Assignment 2

Population Distribution, Transportation Infrastructure, and Market Price Analysis Across Africa

Samuel Gerard Fraley, Selve Rajayogan, Victor Ruiz Perez, Xianrui Cao

February 03, 2026

Contents

Part 1: Global Population and Transportation Analysis

Overview

This section combines global population data from Natural Earth with administrative boundaries to create visualizations of population distribution and accessibility to transportation infrastructure (ports and airports). The analysis identifies patterns in population density across continents and examines average distances from populated places to key transportation hubs.

Data Sources

- **World Boundaries**: Natural Earth 50m administrative boundaries (ne_50m_admin_0_countries)
- **Population Data**: Natural Earth 50m populated places (ne_50m_populated_places) with POP_MAX values
- **Ports**: Natural Earth 10m ports dataset (ne_10m_ports)
- **Airports**: Natural Earth 10m airports dataset (ne_10m_airports)

Methodology

For continent-level distributions, the Antarctica and Seven seas (open ocean) continents are removed for visibility.

Task 1: Map of Total Population by Country

Population totals were computed by aggregating POP_MAX values from populated places at the country level using the ADM0_A3 country code identifier. Countries are visualized using a logarithmic color scale to handle the wide range of population values across nations.

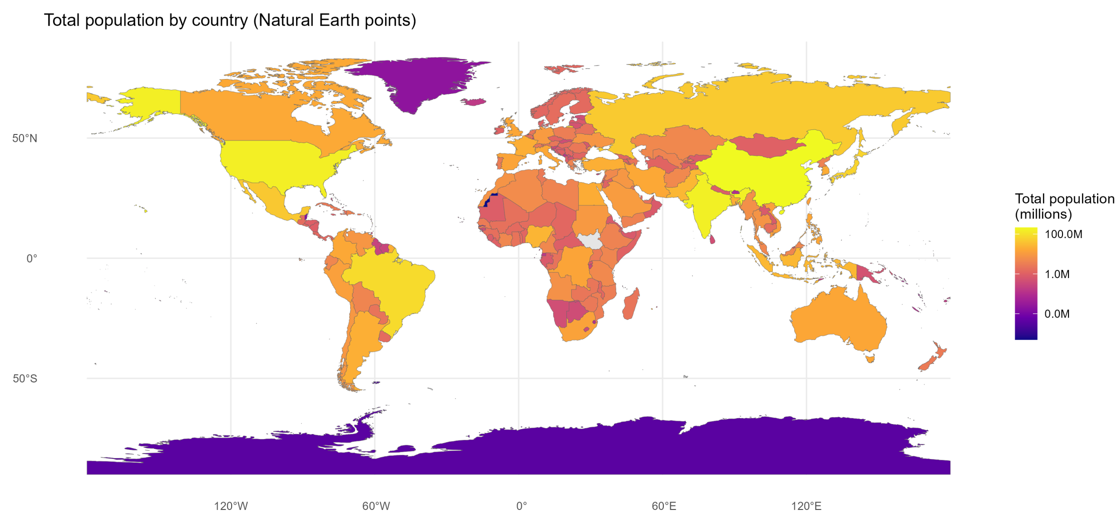

Figure 1: Global population distribution by country. Colors represent log-scaled total population, with darker shades indicating higher populations. Data aggregated from Natural Earth populated places.

Task 2: Country Population Distribution by Continent

The histogram below shows the distribution of country-level total populations across continents. Free y-axis scales reveal that continents differ substantially in the number of countries and their population distributions. Africa and Asia show the greatest diversity in country sizes.

Key Observations:

- **Africa** shows moderate spread with several large population centers (Nigeria, Egypt, Ethiopia)
- **Asia** exhibits the widest range, from small island nations to mega-populated countries (China, India)
- **Europe** is dominated by smaller populations with relatively uniform distribution

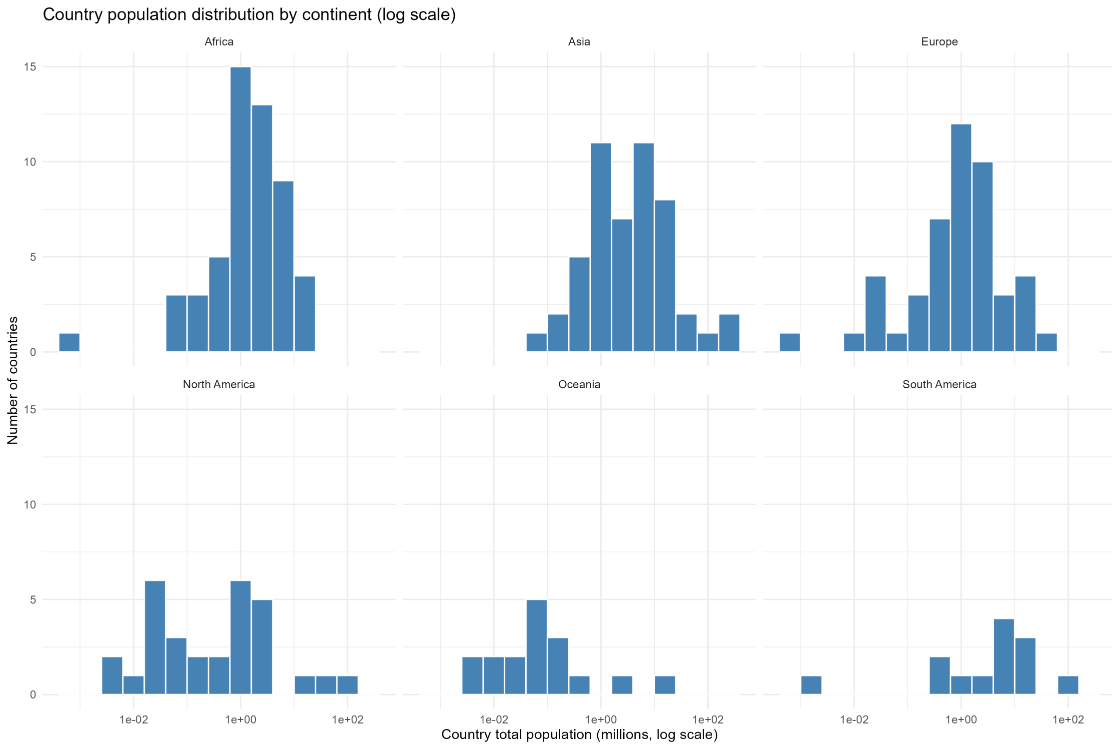

Figure 2: Distribution of country populations by continent. Histograms use free y-axis scales to accommodate differences in country numbers across continents.

