Assignment 1

Maps Replication

Sebastian Dong Uk Paik Sohn Michael Duarte Gonçalves Gal · la Gelpí Buxadé
Maria Victoria Suriel Nuñez

January 28, 2026

Contents

1 Part 1: Fried & Lagakos (2021) — Figure 4

1.1 What the map shows

Figure 4 displays the location of surveyed villages in Ethiopia, overlaid with the country's road network and electricity grid. The blue gradient represents population density, roads appear as black lines (thicker for major highways), the high-voltage grid is shown in red, and hollow circles mark each village from the ERSS sample. The authors use this map to show where their data comes from and to highlight the variation they exploit—some villages sit near power lines while others are far away.

1.2 Data sources

We gathered all layers from public sources:

- Electricity grid — energydata.info
- Power plants — energydata.info
- Administrative boundaries — Africa GeoPortal
- Roads — Humanitarian Data Exchange (OpenStreetMap extract)
- ERSS Wave 1 — World Bank Microdata (2011/12)
- ERSS Wave 2 — World Bank Microdata (2013/14)

1.3 Why our map differs

We could not fully replicate the original for a few reasons. First, the paper mentions an "ERSS-based proxy surface" for population density but does not explain how they built it, so we used kernel density estimation instead, which gives a similar visual effect. Second, the original appears to use an Esri terrain basemap, which we do not have access to—we used administrative boundaries only. Third, we do not know the exact vintage of the road network they used, so we relied on OpenStreetMap data from HDX. Finally, we omitted geographic labels (Somalia, Djibouti, etc.) to keep the focus on Ethiopia.

1.4 Setup

```
if (!require("pacman")) install.packages("pacman")

pacman::p_load(
 sf, dplyr, readr, haven, stringr,
 ggplot2, ggnewscale, stars, MASS,
 knitr, kableExtra
)
```

```
zip_w1 <- "./data/Fried_Lagakos_2021/ETH_2011_ERSS_v02_M_CSV.zip"
zip_w2 <- "./data/Fried_Lagakos_2021/ETH_2013_ESS_v03_M_STATA.zip"
zip_admin <- "./data/Fried_Lagakos_2021/Ethiopia_AdminBoundaries-shp.zip"
zip_roads <- "./data/Fried_Lagakos_2021/hotosm_eth_roads_lines_shp.zip"
zip_grid <- "./data/Fried_Lagakos_2021/ethiopia-electricity-transmission-network.zip"
zip_plants <- "./data/Fried_Lagakos_2021/eth_powerplants.zip"
```

```r
tmp_w1 <- file.path(tempdir(), "w1")
tmp_w2 <- file.path(tempdir(), "w2")
tmp_admin <- file.path(tempdir(), "admin")
tmp_roads <- file.path(tempdir(), "roads")
tmp_grid <- file.path(tempdir(), "grid")
tmp_plants <- file.path(tempdir(), "plants")

lapply(
 c(tmp_w1, tmp_w2, tmp_admin, tmp_roads, tmp_grid, tmp_plants),
 dir.create,
 showWarnings = FALSE
)
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
##
## [[5]]
## [1] TRUE
##
## [[6]]
## [1] TRUE
```

```r
unzip(zip_w1, exdir = tmp_w1)
unzip(zip_w2, exdir = tmp_w2)
unzip(zip_admin, exdir = tmp_admin)
unzip(zip_roads, exdir = tmp_roads)
unzip(zip_grid, exdir = tmp_grid)
unzip(zip_plants, exdir = tmp_plants)
```

```r
w1_csv <- file.path(
 tmp_w1,
 "ETH_2011_ERSS_v02_M_CSV",
 "pub_eth_householdgeovariables_y1.csv"
)
w2_dta <- file.path(tmp_w2, "Pub_ETH_HouseholdGeovars_Y2.dta")

find_shp <- function(dir) {
 list.files(dir, pattern = "\\.shp$", full.names = TRUE, recursive = TRUE)[1]
}

admin_shp <- find_shp(tmp_admin)
roads_shp <- find_shp(tmp_roads)
grid_shp <- find_shp(tmp_grid)
plants_shp <- find_shp(tmp_plants)
```

```r
admin <- st_read(admin_shp, quiet = TRUE)
roads <- st_read(roads_shp, quiet = TRUE)
grid <- st_read(grid_shp, quiet = TRUE)
plants <- st_read(plants_shp, quiet = TRUE)
```

1.5 Processing the survey data

```r
w1 <- read_csv(w1_csv, show_col_types = FALSE) |>
 mutate(
   lat_dd_mod = as.numeric(LAT_DD_MOD),
   lon_dd_mod = as.numeric(LON_DD_MOD),
   h2011_tot = as.numeric(h2011_tot),
   qa_type = as.integer(qa_type),
   ea_id11 = str_pad(ea_id, width = 11, side = "left", pad = "0")
 )
```

```r
w2 <- read_dta(w2_dta) |>
 mutate(
   lat_dd_mod = as.numeric(lat_dd_mod),
   lon_dd_mod = as.numeric(lon_dd_mod),
   ea_id11 = str_pad(ea_id, width = 11, side = "left", pad = "0")
 )
```

```r
w1_ea <- w1 |>
 group_by(ea_id11) |>
 summarise(
   lat = mean(lat_dd_mod, na.rm = TRUE),
   lon = mean(lon_dd_mod, na.rm = TRUE),
   qa_type = first(qa_type),
   pop2011 = mean(h2011_tot, na.rm = TRUE),
   .groups = "drop"
 )

w2_ea <- w2 |>
 group_by(ea_id11) |>
 summarise(
   lat = mean(lat_dd_mod, na.rm = TRUE),
   lon = mean(lon_dd_mod, na.rm = TRUE),
   .groups = "drop"
 )

cat("Wave 1:", nrow(w1_ea), "villages\n")
```

```
## Wave 1: 333 villages
```

```r
cat("Wave 2:", nrow(w2_ea), "villages\n")
```

```
## Wave 2: 334 villages
```

```r
panel_ids <- intersect(w1_ea$ea_id11, w2_ea$ea_id11)
panel <- w1_ea |> filter(ea_id11 %in% panel_ids)
cat("Panel villages:", length(panel_ids), "\n")
```

```
## Panel villages: 333
```

```r
haversine_km <- function(lon1, lat1, lon2, lat2) {
 r <- 6371
 rad <- pi / 180
 d_lat <- (lat2 - lat1) * rad
 d_lon <- (lon2 - lon1) * rad
 a <- sin(d_lat / 2)^2 + cos(lat1 * rad) * cos(lat2 * rad) * sin(d_lon / 2)^2
 2 * r * asin(sqrt(a))
}

panel <- panel |>
 mutate(dist_addis_km = haversine_km(lon, lat, 38.74, 9.03))
```

```r
villages <- panel |>
 filter(qa_type == 1, dist_addis_km >= 25, pop2011 <= 10000)

cat("Final sample:", nrow(villages), "villages\n")
```

```
## Final sample: 315 villages
```

1.6 Creating the population density surface

```r
crs_proj <- 3857

vill_sf <- villages |>
 st_as_sf(coords = c("lon", "lat"), crs = 4326) |>
 st_transform(crs_proj)

ea_sf <- w1_ea |>
 st_as_sf(coords = c("lon", "lat"), crs = 4326) |>
 st_transform(crs_proj)

admin_proj <- st_transform(admin, crs_proj)
roads_proj <- st_transform(roads, crs_proj)
grid_proj <- st_transform(grid, crs_proj)

st_crs(plants) <- 4326
plants_proj <- st_transform(plants, crs_proj)
```

```r
xy <- st_coordinates(ea_sf)
weights <- ea_sf$pop2011
weights[is.na(weights)] <- 0
```

```r
bbox <- st_bbox(admin_proj)

weights_scaled <- pmax(1, round(scales::rescale(
 weights,
 to = c(1, 10),
 from = quantile(weights, c(0.05, 0.95), na.rm = TRUE)
)))
xy_rep <- xy[rep(seq_len(nrow(xy)), weights_scaled), ]

kde <- kde2d(
 x = xy_rep[, 1],
 y = xy_rep[, 2],
 n = c(360, 280),
 lims = c(bbox["xmin"], bbox["xmax"], bbox["ymin"], bbox["ymax"])
)
```

```r
dens_stars <- st_as_stars(
 list(density = kde$z),
 dimensions = st_dimensions(x = kde$x, y = kde$y)
)
st_crs(dens_stars) <- crs_proj

dens_df <- as.data.frame(dens_stars, xy = TRUE)
names(dens_df) <- c("x", "y", "density")
```

