hw10.R

joebrew

Mon Nov 10 09:59:42 2014

```
######
# Attach packages
#######
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
##
## The following object is masked from 'package:stats':
##
##     nobs
##
## The following object is masked from 'package:utils':
##
##     object.size
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(maptools)
```

```
## Loading required package: sp
## Checking rgeos availability: TRUE
```

```
library(rgdal)
```

```
## rgdal: version: 0.9-1, (SVN revision 518)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 1.10.1, released 2013/08/26
## Path to GDAL shared files: /usr/share/gdal/1.10
## Loaded PROJ.4 runtime: Rel. 4.8.0, 6 March 2012, [PJ_VERSION: 480]
## Path to PROJ.4 shared files: (autodetected)
```

```
library(spatstat)
```

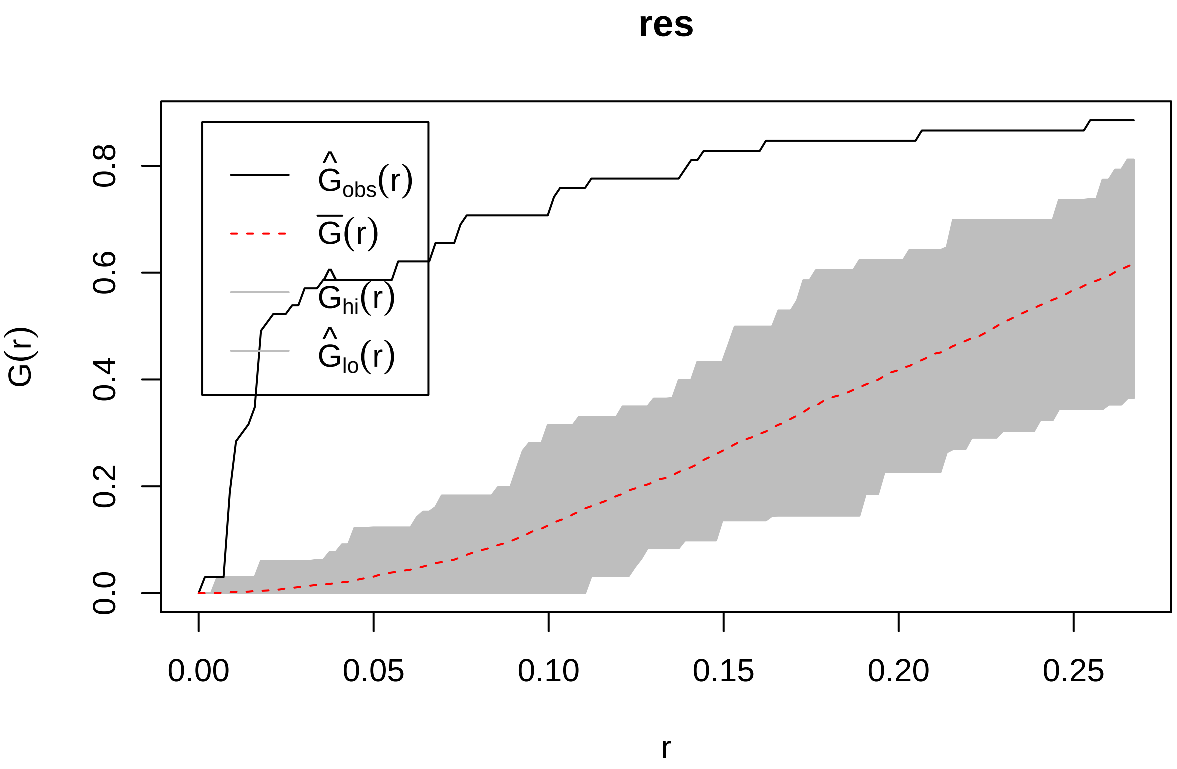
```
##
## spatstat 1.38-1       (nickname: 'Le Hardi')
## For an introduction to spatstat, type 'beginner'
```

```
######
# Set working directory to the haiti directories on local machine
######
if(Sys.info()["sysname"] == "Windows"){
  wd <- 'C:/Users/BrewJR/Documents/uf/phc6194/hw10'
} else {
  wd <- '/home/joebrew/Documents/uf/phc6194/hw10'
}
setwd(wd)

##################
# DIRECTIONS
##################
# For this assignment you will investigate clustering of P. falciparum parasite
# rates in an African country. Data and appropriate shapefiles are provided
# through Sakai. Please perform the following tasks and include all requested
# outputs and answers in a separate Word file numbered according to step:

