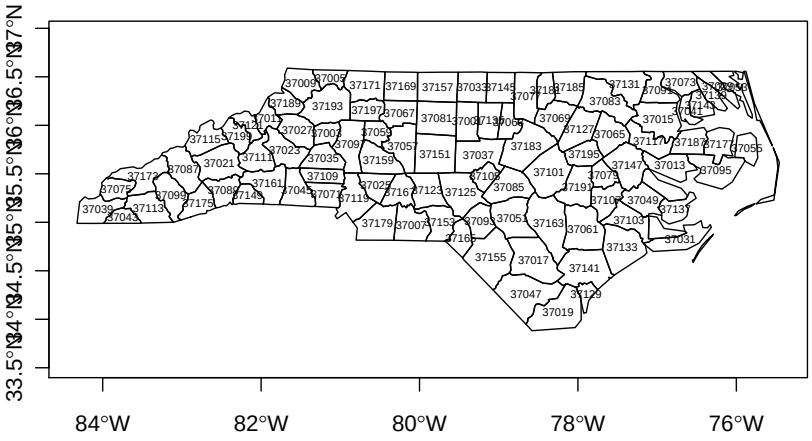

Figure 1: County boundaries and polygon centroids, North Carolina

We can examine the names of the columns of the data frame to see what it contains — in fact some of the same columns that we will be examining below, and some others which will be useful in cleaning the data set.

```
> names(nc)
```

```
 [1] "SP_ID"     "CNTY_ID"   "east"      "north"     "L_id"
 [6] "M_id"      "names"     "AREA"      "PERIMETER" "CNTY_"
[11] "NAME"      "FIPS"      "FIPSNO"    "CRESS_ID"  "BIR74"
[16] "SID74"     "NWBIR74"   "BIR79"     "SID79"     "NWBIR79"
```

```
> summary(nc)
```

```
Object of class SpatialPolygonsDataFrame
Coordinates:
        min       max
x -84.32385 -75.45698
y  33.88199  36.58965
Is projected: FALSE
proj4string : [+proj=longlat +datum=NAD27]
Data attributes:
     SP_ID       CNTY_ID          east            north
 37001  : 1   Min.   :1825   Min.   : 19.0   Min.   :  6.0
 37003  : 1   1st Qu.:1902   1st Qu.:178.8   1st Qu.: 97.0
 37005  : 1   Median :1982   Median :285.0   Median :125.5
 37007  : 1   Mean   :1986   Mean   :271.3   Mean   :122.1
 37009  : 1   3rd Qu.:2067   3rd Qu.:361.2   3rd Qu.:151.5
 37011  : 1   Max.   :2241   Max.   :482.0   Max.   :182.0
 (Other):94
      L_id            M_id            names         AREA
 Min.   :1.00   Min.   :1.00   Alamance : 1   Min.   :0.0420
 1st Qu.:1.00   1st Qu.:2.00   Alexander: 1   1st Qu.:0.0910
 Median :2.00   Median :3.00   Alleghany: 1   Median :0.1205
 Mean   :2.12   Mean   :2.67   Anson    : 1   Mean   :0.1263
```

```
3rd Qu.:3.00   3rd Qu.:3.25   Ashe     : 1   3rd Qu.:0.1542
Max.   :4.00   Max.   :4.00   Avery    : 1   Max.   :0.2410
                              (Other)  :94
  PERIMETER         CNTY_             NAME         FIPS
Min.   :0.999   Min.   :1825   Alamance : 1   37001  : 1
1st Qu.:1.324   1st Qu.:1902   Alexander: 1   37003  : 1
Median :1.609   Median :1982   Alleghany: 1   37005  : 1
Mean   :1.673   Mean   :1986   Anson    : 1   37007  : 1
3rd Qu.:1.859   3rd Qu.:2067   Ashe     : 1   37009  : 1
Max.   :3.640   Max.   :2241   Avery    : 1   37011  : 1
                               (Other)  :94   (Other):94
    FIPSNO         CRESS_ID          BIR74           SID74
Min.   :37001   Min.   :  1.00   Min.   :  248   Min.   : 0.00
1st Qu.:37050   1st Qu.: 25.75   1st Qu.: 1077   1st Qu.: 2.00
Median :37100   Median : 50.50   Median : 2180   Median : 4.00
Mean   :37100   Mean   : 50.50   Mean   : 3300   Mean   : 6.67
3rd Qu.:37150   3rd Qu.: 75.25   3rd Qu.: 3936   3rd Qu.: 8.25
Max.   :37199   Max.   :100.00   Max.   :21588   Max.   :44.00

   NWBIR74           BIR79           SID79          NWBIR79
Min.   :   1.0   Min.   :  319   Min.   : 0.00   Min.   :    3.0
1st Qu.: 190.0   1st Qu.: 1336   1st Qu.: 2.00   1st Qu.:  250.5
Median : 697.5   Median : 2636   Median : 5.00   Median :  874.5
Mean   :1050.8   Mean   : 4224   Mean   : 8.36   Mean   : 1352.8
3rd Qu.:1168.5   3rd Qu.: 4889   3rd Qu.:10.25   3rd Qu.: 1406.8
Max.   :8027.0   Max.   :30757   Max.   :57.00   Max.   :11631.0
```