1.7 Preparing the roads

```r
road_classes <- c("motorway", "trunk", "primary", "secondary", "tertiary")

roads_filtered <- roads_proj |>
 filter(highway %in% road_classes) |>
 mutate(
   road_type = factor(highway, levels = road_classes),
   line_width = case_when(
     highway %in% c("motorway", "trunk") ~ 1.0,
     highway == "primary" ~ 0.7,
     highway == "secondary" ~ 0.5,
     highway == "tertiary" ~ 0.3
   )
 )
```

1.8 Building the map

```
ethiopia_boundary <- st_union(admin_proj)
dens_pts <- st_as_sf(dens_df, coords = c("x", "y"), crs = crs_proj)
inside <- st_intersects(dens_pts, ethiopia_boundary, sparse = FALSE)[, 1]
dens_df <- dens_df[inside, ]

grid_leg <- grid_proj |> mutate(layer = "High-Voltage Grid")
plants_leg <- plants_proj |> mutate(layer = "Power Plants")
vill_leg <- vill_sf |> mutate(layer = "ERSS Sample Villages")

ggplot() +
 geom_raster(data = dens_df, aes(x, y, fill = density), alpha = 0.85) +
 scale_fill_gradient(
   low = "#d4e6f1", high = "#1a5276",
   name = "Population\nDensity",
   guide = guide_colorbar(
     title.position = "top", title.hjust = 0.5,
     barwidth = 1.2, barheight = 6,
     frame.colour = "grey30", ticks.colour = "grey30",
     order = 1
   ),
   labels = c("Low", "", "High"),
   breaks = function(x) c(min(x), mean(x), max(x))
 ) +
 ggnewscale::new_scale_fill() +
 geom_sf(data = admin_proj, fill = NA, color = "white", linewidth = 0.3) +
 geom_sf(
   data = roads_filtered,
   aes(linewidth = line_width),
   color = "grey20", alpha = 0.6
 ) +
 scale_linewidth_identity() +
 geom_sf(data = grid_leg, aes(color = layer), linewidth = 0.6, alpha = 0.9) +
 scale_color_manual(
   name = NULL,
   values = c("High-Voltage Grid" = "#c0392b"),
   guide = guide_legend(order = 2, override.aes = list(linewidth = 1))
 ) +
 geom_sf(
   data = plants_leg,
   aes(shape = layer),
   color = "#d35400",
   fill = "#f39c12",
   size = 3.5,
   stroke = 0.5
 ) +
 scale_shape_manual(
   name = NULL,
   values = c("Power Plants" = 23),
   guide = guide_legend(order = 3, override.aes = list(size = 4))
 ) +
 ggnewscale::new_scale("shape") +
 geom_sf(
   data = vill_leg, aes(shape = layer),
```

```
  fill = "white", color = "black", size = 2.2, stroke = 0.5
) +
scale_shape_manual(
  name = NULL,
  values = c("ERSS Sample Villages" = 21),
  guide = guide_legend(order = 4, override.aes = list(size = 4))
) +
coord_sf(
  crs = crs_proj, datum = NA,
  xlim = bbox[c("xmin", "xmax")],
  ylim = bbox[c("ymin", "ymax")]
) +
labs(
  title = "Figure 4: Ethiopian Population Density and ERSS Sample Villages",
  caption = paste(
    "Data: ERSS (World Bank), Roads (OpenStreetMap/HDX), Grid & Power Plants (energydata.info)",
    "Road thickness: motorway/trunk > primary > secondary > tertiary",
    sep = "\n"
  )
) +
theme_void(base_size = 11) +
theme(
  plot.title = element_text(
    size = 14, face = "bold", hjust = 0.5, margin = margin(b = 5)
  ),
  plot.caption = element_text(
    size = 8, color = "grey40", hjust = 0, margin = margin(t = 10)
  ),
  legend.position = "right",
  legend.box = "vertical",
  legend.title = element_text(size = 10, face = "bold"),
  legend.text = element_text(size = 9),
  legend.spacing.y = unit(0.3, "cm"),
  legend.margin = margin(l = 10),
  plot.margin = margin(15, 15, 15, 15)
)
```

Figure 4: Ethiopian Population Density and ERSS Sample Villages

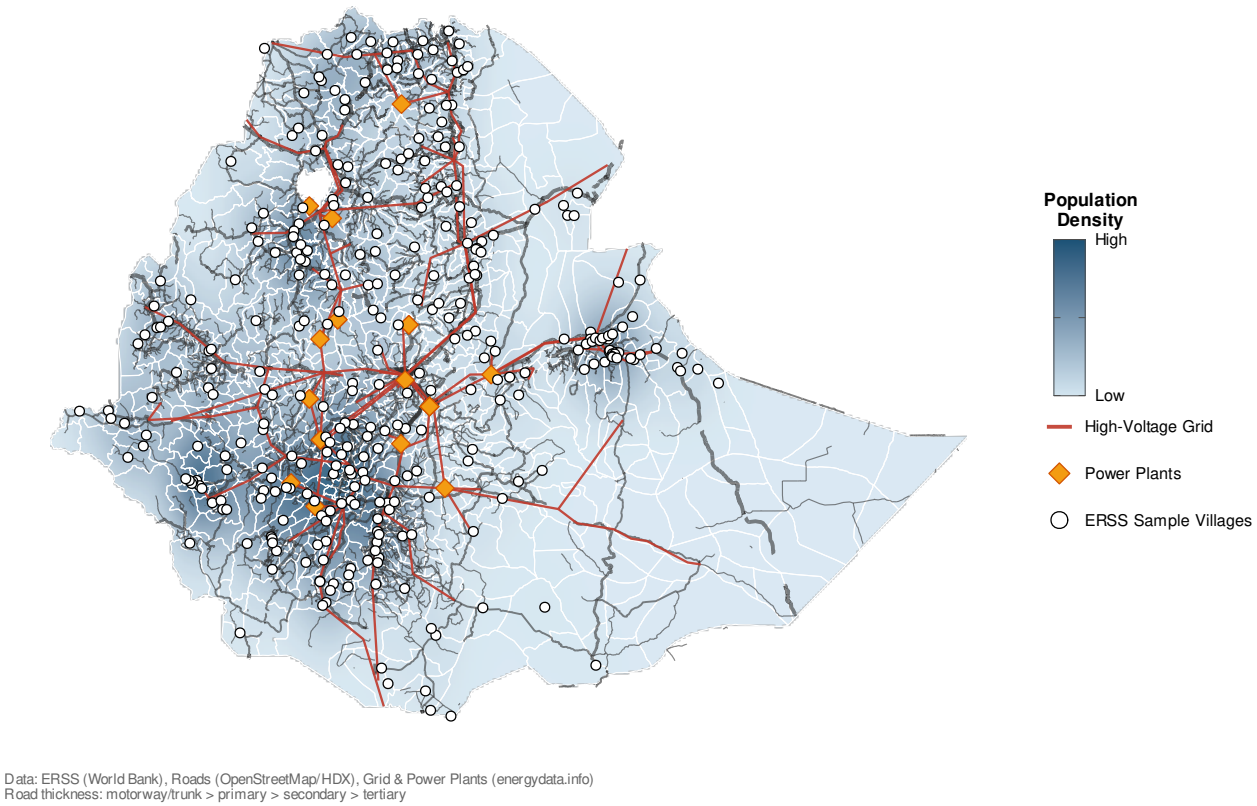


Figure 1: Our replication of Figure 4 from Fried & Lagakos (2021).

