# Set working directory  
setwd("D:/SPATIAL DATA ANALYSIS/ASSIGNMENT2")  
  
# Load required libraries  
libraries <- c("sf", "tmap", "leaflet", "dplyr", "ggplot2", "rgdal", "terra", "readxl", "ncdf4", "chron", "lattice", "RColorBrewer", "gstat")  
install.packages(setdiff(libraries, rownames(installed.packages())))  
lapply(libraries, require, character.only = TRUE)  
  
# Read shape file and Excel data  
Indonesia\_map <- st\_read("idn\_admbnda\_adm0\_bps\_20200401.shp")  
power\_plants <- read\_excel("power plant Indonesia2.xlsx")  
  
# Define plot function  
plot\_power\_plants <- function(data, title) {  
  ggplot() +  
    geom\_sf(data = Indonesia\_map, fill = "lightblue", color = "grey") +  
    geom\_point(data = data, aes(x = longitude, y = latitude, size = capacity\_mw, shape = type, color = status)) +  
    scale\_shape\_manual(values = c("solar" = 15, "Geothermal" = 17, "Hydro" = 16, "wind" = 18)) +  
    scale\_size\_continuous(range = c(1, 12)) +  
    labs(title = title, x = "Longitude", y = "Latitude") +  
    theme\_minimal() +  
    theme(legend.position = "right")  
}  
  
# Plot renewable power plants  
print(plot\_power\_plants(power\_plants, "Renewable Power Plants in Indonesia"))  
  
# Function to process SSRD data  
process\_ssrd\_data <- function(file\_name, title) {  
  era <- nc\_open(file\_name)  
  lon <- ncvar\_get(era, "longitude")  
  lat <- ncvar\_get(era, "latitude")  
  time <- ncvar\_get(era, "time")  
  tunits <- ncatt\_get(era, "time", "units")  
  
  # Convert time  
  tustr <- strsplit(tunits$value, " ")  
  tdstr <- strsplit(unlist(tustr)[3], "-")  
  tyear <- as.integer(unlist(tdstr)[1])   
  tmonth <- as.integer(unlist(tdstr)[2])  
  tday <- as.integer(unlist(tdstr)[3])  
  times <- chron(time / 24, origin = c(tmonth, tday, tyear))  
  
  # Process SSRD  
  ssrd\_array <- ncvar\_get(era, "ssrd")  
  ssrd\_slice <- ssrd\_array[,,2]   
  image(ssrd\_slice, col = rev(brewer.pal(10, "RdBu")), main = title)  
  
  # Convert to sf object  
  lonlat <- as.matrix(expand.grid(lon, lat))  
  ssrd\_vec <- as.vector(ssrd\_slice)  
  ssrd\_df <- na.omit(data.frame(cbind(lonlat, ssrd\_vec)))  
  colnames(ssrd\_df) <- c("lon", "lat", "ssrd")  
  ssrd\_sf <- st\_as\_sf(ssrd\_df, coords = c("lon", "lat"), crs = 4326)  
  st\_transform(ssrd\_sf, 4326)  
}  
  
# Process and visualize SSRD data  
process\_ssrd\_data("SSRD [data.nc](http://data.nc)", "Surface Solar Radiation Downwards")  
  
# Additional code for specific SSRD calculations and visualizations can be added here following the same pattern.

================================

# Set working directory and load required libraries  
setwd("D:/SPATIAL DATA ANALYSIS/ASSIGNMENT2")  
library(sf)  
library(tmap)  
library(leaflet)  
library(dplyr)  
library(ggplot2)  
library(terra)  
library(raster)  
library(readxl)  
  
# Define a general function for quick masking of raster data  
quick\_mask\_raster <- function(raster\_data, masking\_vector) {  
  masking\_vector <- st\_transform(masking\_vector, st\_crs(raster\_data))  
  return(mask(raster\_data, masking\_vector))  
}  
  
# Read the shape file for Indonesia  
Indonesia\_map <- st\_read("idn\_admbnda\_adm0\_bps\_20200401.shp")  
  
# Create a reusable raster template  
raster\_template <- rast(resolution = 0.05, xmin = 95.01079, ymin = -11.00762, xmax = 141.01940, ymax = 6.07693, crs = st\_crs(Indonesia\_map)$wkt)  
  
# Define a general function for plotting raster data  
plot\_raster\_data <- function(raster\_data, title = "Raster Plot") {  
  plot(raster\_data, main = title)  
}  
  
# 1. Land Cover Analysis  
landcover <- rast("D:/SPATIAL DATA ANALYSIS/ASSIGNMENT2/IDN\_msk\_cov.vrt")  
# ... [additional land cover processing steps] ...  
plot\_raster\_data(land\_potential, "Land Cover Potential")  
  
# 2. Peatland Analysis  
peatland <- st\_read("D:/SPATIAL DATA ANALYSIS/ASSIGNMENT2/Indonesia\_peat\_lands.shp")  
# ... [peatland processing steps] ...  
plot\_raster\_data(peatland\_potential, "Peatland Potential")  
  