##################
# 1. Choose a country to investigate, there are four options: Brazil,
#    Ghana, Kenya, and Tanzania. State your choice in your word file.
##################

# I choose.... Ghana!

##################
# 2. Import the World and country shapefile of choice into ArcGIS.
#    Import the country's data.
##################
ghana_map <- readOGR("Ghana", "Ghana")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "Ghana", layer: "Ghana"
## with 3 features and 8 fields
## Feature type: wkbPolygon with 2 dimensions
```

```
ghana_data <- read.csv("Ghana/Ghana_MAP_CSV.csv")

##################
# 3. Perform a global autocorrelation on the parasite rate
#    variable with either ArcGIS or GeoDa. If using GeoDa, right
#    click on plot and select "Display Statistics". Copy html
#    output if using ArcGIS or the Moran's I plot if using GeoDa.
#    Interpret these statistics.
##################
```

```
# Basically, see if the points I have are significantly different from
# random points

# Create point pattern dataset
coords.ppp <- ppp(x = ghana_data$longitude,
                  y = ghana_data$latitude,
                  xrange = range(ghana_data$longitude),
                  yrange = range(ghana_data$latitude))

# Define number of points
n <- coords.ppp$n

# Generate random points to compare with observed
ex <- expression(runifpoint(n, win = owin(c(range(ghana_data$longitude)),
                                            c(range(ghana_data$latitude)))))


# Set a seed to make reproducible
set.seed(130920)

# Use Gest to compute nearest neighbor distance function (g(r))
res <- envelope(coords.ppp,
                Gest,
                nsim = 99,
                simulate = ex,
                verbose = FALSE,
                savefuns = TRUE)

# Plot
plot(res)
```



```
##       lty col   key          label
## obs     1   1   obs hat(G)[obs](r)
## mmean   2   2 mmean       bar(G)(r)
## hi      1   8    hi  hat(G)[hi](r)
## lo      1   8    lo  hat(G)[lo](r)
##                                            meaning
## obs             observed value of G(r) for data pattern
## mmean              sample mean of G(r) from simulations
## hi    upper pointwise envelope of G(r) from simulations
## lo    lower pointwise envelope of G(r) from simulations
```

```
# In the above plot, the fact that the black line (observed)
# does not remain in the grey confidence area means
# that THERE IS SPATIAL AUTOCORRELATION.

# But the above demonstrates correlation between the points,
# not between the data values (PR)

# Moran's I (http://www.ats.ucla.edu/stat/r/faq/morans_i.htm)
library(ape)
```

```
##
## Attaching package: 'ape'
##
## The following objects are masked from 'package:spatstat':
##
##     edges, rotate
```

```r
# First generate a matrix of inverse distance weights
ghana_dists <- as.matrix(dist(cbind(ghana_data$longitude, ghana_data$latitude)))
ghana_dists_inv <- 1/ghana_dists
diag(ghana_dists_inv) <- 0