2 Part 2: Pellegrina & Sotelo (2025) — Figure 2

2.1 What the map shows

Figure 2 consists of three choropleth maps showing Brazil's population distribution by mesoregion in 1950, 1980, and 2010. Darker greens indicate higher population shares. A black line marks Brazil's border, and a red outline defines the "West" region—the agricultural frontier that received migrants during the so-called "March to the West." The maps illustrate how population shifted from the Atlantic coast towards the interior over six decades.

2.2 Data sources

All data come from the authors' replication files, available through the journal's supplementary materials:

- Mesoregions shapefile (137 administrative units)
- West boundary shapefile
- Brazil outline shapefile
- Population shares for each year (CSV)

2.3 Why our map differs

Our replication matches the original quite closely since we used the exact same data the authors provided. The only minor differences are styling choices—we used a grey palette instead of blue, and our legend formatting is slightly different. The geographic patterns and data are identical.

2.4 Setup

```
if (!require("pacman")) install.packages("pacman")

pacman::p_load(
  dplyr,
  ggplot2,
  rmapshaper,
  sf
)
```

2.5 Loading the Data

```
brazil <- st_read(
  "./data/Pellegrina_Sotelo_2025/raw/maps/Artisanal",
  layer = "mesoregions",
  quiet = TRUE
)
```

```
west <- st_read(
  "./data/Pellegrina_Sotelo_2025/raw/maps/Artisanal",
  layer = "westregions",
  quiet = TRUE
)

brazil_all <- st_read(
  "./data/Pellegrina_Sotelo_2025/raw/maps/Artisanal",
  layer = "brazil",
  quiet = TRUE
)
```

```
brazil <- ms_simplify(brazil, keep = 0.05, keep_shapes = TRUE)
west <- ms_simplify(west, keep = 0.05, keep_shapes = TRUE)
brazil_all <- ms_simplify(brazil_all, keep = 0.05, keep_shapes = TRUE)
```

```
table_path <- "./data/Pellegrina_Sotelo_2025/output/fact_maps.csv"

data_fact <- read.table(
  table_path,
  sep = ",",
  header = FALSE,
  na.strings = "NaN",
  col.names = c("meso_code", "pop1950", "pop1980", "pop2010")
) |>
  mutate(meso_code = as.character(meso_code))
```

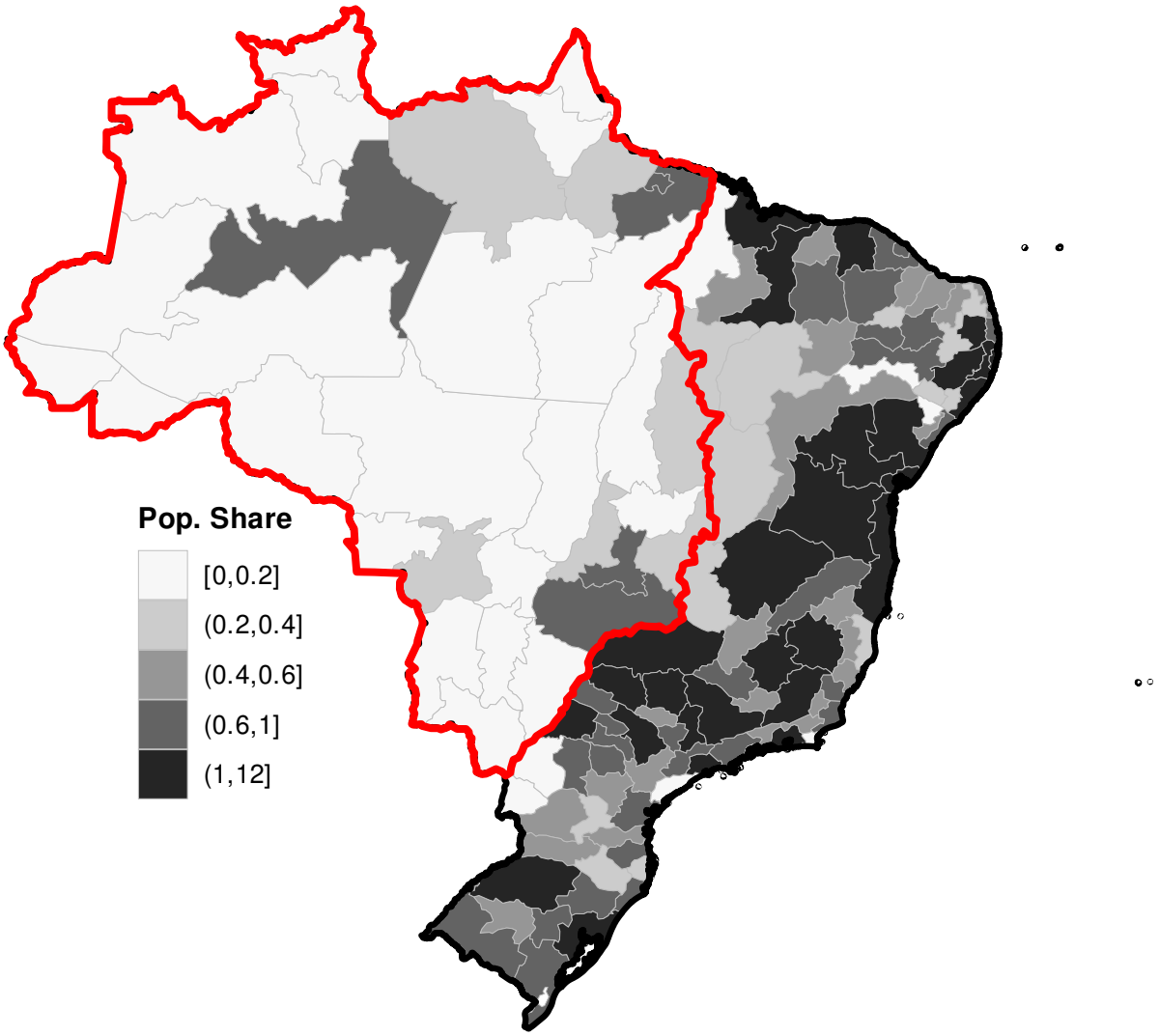
2.6 Merging and plotting

```
map_df <- brazil |>
  left_join(data_fact, by = c("mesocodefo" = "meso_code"))

outline <- st_union(brazil_all)
west_outline <- st_union(west)
```

```
years <- c("1950", "1980", "2010")
breaks <- c(0, 0.2, 0.4, 0.6, 1, 12)

for (year in years) {

  map_df <- map_df |>
    mutate(
      value = get(paste0("pop", year)),
      quintiles = cut(value, breaks = breaks, include.lowest = TRUE)
    )

  plot_map <- ggplot(data = map_df) +
    geom_sf(aes(fill = quintiles), color = "grey", linewidth = 0.1) +
```

```
    geom_sf(data = outline, color = "black", fill = NA, linewidth = 1.00) +
    geom_sf(data = west_outline, color = "red", fill = NA, linewidth = 1.25) +
    scale_fill_brewer(
      palette = "Greys",
      name = "Pop. Share",
      na.value = "grey90"
    ) +
    labs(
      title = paste("Figure 2: Population Distribution in Brazil,", year),
      caption = "Source: Pellegrina & Sotelo (2025) replication data"
    ) +
    theme_void(base_size = 11) +
    theme(
      plot.title = element_text(
        size = 14, face = "bold", hjust = 0.5, margin = margin(b = 10)
      ),
      plot.caption = element_text(
        size = 8, color = "grey40", hjust = 0, margin = margin(t = 10)
      ),
      legend.position = c(0.15, 0.25),
      legend.justification = c(0, 0),
      legend.background = element_rect(fill = "white", color = NA),
      legend.title = element_text(size = 10, face = "bold"),
      legend.text = element_text(size = 9)
    )

  print(plot_map)
}
```