We will now examine the data set reproduced from Cressie and collaborators, included in **spdep**, and add the neighbour relationships used in Cressie and Chan (1989) to the background map as a graph shown in Figure 2:

```
> gal_file <- system.file("etc/weights/ncCR85.gal", package = "spdep")[1]
> ncCR85 <- read.gal(gal_file, region.id = nc$FIPSNO)
> ncCR85

Neighbour list object:
Number of regions: 100
Number of nonzero links: 492
Percentage nonzero weights: 4.92
Average number of links: 4.92

> gal_file <- system.file("etc/weights/ncCC89.gal", package = "spdep")[1]
> ncCC89 <- read.gal(gal_file, region.id = nc$FIPSNO)
> ncCC89

Neighbour list object:
Number of regions: 100
Number of nonzero links: 394
Percentage nonzero weights: 3.94
Average number of links: 3.94
2 regions with no links:
37055 37095

> plot(nc, border = "grey")
> plot(ncCC89, coordinates(nc), add = TRUE, col = "blue")
```

Printing the neighbour object shows that it is a neighbour list object, with a very sparse structure — if displayed as a matrix, only 3.94% of cells would be filled. Objects of class `nb` contain a list as long as the number of counties; each component of the list is a vector with the index numbers of the neighbours of the county in question, so that the neighbours of the county with `region.id` of `"37001"` can be retreived by matching against the indices. More information can be obtained by using `summary()` on an `nb` object. Finally, we associate a vector of names with the neighbour list, through the `row.names` argument. The names should be unique, as with data frame row names.

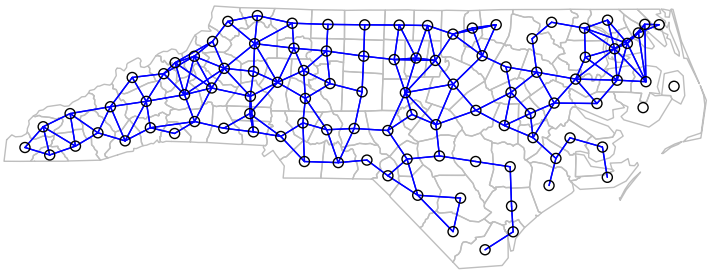

Figure 2: Overplotting county boundaries with 30 mile neighbour relations as a graph.

```
> ncCC89

Neighbour list object:
Number of regions: 100
Number of nonzero links: 394
Percentage nonzero weights: 3.94
Average number of links: 3.94
2 regions with no links:
37055 37095

> r.id <- attr(ncCC89, "region.id")
> ncCC89[[match("37001", r.id)]]

[1] 17 19 32 41 68

> r.id[ncCC89[[match("37001", r.id)]]]

[1] 37033 37037 37063 37081 37135
```

The neighbour list object records neighbours by their order in relation to the list itself, so the neighbours list for the county with `region.id` "37001" are the seventeenth, nineteenth, thirty-second, forty-first and sixty-eighth in the list. We can retreive their codes by looking them up in the `region.id` attribute.

```
> as.character(nc$NAME)[card(ncCC89) == 0]

[1] "Dare" "Hyde"
```

We should also note that this neighbour criterion generates two counties with no neighbours, Dare and Hyde, whose county seats were more than 30 miles from their nearest neighbours. The `card()` function returns the cardinality of the neighbour set. We need to return to methods for handling no-neighbour objects later on. We will also show how new neighbours lists may be constructed in R, and compare these with those from the literature.

2.1 Probability mapping

Rather than review functions for measuring and modelling spatial dependence in the **spdep** package, we will focus on probability mapping for disease rates data. Typically,

we have counts of the incidence of some disease by spatial unit, associated with counts of populations at risk. The task is then to try to establish whether any spatial units seem to be characterised by higher or lower counts of cases than might have been expected in general terms (Bailey and Gatrell, 1995).

