HW3: Discrete, fixed spatial index

Fern Bromley

October 17, 2025

I. Mathematical [6 pts]

(1) [6 pts]

(Option 2) Ch. 7.9 Exercise 12. (a) from Zimmerman and Ver Hoef:

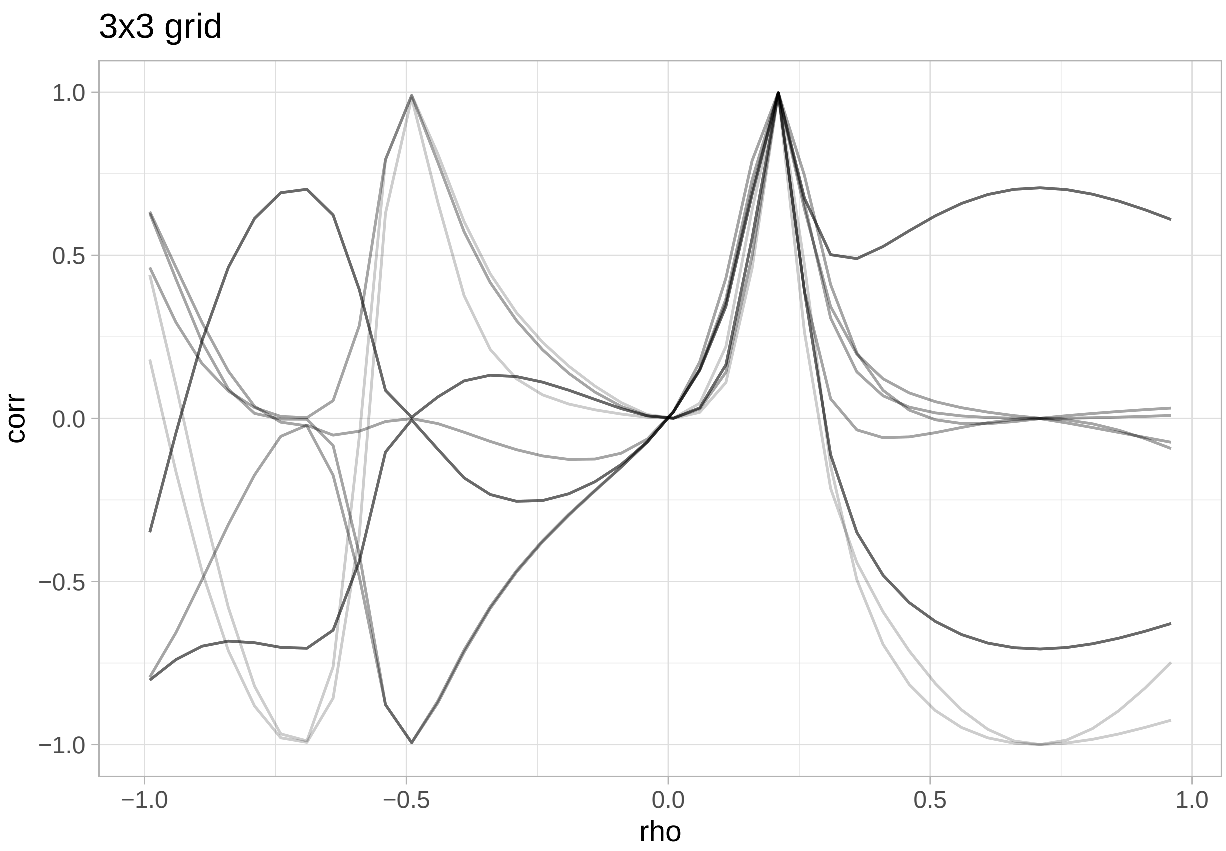
12. Construct figures analogous to Figure 7.3, showing the marginal correlations corresponding to adjacent site-pairs in 3×3 and 6×6 square lattices, for each of the following, and comment on each figure:

(a) SAR models with $\mathbf{W}$ a binary adjacency matrix and $-1 \leq \rho_{\text{SAR}} \leq 1$

```r
# Make grids:

library(sf)
library(spdep)

points3 <- st_as_sf(data.frame(x = c(1, 2, 3, 4),
                               y = c(1, 2, 3, 4)),
                    coords = c("x", "y"))
grid3 <- st_make_grid(points3, cellsize = 1)

points6 <- st_as_sf(data.frame(x = c(1, 2, 3, 4, 5, 6, 7),
                               y = c(1, 2, 3, 4, 5, 6, 7)),
                    coords = c("x", "y"))
grid6 <- st_make_grid(points6, cellsize = 1)

sig <-  1 # set variance

library(dplyr)
library(ggplot2)

# Compute weight matrices:

w3 <- st_intersects(grid3, remove_self = T, sparse = F)*1
w6 <- st_intersects(grid6, remove_self = T, sparse = F)*1

rho <- seq(from = -0.99, to = 0.99, by = 0.05)
```

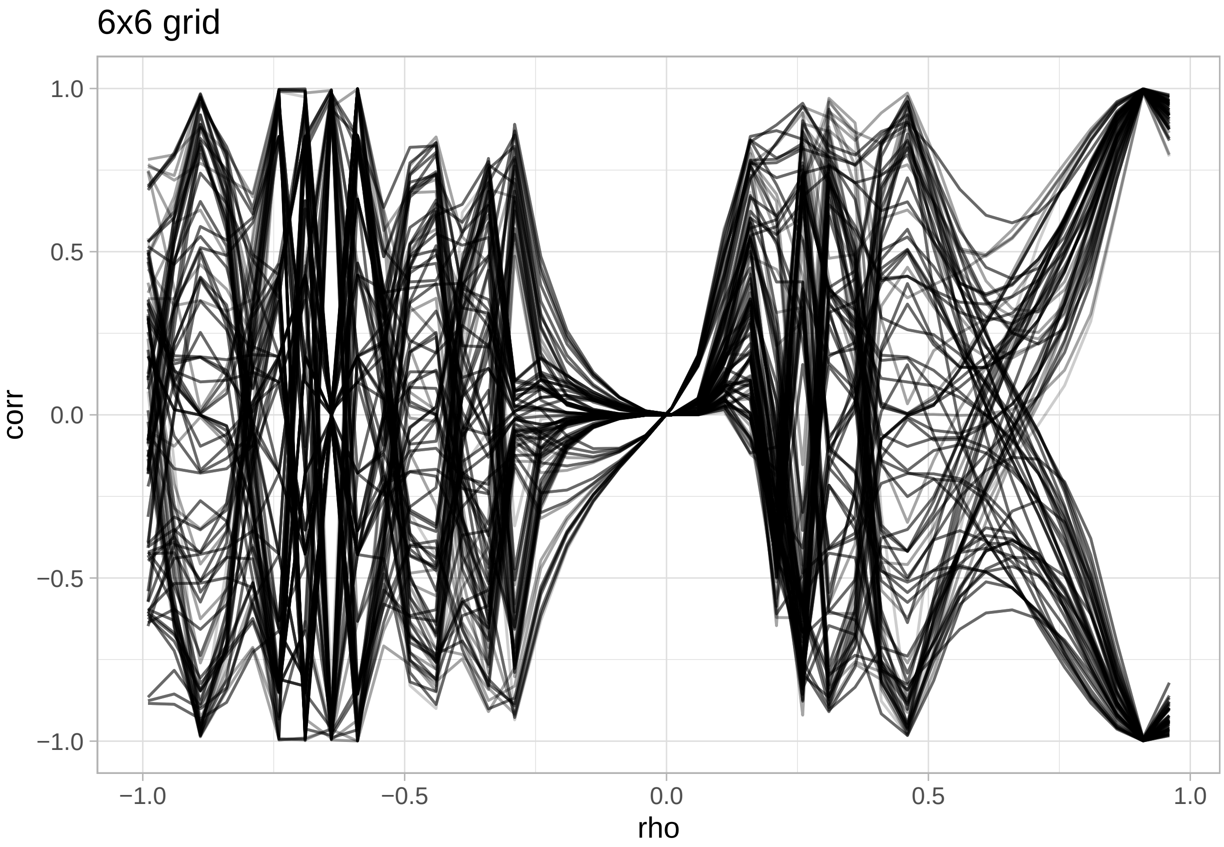
```r
# for 3x3 grid:
n3 <- length(grid3)
# w3_s <- w3/rowSums(w3)

cor_data3 <- data.frame()
```

```r
for(k in 1:length(rho)){
  inv_IminusB <- solve(diag(n3) - (rho[k]*w3))
  sigma3 <- sig*(inv_IminusB %*% t(inv_IminusB))
  cor3 <- cov2cor(sigma3)

      for(i in 1:n3){
        for(j in (i+1):n3){
          if (j <= n3){
                  cor_data3 <- rbind(cor_data3, data.frame(
                      rho = rho[k],
                      site1 = i,
                      site2 = j,
                      corr = cor3[i, j]))
                    }
              }
       }
}

cor_data3 <- cor_data3 %>%
  filter(site1 != site2) %>%
  mutate(site_pair = paste(site1, "-", site2))

ggplot(cor_data3, aes(x = rho, y = corr))+
  geom_line(aes(group = site_pair), alpha = 0.1)+
  ggtitle("3x3 grid")+
  theme_light()
```



```
# for 6x6 grid:
n6 <- length(grid6)
# w6_s <- w6/rowSums(w6)

cor_data6 <- data.frame()
for(k in 1:length(rho)){
  inv_IminusB <- solve(diag(n6) - (rho[k]*w6))
  sigma6 <- sig*(inv_IminusB %*% t(inv_IminusB))
  cor6 <- cov2cor(sigma6)

      for(i in 1:n6){
        for(j in (i+1):n6){
          if (j <= n6){
                  cor_data6 <- rbind(cor_data6, data.frame(
                      rho = rho[k],
                      site1 = i,
                      site2 = j,
                      corr = cor6[i, j]))
                    }
              }
       }
}

cor_data6 <- cor_data6 %>%
  filter(site1 != site2) %>%
  mutate(site_pair = paste(site1, "-", site2))
```

```r

ggplot(cor_data6, aes(x = rho, y = corr))+
  geom_line(aes(group = site_pair), alpha = 0.1)+
  ggtitle("6x6 grid")+
  theme_light()
```