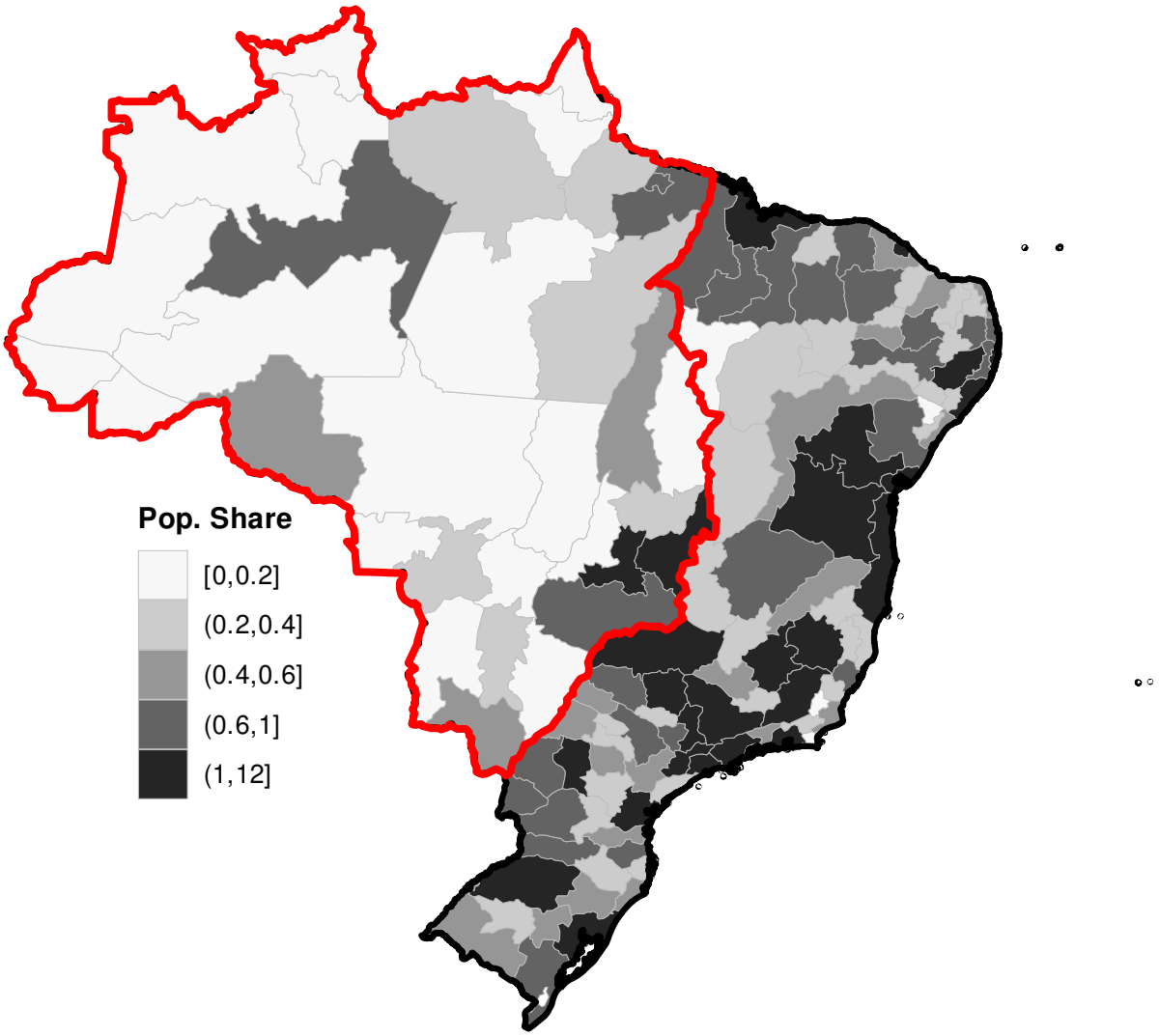
Figure 2: Population Distribution in Brazil, 1950



Source: Pellegrina & Sotelo (2025) replication data

Figure 2: Population share maps for Brazil's mesoregions in 1950, 1980, and 2010. The red outline indicates the 'West' region.

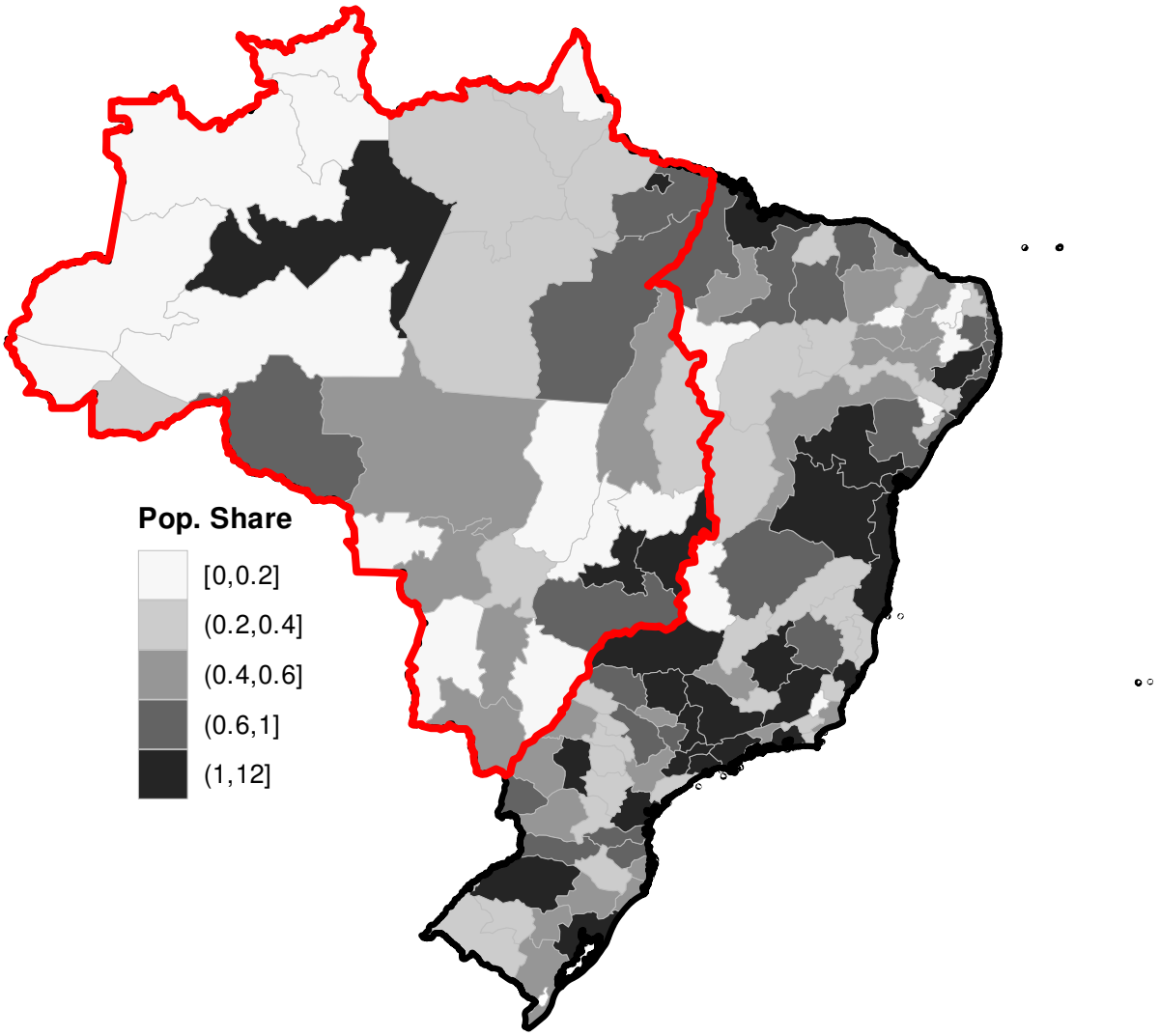
Figure 2: Population Distribution in Brazil, 1980



Source: Pellegrina & Sotelo (2025) replication data

Figure 3: Population share maps for Brazil's mesoregions in 1950, 1980, and 2010. The red outline indicates the 'West' region.