# 3. Population Analysis  
population <- rast("D:/SPATIAL DATA ANALYSIS/ASSIGNMENT2/IDN\_msk\_pop.vrt")  
# ... [population processing steps] ...  
plot\_raster\_data(pop\_raster, "Population Density")  
  
# 4. Protected Areas Analysis  
protected <- st\_read("D:/SPATIAL DATA ANALYSIS/ASSIGNMENT2/WDPA\_WDOECM\_Jan2024\_Public\_IDN\_shp-polygons.shp")  
# ... [protected areas processing steps] ...  
plot\_raster\_data(protected\_raster, "Protected Areas")  
  
# Constrained Areas Analysis  
# ... [constrained areas processing steps] ...  
plot\_raster\_data(constrained\_mask, "Constrained Areas")

================

# Set up the working directory as a variable for easy changes

working\_directory <- "D:/SPATIAL DATA ANALYSIS/ASSIGNMENT2"

setwd(working\_directory)

# Load required libraries

library(sf)

library(tmap)

library(leaflet)

library(dplyr)

library(ggplot2)

library(terra)

library(raster)

library(readxl)

# Install packages if not already installed

packages\_to\_install <- c("rgdal", "terra", "readxl")

new\_packages <- packages\_to\_install[!(packages\_to\_install %in% installed.packages()[,"Package"])]

if(length(new\_packages)) install.packages(new\_packages)

# Read the shape file

Indonesia\_map <- st\_read(file.path(working\_directory, "idn\_admbnda\_adm0\_bps\_20200401.shp"))

# Define a function to quickly mask a raster using a vector

quick\_mask\_raster <- function(raster\_data, masking\_vector) {

masking\_vector <- st\_transform(masking\_vector, st\_crs(raster\_data))

return(mask(raster\_data, masking\_vector))

}

# Define a template raster for cropping and resampling

raster\_template <- rast(resolution = 0.05, xmin = 95.01079, ymin = -11.00762, xmax = 141.01940, ymax = 6.07693, crs = st\_crs(Indonesia\_map)$wkt)

# Define a function to read and process raster data

process\_raster\_data <- function(raster\_path, classification\_matrix) {

landcover <- rast(raster\_path)

land\_crop <- crop(landcover, raster\_template)

land\_crop <- resample(land\_crop, raster\_template)

land\_potential <- classify(land\_crop, classification\_matrix, include.lowest = TRUE)

return(quick\_mask\_raster(land\_potential, Indonesia\_map))

}

# Process Land Cover data

land\_potential <- process\_raster\_data(file.path(working\_directory, "IDN\_msk\_cov.vrt"),

matrix(c(0, 10, 0, 10, 15, 1, 15, 16, 2, 16, 20, 3, 20, 25, 4), ncol = 3, byrow = TRUE))

plot(land\_potential)

# Process Peat Land data

peatland <- st\_read(file.path(working\_directory, "Indonesia\_peat\_lands.shp"))

peatland\_raster <- rasterize(peatland, raster\_template)

peatland\_potential <- process\_raster\_data(file.path(working\_directory, "Indonesia\_peat\_lands.shp"),

matrix(c(0, 2000000000, 0, 2000000000, 4000000000, 1, 4000000000, 6000000000, 2, 6000000000, 8000000000, 3, 8000000000, 10000000000, 4, 10000000000, 15000000000, 5), ncol = 3, byrow = TRUE))

plot(peatland\_potential)

# Process Population data

pop\_raster <- process\_raster\_data(file.path(working\_directory, "IDN\_msk\_pop.vrt"),

matrix(c(0, 10, 0, 10, 15, 1, 15, 20, 2, 20, 25, 3, 25, 30, 4), ncol = 3, byrow = TRUE))

plot(pop\_raster)

# Process Protected Areas data

protected <- st\_read(file.path(working\_directory, "WDPA\_WDOECM\_Jan2024\_Public\_IDN\_shp-polygons.shp"))

protected\_raster <- rasterize(protected, raster\_template)

plot(protected\_raster)

# Create a constrained map

constrained\_map <- stack(protected\_raster, pop\_raster, land\_potential, peatland\_potential)

names(constrained\_map) <- c("protected\_raster", "pop\_raster", "land\_potential", "peatland\_potential")

# Apply constraints and mask

constrained\_map\_df <- as.data.frame(constrained\_map, xy = TRUE)

id <- which(constrained\_map\_df$protected\_raster == 1 | constrained\_map\_df$pop\_raster <= 1 | constrained\_map\_df$land\_potential <= 2 | constrained\_map\_df$peatland\_potential == 1)

constrained <- raster\_template

constrained[id] <- 1

constrained\_mask <- quick\_mask\_raster(constrained, Indonesia\_map)

constrained[is.na(constrained)] <- 0 # Replace NA with 0 to indicate non-constrained areas

plot(constrained\_mask)