Spatial Statistics | Homework 1

Brian Cervantes Alvarez

January 20, 2025

Problem 1

Part A

We need to show that,

$$\det(\Sigma)^{-1/2} = \frac{1}{\sigma_1\sigma_2\sqrt{1-\rho^2}}$$

Now, recall the determinant formula for a 2x2 matrix,

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc.$$

Here, $a = \sigma_1^2$, $b = \rho\sigma_1\sigma_2$, $c = \rho\sigma_1\sigma_2$, and $d = \sigma_2^2$.

Then, we can plug these into the determinant formula and yield,

$$\det(\Sigma) = (\sigma_1^2)(\sigma_2^2) - (\rho\sigma_1\sigma_2)^2.$$

After this, we simplify,

$$\det(\Sigma) = \sigma_1^2\sigma_2^2(1-\rho^2).$$

Next, we take the inverse square root and gives us,

$$\det(\Sigma)^{-1/2} = \frac{1}{\sqrt{\sigma_1^2\sigma_2^2(1-\rho^2)}}.$$

Finally, we can split the terms under the square root,

$$\det(\Sigma)^{-1/2} = \frac{1}{\sigma_1\sigma_2\sqrt{1-\rho^2}}.$$

Part B (skipped)

Part C

We need to arrive to this,

$$(\mathbf{y}-\mu)'\Sigma^{-1}(\mathbf{y}-\mu) = \frac{1}{1-\rho^2}\left(\left(\frac{y_1}{\sigma_1}\right)^2 + 2\rho\frac{y_1 y_2}{\sigma_1\sigma_2} + \left(\frac{y_2}{\sigma_2}\right)^2\right).$$

First, can write the inverse of the covariance matrix as follows,

$$\Sigma^{-1} = \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)}\begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix}.$$

Then, substitute $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ and $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ into $(\mathbf{y}-\mu)$:

$$(\mathbf{y}-\mu) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

Lastly, we expand $(\mathbf{y}-\mu)'\Sigma^{-1}(\mathbf{y}-\mu)$ and arrive to,

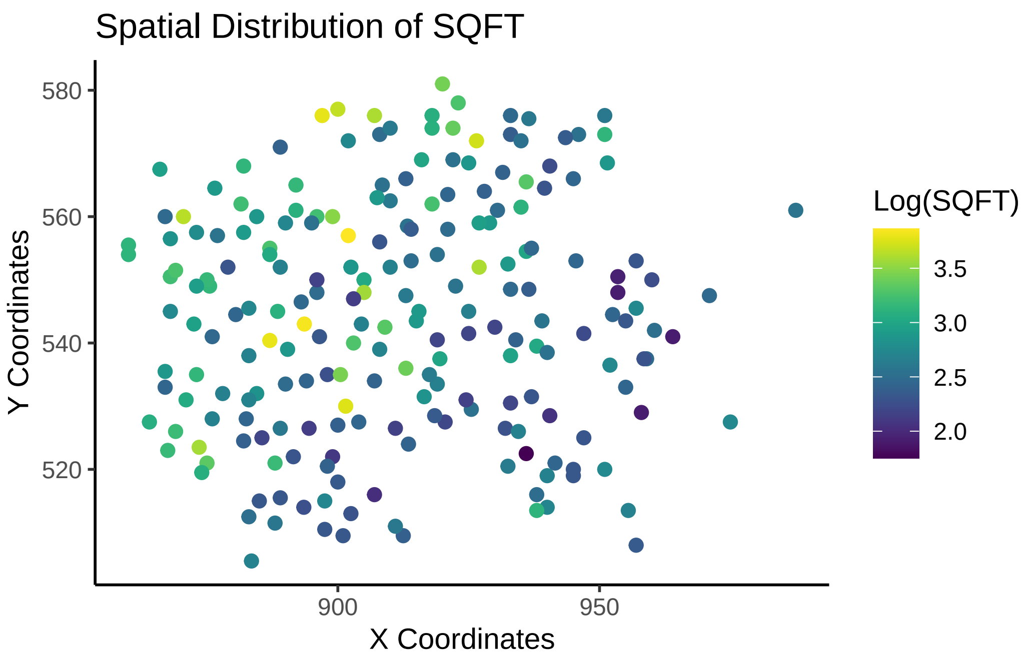
$$(\mathbf{y}-\mu)'\Sigma^{-1}(\mathbf{y}-\mu) = \frac{1}{1-\rho^2}\left[\frac{y_1^2}{\sigma_1^2} + 2\rho\frac{y_1 y_2}{\sigma_1\sigma_2} + \frac{y_2^2}{\sigma_2^2}\right].$$