Figure 2: Population Distribution in Brazil, 2010



Source: Pellegrina & Sotelo (2025) replication data

Figure 4: Population share maps for Brazil's mesoregions in 1950, 1980, and 2010. The red outline indicates the 'West' region.

3 Part 3: Morten & Oliveira (2024) — Figure 1

3.1 What the map shows

Figure 1 displays Brazil's federal highway network as of 2000, distinguishing between radial highways (those emanating from Brasília, shown as solid lines), non-radial highways (dashed lines), and the minimum spanning tree instrument (dotted lines). The MST represents the hypothetical road network that would connect Brasília to state capitals using least-cost paths—this serves as the authors' instrument for actual road placement. Capital cities appear as points, with Brasília highlighted as the hub.

3.2 Data sources

The authors provide all data through openICPSR:

- Replication package — openICPSR #183316

This includes:

- State boundaries (1940 vintage)
- Highway network (2000)
- MST instrument
- Capital city locations

3.3 Why our map differs

We used the exact same shapefiles from the replication package, so the geographic content is identical. We made a few aesthetic changes: we used colors (amber for MST, steel-blue for non-radial, dark charcoal for radial) instead of all-grey tones, added a halo effect around Brasília, and used `ggrepel` for cleaner labels. These changes make it easier to distinguish the three network types at a glance.

3.4 Setup

```
pacman::p_load(
  dplyr,
  ggplot2,
  ggrepel,
  rmapshaper,
  sf
)
```

```
gis_path <- "./data/Morten_Oliveira_2024/GIS_data"
```

3.5 Loading the Data

```r
states <- st_read(
  file.path(gis_path, "uf1940/uf1940_prj.shp"),
  quiet = TRUE
)

states_simple <- ms_simplify(states, keep = 0.01, keep_shapes = TRUE)
```

```r
all_highways <- st_read(
  file.path(gis_path, "roads/2000/highways_2000_prj.shp"),
  quiet = TRUE
)

all_highways_simple <- ms_simplify(all_highways, keep = 0.01, keep_shapes = TRUE)

radial_highways <- all_highways_simple |>
  filter(dm_anlys_p == 1 & dm_radial == 1)

nonradial_highways <- all_highways_simple |>
  filter(dm_anlys_p == 1 & dm_radial == 0)
```

```r
mst_pie <- st_read(
  file.path(gis_path, "mst/mst_pie_prj.shp"),
  quiet = TRUE
)

mst_pie_simple <- ms_simplify(mst_pie, keep = 0.01, keep_shapes = TRUE)
```

```r
capital_cities <- st_read(
  file.path(gis_path, "cities/brazil_capital_cities_prj.shp"),
  quiet = TRUE
)

cities_xy <- capital_cities |>
  cbind(st_coordinates(capital_cities)) |>
  rename(lng = X, lat = Y)

brasilia <- cities_xy |>
  filter(grepl("Bras", CITY_NAME, ignore.case = TRUE))

other_cities <- cities_xy |>
  filter(!grepl("Bras", CITY_NAME, ignore.case = TRUE))
```

3.6 Building the Map

```r
colors_mo <- c(
  "Minimum spanning tree" = "#D4A03E",
  "Non-radial highways (2000)" = "#7B8D9E",
```

```
  "Radial highways (2000)" = "#2C2C2C"
)

fig_1 <- ggplot() +
  geom_sf(
    data = states_simple,
    fill = "#FDFBF7",
    color = "#D0D0D0",
    linewidth = 0.3
  ) +
  geom_sf(
    data = mst_pie_simple,
    aes(color = "Minimum spanning tree"),
    linewidth = 0.9,
    linetype = "dotted",
    show.legend = "line"
  ) +
  geom_sf(
    data = nonradial_highways,
    aes(color = "Non-radial highways (2000)"),
    linewidth = 0.6,
    linetype = "dashed",
    show.legend = "line"
  ) +
  geom_sf(
    data = radial_highways,
    aes(color = "Radial highways (2000)"),
    linewidth = 1.0,
    linetype = "solid",
    show.legend = "line"
  ) +
  geom_point(
    data = brasilia,
    aes(x = lng, y = lat),
    size = 7,
    color = "#D4A03E",
    alpha = 0.35
  ) +
  geom_point(
    data = cities_xy,
    aes(x = lng, y = lat),
    size = 2.0,
    color = "#2C2C2C"
  ) +
  geom_point(
    data = brasilia,
    aes(x = lng, y = lat),
    size = 4.0,
    color = "#2C2C2C"
  ) +
  geom_text_repel(
    data = other_cities,
    aes(x = lng, y = lat, label = CITY_NAME),
```

```
  size = 2.7,
  color = "#3A3A3A",
  segment.color = "#BBBBBB",
  segment.size = 0.25,
  box.padding = 0.35,
  point.padding = 0.25,
  max.overlaps = 30,
  seed = 42
) +
geom_text_repel(
  data = brasilia,
  aes(x = lng, y = lat, label = CITY_NAME),
  size = 3.5,
  fontface = "bold",
  color = "#1A1A1A",
  nudge_y = 1.8,
  segment.color = "#999999",
  seed = 42
) +
scale_color_manual(
  name = NULL,
  values = colors_mo,
  guide = guide_legend(
    override.aes = list(
      linetype = c("dotted", "dashed", "solid"),
      linewidth = c(1.3, 1.0, 1.3)
    )
  )
) +
coord_sf(
  crs = st_crs(states_simple),
  datum = NA
) +
theme_minimal(base_size = 11) +
theme(
  axis.title = element_blank(),
  axis.text = element_blank(),
  axis.ticks = element_blank(),
  panel.grid = element_blank(),
  panel.background = element_rect(fill = "#EDF3F7", color = NA),
  legend.position = c(0.17, 0.20),
  legend.justification = c(0, 0),
  legend.background = element_rect(
    fill = alpha("white", 0.92),
    color = "#CCCCCC",
    linewidth = 0.3
  ),
  legend.key.width = unit(1.4, "cm"),
  legend.text = element_text(size = 9),
  legend.margin = margin(6, 10, 6, 8),
  plot.margin = margin(5, 5, 5, 5)
)
```

```
print(fig_1)
```

Figure 5: Replication of Figure 1 from Morten & Oliveira (2024). Data source: openICPSR #183316.

4 Part 4: Mettetal (2019) — Figure 2

4.1 What the map shows

Figure 2 displays average river gradient by magisterial district in South Africa, with darker shades indicating steeper rivers. Black dots mark the location of irrigation dams. The map illustrates the paper's identification strategy: dams tend to cluster in areas with gentler river slopes (lighter shades), particularly on the eastern side of the country where most irrigated agriculture is concentrated.

4.2 Data sources

We assembled layers from several public sources:

- Digital elevation model — geodata R package (90m SRTM)
- District boundaries — GADM via geodata package
- Dam locations — South Africa Dam Safety Office (KML file)
- River network — SA Department of Water Affairs
- Dam registry — SA Department of Water Affairs (Excel file with dam purposes)

4.3 Why our map differs

The main difference is that we could not access the exact district boundaries and river gradient calculations the author used. We computed river gradient ourselves from SRTM elevation data, using a 1.5 km buffer around rivers to extract slope values. Our sextile classification may differ slightly from the original due to differences in the underlying DEM and the exact method used to aggregate slopes to district level. However, the overall pattern—steeper rivers in the west, gentler rivers and more dams in the east—matches the original figure well.

