**Week 10 Lab**

**Spatial Analysis of Forest Inventory Data**

**Objective**

This lab focuses on spatial analysis of forest inventory data using a Digital Terrain Model (DEM). You will extract slope and aspect from the DEM, visualize spatial patterns, and analyze the distribution of forest attributes (e.g., basal area, biomass) across the study area.

**Materials Needed**

1. Ground Digital Terrain Model (DEM) of study unit 2 (`unit2.img`).
2. Summary table from Week 9 Lab (`sum\_u2.csv`).
3. Shapefile of sample plot locations (`HEE\_Overstory\_Survey\_Points\_2017 - Copy `; ensure all 7 related files are saved in one folder).
4. You may need to install and load ‘terra’, ‘tmap’, and ‘sf’ packages.

**Procedure**

1. **Read and Explore the DEM**

- Use the `rast` function from the `terra` package to read the DEM file.

- Inspect the DEM.

- If you want to view the dem values for each individual pixel, you need to use  
`as.data.frame()` to convert the dem to a readable dataframe format (we won’t use this command in this lab).

*[Example Code]*

*# Read inventory data*

*filepath <-‘your working directory/’*

*library(terra)*

*dem <- rast(paste0(filepath, "unit2.img"))*

*dem*

*#df\_dem <- as.data.frame(dem, xy = TRUE)*

1. **Extract Slope and Aspect**

**a. Extract Slope:**

- Use the `terrain` function from the `terra` package with opt = "slope", unit = "degrees", and neighbors = 8.

- Note: Think of having a 3\*3 cells (square shape). If we use the center cell (pixel) as the reference, we can calculate the difference between the center cell and any cell around. That difference is the height difference, which is used to determine “slope”. You could just calculate the difference between the center cell and the cells that share sides with the center cell. In this case, the center cell has 4 neighbors and it is called “4-cell rule”; Also, you could do this for cells that share either sides or edges (points). In this case, you have 8 neighbors, and this is called “8-cell rule”.

*[Example Code]*

*slope <- terrain(dem, v = "slope", unit = "degrees", neighbors = 8)*

**b. Extract Aspect:**

- Use the `terrain` function with `opt = "aspect"` and `unit = "degrees"`. *[Example Code]*

*aspect <- terrain(dem, v = "aspect", unit = "degrees")*

**c. Visualize Slope and Aspect:**

- Use `tmap` to visualize slope and aspect. Load the `tmap` package.

*[Example Code]*

*#install.packages(‘tmap’)*

*library(tmap)*

*ttm() # Switch to interactive mode*

*tm\_shape(slope, alpha = 0.5) +*

*tm\_raster(style = "cont", alpha = 0.6, title = "Slope (deg)")*

*tm\_shape(aspect) +*

*tm\_raster(style = "cont")*

- Check the image of aspect. You will see the range of aspect is 0-360 degrees (angles of azimuth). An azimuth starts from North and goes clockwise. Now we want to further combine it to 4 categories: north, east, south, and west.

**Question 1**: What is the degree range of each direction (North, East, South, West)?

1. **Reclassify Aspect**

**a. Create Aspect Classification Matrix:**

- Based on your answer to Question 1, create a matrix with 8 rows and 3 columns.

- The first two columns should represent the minimum and maximum range of each direction, while the third column should contain numbers from 1 to 4, representing the four directions (1= North; 2=East; 3=South; 4=West).

*[Example Code]*

*asp\_class <- matrix(c(*

*0, 45, 1,*

*45, 90, 2,*

*90, 175, 2,*

*175, 180, 3,*

*180, 225, 3,*

*225, 270, 4,*

*270, 315, 4,*

*315, 360, 1*

*), ncol = 3, byrow = TRUE)*

**b. Reclassify Aspect:**

- Use the `classify` function from the `terra` package to transform aspect into 4 directions.

*[Example Code]*

*asp <- classify(aspect, asp\_class)*

**c. Visualize Reclassified Aspect:**

- Use `tmap` to visualize the reclassified aspect.

*[Example Code]*

*ttm()*

*tm\_shape(asp) +*

*tm\_raster(style = "cat", palette = c("white", "blue", "green", "yellow", "red"),*

*labels = c(NA, "North", "East", "South", "West"), alpha = 0.2)*

**Question 2:** Paste a screenshot of your output image.

1. **Visualize Sample Forest Inventory Plots**

**a. Read Summary Table and Shapefile:**

- Now we want to visualize our circular sample forest inventory plots on top of both aspect and slope images. Since the summary table currently has no spatial information, we will assign plot center coordinates to every plot.

- Before doing this, we may want to check if the number of plots from the inventory data table matches with the one from the overstory sample point shapefile layer.

*[Example Code]*

*sum\_u2 <- read.csv(paste0(filepath, "sum\_u2.csv"))*

*library(sf)*

*svy\_pts <- st\_read(paste0(filepath, "HEE\_Overstory\_Survey\_Points\_2017 - Copy.shp"))*

*svy\_pts <- st\_transform(svy\_pts, 32616) # Project to WGS 84 UTM 16 N*

*survey\_pts <- subset(svy\_pts, Unit == '2') # Subset for unit 2*

**b. Merge Summary Table with Plot Locations:**

- Merge `sum\_u2` with `survey\_pts` using `merge.data.frame`.