For both sets of grids using non-standardized adjacency matrices, all site pairs have marginal correlations of zero at rho = 0. However, there does not seem to be a consistent pattern in marginal correlations tending to decrease or increase as rho > 0. For the 3x3 grid, marginal correlations are > 0 when 0 < rho < approx. 0.2, and for the 6x6 grid, marginal correlations are > 0 when 0 < rho < approx 0.1. This is admittedly not intuitive to me (how can positive spatial dependence lead to negative marginal correlations?) so I'm hoping I coded this right. When I row standardize the adjacency matrices, which I tend to interpret as dependency getting spread across neighbors, the plots look more reasonable.

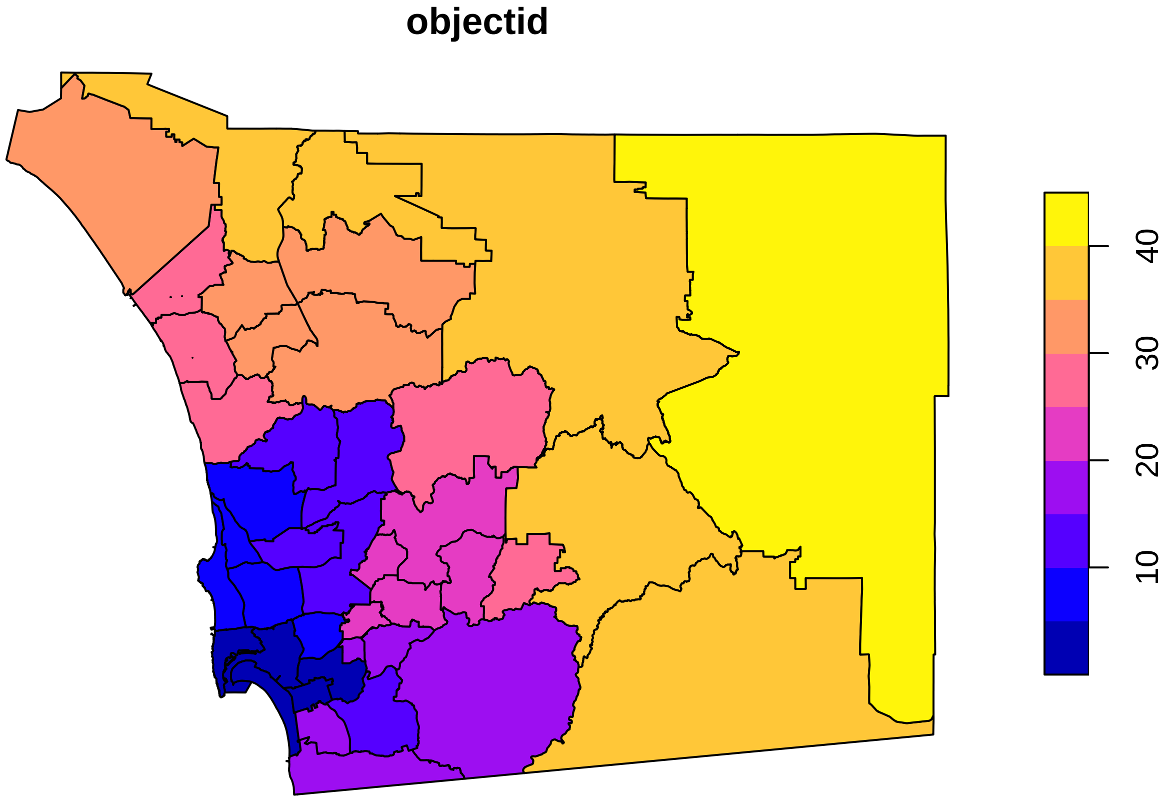
I. Exploring the data [17 pts]