4.4 Setup

```
pacman::p_load(
  sf,
  terra,
  tidyverse,
  geodata,
  exactextractr,
  readxl
)
```

```
dir_data <- "./data/Mettetal_2019/"
dir.create(dir_data, showWarnings = FALSE, recursive = TRUE)
```

4.5 Loading the Data

```r
sa_boundary <- geodata::gadm(country = "ZAF", level = 0, path = dir_data)

dem_raw <- geodata::elevation_global(res = 0.5, country = "ZAF", path = dir_data)
dem <- terra::crop(dem_raw, sa_boundary, mask = TRUE)

rm(dem_raw)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  8389923 448.1   12184651  650.8  12184651  650.8
## Vcells 59630149 455.0  178176326 1359.4 143633274 1095.9
```

```r
sf_districts <- st_read(
  file.path(dir_data, "Magisterialdistricts2005"),
  "District municipalities 2005"
) |>
  st_make_valid()
```

```
## Reading layer 'District municipalities 2005' from data source
##   '/home/michael/Desktop/BSE_DSM/SecondTerm/GeospatialDataScience/Assignments/Assignment_1/papers/da
##   using driver 'ESRI Shapefile'
## Simple feature collection with 371 features and 18 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 16.45485 ymin: -34.83305 xmax: 32.89128 ymax: -22.12595
## Geodetic CRS:  GCS_Assumed_Geographic_1
```

```r
sf_dams <- st_read(file.path(dir_data, "doc.kml")) |>
  st_make_valid() |>
  st_set_crs(4326)
```

```
## Reading layer 'GEarth dams' from data source
##   '/home/michael/Desktop/BSE_DSM/SecondTerm/GeospatialDataScience/Assignments/Assignment_1/papers/da
##   using driver 'LIBKML'
## Simple feature collection with 5744 features and 21 fields
## Geometry type: POINT
## Dimension:     XYZ
## Bounding box:  xmin: 17.5755 ymin: -34.67222 xmax: 33.65056 ymax: -18.81806
## z_range:       zmin: 0 zmax: 0
## Geodetic CRS:  WGS 84
```

```r
sf_rivers <- st_read(file.path(dir_data, "All"), "wriall500") |>
  st_make_valid()
```

```
## Reading layer 'wriall500' from data source
##   '/home/michael/Desktop/BSE_DSM/SecondTerm/GeospatialDataScience/Assignments/Assignment_1/papers/da
##   using driver 'ESRI Shapefile'
## Simple feature collection with 10352 features and 26 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: 15.98333 ymin: -34.76999 xmax: 33.99925 ymax: -19.79143
## CRS:           NA
```

```r
dams_info <- read_excel(file.path(dir_data, "List of Registered Dams Jul2025.xlsx"))

dams_irrigation <- dams_info |>
  distinct() |>
  filter(str_detect(Purpose, regex("IRRIGATION", ignore_case = TRUE))) |>
  distinct(`No of dam`, .keep_all = TRUE)

sf_dams_filtered <- sf_dams |>
  inner_join(dams_irrigation, by = c("Name" = "Name of dam"))
```

4.6 Computing River Gradient

```r
if (is.na(st_crs(sf_districts))) {
  sf_districts <- st_set_crs(sf_districts, 4326)
}

if (is.na(st_crs(sf_rivers))) {
  sf_rivers <- st_set_crs(sf_rivers, 4326)
}

if (is.na(st_crs(sf_dams_filtered))) {
  sf_dams_filtered <- st_set_crs(sf_dams_filtered, 4326)
}

rivers_vect <- project(vect(sf_rivers), dem)
districts_vect <- project(vect(sf_districts), dem)
dams_vect <- project(vect(sf_dams_filtered), dem)
```

```r
rivers_buf <- buffer(rivers_vect, width = 1500)
river_mask <- rasterize(rivers_buf, dem, field = 1)
dem_rivers_only <- mask(dem, river_mask)

rm(rivers_buf, river_mask, rivers_vect)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  8527960 455.5   15911407  849.8  12184651  650.8
## Vcells 64586449 492.8  178176326 1359.4 143633274 1095.9
```

```r
slope_deg <- terrain(dem_rivers_only, "slope", unit = "degrees")
slope_pct <- tan(slope_deg * pi / 180) * 100
```

```r
steep_rivers <- slope_pct > 6
sf_districts$fraction_steep <- exact_extract(steep_rivers, sf_districts, "mean")
```

```
##   |                                                                      |
```

4.7 Classifying Districts

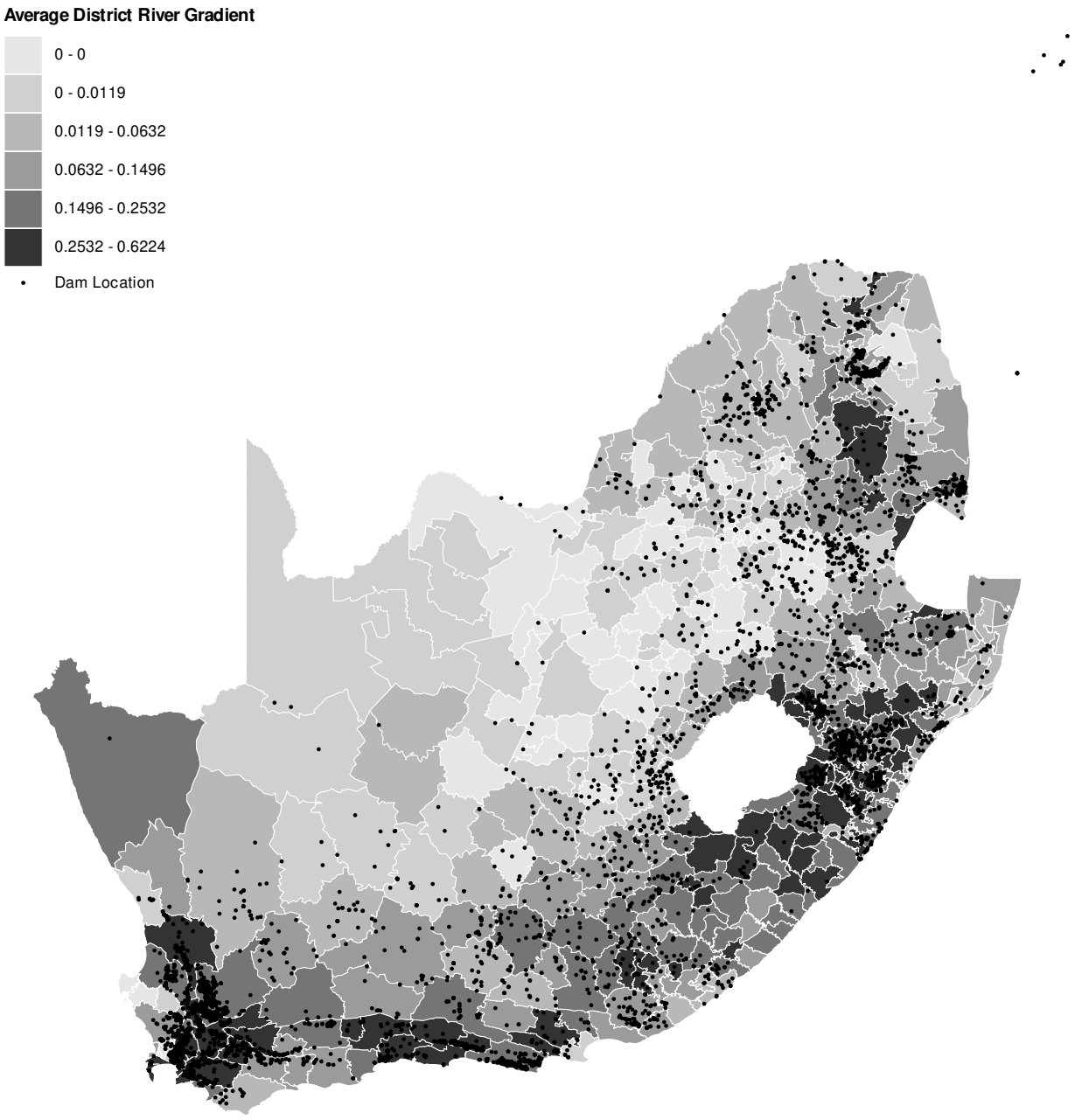
```
sf_districts_clean <- sf_districts |>
  filter(!is.na(fraction_steep))

sf_districts_clean$fraction_jitter <- jitter(
  sf_districts_clean$fraction_steep,
  amount = 0.00001
)

probs <- seq(0, 1, length.out = 7)
breaks <- quantile(sf_districts_clean$fraction_jitter, probs = probs, na.rm = TRUE)

labels_percentile <- paste0(
  head(round(breaks, 4), -1),
  " - ",
  tail(round(breaks, 4), -1)
)

sf_districts_clean <- sf_districts_clean |>
  mutate(
    slope_percentile = cut(
      fraction_jitter,
      breaks = breaks,
      labels = labels_percentile,
      include.lowest = TRUE
    )
  )
```

