

TRhizo-localAdaptation

Urbanization Metrics

David Murray-Stoker

Contents

Load Packages & Data	2
Distance from the Urban Center	3
Impervious Surface Cover	4
Human Influence Index	6
Data Management	7
R Session Information	8

Load Packages & Data

```
## Load the tidyverse
library(tidyverse)

## Packages for analysis
library(geosphere)
library(raster)

## Read in data
site.data <- read_csv(
  "data/localAdaptation-site_information.csv",
  col_types = c("fnn"),
  show_col_types = FALSE
)
```

Distance from the Urban Center

```
distance.data <- data.frame("Distance" = numeric(length = 40))

for (x in 1:40) {
  site.from.UC <- distm(
    x = site.data[x, c(3, 2)],
    y = c(-79.383276, 43.651536),
    fun = distGeo
  )

  site.from.UC.km <- site.from.UC / 1000

  distance.data[x, ] <- site.from.UC.km
}

## Add a Population identifier to each row
distance.data$Population <- site.data$Population

## Set column structure
distance.data$Population <- as_factor(distance.data$Population)
```

Impervious Surface Cover

```
## Set the path to the ISC raster
ISC_raster_path <- "~/Downloads/CAN_gmis_impervious_surface_percentage_geographic_30m.tif"
# Change to the directory of the raster file

## Set the function to extract ISC values
calculate_ISC <- function(df, raster_path, buffer = 250) {
  # Load in raster for country
  raster <- raster::raster(raster_path)

  # Create spatial point dataframe from latitude and longitude
  spdf <- sp::SpatialPointsDataFrame(
    coords = df %>%
      dplyr::select(Longitude, Latitude),
    proj4string = raster@crs,
    data = df
  )

  # Calculate ISC data for population
  ISC_data <- raster::extract(x = raster, y = spdf, method = "simple", buffer = buffer)
  ISC_data <- ifelse(is.na(ISC_data), 255, ISC_data) # Missing data needs to be 255

  # Take mean ISC across all cells included within buffer
  ISC_data_mod <- ISC_data %>%
    # Convert ISC values of 200 to 0, as per documentation
    # Convert ISC values of 255 to missing
    map(., function(x) {
      case_when(
        x == 200 ~ 0,
        x == 255 ~ NA_real_,
        TRUE ~ x
      )
    }) %>%
    map(., mean, na.rm = TRUE) %>% # Ignore cells with missing ISC values when calculating mean
    unlist()

  # Add column with ISC values
  df_out <- df %>%
    mutate(
      Mean_ISC = ISC_data_mod,
      Mean_ISC = ifelse(Mean_ISC == 255, NA, Mean_ISC)
    )

  # Reorganize the data
  df_out <- df_out %>%
    dplyr::select(Population, Mean_ISC)

  return(df_out)
}

## Calculate ISC data in a dataframe
ISC.data <- calculate_ISC(df = site.data, raster_path = ISC_raster_path) %>%
  as_tibble()
```


Human Influence Index

```
## Set the path to the HII raster
HII_raster_path <- "~/Downloads/HII_raster/hii_v2geo/w001001.adf"
# Change to the directory of the raster file

## Set the function to extract HII values
calculate_HII <- function(df, raster_path) {
  # Load in the raster
  raster <- raster::raster(raster_path)

  # Create spatial point dataframe from latitude and longitude
  spdf <- sp::SpatialPointsDataFrame(
    coords = df %>%
      dplyr::select(Longitude, Latitude),
    proj4string = raster@crs,
    data = df
  )

  # Calculate HII data for each population
  HII_data <- raster::extract(x = raster, y = spdf, method = "simple")

  # Add column with HII values
  df_out <- df %>%
    mutate(
      Human_Influence_Index = HII_data,
      Human_Influence_Index = ifelse(Human_Influence_Index == 255, NA, Human_Influence_Index)
    )

  # Reorganize the data
  df_out <- df_out %>%
    dplyr::select(Population, Human_Influence_Index)

  return(df_out)
}

## Calculate HII data in a dataframe
HII.data <- calculate_HII(df = site.data, raster_path = HII_raster_path) %>%
  as_tibble()
```

Data Management

```
## Full urbanization data
full.urbanization.data <- site.data %>%
  full_join(distance.data, by = "Population") %>%
  full_join(HII.data, by = "Population") %>%
  full_join(ISC.data, by = "Population")

## Export full urbanization data
write_rds(full.urbanization.data, file = "data/full_urbanization_data.rds")
```

R Session Information

Table 1: Packages required for data management and analysis.

Package	Loaded Version	Date
dplyr	1.1.4	2023-11-17
forcats	1.0.0	2023-01-29
geosphere	1.5-18	2022-11-15
ggplot2	3.5.0	2024-02-23
kableExtra	1.4.0	2024-01-24
knitr	1.45	2023-10-30
lubridate	1.9.3	2023-09-27
purrr	1.0.2	2023-08-10
raster	3.6-26	2023-10-14
readr	2.1.5	2024-01-10
sp	2.1-3	2024-01-30
stringr	1.5.1	2023-11-14
tibble	3.2.1	2023-03-20
tidyr	1.3.1	2024-01-24
tidyverse	2.0.0	2023-02-22