For this homework, you'll be analyzing data related to cyclist injuries in San Diego County. The County reports counts of cyclist injuries for "subregional areal" (SRA) units. SRAs are defined by The County for their own use. Roughly, each areal unit is made up of a union of several census tracts (Figure 1; SEE PDF ON GRADESCOPE). The shapefiles for this assignment were obtained **here** (the **San Diego Regional Data Warehouse** has several other publicly available spatial datasets). Download the file `Subregional_Areas_2020_shapefile.zip` from D2L and unzip it in your working directory.

(2) [2 pts] **Provide executable code** (**PEC**) that uses the `sf` package to read the shapefiles in the `Subregional_Areas_2020_shapefile/` directory into R as a simple features object called `SRA`. How

many SRAs are there? Verify your SRA boundaries look like the ones in Figure 1 (SEE PDF ON GRADESCOPE).

```r
library(sf)
SRA <- st_read("Subregional_Areas_2020.shp") %>%
  st_as_sf()

plot(SRA[1])
```



```r
length(unique(SRA$geometry))
```

There are 41 subregional areas.

(3) [1 pt] Use the spdep package to create a nb object that defines neighbors based on adjacency. **PEC**. What is the most neighbors any areal unit has? Fewest?

```r
library(spdep)

nbrs <- poly2nb(SRA)

x <- vector(length = 41)
for(i in seq(from = 1, to = 41, by = 1)){
  x[i] <- length(nbrs[[i]])
}
```

```r

max(x)
min(x)
```

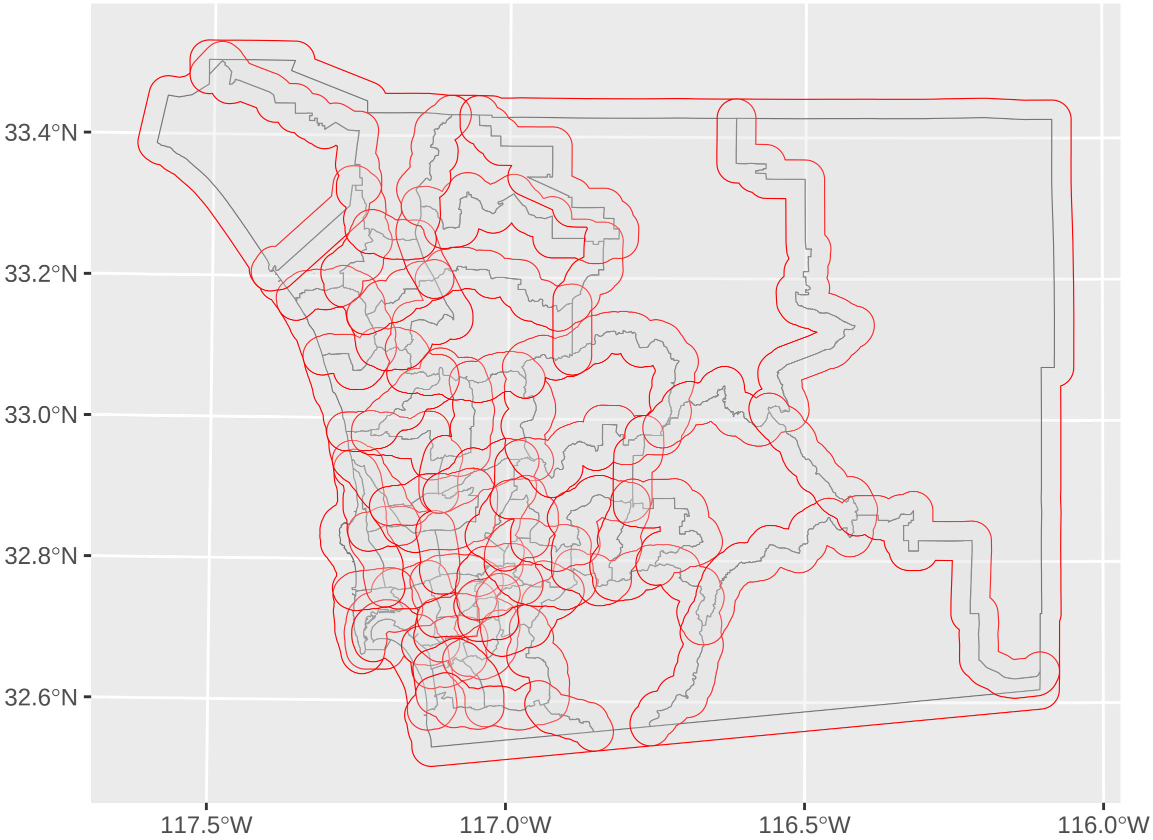
The largest number of neighbors for a single unit is 8, and the fewest is 2.

