

# Densidades Populacionais

Rodrigo

2024-12-19

```
library(geobr)
library(sf)
library(tidyverse)
library(ggthemes)
library(nngeo)
library(spdep)
library(lwgeom)
library(basemaps)
library(ggspatial)
```

```
#Arquivo SHP
censo = read_sf("C:/Users/rodri/OneDrive - Insper/CensoBR_2022/BR_setores_CD2022.shp")

#Filtros e procedimentos
censo <- censo %>%
  filter(NM_MUN == "São Paulo") %>%
  mutate(geometry = st_make_valid(geometry)) %>%
  mutate(Area_Ha = (st_area(geometry)/10000)) %>%
  mutate(Densidade_Populacional = (v0001/Area_Ha)) %>%
  mutate(Area_Ha = as.numeric(Area_Ha)) %>%
  mutate(Densidade_Populacional = as.numeric(Densidade_Populacional))
```

```
# Criação da matriz de vizinhança (k-nearest neighbors)
coords <- st_coordinates(st_centroid(censo)) # Coordenadas dos centros
nb <- knn2nb(knearneigh(coords, k = 1)) # 1 vizinho mais próximo

# Preencher os valores ausentes
censo <- censo %>%
  mutate(Densidade_Populacional = ifelse(
    is.na(Densidade_Populacional),
    sapply(1:nrow(censo), function(i) {
      if (is.na(Densidade_Populacional[i])) {
        vizinho <- nb[[i]]
        if (length(vizinho) > 0) {
          Densidade_Populacional[vizinho[1]]
        } else {
          NA # Caso não haja vizinhos
        }
      } else {
        Densidade_Populacional[i]
      }
    })
  )
```

```

    }),

  Densidade_Populacional
))

summary(censo$Densidade_Populacional)

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
##      0.0   107.9  183.3  281.4  342.8 11596.4

# Calcular os deciles e criar categorias com rótulos personalizados
censo <- censo %>%
  mutate(
  Densidade_Populacional = ifelse(is.na(Densidade_Populacional), 0, Densidade_Populacional),
  Decil = ntile(Densidade_Populacional, 10), # Divide em 10 categorias
  Densidade_Categorias = case_when(
    Decil == 1 ~ "0-10%",
    Decil == 2 ~ "10-20%",
    Decil == 3 ~ "20-30%",
    Decil == 4 ~ "30-40%",
    Decil == 5 ~ "40-50%",
    Decil == 6 ~ "50-60%",
    Decil == 7 ~ "60-70%",
    Decil == 8 ~ "70-80%",
    Decil == 9 ~ "80-90%",
    Decil == 10 ~ "90-100%",
    TRUE ~ NA_character_
  )
)

# Criar o mapa
sp = ggplot(data = censo) +
  geom_sf(aes(fill = Densidade_Categorias), color = NA) +
  scale_fill_viridis_d(name = "Densidade Populacional por Hectare") +
  labs(title = "Densidade Populacional em São Paulo") +
  theme_fivethirtyeight() +
  theme(
    panel.grid = element_blank(),
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank()
)

```

sp

## Densidade Populacional em São Pa

dade Populacional por Hectare



```
# Com BaseMap
```

```
# Criar o basemap (exemplo com Esri basemap)
basemap <- basemap_raster(
  ext = st_bbox(censo), # Extensão da área de interesse
  map_service = "osm", # Serviço do basemap (Ex.: "osm", "esri", "stamen")
  map_type = "streets" # Tipo de basemap (Ex.: "streets", "terrain", "satellite")
)
```

```
## Warning: Transforming 'ext' to Web Mercator (EPSG: 3857), since 'ext' has a
## different CRS. The CRS of the returned basemap will be Web Mercator, which is
## the default CRS used by the supported tile services.
```

```
## Loading basemap 'streets' from map service 'osm'...
```

```
# Criar o mapa com o basemap
sp <- ggplot() +
  layer_spatial(basemap) + # Adiciona o basemap
  geom_sf(data = censo, aes(fill = Densidade_Categorias), color = NA, alpha = 0.5) +
  scale_fill_viridis_d(name = "Densidade Populacional por Hectare") +
  labs(title = "Densidade Populacional em São Paulo") +
  theme_fivethirtyeight() +
  theme(
    panel.grid = element_blank(),
```

```
    axis.title = element_blank(),  
    axis.text = element_blank(),  
    axis.ticks = element_blank()  
)
```

```
sp
```