An early approach by Choynowski (1959), described by Cressie and Read (1985) and Bailey and Gatrell (1995), assumes, given that the true rate for the spatial units is small, that as the population at risk increases to infinity, the spatial unit case counts are Poisson with mean value equal to the population at risk times the rate for the study area as a whole. Choynowski's approach folds the two tails of the measured probabilities together, so that small values, for a chosen α, occur for spatial units with either unusually high or low rates. For this reason, the high and low counties are plotted separately in Figure 3.

```
> ch <- choynowski(nc$SID74, nc$BIR74)
> nc$ch_pmap_low <- ifelse(ch$type, ch$pmap, NA)
> nc$ch_pmap_high <- ifelse(!ch$type, ch$pmap, NA)
> spplot(nc, c("ch_pmap_low", "ch_pmap_high"), at = c(0,
+     0.001, 0.01, 0.05, 0.1, 1), col.regions = grey.colors(5))
```

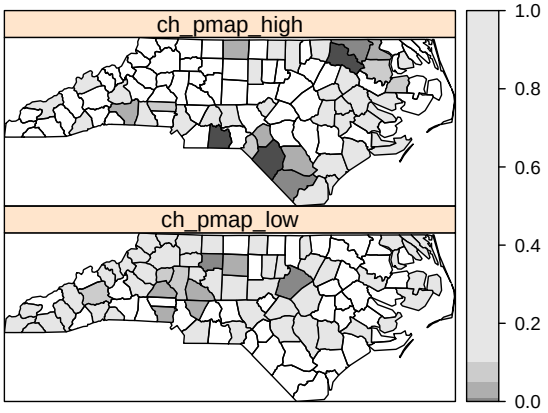

Figure 3: Probability map of North Carolina counties, SIDS cases 1974–78, $\alpha = 0.05$, reproducing Cressie and Read (1985), Figure 1.

For more complicated thematic maps, it may be helpful to use ColorBrewer (`http://colorbrewer.org`) colour palettes. Here we will only use the grey sequential palette, available in R in the **RColorBrewer** package (the colours are copied here to avoid loading the package).

While the `choynowski()` function only provides the probability map valyes required, the `probmap()` function returns raw (crude) rates, expected counts (assuming a constant rate across the study area), relative risks, and Poisson probability map values calculated using the standard cumulative distribution function `ppois()`. This does not fold the tails together, so that counties with lower observed counts than expected, based on population size, have values in the lower tail, and those with higher observed counts than expected have values in the upper tail, as Figure 4 shows.

```
> pmap <- probmap(nc$SID74, nc$BIR74)
> nc$pmap <- pmap$pmap
```

```
> brks <- c(0, 0.001, 0.01, 0.025, 0.05, 0.95, 0.975, 0.99,
+     0.999, 1)
> library(RColorBrewer)
> spplot(nc, "pmap", at = brks, col.regions = rev(brewer.pal(9,
+     "RdBu")))
```

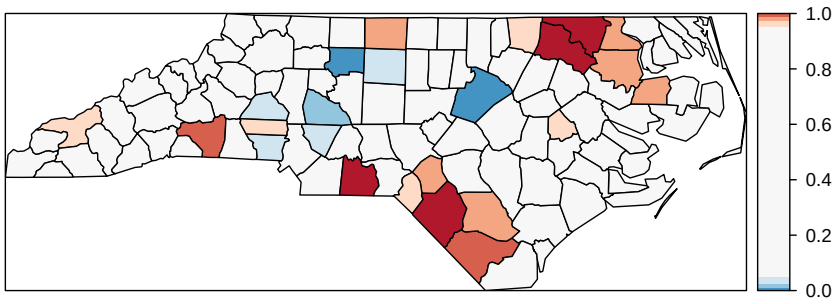

Figure 4: Probability map of North Carolina counties, SIDS cases 1974–78, reproducing Kaluzny et al. (1996), p. 57, Figure 3.28.

Marilia Carvalho (personal communication) and Virgilio GÃşmez Rubio (Gómez Rubio, Ferrándiz and López, 2003) have pointed to the unusual shape of the distribution of the Poisson probability values (Figure 5), repeating the doubts about probability mapping voiced by Cressie (1991, p. 392): "an extreme value . . . may be more due to its lack of fit to the Poisson model than to its deviation from the constant rate assumption". There are many more high values than one would have expected, suggesting perhaps overdispersion, that is that the ratio of the mean and variance is larger than unity.

```
> hist(nc$pmap, main = "")
```

One ad-hoc way to assess the impact of the possible failure of our assumption that the counts follow the Poisson distribution is to estimate the dispersion by fitting a general linear model of the observed counts including only the intercept (null model) and offset by the observed population at risk (suggested by Marilia Carvalho and associates):

```
> res <- glm(SID74 ~ offset(log(BIR74)), data = nc, family = "quasipoisson")
> nc$stdres <- rstandard(res)
> brks <- c(-4, -3, -2, -1.5, -1, -0.5, 0.5, 1, 1.5, 2,
+     3, 4)
> spplot(nc, "stdres", at = brks, col.regions = rev(brewer.pal(11,
+     "RdBu")))
```

The dispersion is equal to 2.2786, much greater than unity; we calculate the corrected probability map values by taking the standardised residuals of the model, taking the size of the dispersion into account; the results are shown in Figure 6. Many fewer counties appear now to have unexpectedly large or small numbers of cases. This is an ad-hoc adjustment made because R provides access to a wide range of model-fitting