(4) [2 pts] Choose another method besides spatial adjacency for defining a binary neighborhood structure. Describe your method and **PEC** that creates an associated `nb` object or adjacency matrix, $\mathbf{W}$. What is the most neighbors any areal unit has? Fewest?

I've decided to designate SRAs as neighbors if they are within 10 km of each other.

```r
nbrs2 <- st_is_within_distance(SRA, SRA, dist = 10000)

x <- vector(length = 41)
for(i in seq(from = 1, to = 41, by = 1)){
  x[i] <- length(nbrs2[[i]])
}
max(x)
min(x)
```

```r
sra_buff <- st_buffer(SRA, dist = 10000)

# just kinda seeing what this looks like...
library(ggplot2)
ggplot()+
  geom_sf(data = SRA, alpha = 0.2)+
  geom_sf(data = sra_buff, color = "red", alpha = 0.2)
```



The largest number of neighbors is 13, and the smallest is 3.

Next, you'll incorporate data reported by the county on the rates of injuries to cyclists caused by vehicles in 2017. These data were obtained from the County of San Diego's public data repository **here**. The following lines of code read the data into R, and then isolate rows corresponding to SRAs and columns corresponding to variables of interest.

```r
bikes_full <- read.csv("Motor_Vehicle_Injuries_to_Pedalcyclists.csv")
bikes <- bikes_full[bikes_full$GeoType == "SRA" & !is.na(bikes_full$GeoType),
                    c("OUTCOME", "GeoName", "GeoID", "Total", "TotalRate")]
```

The variable `OUTCOME` corresponds to one of six outcomes for the cyclist (you will focus on three of them); `GeoName` gives the name assigned to an individual SRA; `GeoID` gives a unique integer label to each SRA that matches with the values in `SRA$OBJECTID`; `Total` gives the total number incidents that of the corresponding outcome type that occurred in the given SRA; `TotalRate` = `Total`/population of SRA * 100,000 gives the number of incidents per 100,000 people. Values of `NA` in `Total` and `TotalRate` correspond to incident counts for the SRA less than 5.

(5) [2 pts] How many total deaths occurred? How many total hospitalizations? How many total emergency room discharges?

```r
library(dplyr)

bikes_lower <- bikes %>%
  mutate(Total = case_when(is.na(Total) ~ 0,
                           .default = Total))

```

```r
bikes_upper <- bikes %>%
  mutate(Total = case_when(is.na(Total) ~ 4,
                           .default = Total))

outcomes_lower <- bikes_lower %>%
  group_by(OUTCOME) %>%
  summarise(total_deaths = sum(Total))
outcomes_upper <- bikes_upper %>%
  group_by(OUTCOME) %>%
  summarise(total_deaths = sum(Total))
```

There were as many as zero, 88, and 961, and as few as 164, 208, and 1001 deaths, hospitilizations, and ED discharges respectively.