Problem 2

Part A

```r
# Run this first and install required libraries
required_packages <- c("spData", "sp", "gstat", "dplyr", "ggplot2",
  "viridis", "sf")
installed_packages <- rownames(installed.packages())
for (pkg in required_packages) {
  if (!(pkg %in% installed_packages)) {
    install.packages(pkg, dependencies = TRUE)
  }
}
```

```r
library(spData)
library(sp)
library(gstat)
library(dplyr)
library(viridis)
library(ggplot2)
library(sf)

data("baltimore")

ggplot(baltimore, aes(x = X, y = Y, color = log(SQFT))) +
  geom_point(size = 2) +
  scale_color_viridis(name = "Log(SQFT)") +
  labs(title = "Spatial Distribution of SQFT",
       x = "X Coordinates",
       y = "Y Coordinates") +
  theme_classic() +
  theme(legend.position = "right")
```



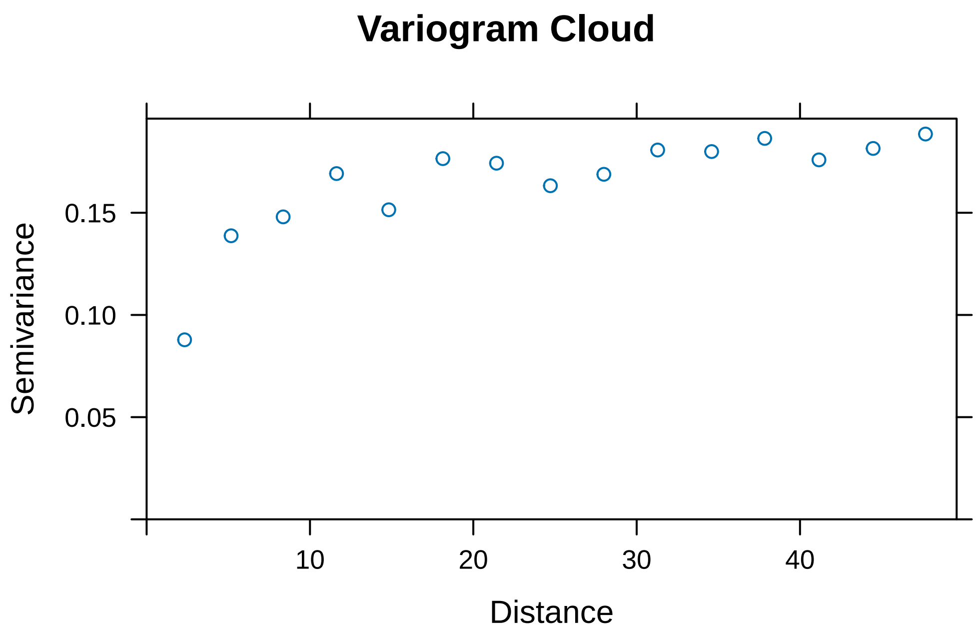
Spatial Distribution of SQFT
580
560
540
520
Y Coordinates
900
950
X Coordinates
Log(SQFT)
3.5
3.0
2.5
2.0