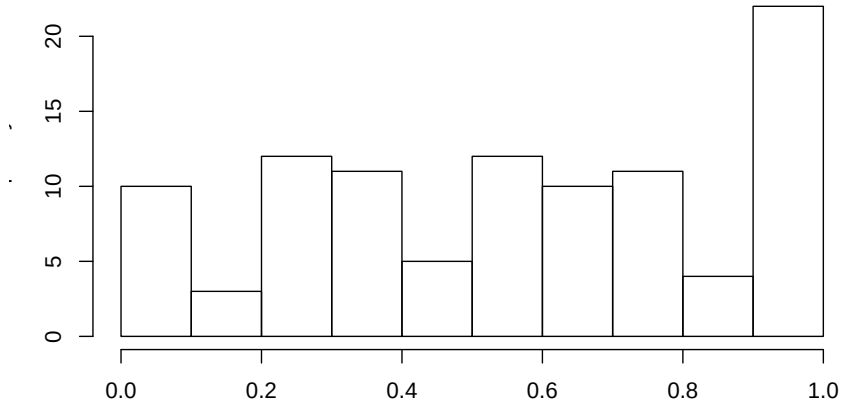

Figure 5: Histogram of Poisson probability values.

functions that can be used to help check our assumptions. Gómez Rubio, Ferrándiz and López (2003) chose rather to construct a probability map under the hypothesis that data are drawn from a Negative Binomial.

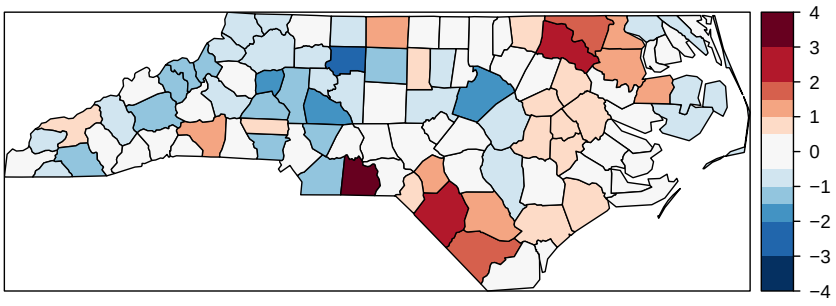

Figure 6: Standardised residual values from the fit of a quasi-Poisson fit of the null model for SIDS rates 1974-78, North Carolina counties.

So far, none of the maps presented have made use of the spatial dependence possibly present in the data. A further elementary step that can be taken is to map Empirical Bayes estimates of the rates, which are smoothed in relation to the raw rates. The underlying question here is linked to the larger variance associated with rate estimates for counties with small populations at risk compared with counties with large populations at risk. Empirical Bayes estimates place more credence on the raw rates of counties with large populations at risk, and modify them much less than they modify

rates for small counties. In the case of small populations at risk, more confidence is placed in either the global rate for the study area as a whole, or for local Empirical Bayes estimates, in rates for a larger moving window including the neighbours of the county being estimated. The function used for this in **spdep** is `EBlocal()`, initially contributed by Marilia Carvalho. It parallels a similar function in GeoDa, but uses the Bailey and Gatrell (1995) interpretation of Marshall (1991), rather than that in GeoDa (Anselin, Syabri and Smirnov, 2002).

```
> global_rate <- sum(nc$SID74)/sum(nc$BIR74)
> nc$Expected <- global_rate * nc$BIR74
> res <- EBlocal(nc$SID74, nc$Expected, ncCC89, zero.policy = TRUE)
> nc$EB_loc <- res$est
> brks <- c(0, 0.25, 0.5, 0.75, 1, 2, 3, 4, 5)
> spl <- list("sp.text", loc = coordinates(nc)[card(ncCC89) ==
+     0, ], txt = rep("*", 2), cex = 1.2)
> spplot(nc, "EB_loc", at = brks, col.regions = rev(brewer.pal(8,
+     "RdBu")), sp.layout = spl)
```

The results are shown in Figure 7. Like other relevant functions in **spdep**, `EBlocal()` takes a `zero.policy` argument to allow missing values to be passed through. In this case, no local estimate is available for the two counties with no neighbours, marked by stars.

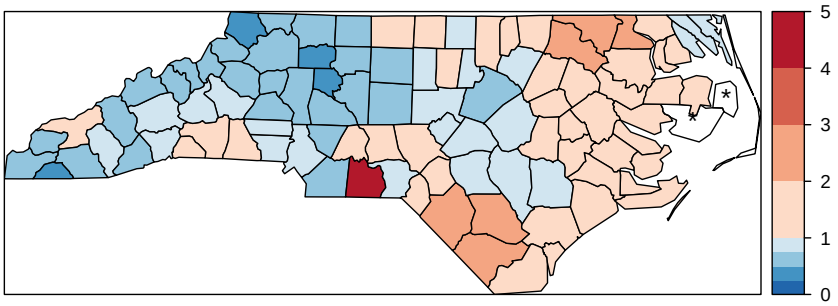

Figure 7: Local Empirical Bayes estimates for SIDS rates per 1000 using the 30 mile county seat neighbours list.

