

CSSS 554: Homework 5

Department of Biostatistics @ University of Washington

Alejandro Hernandez

Winter Quarter 2025

1. Continuing on with your DHS analysis, in this question you will investigate whether it is important to account for the urban/rural stratification in the DHS design. Implement the stratification pipeline, the instructions for which can be found in the document on the class website. Run through the pipeline, and include the outputs in your report.

I was not able to complete this question because I could not get the `step2_UR_surface.R` script to run successfully. Everything before went well, and I went through the steps for both Benin and Malawi. I am uploading the files from Benin. Below I show the files generated from “Step 6. Create Reference DataFrames”. I believe the problem is from trying to access training data.

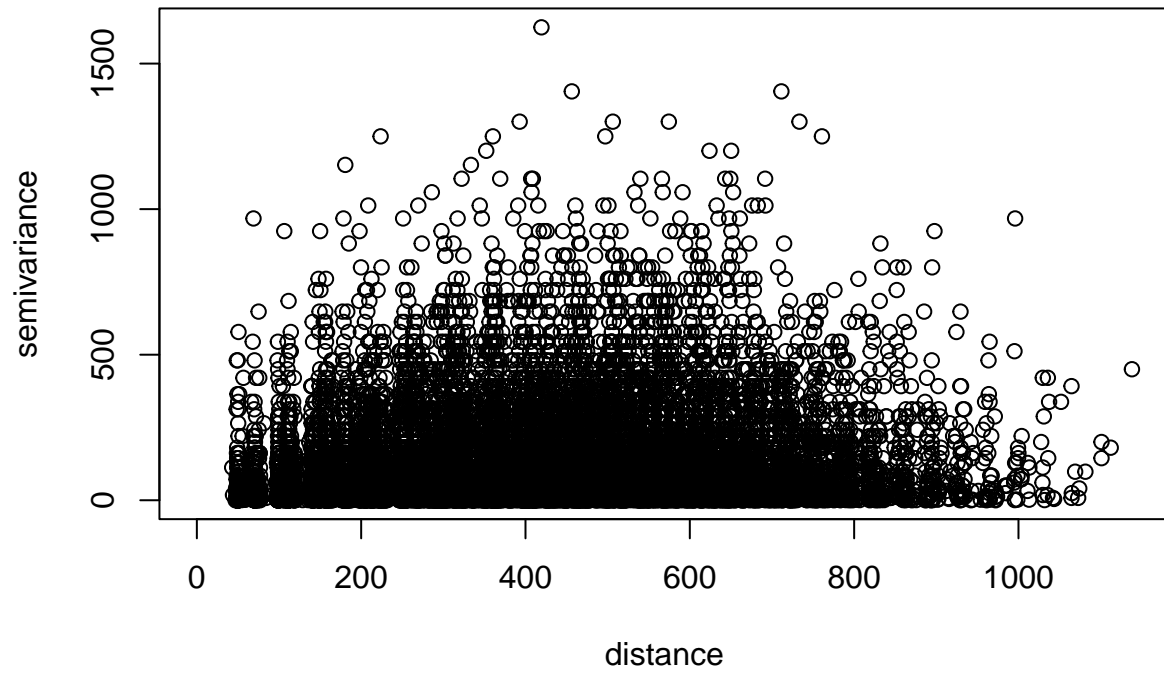
```
# Load administrative data file
getwd()
setwd("C:/Users/alega/Downloads/UR-Pipeline-Demo/UR-Pipeline-Demo/Data/Benin_2017/shapeFiles_gadm/")
country_shp_analysis <- readRDS("country_shp_analysis.rds")
head(country_shp_analysis[["Admin-1"]])

# Load the urban/rural reference tables
setwd("C:/Users/alega/Desktop/RProjects/spatial-analysis/data/Benin20172018")
ben_ref_tab <- readRDS("ben_ref_tab.rds")
ben_sample_ea <- readRDS("ben_sample_ea.rds")
ben_frame_ea <- readRDS("ben_frame_ea.rds")

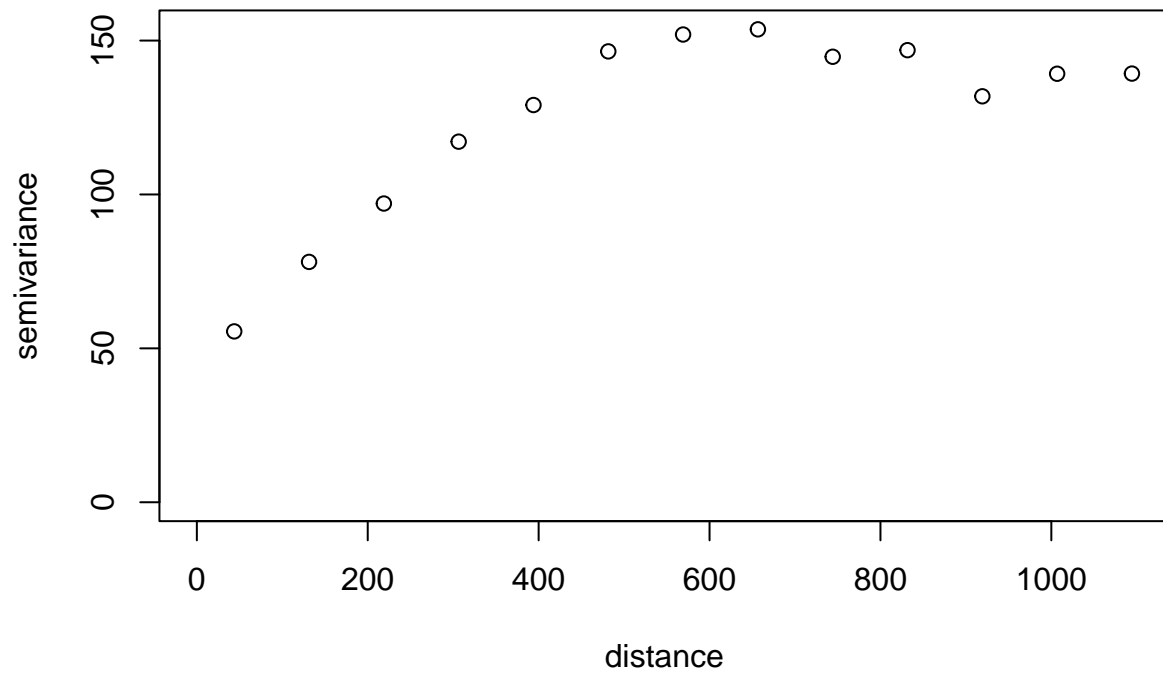
head(ben_ref_tab)
head(ben_sample_ea)
head(ben_frame_ea)
```