(6) [2 pts] Explain in words what the following 7 lines of code do. Make sure you address all 7 lines. What is the meaning of the column `TotalRateInjury` in SRA?

```r
1 SRA <- merge(SRA, bikes[bikes$OUTCOME == "Hospitalization", c("TotalRate", "GeoID")],
2              by.x = "sra", by.y = "GeoID")
3 SRA <- merge(SRA, bikes[bikes$OUTCOME == "ED Discharge", c("TotalRate", "GeoID")],
4              by.x = "sra", by.y = "GeoID", suffixes = c("_Hospital", "_EDD"))
5 SRA$TotalRate_Hospital[is.na(SRA$TotalRate_Hospital)] <- 0
6 SRA$TotalRate_EDD[is.na(SRA$TotalRate_EDD)] <- 0
7 SRA$TotalRateInjury <- SRA$TotalRate_EDD + SRA$TotalRate_Hospital
```

Lines 1 and 2 adds attributes about hospitilizations to the original SRA sf object, by matching values in the 'sra' column of SRA and the 'GeoID' column of bikes. Similarly, lines 3 and 4 adds attributes about ED discharges to the modified SRA sf object from lines 1&2. However, suffixes are also added to the column names that are the same between those that are merged to SRA. Lines 5 and 6 set the rates of hospitilization and ED discharge to 0 if they equal NA respectively. Line 7 adds together the amount of incidents resulting in hospitilization or an ED visit, giving us a total number of injuries for each feature in SRA.

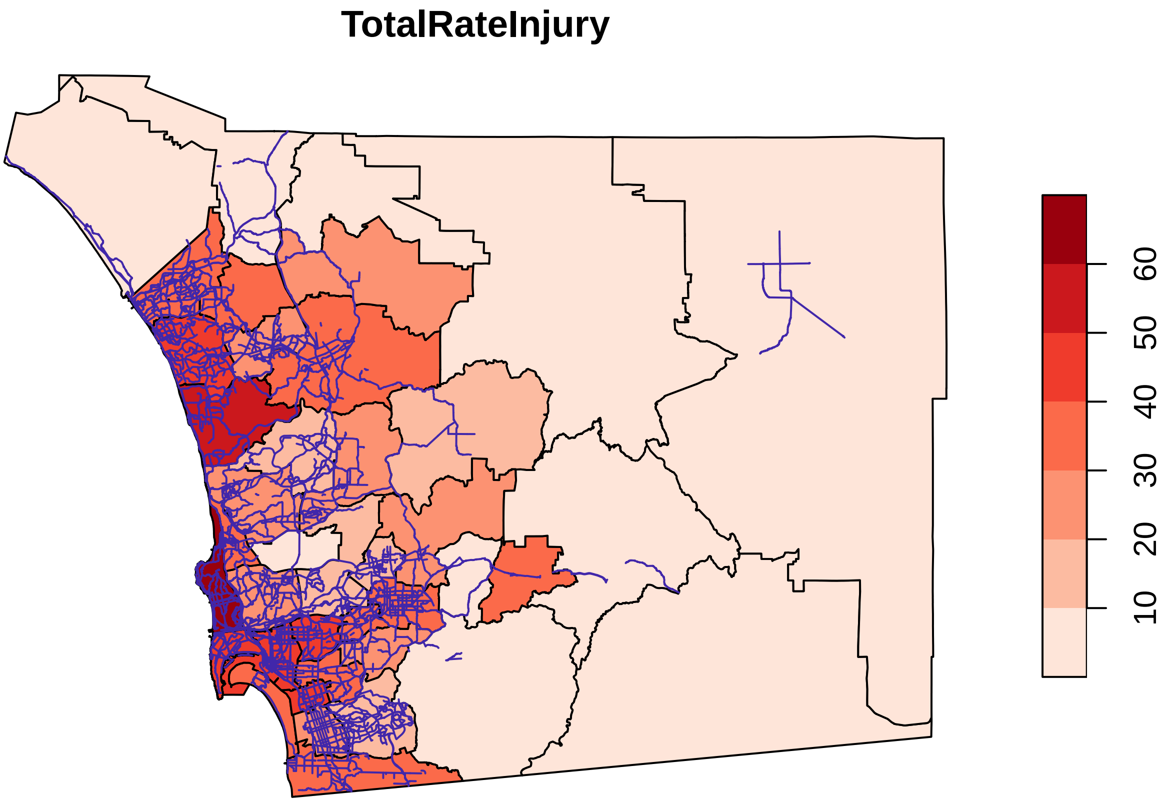
One potentially relevant variable that could help explain the number of vehicle-caused cyclist injuries is the total length of bicycle routes within each SRA. The file `BIKE_ROUTES.zip` on D2L contains shapefiles for all the bicycle routes in San Diego.