- **Americas** show high variability with a few large nations (USA, Brazil, Mexico)
- **Oceania** consists mostly of small-population island nations

Task 3: Average Distance to Airports by Continent

For each populated place, we calculated the minimum distance to the nearest airport from the top 200 airports globally. Country-level averages were then computed and grouped by continent.

Key Findings:

- **Africa**: High average distances (200-1000+ km in some countries), reflecting limited airport infrastructure in inland regions
- **Europe & North America**: Short average distances (<300 km), indicating dense airport coverage
- **Asia**: Highly variable, reflecting mix of developed regions with extensive airport networks and remote areas
- **South America**: Moderate to high distances, particularly in Amazon basin regions

Task 4: Average Distance to Ports by Continent

Similar analysis using the top 200 ports globally. Distances measured from each populated place to the nearest port.

Key Findings:

- **Africa**: Highly variable distances; coastal nations have short distances while interior nations face 500-2000+ km distances

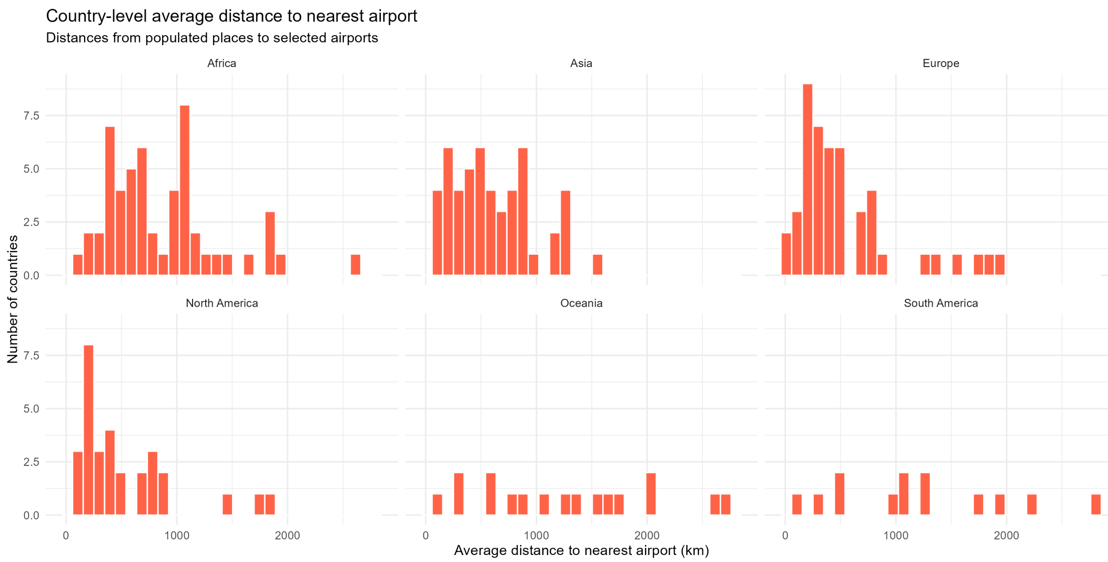

Figure 3: Country-level average distance to nearest airport by continent. Distances calculated from populated places to the 200 largest airports globally.

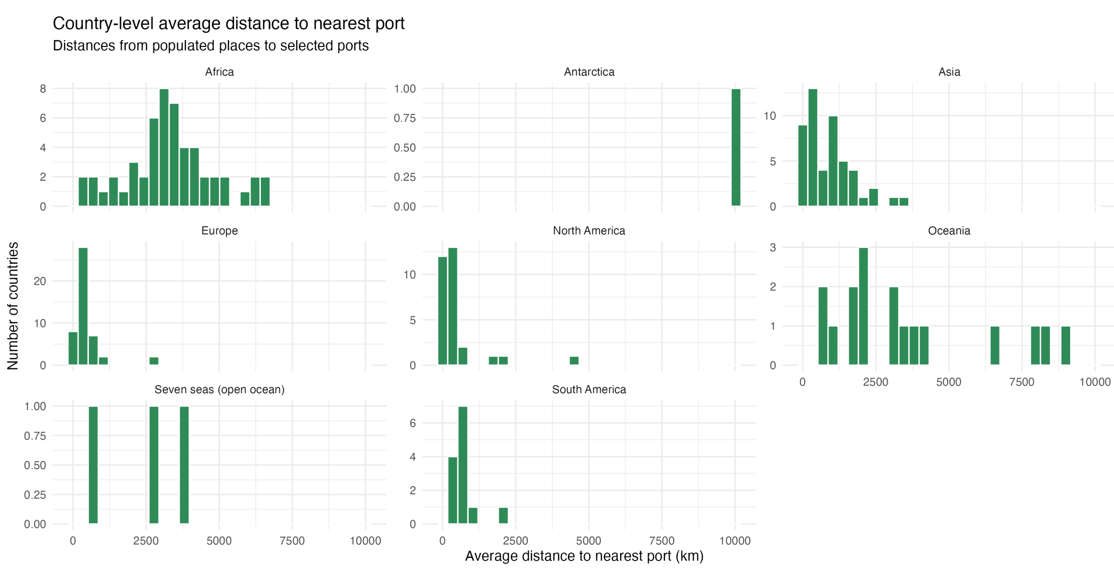

Figure 4: Country-level average distance to nearest port by continent. Distances calculated from populated places to the 200 largest ports globally.