In addition to Empirical Bayes smoothing globally, used both for disease mapping and the AssunÃğÃčo and Reis correction to Moran's *I* for rates data (to shrink towards the global rate when the population at risk is small, here as a Monte Carlo test), lists of local neighbours can be used to shrink towards a local rate.

```
> EBImoran.mc(nc$SID74, nc$BIR74, nb2listw(ncCC89, style = "B",
+     zero.policy = TRUE), nsim = 999, zero.policy = TRUE)

        Monte-Carlo simulation of Empirical Bayes Index

data:  cases: nc$SID74, risk population: nc$BIR74
weights: nb2listw(ncCC89, style = "B", zero.policy = TRUE)
number of simulations + 1: 1000
```

```
statistic = 0.254, observed rank = 999, p-value = 0.001
alternative hypothesis: greater
```

3 Exploration and modelling of the data

One of the first steps taken by Cressie and Read (1985) is to try to bring out spatial trends by dividing North Carolina up into 4×4 rough rectangles. Just to see how this works, let us map these rough rectangles before proceeding further (see Figure 8).

```
> nc$both <- factor(paste(nc$L_id, nc$M_id, sep = ":"))
> nboth <- length(table(unclass(nc$both)))
> spplot(nc, "both", col.regions = sample(rainbow(nboth)))
```

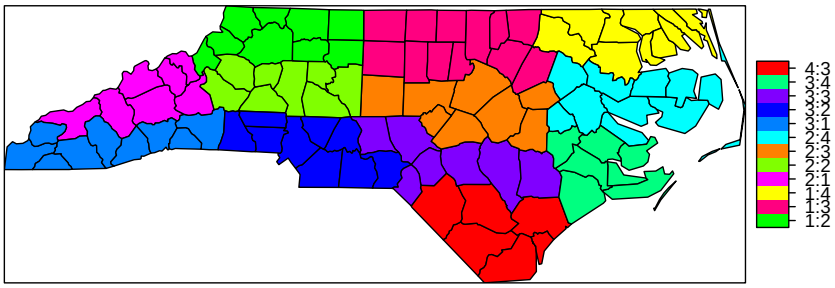

Figure 8: Rough rectangles used by Cressie and Read (1985) to bring out spatial trends.

Cressie constructs a transformed SIDS rates variable, 1974–78, for his analyses (with co-workers). We can replicate his stem-and-leaf figure on p. 396 in the book, taken from Cressie and Read (1989):

```
> nc$ft.SID74 <- sqrt(1000) * (sqrt(nc$SID74/nc$BIR74) +
+     sqrt((nc$SID74 + 1)/nc$BIR74))
> stem(round(nc$ft.SID74, 1), scale = 2)

  The decimal point is at the |

  0 | 9
  1 | 111244
  1 | 567789999
  2 | 0011111222334444
  2 | 55555666677778999999999
  3 | 000111122333333344444444
  3 | 5568999
  4 | 013344
  4 | 555557
  5 | 2
  5 |
  6 | 3
```

3.1 Median polish smoothing

Cressie (1991, pp. 46–48, 393–400) discusses in some detail how smoothing may be used to partition the variation in the data into smooth and rough. In order to try it out on the North Carolina SIDS data set, we will use a coarse gridding into four columns and four rows given by Cressie (1991, pp. 553–554), where four grid cells are empty; these are given by variables `L_id` and `M_id` in object `nc`. Next we aggregate the number of live births and the number of SIDS cases 1974–1978 for the grid cells:

```
> mBIR74 <- tapply(nc$BIR74, nc$both, sum)
> mSID74 <- tapply(nc$SID74, nc$both, sum)
```

Using the same Freeman-Tukey transformation as is used for the county data, we coerce the data into a correctly configured matrix, some of the cells of which are empty. The `medpolish` function is applied to the matrix, being told to remove empty cells; the function iterates over the rows and columns of the matrix using `median` to extract an overall effect, row and column effects, and residuals:

```
> mFT <- sqrt(1000) * (sqrt(mSID74/mBIR74) + sqrt((mSID74 +
+     1)/mBIR74))
> mFT1 <- t(matrix(mFT, 4, 4, byrow = TRUE))
> med <- medpolish(mFT1, na.rm = TRUE, trace.iter = FALSE)
> med

Median Polish Results (Dataset: "mFT1")

Overall: 2.765802

Row Effects:
[1] -0.728192882  0.001560182  0.861279153 -0.001560182

Column Effects:
[1]  0.00000000 -0.03564965  0.44969186  0.00000000

Residuals:
          [,1]      [,2]     [,3]      [,4]
[1,] -0.089076  0.089076  0.33761 -0.089076
[2,]  0.089076 -0.089076 -0.33761  0.089076
[3,]  0.327702 -0.327702 -0.93746  0.327702
[4,] -0.324994  0.324994  0.49479 -0.324994
```

