Population Density and Housing Price Dynamics

Data Analytics Capstone

Christian McIntire, supported by Dr. Alexandre Scarcioffolo

Contents

library(dplyr)  
data <- mtcars  
summary <- data %>%  
 summarise(  
 avg\_mpg = mean(mpg),  
 avg\_hp = mean(hp)  
 )  
summary

### Rayshader Prerender Testing

# Test Rayshader plot  
library(rayshader)  
  
volcano %>%  
 sphere\_shade(texture = "imhof1") %>%  
 plot\_3d(volcano, zscale = 10, theta = 30, phi = 40)  
  
# Render the plot to view it interactively  
rgl::rglwidget()

### NAOR Dataset Cleaning

library(readr)  
library(readxl)  
library(stringr)  
library(tidyverse)  
  
# Define the file path for the Excel file  
file\_path <- "/Users/cmacbook/Documents/Data Analytics Capstone/naorMedianHomePriceData.xlsx"  
  
# Read the Excel data (assuming data is in the first sheet)  
naor\_housing\_data <- read\_xlsx(file\_path, sheet = 1)  
  
# Clean the data  
naor\_housing\_clean <- naor\_housing\_data %>%  
 # Rename columns for ease of use  
 rename(  
 state = State,  
 county = countyName,  
 median\_home\_price = medianHomePriceQ32024,  
 monthly\_payment\_q32024 = monthlyPaymentQ32024,  
 monthly\_payment\_q32023 = monthlyPaymentQ32023  
 ) %>%  
 # Standardize text and clean numeric columns  
 mutate(  
 state = str\_trim(tolower(state)),  
 county = str\_trim(tolower(county)),  
 median\_home\_price = str\_replace\_all(median\_home\_price, "[$,]", "") %>% as.numeric(),  
 monthly\_payment\_q32024 = str\_replace\_all(monthly\_payment\_q32024, "[$,]", "") %>% as.numeric(),  
 monthly\_payment\_q32023 = str\_replace\_all(monthly\_payment\_q32023, "[$,]", "") %>% as.numeric()  
 ) %>%  
 # Ensure each county-state pair is unique  
 distinct(county, state, .keep\_all = TRUE)  
  