Part B

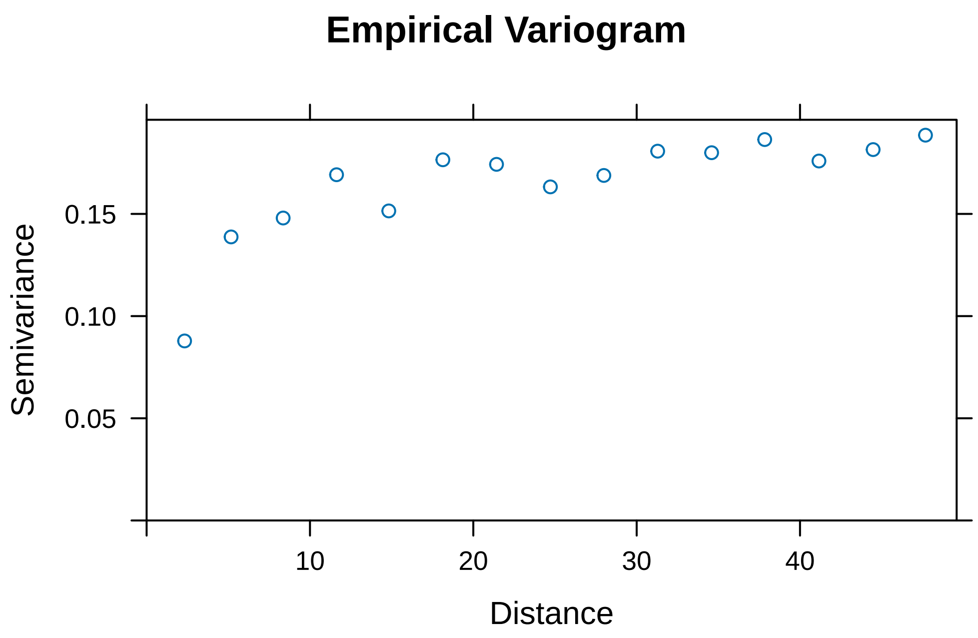
```r
# Log-transform SQFT
baltimore$LOG_SQFT <- log(baltimore$SQFT)
glimpse(baltimore)
```

```
Rows: 211
Columns: 18
$ STATION  <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18~
$ PRICE    <dbl> 47.0, 113.0, 165.0, 104.3, 62.5, 70.0, 127.5, 53.0, 64.5, 145~
$ NROOM    <dbl> 4, 7, 7, 7, 7, 6, 6, 8, 6, 7, 6, 5, 4, 5, 5, 9, 5, 5, 5, 6, 5~
$ DWELL    <dbl> 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1~
$ NBATH    <dbl> 1.0, 2.5, 2.5, 2.5, 1.5, 2.5, 2.5, 1.5, 1.0, 2.5, 2.0, 2.0, 2~
$ PATIO    <dbl> 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0~
$ FIREPL   <dbl> 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ AC       <dbl> 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0~
$ BMENT    <dbl> 2, 2, 3, 2, 2, 3, 3, 0, 3, 3, 2, 0, 3, 3, 2, 2, 2, 3, 2, 3, 3~
$ NSTOR    <dbl> 3, 2, 2, 2, 2, 3, 1, 3, 2, 2, 2, 1, 1, 2, 2, 3, 2, 2, 2, 2, 1~
$ GAR      <dbl> 0, 2, 2, 2, 0, 1, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ AGE      <dbl> 148, 9, 23, 5, 19, 20, 20, 22, 22, 4, 23, 20, 30, 20, 18, 75,~
$ CITCOU   <dbl> 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1~
$ LOTSZ    <dbl> 5.70, 279.51, 70.64, 174.63, 107.80, 139.64, 250.00, 100.00, ~
$ SQFT     <dbl> 11.25, 28.92, 30.62, 26.12, 22.04, 39.42, 21.88, 36.72, 25.60~
$ X        <dbl> 907.0, 922.0, 920.0, 923.0, 918.0, 900.0, 918.0, 907.0, 918.0~
$ Y        <dbl> 534.0, 574.0, 581.0, 578.0, 574.0, 577.0, 576.0, 576.0, 562.0~
$ LOG_SQFT <dbl> 2.420368, 3.364533, 3.421653, 3.262701, 3.092859, 3.674273, 3~
```

```r
# Variogram cloud
vgCloud <- variogram(LOG_SQFT ~ 1, ~X + Y, data = baltimore)
plot(vgCloud, main = "Variogram Cloud",
 xlab = "Distance",
 ylab = "Semivariance")
```

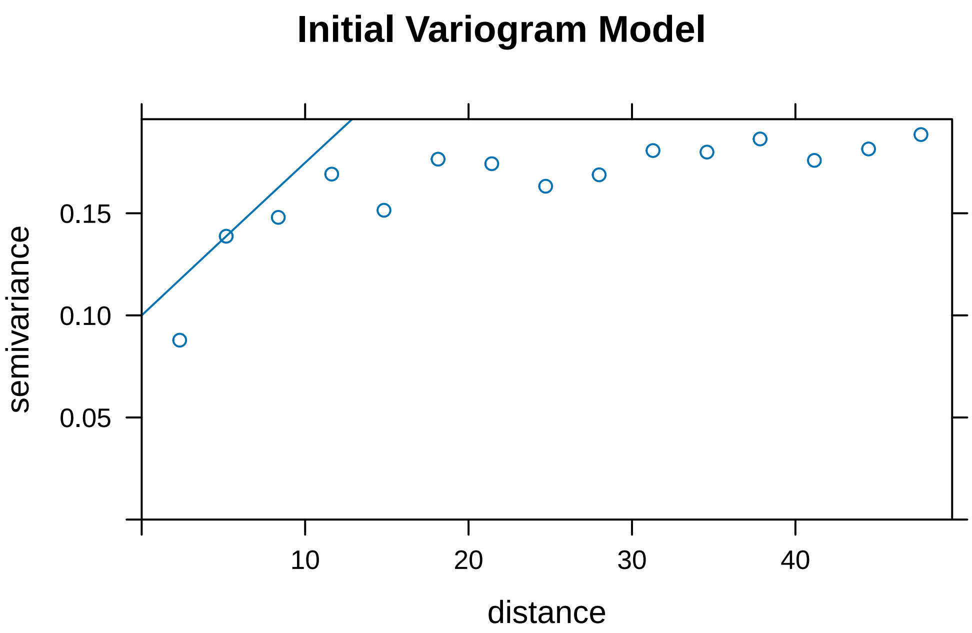



```r
# Empirical variogram
vgEmp <- variogram(LOG_SQFT ~ 1, ~X + Y, data = baltimore)
plot(vgEmp, main = "Empirical Variogram",
 xlab = "Distance",
 ylab = "Semivariance")
```