- **Europe**: Consistently short distances (<500 km), reflecting dense coastline and port infrastructure
- **Asia**: Bimodal distribution reflecting coastal vs. inland countries
- **Americas**: Moderate distances except in interior South America

R Code for Part 1

The following code implements the analyses above:

```r
# Market Locations & Natural Earth Analysis - R Script

# --------------------------------------------------------------
# 0. Libraries
# --------------------------------------------------------------
library(sf)
library(readxl)
library(dplyr)
library(ggplot2)
library(rstudioapi)
library(scales)  # for nicer legend / axis labels

# --------------------------------------------------------------
# 1. Paths
# --------------------------------------------------------------

# Directory where this script is located (a2/scripts/)
script_dir  <- dirname(rstudioapi::getActiveDocumentContext()$path)

# Assumes: a2/scripts/, a2/data/, a2/outputs/
data_dir    <- file.path(dirname(script_dir), "data")
output_dir  <- file.path(dirname(script_dir), "outputs")
if (!dir.exists(output_dir)) dir.create(output_dir, recursive = TRUE)

# Your market coordinates file
mkt_path    <- file.path(data_dir, "MktCoords.xlsx")

# Natural Earth shapefiles: folder + .shp filename INSIDE that folder
# Change these four lines if your names differ
world_path    <- file.path(data_dir,
                           "ne_50m_admin_0_countries",
                           "ne_50m_admin_0_countries.shp")
pop_path      <- file.path(data_dir,
                           "ne_50m_populated_places",
                           "ne_50m_populated_places.shp")
ports_path    <- file.path(data_dir,
                           "ne_10m_ports",
                           "ne_10m_ports.shp")
airports_path <- file.path(data_dir,
                           "ne_10m_airports",
                           "ne_10m_airports.shp")

# --------------------------------------------------------------
# 2. Your market data
# --------------------------------------------------------------
```

```
cat("Reading market data...\n")
df <- read_excel(mkt_path)

cat("\nFirst few rows of market data:\n")
print(head(df))

cat("\nCreating spatial points for markets...\n")
gdf <- st_as_sf(df, coords = c("longitude", "latitude"), crs = 4326)

cat("\nSpatial features created (markets):\n")
print(head(gdf))
cat(sprintf("\nTotal markets: %d\n", nrow(gdf)))
cat(sprintf("CRS: %s\n", st_crs(gdf)))

cat("\nGenerating market points plot...\n")
plot_mkt <- ggplot() +
  geom_sf(data = gdf, color = "red", size = 3, alpha = 0.6) +
  geom_sf_text(data = gdf, aes(label = market),
               nudge_x = 0.3, nudge_y = 0.3,
               size = 3, alpha = 0.7, check_overlap = TRUE) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    panel.grid = element_line(color = "gray90")
  ) +
  labs(
    title = "Market Locations Across Africa",
    x = "Longitude",
    y = "Latitude"
  )

mkt_plot_path <- file.path(output_dir, "market_points_map.png")
ggsave(mkt_plot_path, plot_mkt, width = 14, height = 10, dpi = 300)
cat(sprintf("\nMarket plot saved to: %s\n", mkt_plot_path))

# Save markets as shapefile & GeoJSON
mkt_shp_path <- file.path(output_dir, "MktCoords")
st_write(gdf, mkt_shp_path, driver = "ESRI Shapefile",
         delete_layer = TRUE, quiet = TRUE)
cat(sprintf("Spatial features saved as shapefile: %s\n", mkt_shp_path))

mkt_geojson_path <- file.path(output_dir, "MktCoords.geojson")
st_write(gdf, mkt_geojson_path, driver = "GeoJSON",
         delete_layer = TRUE, quiet = TRUE)
cat(sprintf("Spatial features saved as GeoJSON: %s\n", mkt_geojson_path))

# ---------------------------------------------------------------
# 3. Read Natural Earth datasets
# ---------------------------------------------------------------

cat("\nReading Natural Earth data...\n")
```

```
world    <- st_read(world_path, quiet = TRUE)
pop_pts  <- st_read(pop_path, quiet = TRUE)
ports    <- st_read(ports_path, quiet = TRUE)
airports <- st_read(airports_path, quiet = TRUE)

# Common CRS (WGS84 lon/lat)
world    <- st_transform(world, 4326)
pop_pts  <- st_transform(pop_pts, 4326)
ports    <- st_transform(ports, 4326)
airports <- st_transform(airports, 4326)

cat("\nWorld columns:\n");   print(names(world))
cat("\nPopulated places columns:\n"); print(names(pop_pts))

# --------------------------------------------------------------
# 4. Map of total population by country
# --------------------------------------------------------------
# Assumes:
#   world:   ADM0_A3 (country code), CONTINENT
#   pop_pts: ADM0_A3 (country code), POP_MAX (population)

cat("\nComputing total population per country...\n")

country_pop <- pop_pts |>
  st_drop_geometry() |>
  group_by(ADM0_A3) |>
  summarise(
    total_pop = sum(POP_MAX, na.rm = TRUE),
    .groups = "drop"
  )

world_pop <- world |>
  left_join(country_pop, by = "ADM0_A3")

p_total_pop <- ggplot(world_pop) +
  geom_sf(aes(fill = total_pop), color = "grey40", linewidth = 0.1) +
  scale_fill_viridis_c(
    option = "plasma",
    trans  = "log10",
    na.value = "grey90",
    # show legend in millions of people
    labels = ~ label_number(
      accuracy = 0.1,
      suffix   = "M"
    )(.x / 1e6)
  ) +
  theme_minimal() +
  labs(
    title = "Total population by country (Natural Earth points)",
    fill  = "Total population\n(millions)"
  )

map_total_path <- file.path(output_dir, "map_total_population_country.png")
```

```
ggsave(map_total_path, p_total_pop, width = 12, height = 6, dpi = 300)
cat(sprintf("Map of total population saved to: %s\n", map_total_path))