# Display a summary of the cleaned data  
# view(naor\_housing\_clean)

### Shapefiles for Choropleths - United States Census Bureau

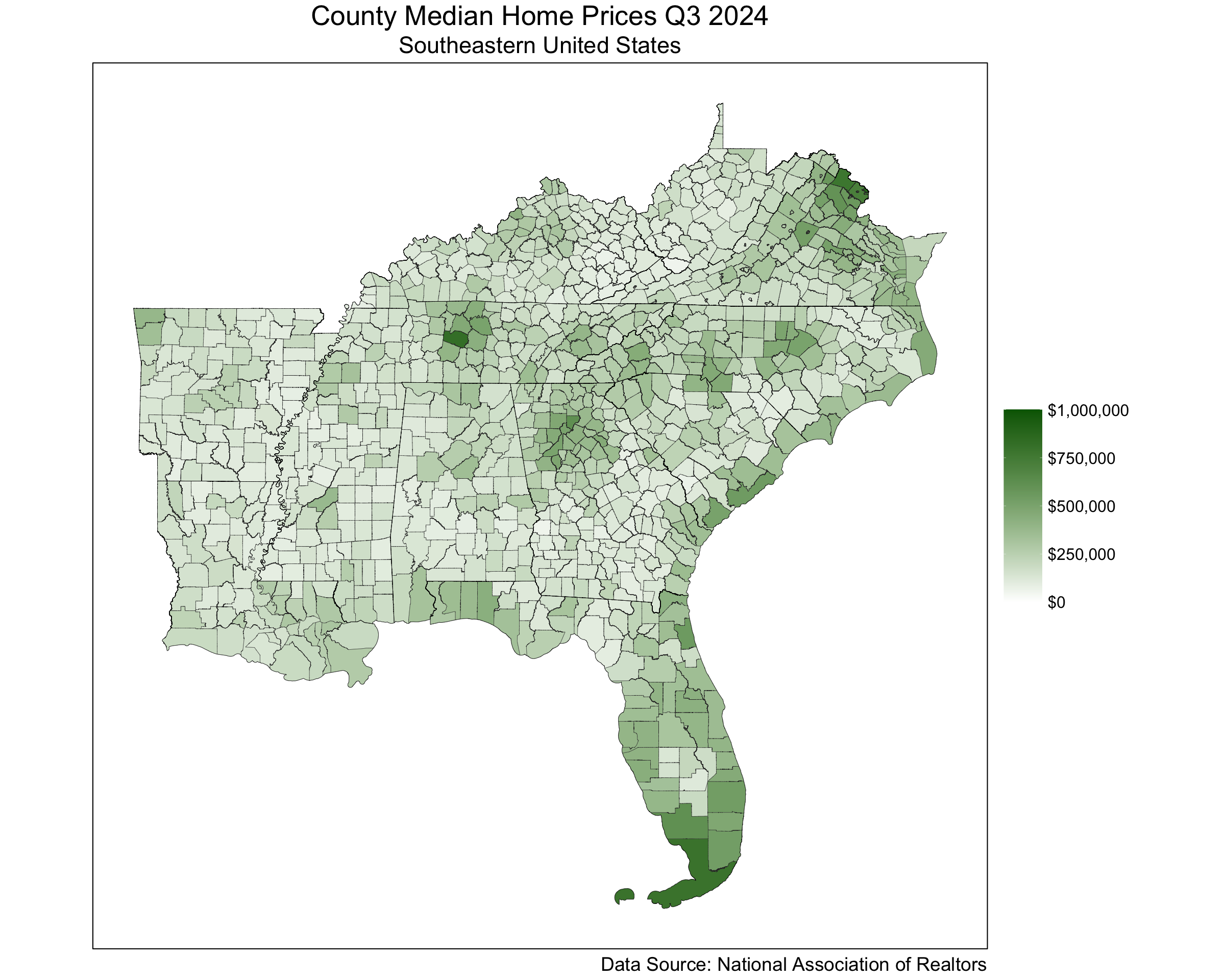
library(sf)  
  
# Define file paths for county and state shapefiles  
county\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_county/tl\_2024\_us\_county.shp"  
state\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_state/tl\_2024\_us\_state.shp"  
  
# Load the shapefiles  
county\_shapes <- st\_read(county\_shapefile, quiet = TRUE)  
state\_shapes <- st\_read(state\_shapefile, quiet = TRUE)

### Southeastern United States Choropleth

library(tidyverse)  
library(sf)  
library(scales)  
library(stringr)  
library(readr)  
  
# --- 1. Use the cleaned dataset (already in memory) ---  
# naor\_housing\_clean has columns: state, county, median\_home\_price, monthly\_payment\_q32024, monthly\_payment\_q32023  
  
# Define the Southeastern states you want to include  
southeast\_states <- c(  
 "alabama", "arkansas", "louisiana", "kentucky", "tennessee",  
 "mississippi", "georgia", "florida", "north carolina",  
 "south carolina", "virginia", "west virginia"  
)  
  
# Filter your housing data for Southeastern states only  
southeast\_data <- naor\_housing\_clean %>%  
 filter(state %in% southeast\_states)  
  
# --- 2. Load Shapefiles ---  
county\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_county/tl\_2024\_us\_county.shp"  
state\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_state/tl\_2024\_us\_state.shp"  
  
# Read county shapefile  
county\_geo <- st\_read(county\_shapefile, quiet = TRUE) %>%  
 mutate(  
 # Convert county name to lowercase, trimmed  
 county = str\_trim(tolower(NAME)))  
  
# Read state shapefile  
state\_geo <- st\_read(state\_shapefile, quiet = TRUE) %>%  
 mutate(  
 # Convert state name to lowercase, trimmed  
 state = str\_trim(tolower(NAME))) %>%  
 select(STATEFP, state)  
  
# Convert state\_geo to a regular data frame by dropping geometry  
state\_geo\_df <- state\_geo %>%   
 st\_drop\_geometry()  
  