2. In the `geoR` library there are data `ca20` which we will explore/model using various geostatistical techniques.
 - (a) Examine cloud and binned semi-variograms and comment on the evidence for spatial dependence.

Variogram Cloud for ca20



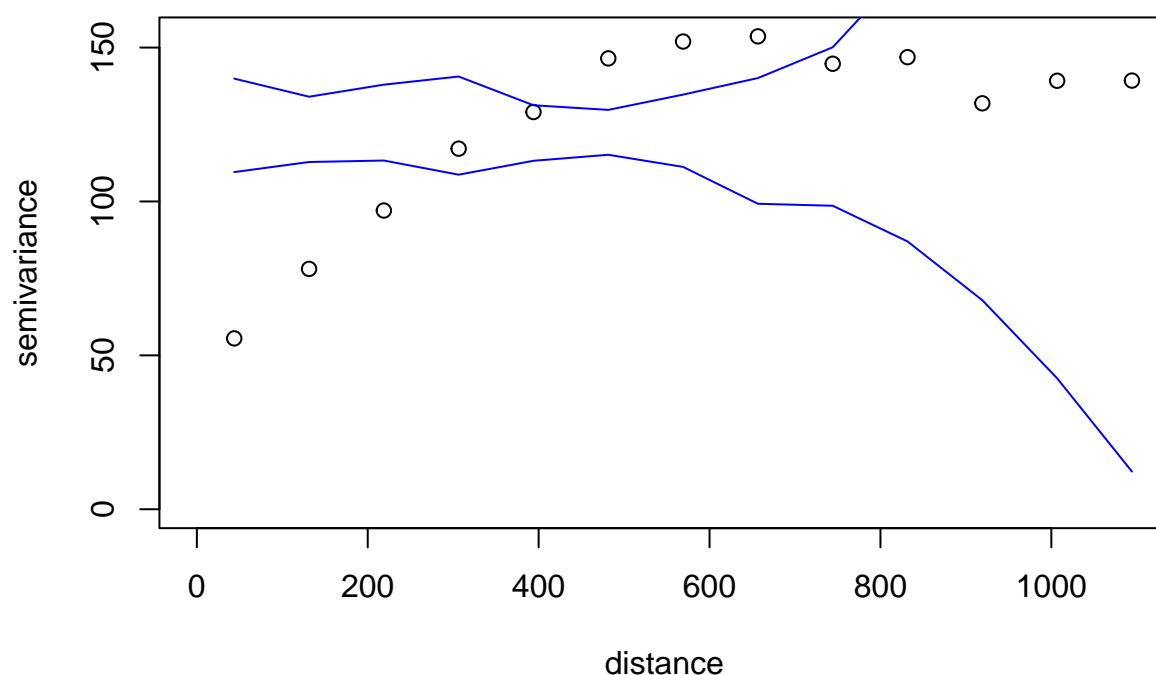
Binned Semi-Variogram for ca20



The semi-variogram, which measures the inverse of correlation, generally is increasing with distance, suggesting there is spatial dependence.

(b) Using the semi-variogram examine Monte Carlo intervals for no spatial dependence.

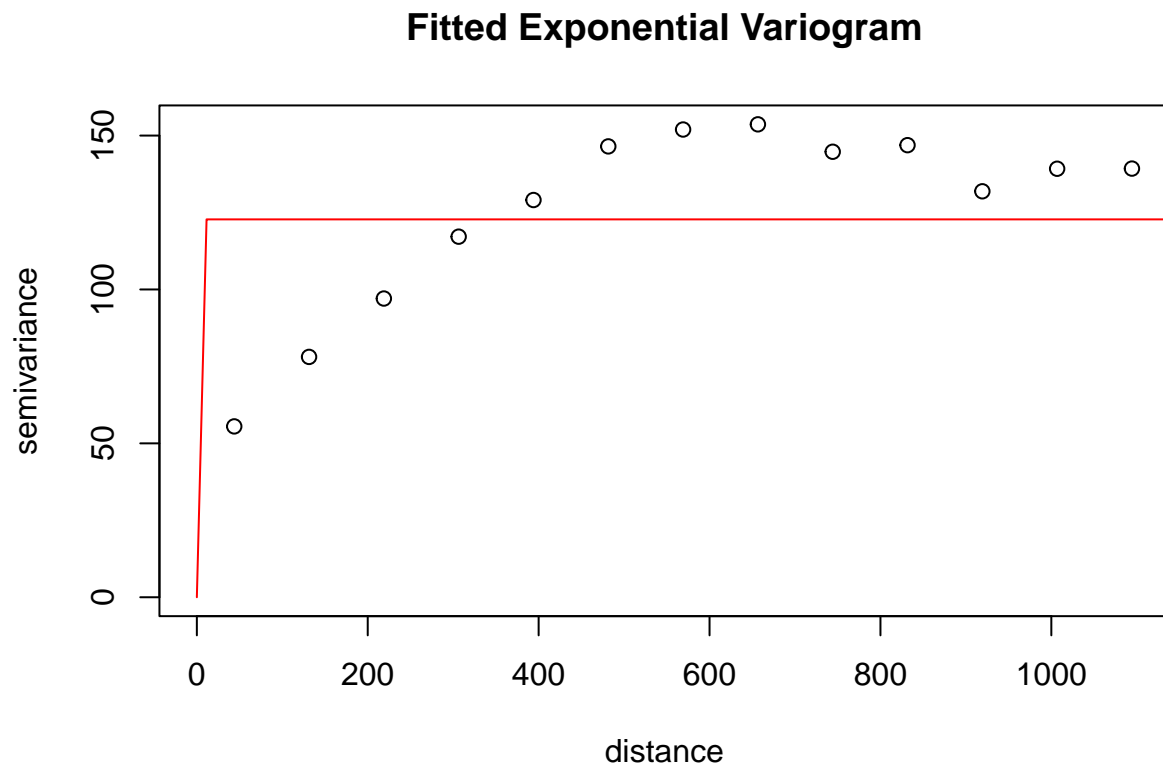
Binned Semi-Variogram with Monte Carlo Envelopes



The semi-variogram does not lie entirely within the envelope, which suggests significant spatial dependence. It does lower into it at high distances, but this may be because there are such few points that are that far from each other.