# -------------------------------------------------------------
# 5. Histogram of country population by continent
# -------------------------------------------------------------

cat("\nCreating histogram of country population by continent...\n")

world_pop_df <- world_pop |>
  st_drop_geometry() |>
  filter(!is.na(total_pop), !is.na(CONTINENT)) |>
  filter(!CONTINENT %in% c("Antarctica", "Seven seas (open ocean)")) |>
  select(ADM0_A3, CONTINENT, total_pop)  # Keep country code for debugging

# Use population in millions on x axis (no log scale, easier to read)
p_hist_pop <- ggplot(world_pop_df,
                     aes(x = total_pop / 1e6)) +
  geom_histogram(binwidth = 0.4, fill = "steelblue", color = "white") +
  scale_x_log10() +
  facet_wrap(~ CONTINENT) +  # Remove scales = "free_y"
  theme_minimal() +
  labs(
    title = "Country population distribution by continent (log scale)",
    x = "Country total population (millions, log scale)",
    y = "Number of countries"
  )

hist_pop_path <- file.path(output_dir, "hist_country_population_by_continent.png")
ggsave(hist_pop_path, p_hist_pop, width = 12, height = 8, dpi = 300)
cat(sprintf("Histogram of country population saved to: %s\n", hist_pop_path))

# -------------------------------------------------------------
# 6. Histogram of country-level average distance to airports
# -------------------------------------------------------------

cat("\nComputing distances to nearest airports...\n")

# Take more airports if you want (e.g. first 200 instead of 20)
airports_top <- airports |>
  slice_head(n = 200)

# Project for distance calculation (meters)
crs_dist <- 3857
pop_proj      <- st_transform(pop_pts, crs_dist)
airports_proj <- st_transform(airports_top, crs_dist)
world_proj    <- st_transform(world, crs_dist)

# Nearest airport for each populated place
nearest_idx <- st_nearest_feature(pop_proj, airports_proj)

# Distances in km
dist_km <- st_distance(pop_proj, airports_proj[nearest_idx, ], by_element = TRUE)
```

```r
dist_km <- as.numeric(dist_km) / 1000
pop_proj$dist_km_airport <- dist_km

# Average distance per country
country_dist_airports <- pop_proj |>
  st_drop_geometry() |>
  group_by(ADM0_A3) |>
  summarise(
    avg_dist_km_airport = mean(dist_km_airport, na.rm = TRUE),
    .groups = "drop"
  )

# Add continent info
dist_continent_airports <- world_proj |>
  st_drop_geometry() |>
  select(ADM0_A3, CONTINENT) |>
  left_join(country_dist_airports, by = "ADM0_A3") |>
  filter(!is.na(avg_dist_km_airport), !is.na(CONTINENT)) |>
  filter(!CONTINENT %in% c("Antarctica", "Seven seas (open ocean)"))

p_hist_dist_airports <- ggplot(dist_continent_airports,
                               aes(x = avg_dist_km_airport)) +
  geom_histogram(bins = 30, fill = "tomato", color = "white") +
  facet_wrap(~ CONTINENT) +
  theme_minimal() +
  labs(
    title = "Country-level average distance to nearest airport",
    subtitle = "Distances from populated places to selected airports",
    x = "Average distance to nearest airport (km)",
    y = "Number of countries"
  )

hist_dist_airport_path <- file.path(output_dir, "hist_avg_distance_airport_by_continent.png")
ggsave(hist_dist_airport_path, p_hist_dist_airports, width = 12, height = 6, dpi = 300)
cat(sprintf("Histogram of average distances to airports saved to: %s\n",
            hist_dist_airport_path))

# -------------------------------------------------------------
# 7. Histogram of country-level average distance to ports
# -------------------------------------------------------------

cat("\nComputing distances to nearest ports...\n")

# Use more ports if desired (e.g. first 200)
ports_top <- ports |>
  slice_head(n = 200)

ports_proj <- st_transform(ports_top, crs_dist)

nearest_idx_ports <- st_nearest_feature(pop_proj, ports_proj)

dist_km_ports <- st_distance(pop_proj, ports_proj[nearest_idx_ports, ], by_element = TRUE)
dist_km_ports <- as.numeric(dist_km_ports) / 1000
```

```r
pop_proj$dist_km_port <- dist_km_ports

country_dist_ports <- pop_proj |>
  st_drop_geometry() |>
  group_by(ADM0_A3) |>
  summarise(
    avg_dist_km_port = mean(dist_km_port, na.rm = TRUE),
    .groups = "drop"
  )

dist_continent_ports <- world_proj |>
  st_drop_geometry() |>
  select(ADM0_A3, CONTINENT) |>
  left_join(country_dist_ports, by = "ADM0_A3") |>
  filter(!is.na(avg_dist_km_port), !is.na(CONTINENT))

p_hist_dist_ports <- ggplot(dist_continent_ports,
                            aes(x = avg_dist_km_port)) +
  geom_histogram(bins = 30, fill = "seagreen", color = "white") +
  facet_wrap(~ CONTINENT, scales = "free_y") +
  theme_minimal() +
  labs(
    title = "Country-level average distance to nearest port",
    subtitle = "Distances from populated places to selected ports",
    x = "Average distance to nearest port (km)",
    y = "Number of countries"
  )

hist_dist_ports_path <- file.path(output_dir, "hist_avg_distance_port_by_continent.png")
ggsave(hist_dist_ports_path, p_hist_dist_ports, width = 12, height = 6, dpi = 300)
cat(sprintf("Histogram of average distances to ports saved to: %s\n",
            hist_dist_ports_path))

cat("\nDone!\n")
```

Part 2: African Market Price and Distance Analysis

Overview

This section analyzes commodity market prices across Africa in relation to transportation infrastructure. Following Porteous (2019), we examine how distances to transportation hubs (coast, roads, airports) correlate with market prices. The analysis uses coordinates and price data from the Porteous study, combined with Natural Earth transportation and coastline layers.