# Now county\_geo is still an sf object, but state\_geo\_df is just a data frame  
county\_geo <- county\_geo %>%  
 left\_join(state\_geo\_df, by = "STATEFP") %>%  
 mutate(state = str\_trim(tolower(state)))  
  
# Filter county\_geo for Southeastern states  
county\_geo <- county\_geo %>%  
 filter(state %in% southeast\_states)  
  
# --- 3. Merge Shapefile with Housing Data ---  
southeast\_geo <- county\_geo %>%  
 left\_join(southeast\_data, by = c("county", "state")) %>%  
 mutate(  
 # Fill missing prices with the median of existing prices  
 median\_home\_price = ifelse(  
 is.na(median\_home\_price),  
 median(median\_home\_price, na.rm = TRUE),  
 median\_home\_price  
 ),  
 price = median\_home\_price # We'll map 'price' in ggplot  
 )

naor\_housing\_clean <- naor\_housing\_clean %>%  
 mutate(  
 # Remove " county" at the end of the string  
 county = str\_replace(county, "\\s+county$", ""),  
 # Remove " parish" at the end of the string (common in Louisiana)  
 county = str\_replace(county, "\\s+parish$", ""),  
 # Remove " city" if needed (common in Virginia independent cities)  
 county = str\_replace(county, "\\s+city$", "")  
 )  
  
southeast\_geo <- county\_geo %>%  
 left\_join(naor\_housing\_clean, by = c("county", "state")) %>%  
 mutate(  
 median\_home\_price = ifelse(  
 is.na(median\_home\_price),  
 median(median\_home\_price, na.rm = TRUE),  
 median\_home\_price  
 ),  
 price = median\_home\_price  
 )  
  
naor\_housing\_clean <- naor\_housing\_clean %>%  
 group\_by(county, state) %>%  
 summarise(  
 median\_home\_price = median(median\_home\_price, na.rm = TRUE),  
 .groups = "drop"  
 )

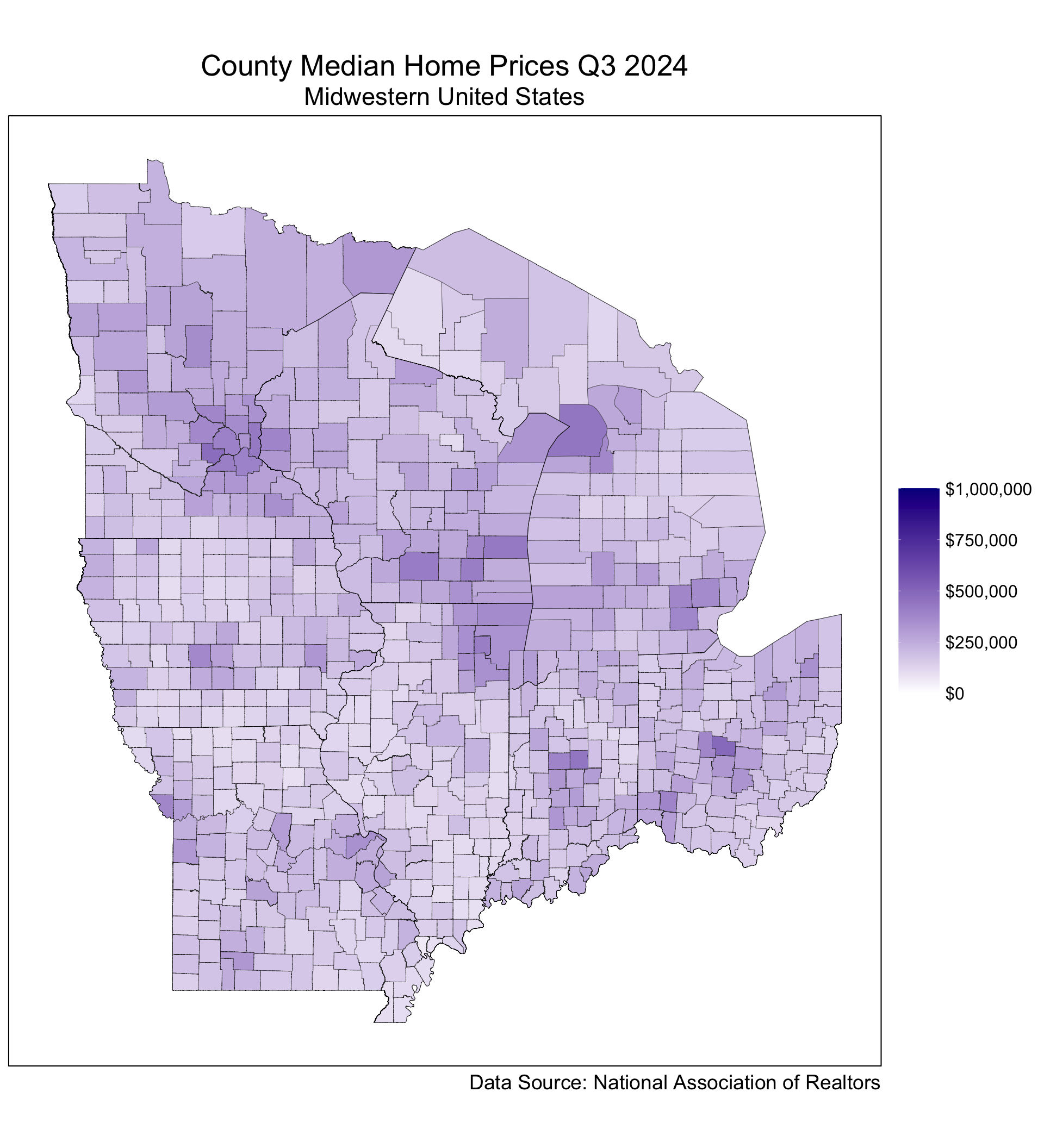
library(tidyverse)  
library(sf)  
library(scales)  
library(stringr)  
library(readr)  
library(grid)  
  