# Now, having created a matrix where each off-diagonal entry (i,j)
# in the matrix equals 1 / distance between points i and j,
# we can calcualte Moran's I.
x <- Moran.I(ghana_data$PR, ghana_dists_inv)
x
```

```
## $observed
## [1] 0.4769296
##
## $expected
## [1] -0.01515152
##
## $sd
## [1] 0.04395428
##
## $p.value
## [1] 0
```

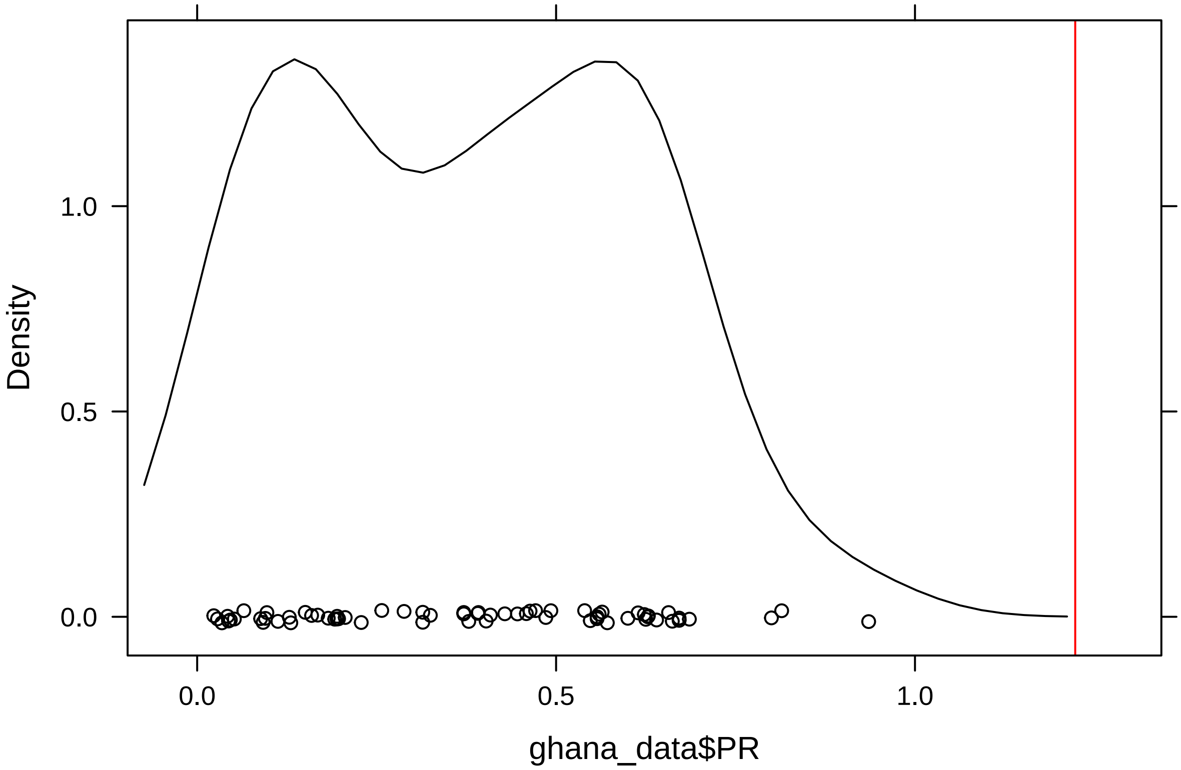
```r
# We can reject the null hypothesis that there is no
# spatial autocorrelation.  In other words,  PARASITE RATE
# IS SPATIALLY AUTOCORRELATED.

##################
# 4. Perform a local autocorrelation to show clusters
#    using either ArcGIS (either local Moran’s I or Getis-Ord Gi)
#    or GeoDa. Add any outputted maps to your Word file. Include
#    legends and titles as appropriate.
##################

# Hotspots
library(hotspots)
```

```
## Loading required package: lattice
##
## Attaching package: 'lattice'
##
## The following object is masked from 'package:spatstat':
##
##     panel.histogram
##
## Loading required package: ineq
```

```r
# Calculate a cutoff for t-distribution
x <- hotspots(ghana_data$PR, p = 0.95, tail = "positive")

# summarize
summary(x)
```

```
##
## Source data: ghana_data$PR
## Distribution and probability: t, 0.95
## Tail: positive hot spots only
## Mean:                 0.37669
## Median:               0.39093
## Min:                  0.02312
## Max:                  0.93548
## mad:                  0.30996
## CV (mad/median):  0.79287
##
## n =  67
##
## positive hot spots:
##   Cutoff  number positive hot spots  % positive hot spots   % sum
##    1.223                          0                      0      0
```

```
# plot
plot(x)
```



```
# Mantel test (http://www.ats.ucla.edu/stat/r/faq/mantel_test.htm)
library(ade4)
```

```
##
## Attaching package: 'ade4'
##
## The following object is masked from 'package:spatstat':
##
##     disc
```

```r
ghana_dists <- dist(cbind(ghana_data$longitude, ghana_data$latitude))
pr_dists <- dist(ghana_data$PR)
x <- mantel.rtest(pr_dists, ghana_dists, nrepet = 9999)
```

```
## Warning in is.euclid(m1): Zero distance(s)
```

```
## Warning in is.euclid(distmat): Zero distance(s)
```

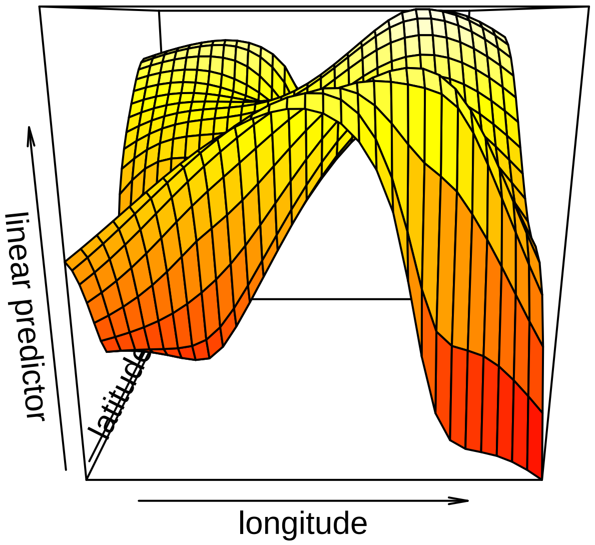
```r
x
```

```
## Monte-Carlo test
## Observation: 0.3128844
## Call: mantel.rtest(m1 = pr_dists, m2 = ghana_dists, nrepet = 9999)
## Based on 9999 replicates
## Simulated p-value: 1e-04
```