Data Sources

- **Market Locations & Prices**: Porteous (2019) study data
  - MktCoords.xlsx: Market locations with mktcode, longitude, latitude

  - PriceMaster4GAMS.xlsx: Price data across crops and time periods

- **Coastline**: Natural Earth coastline data
- **Roads**: Natural Earth roads layer
- **Airports**: Natural Earth airports layer

Reference

Porteous, O., 2019. High trade costs and their consequences: An estimated dynamic model of African agricultural storage and trade. *American Economic Journal: Applied Economics*, 11(4), pp.327-66.

Methodology

Data Preparation

Market coordinates and price data were merged into a single spatial dataset. Average prices were computed across all crops and time periods for each market location. Distance calculations employed a Web Mercator projection (EPSG:3857) for accurate distance measurement.

Task 1: Map of African Market Locations with Prices

Markets are displayed as points, colored by their average price levels. This visualization reveals geographic patterns in commodity prices across the African continent.

Task 2 & 3: Distance Analysis

For each market, we calculated the minimum distance to three transportation infrastructure categories:

1. **Coast**: Minimum distance to any coastline
2. **Roads**: Minimum distance to the nearest road network
3. **Airports**: Minimum distance to the nearest airport

These distances measure market accessibility to regional trade networks and supply chains.

Coast-Price Relationship: Markets closer to the coast show higher average prices, consistent with the hypothesis that coastal access reduces trade costs and increases market integration.

Road-Price Relationship: The relationship between road distance and price is less clear than for coastline, possibly due to variation in road quality and network connectivity. Some distant markets may have better road infrastructure quality.

Airport-Price Relationship: Airport proximity shows weak correlation with prices, suggesting that for bulk agricultural commodities, air transport is less determinant of pricing than maritime or road infrastructure.

R Code for Part 2

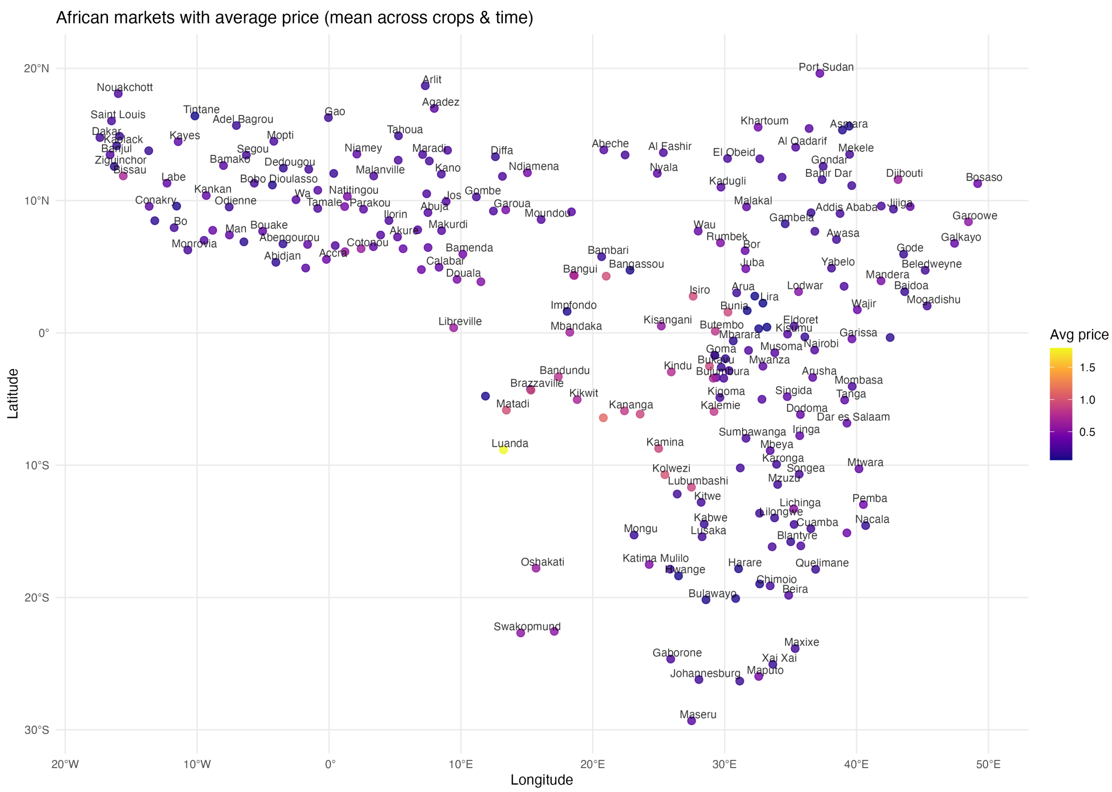

Figure 5: Spatial distribution of African agricultural markets with average commodity prices. Colors indicate the mean price level across crops and time periods. Warmer colors represent higher prices.