# 1. Define Southeastern states  
southeast\_states <- c(  
 "alabama", "arkansas", "louisiana", "kentucky", "tennessee",  
 "mississippi", "georgia", "florida", "north carolina",  
 "south carolina", "virginia", "west virginia"  
)  
  
# 2. Read state shapefile for Southeastern borders  
state\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_state/tl\_2024\_us\_state.shp"  
state\_geo\_southeast <- st\_read(state\_shapefile, quiet = TRUE) %>%  
 mutate(state = str\_trim(tolower(NAME))) %>%  
 filter(state %in% southeast\_states)  
  
# 3. Build the Southeastern Choropleth  
ggSoutheastPrices <- ggplot() +  
 # Counties with grey30 borders  
 geom\_sf(  
 data = southeast\_geo,  
 aes(fill = price),  
 color = "grey20",  
 size = 0.05  
 ) +  
 # State borders on top (thicker black lines)  
 geom\_sf(  
 data = state\_geo\_southeast,  
 fill = NA,  
 color = "black",  
 size = 30  
 ) +  
 scale\_fill\_gradient(  
 name = NULL,   
 low = "white",  
 high = "darkgreen",  
 na.value = "grey",  
 limits = c(0, 1000000),   
 labels = scales::label\_dollar()  
 ) +  
 labs(  
 title = "County Median Home Prices Q3 2024",  
 subtitle = "Southeastern United States",  
 caption = "Data Source: National Association of Realtors"  
 ) +  
 guides(fill = guide\_colorbar(  
 title.theme = element\_text(size = 28, hjust = 0.5, margin = margin(b = 10)),  
 label.theme = element\_text(size = 24),  
 barheight = unit(10, "cm"),  
 barwidth = unit(2, "cm"),  
 title.position = "top"  
 )) +  
 theme\_linedraw(base\_size = 16) +  
 theme(  
 plot.title = element\_text(size = 40, hjust = 0.5),  
 plot.subtitle = element\_text(size = 34, hjust = 0.5),  
 plot.caption = element\_text(size = 28),  
 legend.position = "right",  
 axis.text = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.title = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank()  
 ) +  
 coord\_sf(expand = TRUE, clip = "off")  
  
print(ggSoutheastPrices)