```r
# Using both Moran's I and Mantel (Monte Carlo simulations), we get
# very small P-values.  There is definitely spatial autocorrelation.


# GAM
library(mgcv)
```

```
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
##     collapse
##
## This is mgcv 1.8-3. For overview type 'help("mgcv-package")'.
```

```r
my_gam <- gam(PR ~ s(longitude, latitude),
              data = ghana_data)

# Visualize GAM
vis.gam(my_gam)
```



```
vis.gam(my_gam,
        ticktype = "detailed",
        n.grid = 100,
        plot.type = "contour",
        too.far = 20,
        contour.col = adjustcolor("black", alpha.f=0.4))
```

```
## Warning in plot.window(...): "ticktype" is not a graphical parameter
```

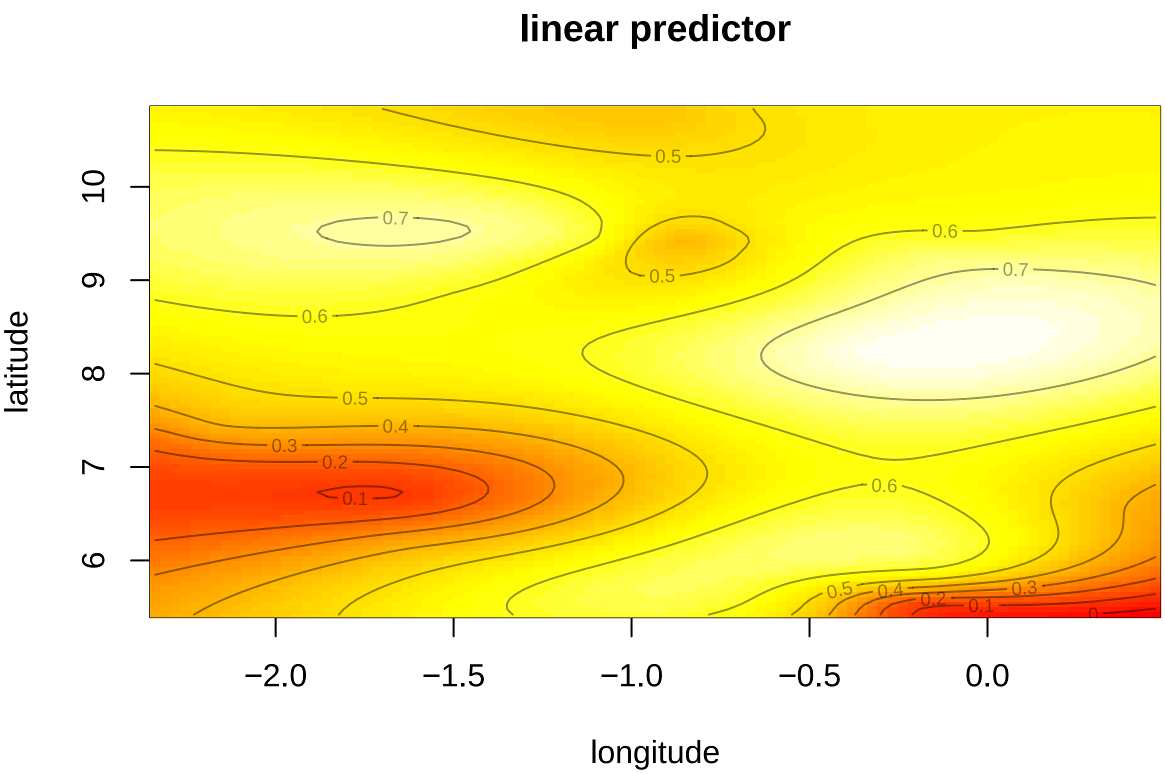
```
## Warning in plot.xy(xy, type, ...): "ticktype" is not a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "ticktype" is
## not a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "ticktype" is
## not a graphical parameter
```

```
## Warning in box(...): "ticktype" is not a graphical parameter
```

```
## Warning in title(...): "ticktype" is not a graphical parameter
```



```
# More visualizations: KRIGING
library(gstat)
```

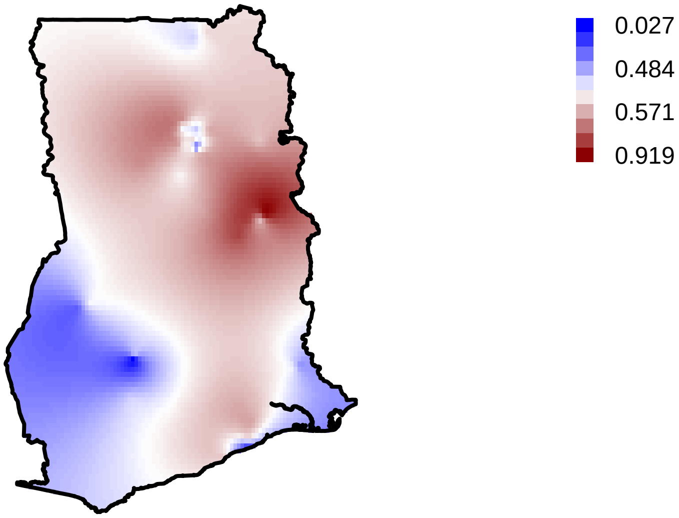
```
##
## Attaching package: 'gstat'
##
## The following object is masked from 'package:spatstat':
##
##     idw
```

```
library(geoR)
```

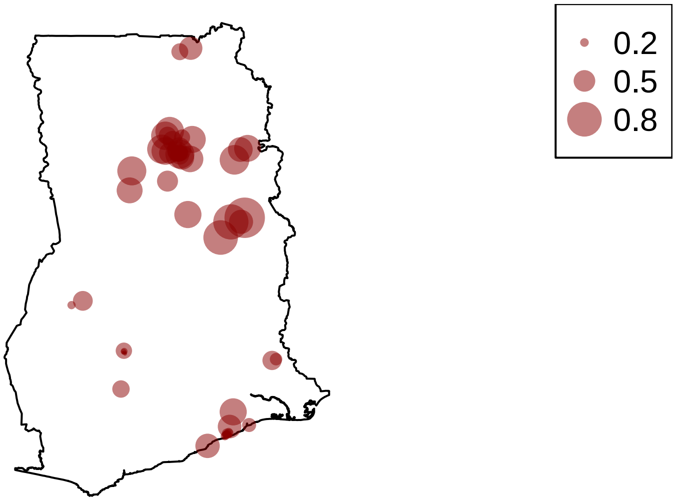
```
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