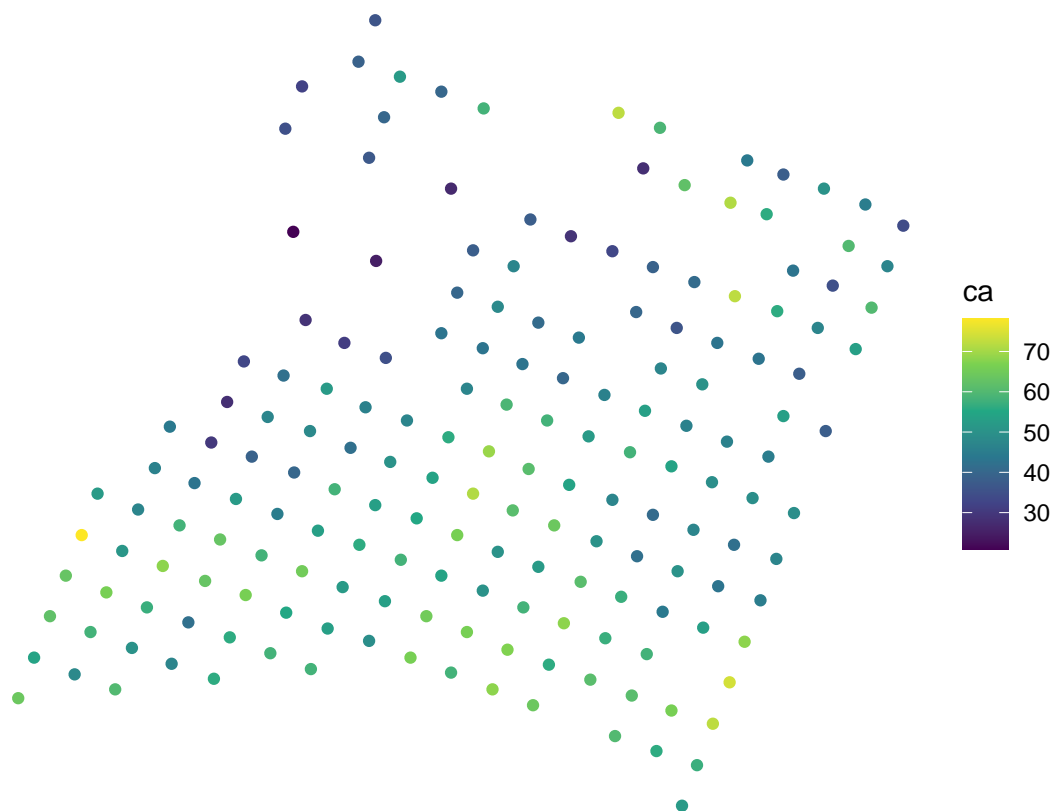
- (c) Fit an exponential variogram model to these data, using maximum likelihood or restricted maximum likelihood.