### Midwestern United States Choropleth

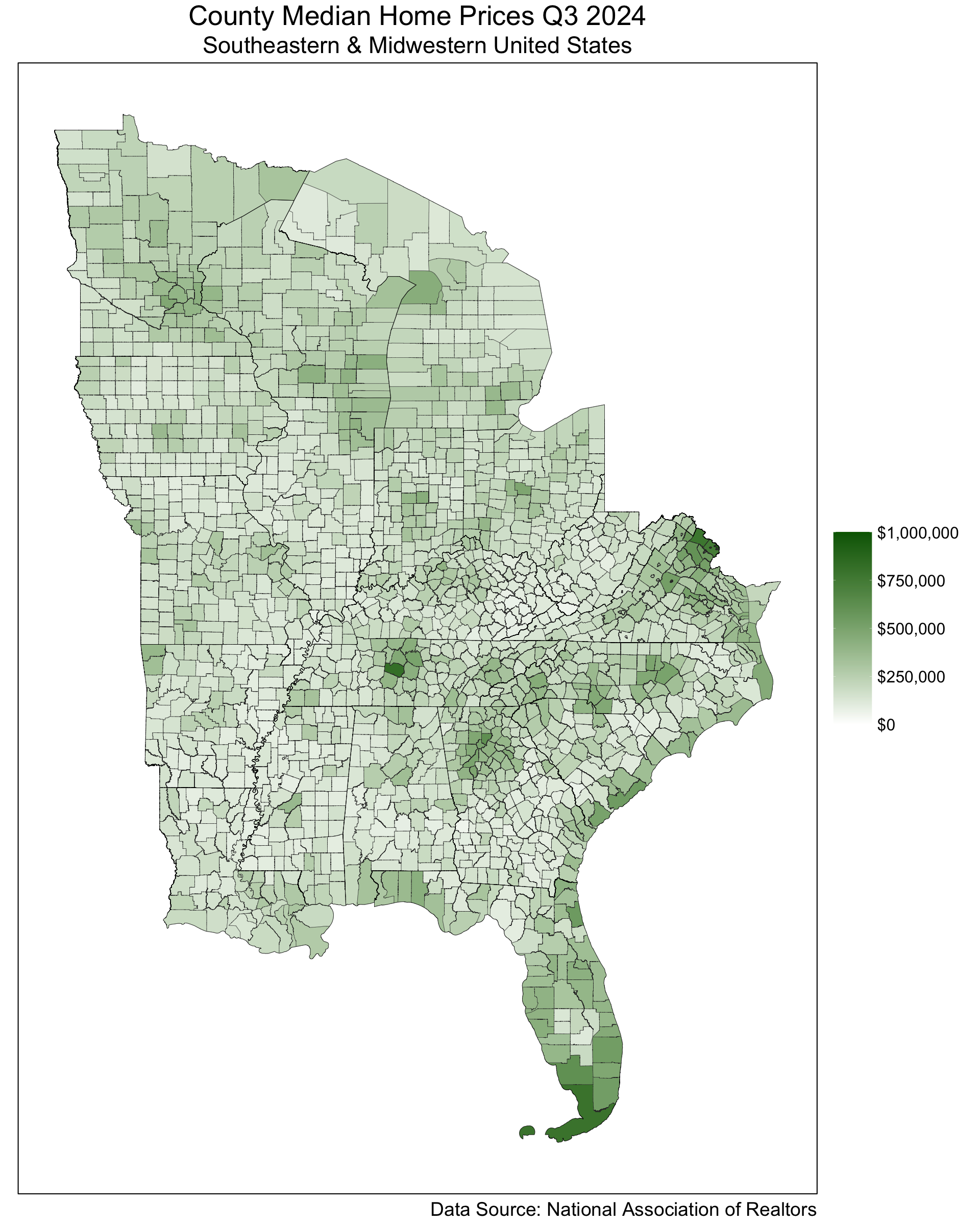
library(tidyverse)  
library(sf)  
library(scales)  
library(stringr)  
library(readr)  
library(grid) # For unit() usage in legend guides  
  