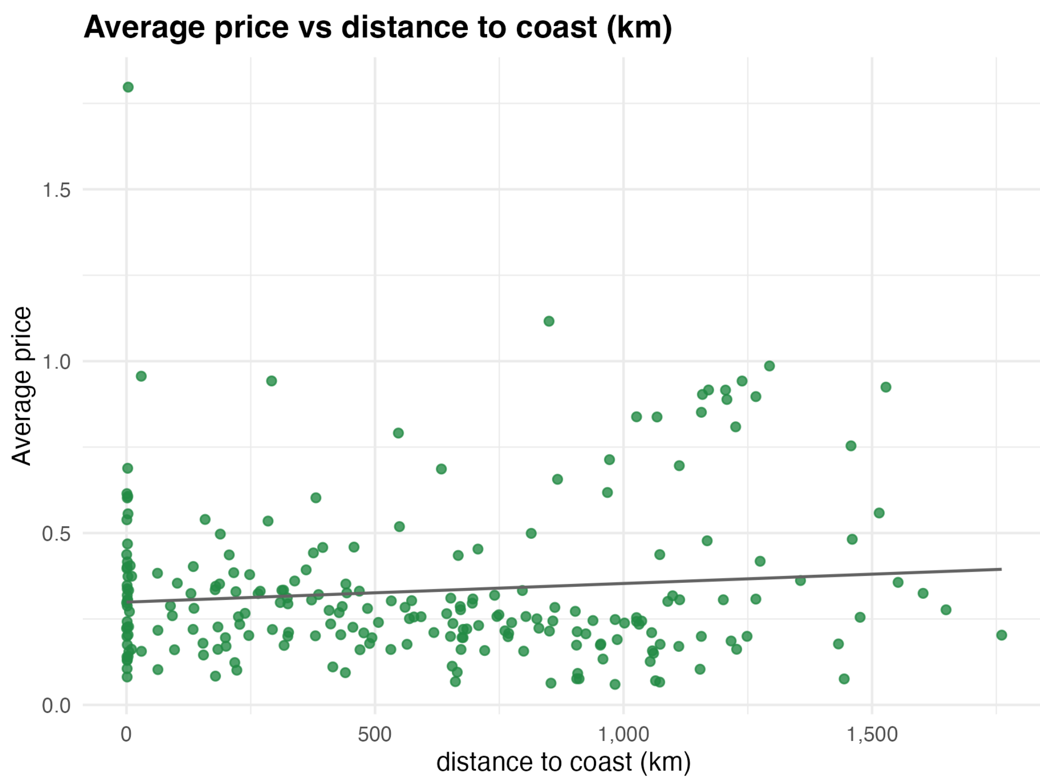

Figure 6: Distance from markets to coastline. Lower values indicate proximity to coastal trade routes.

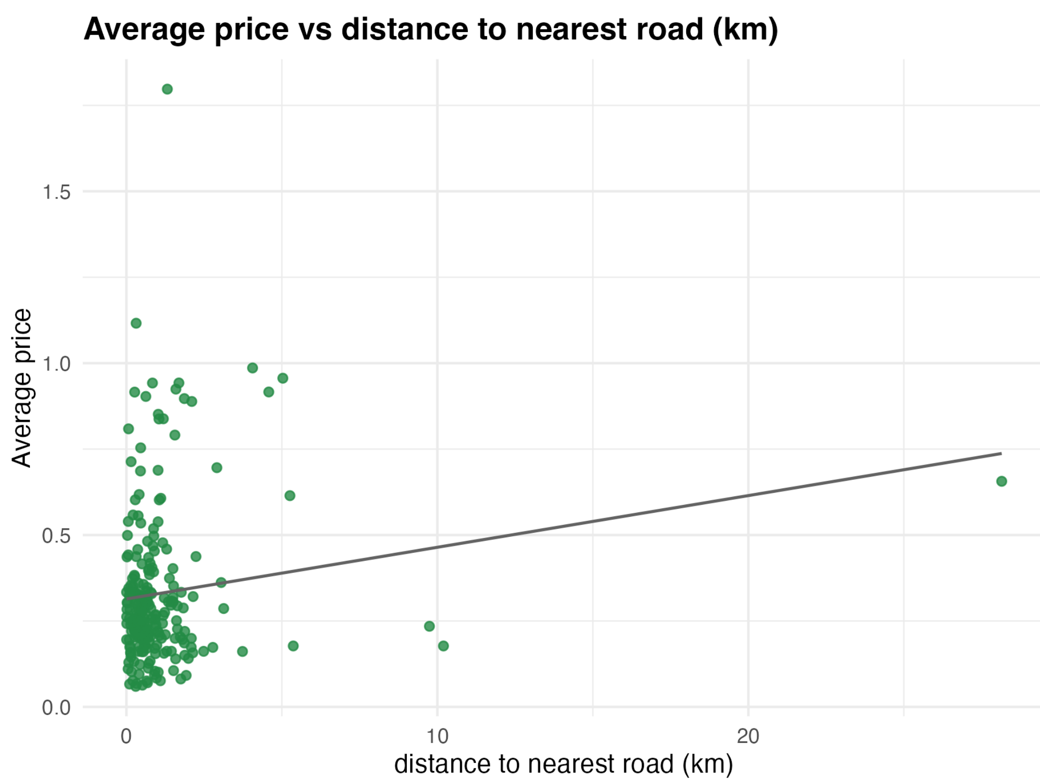

Figure 7: Distance from markets to nearest road. Road access is a key determinant of trade cost variations.