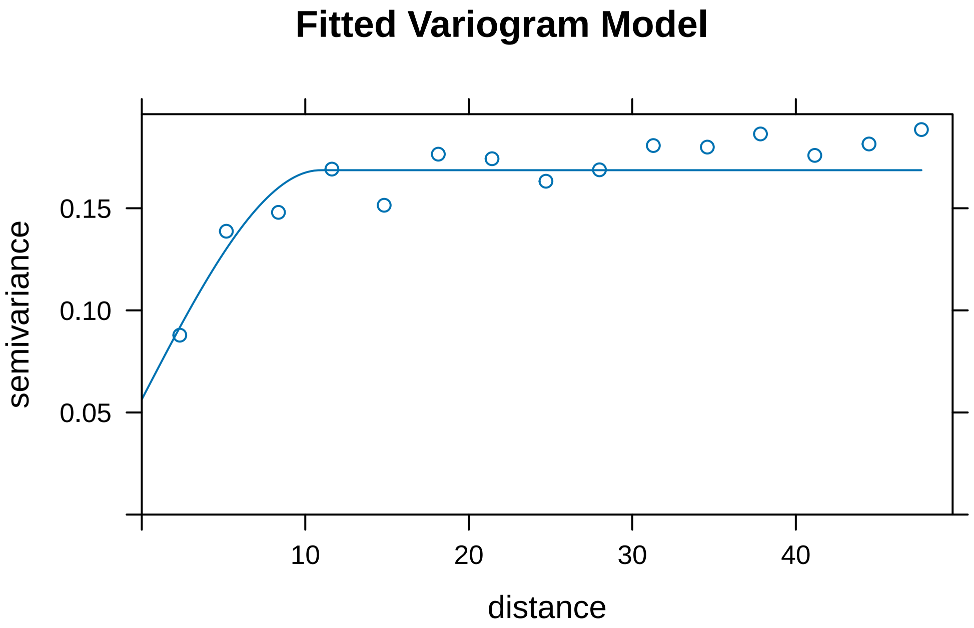
Part C

```r
# Fit an initial variogram model
vgModel <- vgm(psill = 0.5, model = "Sph", range = 100, nugget = 0.1)

# Plot the model
plot(vgEmp, model = vgModel, main = "Initial Variogram Model")
```



Part D

```r
# Fit the variogram
vgFit <- fit.variogram(vgEmp, model = vgModel)

# Plot the fitted model
plot(vgEmp, model = vgFit, main = "Fitted Variogram Model")
```



Part E

```
baltimore <- st_as_sf(baltimore, coords = c("X", "Y"), crs = 32617)

# Define new locations for prediction
locations <- data.frame(X = c(921, 980), Y = c(551, 580))
locations <- st_as_sf(locations, coords = c("X", "Y"), crs = st_crs(baltimore))

# Perform kriging
krigResults <- krige(LOG_SQFT ~ 1, locations = baltimore,
 newdata = locations, model = vgFit)
```

[using ordinary kriging]

```
print(krigResults)
```

```
Simple feature collection with 2 features and 2 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 921 ymin: 551 xmax: 980 ymax: 580
Projected CRS: WGS 84 / UTM zone 17N
  var1.pred  var1.var        geometry
1  2.668296 0.1209852 POINT (921 551)
2  2.686720 0.1703505 POINT (980 580)
```

Part F

i.

The prediction interval at `(980, 580)` is wider because the variance (`0.1703505`) is larger than that of `(921, 551)` (`0.1209852`). This makes sense, as predictions farther from observed data points generally have higher uncertainty due to reduced spatial correlation.