# 1. Define the Midwestern states you want to include  
midwest\_states <- c("minnesota", "wisconsin", "michigan",   
 "ohio", "indiana", "illinois",   
 "iowa", "missouri")  
  
# 2. Filter your housing data for Midwestern states only  
midwest\_data <- naor\_housing\_clean %>%  
 filter(state %in% midwest\_states)  
  
# 3. Read the county shapefile  
county\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_county/tl\_2024\_us\_county.shp"  
county\_geo\_all <- st\_read(county\_shapefile, quiet = TRUE) %>%  
 mutate(county = str\_trim(tolower(NAME)))  
  
# 4. Read the state shapefile  
state\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_state/tl\_2024\_us\_state.shp"  
state\_geo\_all <- st\_read(state\_shapefile, quiet = TRUE) %>%  
 mutate(state = str\_trim(tolower(NAME))) %>%  
 select(STATEFP, state)  
  
# 5. Convert state\_geo\_all to a regular data frame by dropping geometry  
state\_geo\_df\_all <- state\_geo\_all %>%  
 st\_drop\_geometry()  
  
# 6. Attach state names to the county data and filter for Midwestern states  
county\_geo\_all <- county\_geo\_all %>%  
 left\_join(state\_geo\_df\_all, by = "STATEFP") %>%  
 mutate(state = str\_trim(tolower(state))) %>%  
 filter(state %in% midwest\_states)  
  
# 7. Merge the county shapefile with your housing data  
midwest\_geo <- county\_geo\_all %>%  
 left\_join(midwest\_data, by = c("county", "state")) %>%  
 mutate(  
 median\_home\_price = ifelse(  
 is.na(median\_home\_price),  
 median(median\_home\_price, na.rm = TRUE),  
 median\_home\_price  
 ),  
 price = median\_home\_price  
 )  
  
# 8. Create a separate sf object for the state borders (filtered to midwest states)  
state\_geo\_midwest <- st\_read(state\_shapefile, quiet = TRUE) %>%  
 mutate(  
 state = str\_trim(tolower(NAME))  
 ) %>%  
 filter(state %in% midwest\_states)

library(tidyverse)  
library(sf)  
library(scales)  
library(stringr)  
library(readr)  
library(grid)  
  
# Assume that 'midwest\_geo' (the county polygons with housing data for midwestern states)  
# and 'state\_geo\_midwest' (the state shapefile filtered to midwestern states)   
# have already been created.  
  
ggMidwestPrices <- ggplot() +  
 # Plot county geometries with thin borders (black)  
 geom\_sf(  
 data = midwest\_geo,   
 aes(fill = price),   
 color = "grey20",   
 size = 0.1  
 ) +  
 # Overlay state borders using the midwestern state shapefile  
 geom\_sf(  
 data = state\_geo\_midwest,  
 fill = NA,  
 color = "black",  
 size = 3,  
 inherit.aes = FALSE  
 ) +  
 scale\_fill\_gradient(  
 name = NULL, # Remove legend title  
 low = "white",  
 high = "darkblue",  
 na.value = "grey",  
 limits = c(0, 1000000), # Fixed scale: 0 to $1,000,000  
 labels = scales::label\_dollar()  
 ) +  
 labs(  
 title = "County Median Home Prices Q3 2024",  
 subtitle = "Midwestern United States",  
 caption = "Data Source: National Association of Realtors"  
 ) +  
 guides(fill = guide\_colorbar(  
 title.theme = element\_text(size = 28, hjust = 0.5, margin = margin(b = 10)),  
 label.theme = element\_text(size = 24),  
 barheight = unit(10, "cm"),  
 barwidth = unit(2, "cm"),  
 title.position = "top"  
 )) +  
 theme\_linedraw(base\_size = 16) +  
 theme(  
 plot.title = element\_text(size = 40, hjust = 0.5),  
 plot.subtitle = element\_text(size = 34, hjust = 0.5),  
 plot.caption = element\_text(size = 28),  
 legend.position = "right",  
 axis.text = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.title = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank()  
 ) +  
 coord\_sf(expand = TRUE, clip = "off")  
  
print(ggMidwestPrices)