*[Example Code]*

*sum\_u2 <- merge.data.frame(sum\_u2, survey\_pts, all.x = TRUE)*

- Here we use "all.x = T" because we only need plot locations for the summary table (sum\_u2), if you check all the plots for the two files separately, you'll find the shapefile have more plots than the summary table. The “unique” function tells you what unique values your object (or parameter, in this case, plot) has.

*[Example Code]*

*unique(sum\_u2$Plot)*

*unique(survey\_pts$Plot)*

**c. Convert to `sf` Format:**

- Convert the merged table to `sf` format for spatial analysis.

- Don’t worry about sf definition right now. You can regard it as a format to store spatial data. Your new summary table should now have a new column “geometry” that shows the plot center coordinates.

*[Example Code]*

*sum\_u2 <- st\_as\_sf(sum\_u2, coords = c("X", "Y"), crs = 32616)*

*sum\_u2*

1. **Create Circular Plots**

**a. Create Buffer Zones:**

- Right now, we only have point coordinates of plot center, but it’s possible to convert them into polygons (circles) for better visualization. One way to achieve this is creating a buffer zone at a 17.83m radius, the buffered area would then represent the circular plot.

- Use `st\_buffer` from the `sf` package to create circular plots with a 17.83m radius.

*[Example Code]*

*sf\_plot <- st\_buffer(sum\_u2, dist = 17.83)*

1. **Unify Coordinate Systems**

**a. Check CRS:**

- In our lab, there are two raster images (slope and aspect) but only one vector file (plot locations) for visualization purpose, so it’s easier to convert the coordinate reference system (CRS) for plots to the ones used by both raster images.

- First, we need to check CRS.

*[Example Code]*

*crs(sf\_plot* *, proj=T)*

*crs(asp* *, proj=T)*

**b. Transform CRS:**

- Transform the CRS of plots to match the raster images.

*[Example Code]*

*asp\_crs <- crs(asp, proj = TRUE)*

*sf\_plot\_crs <- st\_transform(sf\_plot, crs = asp\_crs)*

1. **Visualization**

**a. Dominant Species by Aspect:**

- Visualize dominant species distribution across aspect categories.

*[Example Code]*

*ttm()*

*tm\_shape(asp, alpha = 0.5) +*

*tm\_raster(style = "cat", palette = c("white", "blue", "green", "yellow", "red"),*

*showNA = FALSE, alpha = 0.2, labels = c(NA, "North", "East", "South", "West")) +*

*tm\_shape(sf\_plot) +*

*tm\_polygons('Common.name') +*

*tm\_layout(legend.outside = TRUE, legend.outside.size = 0.2) +*

*tm\_text("Plot", ymod = -0.9)*

**Question 3:** Display dominant tree species by aspect (paste your image below). Describe if the distribution of sugar maple is related to aspect. Does it favor a particular side? Describe what can potentially cause this pattern (e.g. consider shade tolerance of different species).

**b. Dominant Species by Slope:**

- Visualize dominant species distribution across slope.

*[Example Code]*

*ttm()*

*tm\_shape(slope, alpha = 0.5) +*

*tm\_raster(style = "cont", alpha = 0.6, title = "Slope (deg)") +*

*tm\_shape(sf\_plot) +*

*tm\_polygons('Common.name', title = "Dom\_Species", alpha = 0.6) +*

*tm\_layout(title = "Dominant trees by slope",*

*legend.outside = TRUE, legend.outside.size = 0.2) +*

*tm\_text("Plot", ymod = -0.9, size = 1.2)*

**Question 4:** Display dominant tree species by slope (paste your image below).

**c. Basal Area (BA) Distribution:**

- Visualize BA values with a topographic map background.

*[Example Code]*

*ttm()*

*tm\_shape(sf\_plot) +*

*tm\_polygons('BA', title = "Basal Area (sq\_ft/acre)", palette = "brewer.spectral") +*

*tm\_layout(title = "Basal Area Distribution",*

*legend.outside = TRUE, legend.outside.size = 0.2) +*

*tm\_text("Plot", ymod = -1.5, size = 1.2) +*

*tm\_scale\_bar()*

**Question 5:** Is there a pattern where plots with high/low BA values cluster together? Indicate their relative directions/positions (paste your image below).

**d. Trees Per Acre (TPA) Distribution:**

- Visualize TPA values with a topographic map background.

*[Example Code]*

*ttm()*

*tm\_shape(sf\_plot) +*

*tm\_polygons('TPA', title = "Trees Per Acre", palette = "brewer.spectral") +*

*tm\_layout(title = "TPA Distribution",*

*legend.outside = TRUE, legend.outside.size = 0.2) +*

*tm\_text("Plot", ymod = -1.5, size = 1.2) +*

*tm\_scale\_bar()*

**Question 6:** Is there a pattern where plots with high/low TPA values cluster together? (paste your image below).

**e. Biomass Distribution:**

- Visualize biomass values with a topographic map background.

*[Example Code]*

*ttm()*

*tm\_shape(sf\_plot) +*

*tm\_polygons('bm\_tonpa', title = "Biomass (tons/ac)", palette = "brewer.spectral") +*

*tm\_layout(title = "Biomass Distribution",*

*legend.outside = TRUE, legend.outside.size = 0.2) +*

*tm\_text("Plot", ymod = -1.5, size = 1.2) +*

*tm\_scale\_bar()*

**Question 7:** Is there a pattern where plots with high/low biomass values cluster together? Indicate their relative directions/positions (paste your image below).