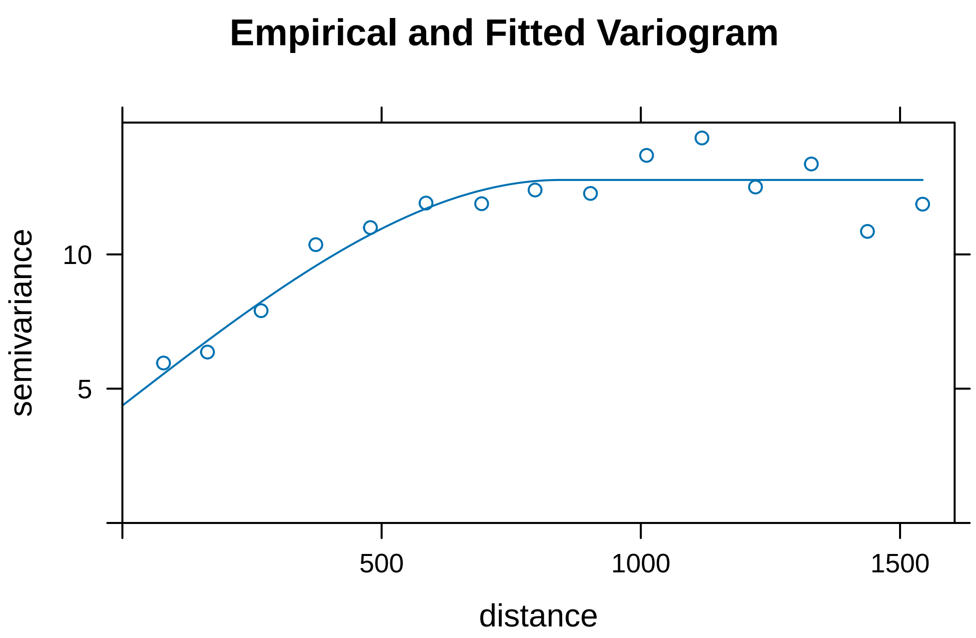
ii.

Predicting square footage based solely on spatial location can make sense if there is strong spatial correlation in the dataset, as houses in close proximity often share similar characteristics. However, square footage is likely influenced by additional factors like the number of rooms, age, or property features, which are ignored in this model.

Problem 3

Part A

```
data("meuse")

tidyMeuse <- meuse[!is.na(meuse$om), ]
tidyMeuse <- st_as_sf(tidyMeuse, coords = c("x", "y"), crs = 32631)

# Compute the empirical variogram
vgEmp <- variogram(om ~ 1, data = tidyMeuse)

# Fit a variogram model
vgModel <- vgm(psill = 0.5, model = "Sph", range = 100, nugget = 0.1)
vgFit <- fit.variogram(vgEmp, model = vgModel)

# Plot the empirical and fitted variogram
plot(vgEmp, model = vgFit, main = "Empirical and Fitted Variogram")
```



Part B

```
# Predict missing "om" values using the fitted variogram
missingMeuse <- meuse[is.na(meuse$om), ]
missingMeuse <- st_as_sf(missingMeuse, coords = c("x", "y"), crs = 32631)

krigResultsA <- krige(om ~ 1, locations = tidyMeuse,
 newdata = missingMeuse, model = vgFit)
```

[using ordinary kriging]

```
krigResultsA
```

```
Simple feature collection with 2 features and 2 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 180451 ymin: 332175 xmax: 180561 ymax: 332193
Projected CRS: WGS 84 / UTM zone 31N
   var1.pred var1.var              geometry
43  2.923869 7.639193 POINT (180561 332193)
44  3.898807 7.109570 POINT (180451 332175)
```

Part C

```
vgModelAlt <- vgm(psill = 0.7, model = "Exp", range = 120, nugget = 0.2)
vgFitAlt <- fit.variogram(vgEmp, model = vgModelAlt)

krigResultsB <- krige(om ~ 1, locations = tidyMeuse,
 newdata = missingMeuse, model = vgFitAlt)
```

```
[using ordinary kriging]
```

```
krigResultsB
```

```
Simple feature collection with 2 features and 2 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 180451 ymin: 332175 xmax: 180561 ymax: 332193
Projected CRS: WGS 84 / UTM zone 31N
   var1.pred var1.var              geometry
43  2.986237 8.021462 POINT (180561 332193)
44  3.711300 7.387392 POINT (180451 332175)
```

```
# Compare predictions from both models
comparison <- data.frame(
  Coordinates = st_coordinates(missingMeuse),
  PredictionModelA = krigResultsA$var1.pred,
  PredictionModelB = krigResultsB$var1.pred
)
comparison
```

```
  Coordinates.X Coordinates.Y PredictionModelA PredictionModelB
1        180561        332193         2.923869         2.986237
2        180451        332175         3.898807         3.711300
```

The predictions differ slightly between the two variogram models, with the differences reflecting the alternative model's higher sill and range, which increase uncertainty.