(7) [2 pts] **PEC** that: (i) reads the bike route shapefiles into R as a simple features object called `routes_sf` and (ii) verifies that the CRS for `routes_sf` matches the one used for the SRAs.

```r
routes_sf <- st_read("BIKE_ROUTES.shp") %>%
  st_as_sf()

st_crs(routes_sf) == st_crs(SRA)
```

(8) [4 pts] **PEC** to make a plot similar to Figure 2 (SEE PDF ON GRADESCOPE). Make sure your map shows the SRAs of San Diego colored according to `TotalRateInjury` and the bicycle routes (hint: you may want to use the `plot()` function twice; once with `reset = FALSE`, and once with `add = TRUE`). Include your map in your submission.

```r
library(RColorBrewer)

breaks <- c(0, 10, 20, 30, 40, 50, 60, 70)

plot.new()
plot(SRA[10], reset = F, pal = brewer.pal(n = 7, name = "Reds"), breaks = breaks)
plot(routes_sf, add = T, col = "#4127AA")
```



Your next goal is to build a model for injury rate as a function of the total length of bicycles routes in a given SRA, so the first thing to do is create that predictor variable. The following code uses the `st_intersection()` and `st_length()` functions from the `sf` package to calculate the length of the bicycles routes in each SRA polygon.

```r
1 routes_SRA_int <- st_intersection(routes_sf['ROUTE'], SRA['sra'])
2 routes_SRA_int$length <- units::set_units(st_length(routes_SRA_int), "mi")
3 SRA_lengths <- aggregate(length ~ sra, data = routes_SRA_int, FUN = sum)
4 SRA <- merge(SRA, SRA_lengths, by = "sra", all.x = T)
5 SRA$length[c(38, 40)] <- 0
```

(9) [2 pts] Explain what each of the 5 lines in the code chunk above do. Be sure to explain why the last line about SRAs #38 and #40 is included.

Line 1 finds pairs bike routes and SRAs that they overlap. Line 2 adds a column which computes the length of each pair of overlaps in miles. Line 3 creates a data frame from summing the lengths of overlapping bike paths for each SRA. Line 4 merges that data frame with the length information to the SRA feature to add these lengths as attributes to each SRA. Line 5 sets the length equal to zero for SRAs with null values resulting in a lack of information generated from line 1 (due to an absence of bike paths).

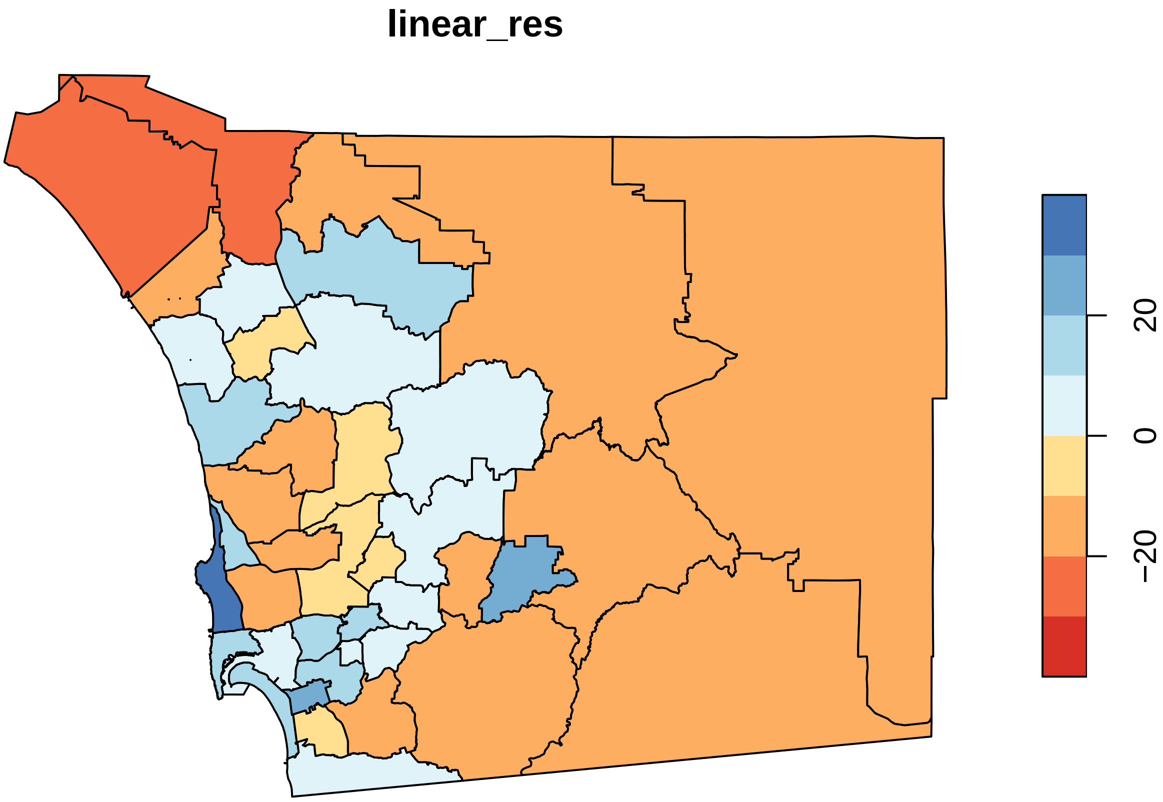
II. Statistical models [17 pts]