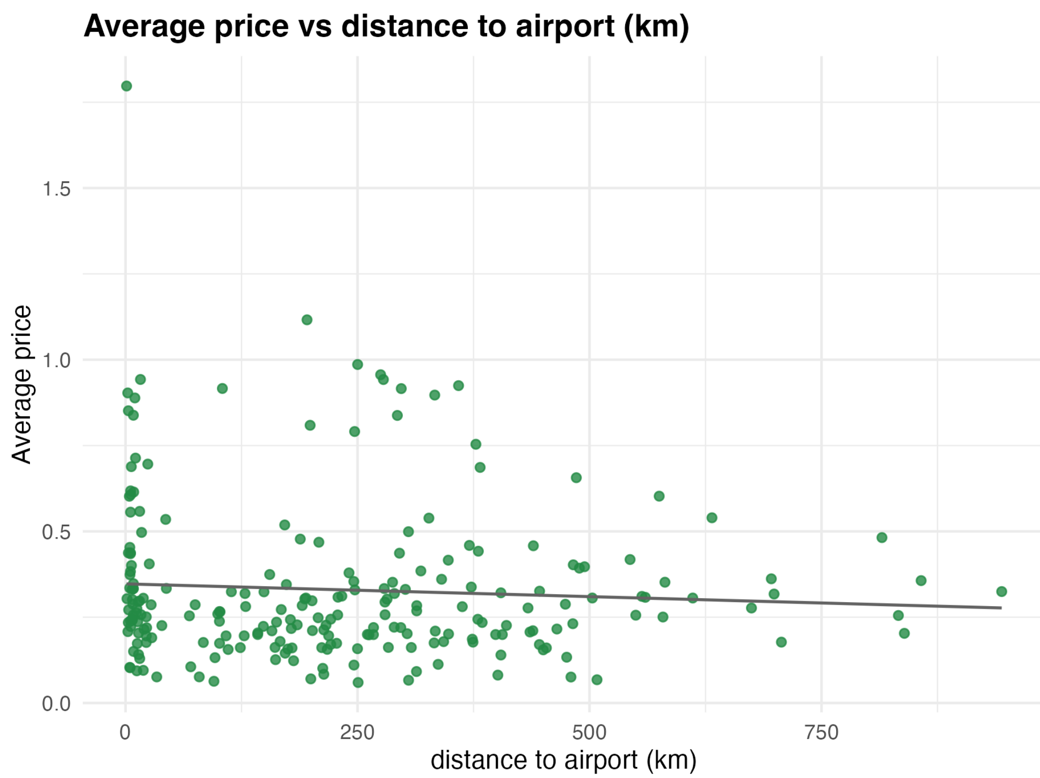

Figure 8: Distance from markets to nearest airport. Aviation access provides alternative trade connectivity.

```
# ---------------------------------------------------------------
# 0. Libraries
# ---------------------------------------------------------------
suppressPackageStartupMessages({
  library(sf)
  library(readxl)
  library(dplyr)
  library(tidyr)
  library(ggplot2)
  library(scales)
  library(rnaturalearth)
  library(rnaturalearthdata)
})

# ---------------------------------------------------------------
# 1. Paths (robust to RStudio / Rscript)
# ---------------------------------------------------------------
script_dir <- tryCatch({
  if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
    dirname(rstudioapi::getActiveDocumentContext()$path)
  } else {
    # Rscript: --file=path/to/script.R
    cmd_args <- commandArgs(trailingOnly = FALSE)
    file_arg <- grep("^--file=", cmd_args, value = TRUE)
    if (length(file_arg) > 0) {
```

```
      script_path <- sub("^--file=", "", file_arg[1])
      if (file.exists(script_path)) dirname(script_path) else getwd()
    } else {
      getwd()
    }
  }
}, error = function(e) getwd())

data_dir <- file.path(dirname(script_dir), "data")
output_dir <- file.path(dirname(script_dir), "outputs")
if (!dir.exists(output_dir)) dir.create(output_dir, recursive = TRUE)

# Helper: load Natural Earth layer (prefer local gpkg/shp, else download, coastline has package fallbac
load_ne <- function(type, category, scale = 10) {
  try_read <- function(path, label) {
    if (!file.exists(path)) return(NULL)
    message(sprintf("Using local %s: %s", label, path))
    tryCatch(st_read(path, quiet = TRUE) |> st_make_valid(),
             error = function(e) {message(sprintf("Read failed for %s: %s", label, e$message)); NULL})
  }

  # Local gpkg / shp (10m then 50m)
  for (sc in c(scale, 50)) {
    gpkg <- file.path(data_dir, sprintf("ne_%dm_%s.gpkg", sc, type))
    obj <- try_read(gpkg, sprintf("%s gpkg (%dm)", type, sc)); if (!is.null(obj)) return(obj)
    shp  <- file.path(data_dir, sprintf("ne_%dm_%s", sc, type), sprintf("ne_%dm_%s.shp", sc, type))
    obj <- try_read(shp, sprintf("%s shp (%dm)", type, sc));  if (!is.null(obj)) return(obj)
  }

  # Download attempt
  for (sc in c(scale, 50)) {
    obj <- tryCatch(
      ne_download(scale = sc, type = type, category = category, returnclass = "sf", destdir = data_dir)
        st_make_valid(),
      error = function(e) {message(sprintf("Could not download %s (%dm): %s", type, sc, e$message)); NUI
    )
    if (!is.null(obj)) return(obj)
  }

  if (type == "coastline") {
    message("Falling back to coastline50 from rnaturalearthdata.")
    return(tryCatch({data("coastline50", package = "rnaturalearthdata", envir = environment()); coastlin
                    error = function(e) NULL))
  }
  NULL
}

# ----------------------------------------------------------------
# 2. Data of PriceMaster
# ----------------------------------------------------------------
coords_path <- file.path(data_dir, "MktCoords.xlsx")
price_path  <- file.path(data_dir, "PriceMaster4GAMS.xlsx")
```

```r
message("Reading market coordinates...")
coords <- read_excel(coords_path)
if (!all(c("mktcode", "longitude", "latitude") %in% names(coords))) {
  stop("MktCoords.xlsx is missing required columns: mktcode, longitude, latitude")
}
markets <- st_as_sf(coords, coords = c("longitude", "latitude"), crs = 4326)

message("Reading price data and computing average price per market...")
price_raw <- read_excel(price_path, sheet = 1)

price_summary <- price_raw |>
  pivot_longer(
    cols = matches("^[0-9]+$"),
    names_to = "period",
    values_to = "price"
  ) |>
  group_by(mktcode) |>
  summarise(avg_price = mean(price, na.rm = TRUE), .groups = "drop")

markets <- markets |>
  left_join(price_summary, by = "mktcode")

message("Downloading reference layers (cached after first run)...")

# 1. Coastline
coast <- load_ne("coastline", "physical")
if (is.null(coast)) {
  coast <- tryCatch({
    data("coastline50", package = "rnaturalearthdata", envir = environment())
    coastline50
  }, error = function(e) NULL)
}
if (is.null(coast)) stop("Coastline layer unavailable. Connect to the internet or place Natural Earth c
message("   Coastline loaded")

# 2. Roads - try multiple scales and sources
message("  Attempting roads download...")
roads <- load_ne("roads", "cultural", scale = 10)
if (is.null(roads)) {
  message("    Scale 10 failed, trying scale 50...")
  roads <- load_ne("roads", "cultural", scale = 50)
}
if (is.null(roads)) {
  message("    Scale 50 failed, trying 'roads_north_america'...")
  roads <- load_ne("roads_north_america", "cultural", scale = 10)
}
if (!is.null(roads)) {
  message("   Roads loaded (", nrow(roads), " features)")
} else {
  message("   Road layer unavailable; dist_road_km will be NA.")