4.8 Building the Map

```
p <- ggplot() +
  geom_sf(
    data = sf_districts_clean,
    aes(fill = slope_percentile),
    color = "white",
    linewidth = 0.1
  ) +
  geom_sf(
    data = st_as_sf(dams_vect),
    aes(color = "Dam Location"),
    size = 0.2
  ) +
  scale_fill_grey(
    name = "Average District River Gradient",
    start = 0.9,
    end = 0.2
  ) +
  scale_color_manual(
```

```
    name = NULL,
    values = c("Dam Location" = "black")
  ) +
  labs(
    title = "Figure 2: River Gradients and Dam Locations - South Africa",
    caption = "Source: SA Department of Water Affairs, GADM, SRTM elevation data"
  ) +
  theme_void() +
  theme(
    plot.title = element_text(
      size = 14, face = "bold", hjust = 0.5, margin = margin(b = 10)
    ),
    plot.caption = element_text(
      size = 8, color = "grey40", hjust = 0, margin = margin(t = 10)
    ),
    legend.position = c(0.02, 0.98),
    legend.justification = c(0, 1),
    legend.title = element_text(size = 8, face = "bold"),
    legend.text = element_text(size = 7),
    legend.spacing.y = unit(-0.05, "cm")
  ) +
  guides(
    fill = guide_legend(order = 1),
    color = guide_legend(order = 2, override.aes = list(linetype = 0))
  )

print(p)
```

Figure 2: River Gradients and Dam Locations - South Africa



Source: SA Department of Water Affairs, GADM, SRTM elevation data

Figure 6: Replication of Figure 2 from Mettetal (2019). Darker shades indicate steeper rivers; black dots show irrigation dams.

5 Part 5: Balboni (2021) — Figure 3

5.1 What the map shows

Figure 3 displays Vietnam's road network in 2000 and 2010, illustrating the spatial distribution of road upgrades during a period of major transport infrastructure investment. Roads are categorized into five types: freeways, dual carriageways, major roads, minor roads, and other roads. The maps reveal how investments were concentrated in the low elevation coastal zone, particularly in the Red River Delta and Mekong River Delta regions. The paper uses this infrastructure data to study whether road investments should continue to favor coastal areas given rising sea levels and climate vulnerability.

5.2 Data sources

Since the original georeferenced road data from the paper is not publicly available, we use OpenStreetMap extracts as proxies:

- Vietnam Roads 2015 — Humanitarian Data Exchange (OSM extract)
- Vietnam Roads 2026 — Geofabrik (current OSM extract)
- Country boundary — GADM via `{geodata}` R package

5.3 Why our map differs

We could not replicate the original maps exactly because the author's manually digitized road network data from ITMB travel maps is not publicly available. Instead, we use OpenStreetMap data from 2015 and Geofabrik for 2026 data as modern approximations. The road classification also differs slightly: we infer road hierarchy from OSM's `highway` tags rather than the author's six-category system based on physical road characteristics. Despite these differences, the maps capture the same essential pattern—a dense network concentrated along the coast and major corridors, with the highest-quality roads connecting major urban centers.

5.4 Setup

```
pacman::p_load(
  sf,
  dplyr,
  ggplot2,
  geodata
)
```

```
colors_balboni <- c("blue", "darkgreen", "red", "orange", "gold")

vehicular_classes <- c(
  "motorway", "motorway_link",
  "trunk", "trunk_link",
  "primary", "primary_link",
  "secondary", "secondary_link",
  "tertiary", "tertiary_link",
```

```
  "road"
)

linewidths <- c(0.8, 0.6, 0.4, 0.2, 0.1)
legend_widths <- c(1.2, 0.9, 0.6, 0.3, 0.15)
```

5.5 Loading the Data

```
vn_border <- gadm(country = "VNM", level = 0, path = tempdir()) |>
  st_as_sf()
```

```
r15_raw <- st_read(
  "./data/Balboni_2021/vnm_rdsl_2015_osm/",
  "vnm_rdsl_2015_OSM",
  quiet = TRUE
)

r15 <- r15_raw |>
  filter(type %in% vehicular_classes) |>
  mutate(
    road_type = case_when(
      type %in% c("motorway", "motorway_link") ~ "Freeway",
      type %in% c("trunk", "trunk_link") ~ "Trunk / Dual Carriageway",
      type %in% c("primary", "primary_link") ~ "Primary Road",
      type %in% c("secondary", "secondary_link") ~ "Secondary Road",
      TRUE ~ "Tertiary / Local Road"
    ),
    road_type = factor(
      road_type,
      levels = c(
        "Freeway",
        "Trunk / Dual Carriageway",
        "Primary Road",
        "Secondary Road",
        "Tertiary / Local Road"
      )
    ),
    road_type = droplevels(road_type)
  )
```

```
r26_raw <- st_read(
  "./data/Balboni_2021/vietnam-260126-free/",
  "gis_osm_roads_free_1",
  quiet = TRUE
)

r26 <- r26_raw |>
  filter(fclass %in% vehicular_classes) |>
  mutate(
```

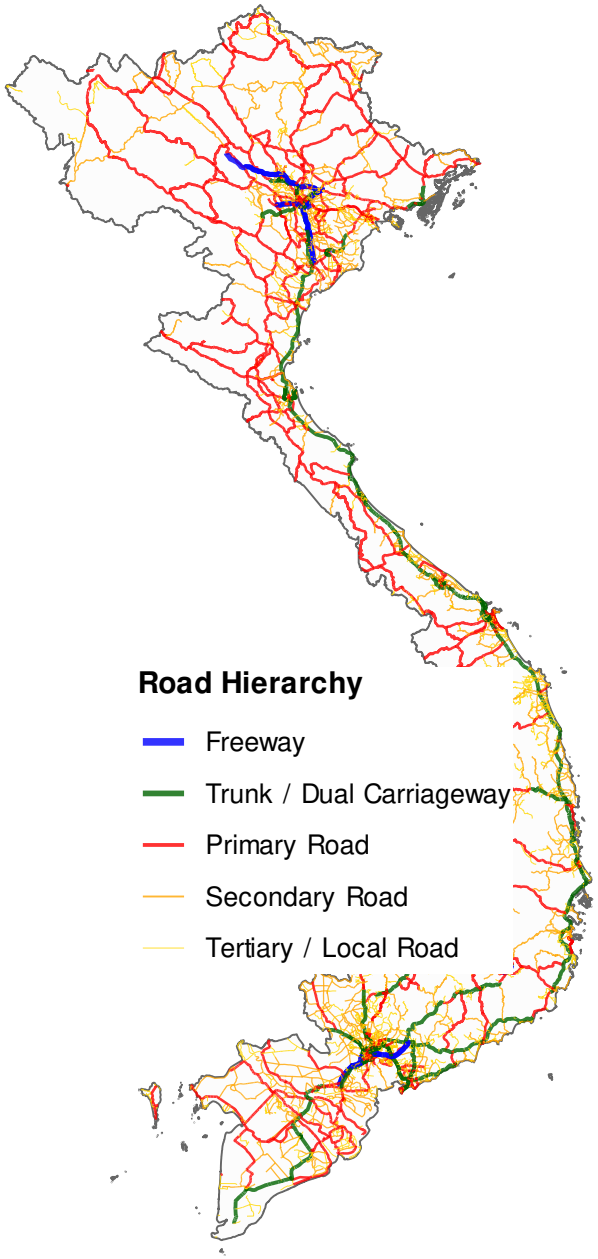
```
    road_type = case_when(
      fclass %in% c("motorway", "motorway_link") ~ "Freeway",
      fclass %in% c("trunk", "primary") & oneway %in% c("T", "B") ~ "Dual carriageway",
      fclass %in% c("trunk", "trunk_link", "primary", "primary_link") ~ "Major roads",
      fclass %in% c("secondary", "secondary_link") ~ "Minor roads",
      TRUE ~ "Other roads"
    ),
    road_type = factor(
      road_type,
      levels = c(
        "Freeway",
        "Dual carriageway",
        "Major roads",
        "Minor roads",
        "Other roads"
      )
    ),
    road_type = droplevels(road_type)
  )
```