(10) [4 pts] **PEC** that fits a traditional linear model for `TotalRateInjury` as function of the miles of bicycle routes in the SRA assuming independent residuals. Give a 95% confidence interval for the effect of miles of bicycles routes. Make a map of the SRAs colored according to the residuals from the linear model. Use a diverging color scheme centered at 0 (hint: you may need to set the `breaks` argument in `plot()`).

```r
library(spmodel)
linmod <- splm(TotalRateInjury ~ length, data = SRA, spcov_type = "none")
confint(linmod)
```

For every additional mile of bicycle routes, there were an additional 0.14 - 0.39 deaths per 100,000 people.

```r
SRA$linear_res <- residuals(linmod)
plot(SRA['linear_res'],
     pal = brewer.pal(n = 8, name = "RdYlBu"),
     breaks = c(-40, -30, -20, -10, 0, 10, 20, 30, 40))
```



(11) [6 pts] **PEC** that fits the same linear model while accounting for residual spatial dependence using a CAR model based on the adjacency network from question (2). Compare the models with and without spatial dependence using AIC and leave-one-out cross-validation. Is there evidence in favor of including the “small” scale spatial effect?

```
nbrs1 <- st_intersects(SRA, remove_self = T)

carmod <- spautor(TotalRateInjury ~ length, data = SRA,
                  spcov_type = "car", W = as.matrix(nbrs1))

rbind(c("linear", loocv(linmod)), c("car", loocv(carmod)))
AIC(linmod, carmod)
```

There is some evidence that the CAR model performed better. Compared to the linear model, the AIC and prediction error are lower and the "R2"(ish) is higher.

(12) [3 pts] Give a 95% confidence interval for the effect of miles of bicycle routes based on the CAR model from (10). How does it compare to the interval you found using the linear model *without* accounting for residual spatial dependence?

```
confint(carmod)
```

CAR modeling estimates that for each additional bike route mile, there are 0.06 - 0.29 additional deaths per 100,000 people. This model produced a slightly narrower confidence interval, and the mean of the estimate was lower.

(13) [4 pts] Fit another linear model with spatial dependence (could be another CAR model, but need not be) based on your alternative neighborhood structure from question (3). Compare the two model fits using a predictive score of your choice. Point out differences/similarities you notice in both the "large" and "small" scale effects.

```
carmod2 <- spautor(TotalRateInjury ~ length, data = SRA,
                  spcov_type = "car", W = as.matrix(nbrs2))

rbind(loocv(carmod), loocv(carmod2))
AIC(carmod, carmod2)
summary(carmod);summary(carmod2)
```

My alternative neighborhood structure made little difference in the predictive power of the model (AIC). My inclusion of extra neighbors led to a decreased estimate of the effect of length, however much less variance was explained compared to the original neighborhood structure (R2). Including more neighbors increased the spatial dependence and range parameters, confirming that this structure implies more vast spatial dependence.