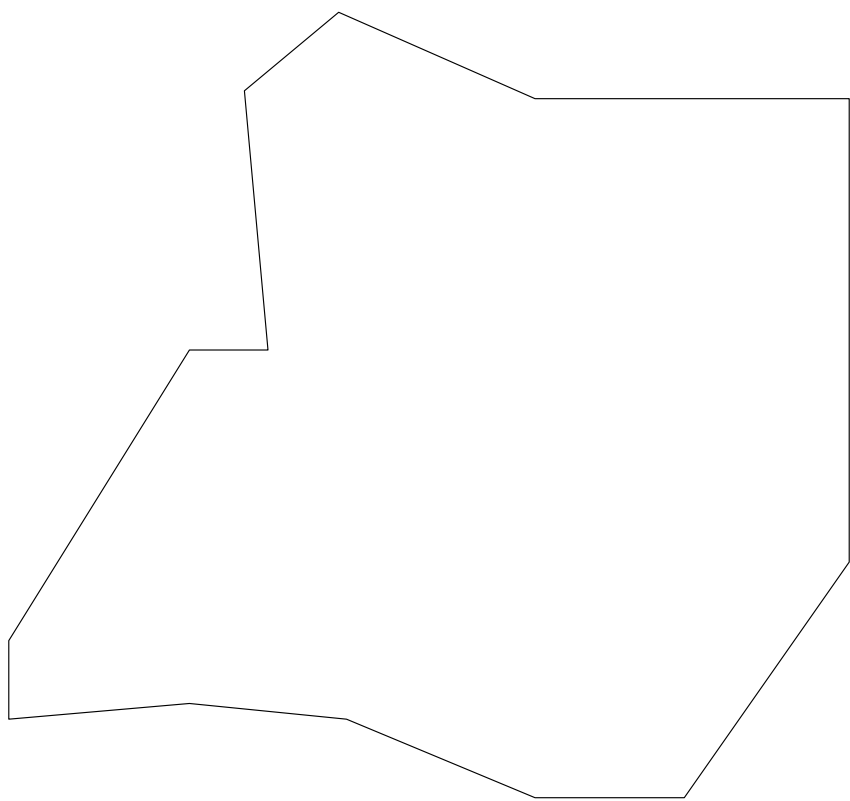
Below shows an exponential variogram model using restricted maximum likelihood, which is useful for smaller datasets.

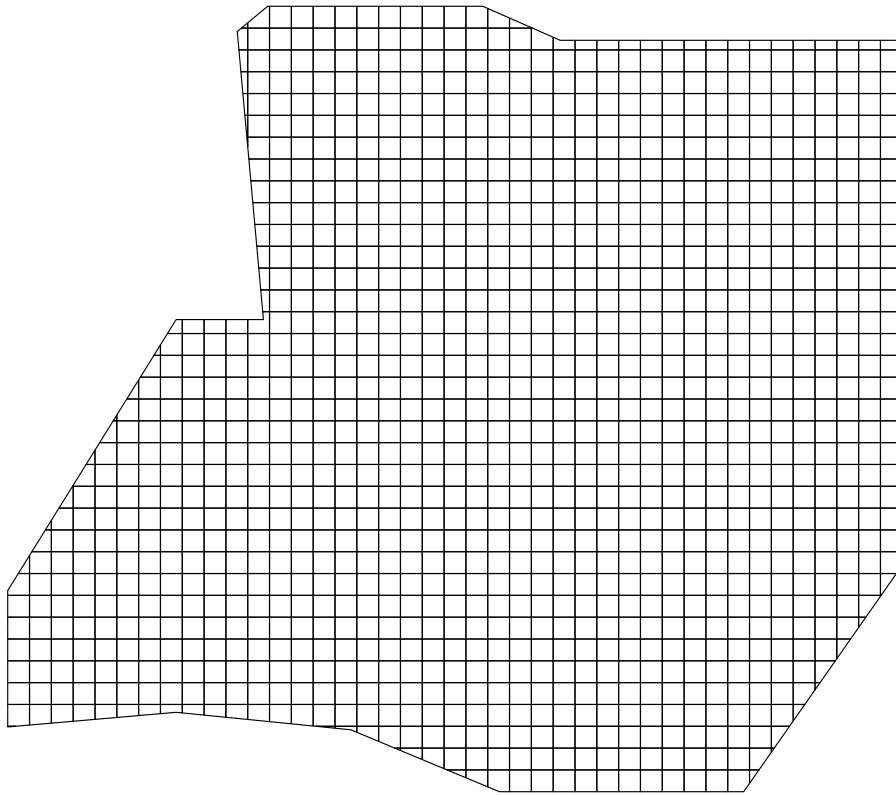


- (d) Carry out kriging and examine the resultant surface, both in terms of the mean and the standard deviation.

There were a number of points that fell outside or on the border, and their results were not calculated. I could not identify them and did not complete the last part of the question.







End of report. Code appendix begins on the next page.