Returning to the factors linking rows and columns to counties, and generating matrices of dummy variables using `model.matrix`, we can calculate fitted values of the Freeman-Tukey adjusted rate for each county, and residuals by subtracting the fitted value from the observed rate. Naturally, the fitted value will be the same for counties in the same grid cell:

```
> mL_id <- model.matrix(~as.factor(nc$L_id) - 1)
> mM_id <- model.matrix(~as.factor(nc$M_id) - 1)
> nc$pred <- c(med$overall + mL_id %*% med$row + mM_id %*%
+     med$col)
> nc$mp_resid <- nc$ft.SID74 - nc$pred
```

```
> cI_ft <- pretty(nc$ft.SID74, n = 9)
> pal_ft <- colorRampPalette(brewer.pal(6, "YlOrBr"))(length(cI_ft) -
+     1)
> p1 <- spplot(nc, c("ft.SID74"), col.regions = pal_ft,
+     at = cI_ft, col = "grey30", main = "FT transformed SIDS rate")
> p2 <- spplot(nc, c("pred"), col.regions = pal_ft, at = cI_ft,
+     col = "grey30", main = "Median-polish fit")
> atn <- pretty(nc$mp_resid[nc$mp_resid < 0])
> atp <- pretty(nc$mp_resid[nc$mp_resid >= 0])
```

```
> pal <- c(rev(brewer.pal(length(atn - 1), "YlOrRd")),
+     brewer.pal(length(atp[-1]), "YlGnBu")[-1])
> p3 <- spplot(nc, "mp_resid", at = c(atn, atp[-1]), col.regions = pal,
+     col = "grey30", main = "Median-polish residuals")
> plot(p1, split = c(1, 1, 1, 3), more = TRUE)
> plot(p2, split = c(1, 2, 1, 3), more = TRUE)
> plot(p3, split = c(1, 3, 1, 3), more = FALSE)
```

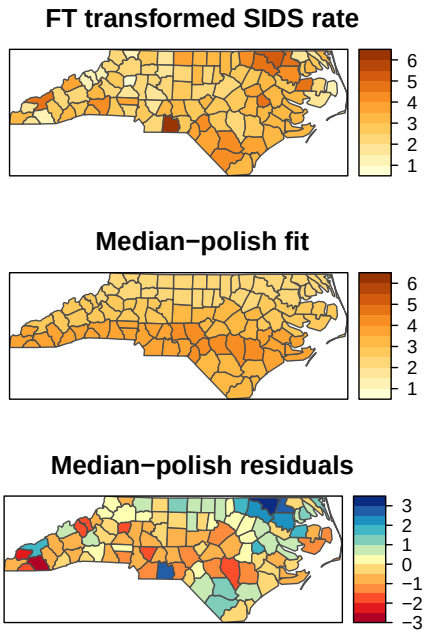

Figure 9: Freeman-Tukey transformed SIDS rates, fitted smoothed values, residuals, and Tukey additivity plot.

Figure 9 shows the median polish smoothing results as three maps, the observed Freeman-Tukey transformed SIDS rates, the fitted smoothed values, and the residuals. In addition, a plot for the median polish object is also shown, plotting the smooth residuals against the outer product of the row and column effects divided by the overall effect, which would indicate a lack of additivity between row and column if this was the case — this is more relevant for analysis of tables of covariates rather than geographical grids.

3.2 CAR model fitting

We will now try to replicate three of the four models fitted by (Cressie and Chan, 1989) to the transformed rates variable. The first thing to do is to try to replicate their 30 mile distance between county seats neighbours, which almost works. From there we try to reconstruct three of the four models they fit, concluding that we can get quite close, but that a number of questions are raised along the way.

Building the weights is much more complicated, because they use a combination of distance-metric and population-at-risk based weights, but we can get quite close (see

also Kaluzny et al., 1996):

```
> sids.nhbr30.dist <- nbdists(ncCC89, cbind(nc$east, nc$north))
> sids.nhbr <- listw2sn(nb2listw(ncCC89, glist = sids.nhbr30.dist,
+     style = "B", zero.policy = TRUE))
> dij <- sids.nhbr[, 3]
> n <- nc$BIR74
> el1 <- min(dij)/dij
> el2 <- sqrt(n[sids.nhbr$to]/n[sids.nhbr$from])
> sids.nhbr$weights <- el1 * el2
> sids.nhbr.listw <- sn2listw(sids.nhbr)
```

The first model (I) is a null model with just an intercept, the second (II) includes all the 12 parcels of contiguous counties in 4 east-west and 4 north-south bands, while the fourth (IV) includes the transformed non-white birth-rate:

```
> nc$ft.NWBIR74 <- sqrt(1000) * (sqrt(nc$NWBIR74/nc$BIR74) +
+     sqrt((nc$NWBIR74 + 1)/nc$BIR74))
```