##
## --------------------------------------------------------------
##  Analysis of geostatistical data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-4.1 (built on 2012-06-29) is now loaded
## --------------------------------------------------------------
```

```
library(rgdal)
library(RColorBrewer)
# Define color vector
my_colors <- colorRampPalette(c("blue", "white", "darkred"))(100)

SurfaceFun <- function(var = "PR",
                       boundary_shape = ghana_map){


  # getting coordinates of alachua boundary
  boundary_points <- boundary_shape@polygons[[1]]@Polygons
  boundary_points <- boundary_points[[1]]@coords

  # Get trap locations and data values
  a <- data.frame("x" = ghana_data$longitude,
                  "y" = ghana_data$latitude,
                  "z" = ghana_data[,var])
  # Make into a geodata object
  b <- as.geodata(a)

  # Predict multiple points in Alachua County's boundary
  x <- seq(min(boundary_points[,1]), max(boundary_points[,1]), length = 100)
  y <- seq(min(boundary_points[,2]), max(boundary_points[,2]), length = 100)

  # Make a grid of those points
  pred.grid <- expand.grid(x,y)

  # kriging calculations
  kc <- krige.conv(geodata = b, coords = b$coords, data = b$data,
                   locations = pred.grid,
                   borders = boundary_points,
                   #borders = boundary@polygons,
                   # borders = ALACHUA BORDERS!,
                   krige = krige.control(type.krige = "ok",
                                         cov.pars = c(10, 3.33)))