Code Appendix

```
# Clear environment
rm(list=ls())

# Setup options
knitr::opts_chunk$set(echo=FALSE, warning=FALSE, message=FALSE, results='hide')
options(knitr.kable.NA = '-', digits = 2)
options(survey.lonely.psu="adjust")
labs = knitr::all_labels()
labs = labs[!labs %in% c("setup", "allcode")]

# Load relevant packages
library(geoR)
library(sf)
library(sp)
library(ggplot2)

## Load data
data(ca20)
summary(ca20)

# Load administrative data file
getwd()
setwd("C:/Users/alega/Downloads/UR-Pipeline-Demo/UR-Pipeline-Demo/Data/Benin_2017/shapeFiles_gadm/")
country_shp_analysis <- readRDS("country_shp_analysis.rds")
head(country_shp_analysis[["Admin-1"]])

# Load the urban/rural reference tables
setwd("C:/Users/alega/Desktop/RProjects/spatial-analysis/data/Benin20172018")
ben_ref_tab <- readRDS("ben_ref_tab.rds")
ben_sample_ea <- readRDS("ben_sample_ea.rds")
ben_frame_ea <- readRDS("ben_frame_ea.rds")

head(ben_ref_tab)
head(ben_sample_ea)
head(ben_frame_ea)

# Compute the variogram cloud
variogram_cloud <- geoR::variog(ca20, option = "cloud")
plot(variogram_cloud, main = "Variogram Cloud for ca20")

# Compute and plot the binned semi-variogram
variogram_binned <- variog(ca20, option = "bin")
plot(variogram_binned, main = "Binned Semi-Variogram for ca20")

# Generate Monte Carlo envelopes
mc_env <- variog.mc.env(ca20, obj.variog = variogram_binned)

# Plot with Monte Carlo envelopes
plot(variogram_binned, main = "Binned Semi-Variogram with Monte Carlo Envelopes")
lines(mc_env, col = "blue", lty = 2)
```

```

# Fit using MLE
variogram_model_mle <- likfit(ca20, ini = c(0.1, 0.5), cov.model = "exp")

# Fit using REML
variogram_model_reml <- likfit(ca20, ini = c(0.1, 0.5), cov.model = "exp", method = "REML")

# Print model summaries
summary(variogram_model_reml)

# Plot the fitted variogram
plot(variogram_binned, main = "Fitted Exponential Variogram")
lines(variogram_model_reml, col = "red")

# The MLE method estimates parameters by maximizing the likelihood.
# The REML method accounts for degrees of freedom, useful for small datasets.
# Compare nugget, sill, and range values for model interpretation.
data(ca20)
df_ca <- data.frame(ca = ca20$data, east = ca20$coords[,1], north = ca20$coords[,2])
df_ca <- st_as_sf(df_ca, coords = c("east", "north"))

# plot point data
gg_c_points <- ggplot(df_ca) + geom_sf(aes(color = ca)) +
  theme_void() + scale_color_viridis_c()
gg_c_points

# ca20 borders are already in the geodata object
# we just need to extract them and convert to polygon
borders_ca <- st_polygon(list(as.matrix(ca20$borders)))

# plot border polygon data
gg_c_border <- ggplot(borders_ca) + geom_sf(color = "black", fill = NA) +
  theme_void()
gg_c_border

grid_ca <- st_make_grid(df_ca, cellsize = 25, square = T)
grid_ca <- st_intersection(grid_ca, borders_ca)

# plot grids
gg_c_grid <- ggplot(grid_ca) + geom_sf(color = "black", fill = NA) +
  theme_void()
gg_c_grid

# extract coordinates for centroids to be used for prediction
centroids <- st_centroid(grid_ca)
pred_coords <- as.data.frame(st_coordinates(centroids))
names(pred_coords) <- c("x", "y")
head(pred_coords)

# convert polygon to sf so that we can add predictions to it
grid_ca <- st_sf(grid_ca)
# get predictions
krige_result <- krige.conv(ca20, loc=ca20$coords, krige = krige.control(obj.m = variogram_model_mle))

```

```
summary(krige_result)
# ca20$pred <- krige_result$predict # add kriging predictions

# plot
# ggplot(grid_ca) + geom_sf(aes(fill = pred), color = NA) +
# theme_void() + scale_fill_viridis_c()
```

End of document.