Cressie identifies Anson county as an outlier, and drops it from further analysis. Because the weights are constructed in a complicated way, they will be subsetted by dropping the row and column of the weights matrix:

```
> lm_nc <- lm(ft.SID74 ~ 1, data = nc)
> outl <- which.max(rstandard(lm_nc))
> as.character(nc$names[outl])
```

```
[1] "Anson"
```

```
> W <- listw2mat(sids.nhbr.listw)
> W.4 <- W[-outl, -outl]
> sids.nhbr.listw.4 <- mat2listw(W.4)
> nc2 <- nc[!(1:length(nc$CNTY_ID) %in% outl), ]
```

It appears that both numerical issues (convergence in particular) and uncertainties about the exact spatial weights matrix used make it difficult to reproduce the results of Cressie and Chan (1989), also given in Cressie (1991). We now try to replicate them for the null weighted CAR model (Cressie has intercept 2.838, $\hat{\theta}$ 0.833, for k=1):

```
> ecarIaw <- spautolm(ft.SID74 ~ 1, data = nc2, listw = sids.nhbr.listw.4,
+     weights = BIR74, family = "CAR")
> summary(ecarIaw)
```

```
Call:
spautolm(formula = ft.SID74 ~ 1, data = nc2, listw = sids.nhbr.listw.4,
    weights = BIR74, family = "CAR")

Residuals:
      Min        1Q    Median        3Q       Max
-2.010292 -0.639658 -0.062209  0.443549  2.018065

Coefficients:
            Estimate Std. Error z value  Pr(>|z|)
(Intercept) 2.945323   0.095135  30.959 < 2.2e-16

Lambda: 0.86814 LR test value: 22.83 p-value: 1.7701e-06

Log likelihood: -118.8432
ML residual variance (sigma squared): 1266.5, (sigma: 35.588)
Number of observations: 99
Number of parameters estimated: 3
AIC: 243.69
```

The spatial parcels model also seems to work, with Cressie's $\hat{\theta}$ 0.710, and the other coefficients agreeing more or less by rank:

```
> ecarIIaw <- spautolm(ft.SID74 ~ both - 1, data = nc2,
+     listw = sids.nhbr.listw.4, weights = BIR74, family = "CAR")
> summary(ecarIIaw)

Call:
spautolm(formula = ft.SID74 ~ both - 1, data = nc2, listw = sids.nhbr.listw.4,
    weights = BIR74, family = "CAR")

Residuals:
      Min        1Q    Median        3Q       Max
-2.558961 -0.463383 -0.020350  0.389350  2.056820

Coefficients:
        Estimate Std. Error z value  Pr(>|z|)
both1:2  2.06223    0.20016 10.3031 < 2.2e-16
both1:3  2.91823    0.14139 20.6400 < 2.2e-16
both1:4  4.11486    0.29939 13.7439 < 2.2e-16
both2:1  2.57650    0.26905  9.5762 < 2.2e-16
both2:2  2.17403    0.18222 11.9305 < 2.2e-16
both2:3  2.67397    0.15329 17.4443 < 2.2e-16
both2:4  3.11361    0.24699 12.6062 < 2.2e-16
both3:1  2.94400    0.29893  9.8486 < 2.2e-16
both3:2  2.65391    0.14098 18.8250 < 2.2e-16
both3:3  2.91619    0.17099 17.0552 < 2.2e-16
both3:4  3.20425    0.20349 15.7468 < 2.2e-16
both4:3  3.80286    0.20806 18.2781 < 2.2e-16

Lambda: 0.2109 LR test value: 1.3088 p-value: 0.25261

Log likelihood: -99.25505
ML residual variance (sigma squared): 891.48, (sigma: 29.858)
Number of observations: 99
Number of parameters estimated: 14
AIC: 226.51
```

Finally, the non-white model repeats Cressie's finding that much of the variance of the transformed SIDS rate for 1974–8 can be accounted for by the transformed non-white birth variable (Cressie intercept 1.644, $\hat{b}$ 0.0346, $\hat{\theta}$ 0.640 — not significant):

```
> ecarIVaw <- spautolm(ft.SID74 ~ ft.NWBIR74, data = nc2,
+     listw = sids.nhbr.listw.4, weights = BIR74, family = "CAR")
> summary(ecarIVaw)

Call:
spautolm(formula = ft.SID74 ~ ft.NWBIR74, data = nc2, listw = sids.nhbr.listw.4,
    weights = BIR74, family = "CAR")

Residuals:
     Min       1Q   Median       3Q      Max
-1.99188 -0.44824  0.15464  0.60655  1.95584

Coefficients:
             Estimate Std. Error z value  Pr(>|z|)
(Intercept) 1.4365406  0.2252431  6.3777 1.797e-10
ft.NWBIR74  0.0408552  0.0062916  6.4936 8.379e-11

Lambda: 0.22339 LR test value: 1.1557 p-value: 0.28235

Log likelihood: -114.0284
ML residual variance (sigma squared): 1201.3, (sigma: 34.659)
Number of observations: 99
Number of parameters estimated: 4
AIC: 236.06

> nc2$fitIV <- fitted.values(ecarIVaw)
```

```
> spplot(nc2, "fitIV", cuts = 12, col.regions = grey.colors(13,
+     0.9, 0.3))
```