5.6 Building the Maps

```
n_levels_15 <- length(levels(r15$road_type))

p_2015 <- ggplot() +
  geom_sf(
    data = vn_border,
    fill = "#fcfcfc",
    color = "#666666",
    linewidth = 0.3
  ) +
  geom_sf(
    data = r15,
    aes(color = road_type, linewidth = road_type),
    alpha = 0.8
  ) +
  scale_color_manual(
    values = colors_balboni[1:n_levels_15],
    name = "Road Hierarchy"
  ) +
  scale_linewidth_manual(
    values = linewidths[1:n_levels_15],
    name = "Road Hierarchy"
  ) +
  guides(
    color = guide_legend(
      override.aes = list(linewidth = legend_widths[1:n_levels_15])
    )
  ) +
```

```
  labs(
    title = "Figure 3a: Vietnam Road Network (2015)",
    caption = "Data: OpenStreetMap via Humanitarian Data Exchange"
  ) +
  theme_void(base_size = 11) +
  theme(
    plot.title = element_text(
      size = 14, face = "bold", hjust = 0.5, margin = margin(b = 10)
    ),
    plot.caption = element_text(
      size = 8, color = "grey40", hjust = 0, margin = margin(t = 10)
    ),
    legend.position = c(0.25, 0.25),
    legend.justification = c(0, 0),
    legend.background = element_rect(fill = "white", color = NA),
    legend.title = element_text(size = 10, face = "bold"),
    legend.text = element_text(size = 9)
  )

print(p_2015)
```

Figure 3a: Vietnam Road Network (2015)



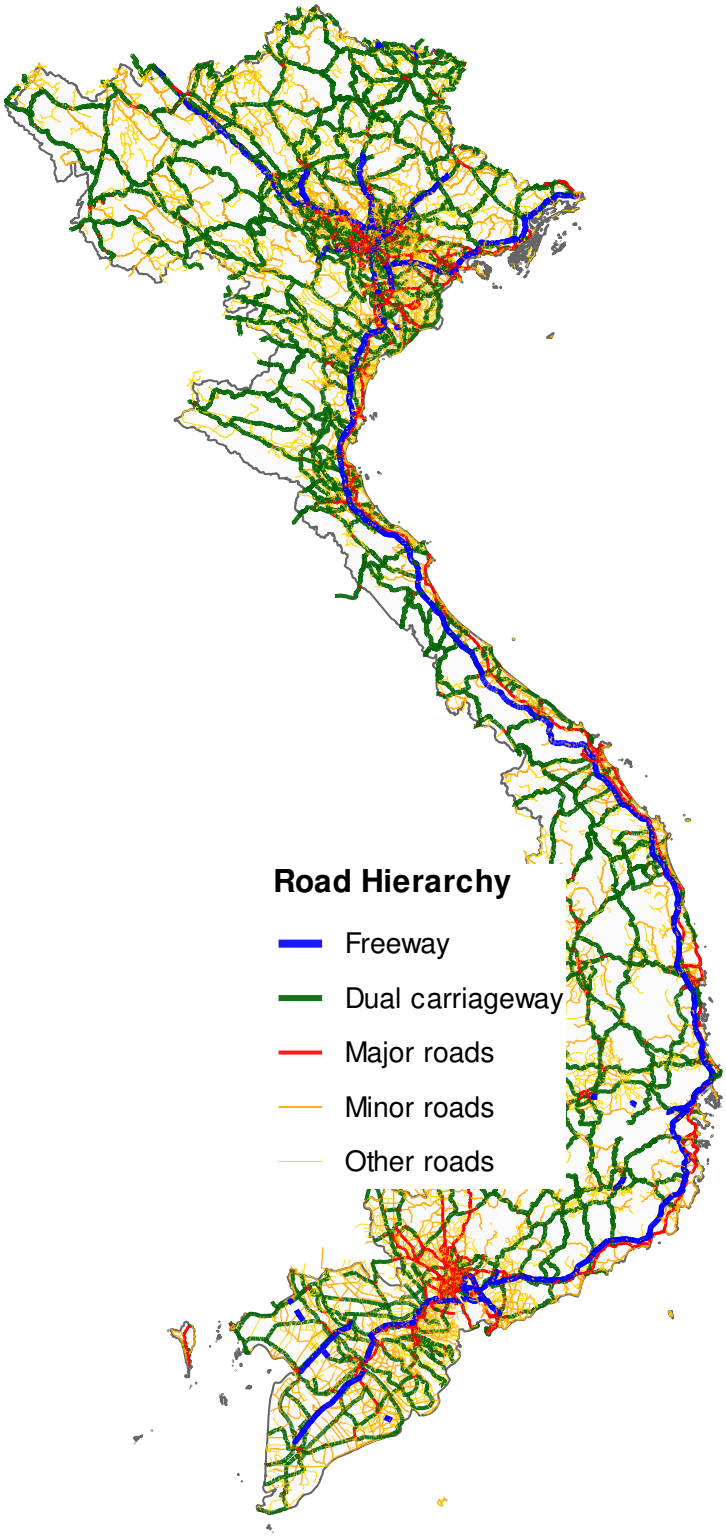
Data: OpenStreetMap via Humanitarian Data Exchange

Figure 7: Vietnam road network (2015 OSM data).

```
n_levels_26 <- length(levels(r26$road_type))

p_2026 <- ggplot() +
```

```
  geom_sf(
    data = vn_border,
    fill = "#fafafa",
    color = "#666666",
    linewidth = 0.3
  ) +
  geom_sf(
    data = r26,
    aes(color = road_type, linewidth = road_type),
    alpha = 0.9
  ) +
  scale_color_manual(
    values = colors_balboni[1:n_levels_26],
    name = "Road Hierarchy"
  ) +
  scale_linewidth_manual(
    values = linewidths[1:n_levels_26],
    name = "Road Hierarchy"
  ) +
  guides(
    color = guide_legend(
      override.aes = list(linewidth = legend_widths[1:n_levels_26])
    )
  ) +
  labs(
    title = "Figure 3b: Vietnam Road Network (2026)",
    caption = "Data: OpenStreetMap via Geofabrik"
  ) +
  theme_void(base_size = 11) +
  theme(
    plot.title = element_text(
      size = 14, face = "bold", hjust = 0.5, margin = margin(b = 10)
    ),
    plot.caption = element_text(
      size = 8, color = "grey40", hjust = 0, margin = margin(t = 10)
    ),
    legend.position = c(0.25, 0.25),
    legend.justification = c(0, 0),
    legend.background = element_rect(fill = "white", color = NA),
    legend.title = element_text(size = 10, face = "bold"),
    legend.text = element_text(size = 9)
  )

print(p_2026)
```

Figure 3b: Vietnam Road Network (2026)



Data: OpenStreetMap via Geofabrik

Figure 8: Vietnam road network (2026 Geofabrik).

6 References

Balboni, C. (2025). In Harm's Way? Infrastructure Investments and the Persistence of Coastal Cities. *American Economic Review*.

Fried, S., & Lagakos, D. (2021). Rural Electrification, Migration and Structural Transformation: Evidence from Ethiopia. *Regional Science and Urban Economics*.

Mettetal, E. (2019). Irrigation dams, water and infant mortality: Evidence from South Africa. *Journal of Development Economics*.

Morten, M., & Oliveira, J. (2024). The Effects of Roads on Trade and Migration: Evidence from a Planned Capital City. *American Economic Journal: Applied Economics*.

Pellegrina, H. S., & Sotelo, S. (2025). Migration, Specialization, and Trade: Evidence from Brazil's March to the West. *Journal of Political Economy*.