ODIAC CO2 data processing

- Since they are all satelite data, the processing method of PM2.5 data (ChinaPM, Modis) is very similar to this.

1. **Format transformation of the raw data (.gz files to .tiff files)**

```
library(R.utils)

# Loop over the years 2000 to 2008
for (year in 2002) {
  # Set the folder containing the gzipped TIF files for the current year
  tif_folder <- paste("E:/Impacts/CO2/", year, sep = "")

  # List all the gzipped TIF files in the folder
  tif_files <- list.files(tif_folder, pattern = ".tif.gz$", full.names =
TRUE)

  # Loop over all TIF files and unzip them
  for (tif_file in tif_files) {
    cat("Unzipping ", tif_file, "\n")

    # Unzip the file
    R.utils::gunzip(tif_file, remove = TRUE)
  }
}
```

2. **zonal city (aggregate pixel values with cities' boundary)**

```
library(terra)

# Load the shapefile with polygon IDs
shapefile_path <- "E:/Impacts/PM2.5/2020china_map/city/city.shp"
polygons <- vect(shapefile_path)

# Get a list of all rasters in a folder
raster_folder <- "E:/Impacts/CO2/2000/"
raster_files <- list.files(raster_folder, pattern = ".tif$", full.names =
TRUE)

# Loop over all raster files and extract zonal statistics
for (i in seq_along(raster_files)) {
  cat("Processing raster file ", i, " of ", length(raster_files), "\n")
  raster_file <- rast(raster_files[i])

  # Create data frames for storing the zonal statistics
  zonal_stats <- zonal(raster_file, polygons,
                       fun = mean, na.rm = TRUE,
                       id = polygons$市代码)
  colnames(zonal_stats)[1] <- "meanPM"
  zonal_stats$city_id <- polygons$市代码

  # modify the name of csv
  year <- sub(".*_(\\d{4})\\.tif$", "\\1", basename(raster_files[i]))
```

```
  output_path <- paste0("E:/Impacts/CO2/calculation/city-", year, ".csv")
  write.csv(zonal_stats, output_path, row.names = FALSE)
}
```

3. **zonal county**

```
library(terra)

# Load the shapefile with polygon IDs
shapefile_path <- "E:/Impacts/PM2.5/2020china_map/county/county.shp"
polygons <- vect(shapefile_path)


# Get a list of all rasters in a folder
raster_folder <- "E:/Impacts/CO2/2002/"
raster_files <- list.files(raster_folder, pattern = ".tif$", full.names =
TRUE)


# Loop over all raster files and extract zonal statistics

for (i in seq_along(raster_files)) {
  cat("Processing raster file ", i, " of ", length(raster_files), "\n")
  raster_file <- rast(raster_files[i])

  # Create data frames for storing the zonal statistics
  zonal_stats <- zonal(raster_file, polygons,
                       fun = mean, na.rm = TRUE,
                       id = polygons$PAC)
  colnames(zonal_stats)[1] <- "meanCO2"
  zonal_stats$county_id <- polygons$PAC

  # modify the name of csv
  year <- sub(".*_(\\d{4})\\.tif$", "\\1", basename(raster_files[i]))
  output_path <- paste0("E:/Impacts/CO2/calculation/county-", year, ".csv")
  write.csv(zonal_stats, output_path, row.names = FALSE)
}
```

4. **aggregate the generated data into one file**

```
library(data.table)
library(haven)

# Set the path to your folder containing city and county files
folder <- "D:/CO2/calculation"
files <- list.files(path = folder, pattern = "*.csv", full.names = TRUE)

city_data_list <- list()
county_data_list <- list()

for (file in files) {
  df <- fread(file)  # Use fread to efficiently read CSV files

  # Extract the base name without extension
  file_base <- tools::file_path_sans_ext(basename(file))
```

```
  # Identify whether it's a city or county file based on the name
  if (grepl("city", file_base, ignore.case = TRUE)) {
    location_type <- "city"
  } else if (grepl("county", file_base, ignore.case = TRUE)) {
    location_type <- "county"
  } else {
    # Handle other cases if needed
    next
  }

  if (location_type == "city") {
    date <- as.character(substring(file_base, first = nchar(location_type) +
2))
    df$year <- as.numeric(paste0("20", substr(date, 1, 2)))
    df$month <- as.numeric(substr(date, 3, 4))
    df$location_type <- location_type
    new_order <- c("year", "month", "location_type","city_id","meanCO2")
    df <- df[, ..new_order]

    city_data_list[[file]] <- df
  }

  else if (location_type == "county") {
    date <- as.character(substring(file_base, first = nchar(location_type) +
2))
    df$year <- as.numeric(paste0("20", substr(date, 1, 2)))
    df$month <- as.numeric(substr(date, 3, 4))
    df$location_type <- location_type
    new_order <- c("year", "month", "location_type","county_id","meanCO2")
    df <- df[, ..new_order]

    county_data_list[[file]] <- df
  }
}

city_data <- rbindlist(city_data_list)  # Combine city data into a single
data.table
county_data <- rbindlist(county_data_list)  # Combine county data into a
single data.table

# Write city and county data to separate CSV files
write_dta(city_data, "D:/CO2/city_data.dta")
write_dta(county_data, "D:/CO2/county_data.dta")
```

5. **summarize the monthly data to yearly level**

```
library(dplyr)

# Load your .dta file (make sure it's in the working directory or provide the
full path)
data <- read_dta("D:/CO2/co2_odiac_county_2000_2021_monthly.dta")

# Summarize the data by year and citycode
summarized_data <- data %>%
  group_by(year, countycode, citycode) %>%
  summarise(co2emission = sum(co2))

write_dta(summarized_data, "D:/CO2/county co2 odiac 2000-2021.dta")
```