}

# 3. Airports - try multiple scales
```

```r
message("  Attempting airports download...")
airports <- load_ne("airports", "cultural", scale = 10)
if (is.null(airports)) {
  message("    Scale 10 failed, trying alternative approach...")
  # Try downloading manually
  airports <- tryCatch({
    ne_download(scale = 10, type = "airports", category = "cultural",
                returnclass = "sf", destdir = data_dir, load = TRUE)
  }, error = function(e) {
    message("    Error: ", e$message)
    NULL
  })
}
if (!is.null(airports)) {
  airports <- airports |> rename_with(tolower)
  message("   Airports loaded (", nrow(airports), " features)")
} else {
  message("   Airport layer unavailable; dist_airport_km will be NA.")
}

message("Computing distances (km)...")
proj_crs <- 3857
markets_p <- st_transform(markets, proj_crs)
coast_p <- if (!is.null(coast)) st_transform(coast, proj_crs) else NULL
roads_p <- if (!is.null(roads)) st_transform(roads, proj_crs) else NULL
airports_p <- if (!is.null(airports)) st_transform(airports, proj_crs) else NULL

nearest_coast <- if (!is.null(coast_p)) st_nearest_feature(markets_p, coast_p) else NULL
nearest_road  <- if (!is.null(roads_p)) st_nearest_feature(markets_p, roads_p) else NULL
nearest_air   <- if (!is.null(airports_p)) st_nearest_feature(markets_p, airports_p) else NULL

dist_km <- function(target, source, idx) {
  if (is.null(source) || is.null(idx)) return(rep(NA_real_, nrow(target)))
  as.numeric(st_distance(target, source[idx, ], by_element = TRUE)) / 1000
}

markets <- markets |>
  mutate(
    dist_coast_km = dist_km(markets_p, coast_p, nearest_coast),
    dist_road_km = dist_km(markets_p, roads_p, nearest_road),
    dist_airport_km = dist_km(markets_p, airports_p, nearest_air)
  )

message("Saving outputs...")
write.csv(st_drop_geometry(markets), file.path(output_dir, "market_distances_with_price.csv"), row.name
st_write(markets, file.path(output_dir, "MktCoords_with_price_and_dist"), driver = "ESRI Shapefile", de
st_write(markets, file.path(output_dir, "MktCoords_with_price_and_dist.geojson"), driver = "GeoJSON", d

message("Plotting market locations...")
map_plot <- ggplot() +
  geom_sf(data = markets, aes(color = avg_price), size = 2.5, alpha = 0.8) +
  scale_color_viridis_c(option = "C", na.value = "gray70") +
  geom_sf_text(data = markets, aes(label = market), nudge_x = 0.5, nudge_y = 0.5,
```

```r
               size = 3, alpha = 0.8, check_overlap = TRUE) +
  theme_minimal(base_size = 11) +
  labs(
    title = "African markets with average price (mean across crops & time)",
    color = "Avg price",
    x = "Longitude", y = "Latitude"
  )
ggsave(file.path(output_dir, "market_points_map_avg_price.png"), map_plot, width = 12, height = 9, dpi =

message("Plotting price vs distance scatter plots...")
make_scatter <- function(xvar, xlabel, filename) {
  p <- ggplot(markets, aes(.data[[xvar]], avg_price)) +
    geom_point(color = "#238b45", alpha = 0.8) +
    geom_smooth(method = "lm", se = FALSE, color = "#636363", linewidth = 0.6) +
    scale_x_continuous(labels = comma) +
    labs(
      title = paste("Average price vs", xlabel),
      x = xlabel,
      y = "Average price"
    ) +
    theme_minimal(base_size = 11) +
    theme(plot.title = element_text(face = "bold"))
  ggsave(file.path(output_dir, filename), p, width = 6.2, height = 4.6, dpi = 300)
}

run_scatter <- function(xvar, xlabel, filename) {
  if (all(is.na(markets[[xvar]]))) {
    message(sprintf("Skipping scatter for %s (no data).", xvar))
  } else {
    make_scatter(xvar, xlabel, filename)
  }
}

run_scatter("dist_coast_km", "distance to coast (km)", "scatter_price_coast.png")
run_scatter("dist_road_km", "distance to nearest road (km)", "scatter_price_road.png")
run_scatter("dist_airport_km", "distance to airport (km)", "scatter_price_airport.png")

print(map_plot)
message("Done. Outputs saved to: ", output_dir)
```

Appendix: Summary of Output Files

All analysis outputs are saved in the `outputs/` directory:

Maps and Visualizations: - `map_total_population_country.png`: Global population distribution - `hist_country_population_by_continent.png`: Country population histograms - `hist_avg_distance_airport_by_contin` Airport accessibility - `hist_avg_distance_port_by_continent.png`: Port accessibility - `market_points_map.png`: African market locations - `market_points_map_avg_price.png`: Markets colored by average price - `scatter_price_coast.png`: Price vs. coastal distance - `scatter_price_road.png`: Price vs. road distance - `scatter_price_airport.png`: Price vs. airport distance

Data Files: - `MktCoords.shp/geojson`: Market locations (spatial) - `MktCoords_with_price_and_dist.shp/geojson`: Markets with computed distances - `market_distances_with_price.csv`: Tabular market data with all computed variables

---

Report generated on February 03, 2026