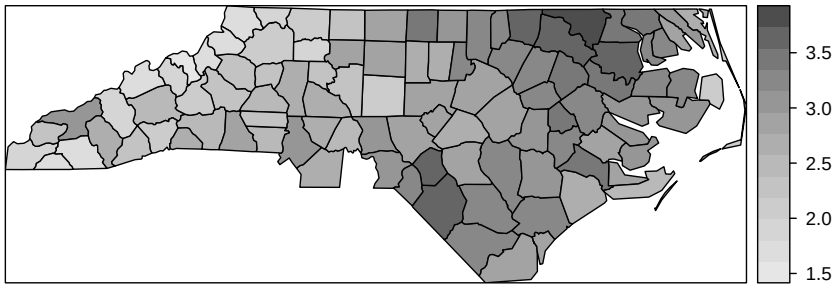

Figure 10: Fitted values for model IV, including covariate.

```
> ecarIawll <- spautolm(ft.SID74 ~ 1, data = nc2, listw = sids.nhbr.listw.4,
+     weights = BIR74, family = "CAR", llprof = seq(-0.1,
+         0.9020532358, length.out = 100))
> plot(ll ~ lambda, ecarIawll$llprof, type = "l")
```

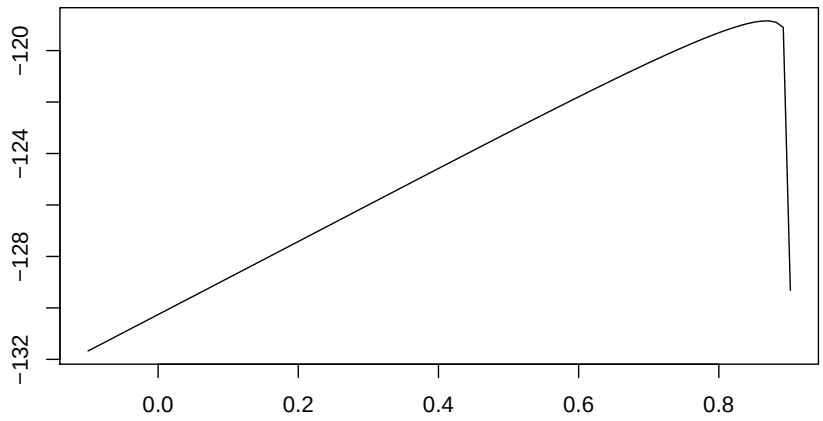

Figure 11: Plot of log likelihood function values by coefficient values, model I.

References

Anselin, L., Syabri, I., Smirnov, O., 2002. Visualizing Multivariate Spatial Correlation with Dynamically Linked Windows. In Anselin, L., Rey, S. (Eds.), Proceed-

ings, CSISS Workshop on New Tools for Spatial Data Analysis, Santa Barbara, CA, May 10-11, 2002. Center for Spatially Integrated Social Science, 20 pp., http://sal.agecon.uiuc.edu/csiss/pdf/multilisa.pdf.

Bailey, T. C., Gatrell, A. C., 1995. Interactive Spatial Data Analysis. Harlow: Longman, 413 pp.

Choynowski, M., 1959 Maps based on probabilities. Journal of the American Statistical Association, 54 (286), 385–388.

Cressie, N., 1991. Statistics for spatial data. New York: Wiley, 900 pp.

Cressie, N., Chan N. H., 1989. Spatial modelling of regional variables. Journal of the American Statistical Association, 84 (406), 393–401.

Cressie, N., Read, T. R. C., 1985. Do sudden infant deaths come in clusters?. Statistics and Decisions, Supplement Issue 2, 333–349.

Cressie, N., Read, T. R. C., 1989. Spatial data-analysis of regional counts. Biometrical Journal, 31 (6), 699–719.

Gómez Rubio, V., Ferrándiz, J., López, A., 2003 Detecting Disease Clusters with R. In: Hornik, K., Leisch, F., Zeilis, A. (Eds), Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria, 15 pp., (http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/GomezRubioEtAl.pdf).

Kaluzny, S. P., Vega, S. C., Cardoso, T. P., Shelly, A. A., 1996. S-PLUS SPATIALSTATS user's manual version 1.0. Seattle: MathSoft Inc., 226 pp.

Marshall, R. M., 1991. Mapping disease and mortality rates using Empirical Bayes Estimators. Applied Statistics, 40 (2), 283–294.

Symons, M. J., Grimson, R. C., Yuan, Y. C., 1983. Clustering of rare events. Biometrics, 39 (1), 193–205.