  # Plot!
  # displaying predicted values
  image(kc, loc = pred.grid,
        col = my_colors,
        xlab=NA, ylab=NA,
        xaxt = "n",
        yaxt = "n",
        xpd = NA,
        bty = "n")

  # Define percentiles for legend
  legtemp <-  round(quantile(kc$predict, probs = seq(0,1,, length = 10)),
                    digits = 3)

  legend(x="topright",
         fill = my_colors[c(1,11,22,33,44,55,66,77,88,100)],
         legend = c(legtemp[1], NA, NA, legtemp[4], NA, NA, legtemp[7], NA, NA, legtemp[10]),
         border = NA,
         bty = "n",
         ncol = 1,
         y.intersp = 0.5,
         #title = "Interpolation",
         cex = 0.75)
}
SurfaceFun()
```

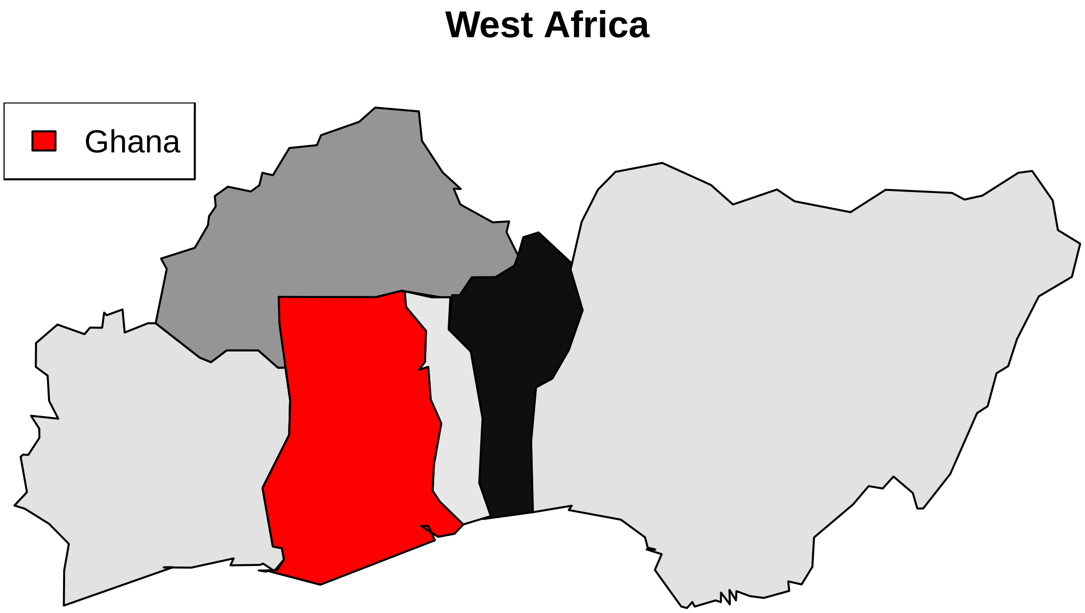
```
## krige.conv: results will be returned only for prediction locations inside the borders
## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood
```



```
# SIMPLE POINT MAP
plot(ghana_map)
points(ghana_data$longitude,
       ghana_data$latitude,
       pch = 16,
       col = adjustcolor("darkred", alpha.f = 0.5),
       cex = ghana_data$PR *3)
legend(x = "topright",
       pch = 16,
       col = adjustcolor("darkred", alpha.f = 0.5),
       pt.cex = c(0.2, 0.5, 0.8) * 3,
       legend = c(0.2, 0.5, 0.8) )
```



```
##################
# 5. Describe you what you see in your maps in a few sentences.
##################
# In all methods, there appears to be a cluster of high parasite rates
# in the eastern part of the country, at the northern age of lake Volta,
# along the Oti River, near the border with Togo.  Specifically, this
# parasite "hotspot" appears to be located in a triangle near
# Domanko, Nakpayili and Kpandai.

library(maps)
map("world", c("ghana", "togo", "burkina faso", "ivory coast",
               "nigeria", "benin"),
    fill = TRUE, col = sample(colorRampPalette(c("white", "black"))(100),6))
map("world", "ghana", fill = TRUE, col = "red", add = TRUE)
title(main = "West Africa")
legend("topleft",
       fill = "red",
       legend = "Ghana")
```


West Africa
🟥 Ghana