### Southeastern and Midwestern United States Choropleth

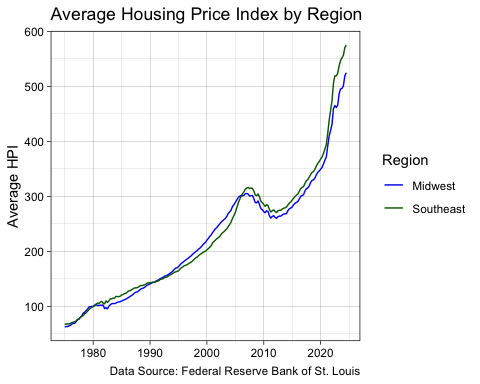
library(tidyverse)  
library(sf)  
library(scales)  
library(stringr)  
library(readr)  
library(grid)  
  
# 1. Define Southeastern and Midwestern states  
selected\_states <- c(  
 # Southeastern  
 "arkansas", "louisiana", "kentucky", "tennessee", "mississippi",  
 "alabama", "georgia", "florida", "north carolina", "south carolina",  
 "virginia", "west virginia",  
 # Midwestern  
 "minnesota", "wisconsin", "michigan", "ohio", "indiana",  
 "illinois", "iowa", "missouri"  
)  
  
# 2. Assume 'naor\_housing\_clean' is your cleaned dataset with columns:  
# state (lowercase), county (lowercase), median\_home\_price (numeric), etc.  
# We'll rename or create a "price" column for consistency.  
all\_housing\_data <- naor\_housing\_clean %>%  
 mutate(price = median\_home\_price) # If not already present  
  
# 3. Read the nationwide county shapefile  
county\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_county/tl\_2024\_us\_county.shp"  
county\_geo\_all <- st\_read(county\_shapefile, quiet = TRUE) %>%  
 mutate(county = str\_trim(tolower(NAME))) # Standardize county name  
  
# 4. Read the nationwide state shapefile  
state\_shapefile <- "/Users/cmacbook/Documents/Data Analytics Capstone/ShapeFiles/tl\_2024\_us\_state/tl\_2024\_us\_state.shp"  
state\_geo\_all <- st\_read(state\_shapefile, quiet = TRUE) %>%  
 mutate(state = str\_trim(tolower(NAME))) %>%  
 select(STATEFP, state)  
  
# 5. Attach state names to county data via a non-spatial join  
state\_geo\_df\_all <- state\_geo\_all %>% st\_drop\_geometry()  
county\_geo\_all <- county\_geo\_all %>%  
 left\_join(state\_geo\_df\_all, by = "STATEFP") %>%  
 mutate(state = str\_trim(tolower(state)))  
  
# 6. Filter counties to just the selected Southeastern & Midwestern states  
county\_geo\_sel <- county\_geo\_all %>%  
 filter(state %in% selected\_states)  
  
# 7. Merge with housing data  
sel\_geo <- county\_geo\_sel %>%  
 left\_join(all\_housing\_data, by = c("county", "state")) %>%  
 mutate(  
 # Fill missing prices with overall median if desired  
 price = ifelse(is.na(price), median(price, na.rm = TRUE), price)  
 )  
  
# 8. Filter the state shapefile to the same states for borders  
state\_geo\_sel <- st\_read(state\_shapefile, quiet = TRUE) %>%  
 mutate(state = str\_trim(tolower(NAME))) %>%  
 filter(state %in% selected\_states)  
  
# 9. Plot the combined SE & MW states  
ggSEMWPrices <- ggplot() +  
 # Counties  
 geom\_sf(  
 data = sel\_geo,  
 aes(fill = price),  
 color = "grey20", # County borders  
 size = 0.1  
 ) +  
 # State borders  
 geom\_sf(  
 data = state\_geo\_sel,  
 fill = NA,  
 color = "black",  
 size = 2  
 ) +  
 scale\_fill\_gradient(  
 name = NULL, # No legend title  
 low = "white",  
 high = "darkgreen",  
 na.value = "grey",  
 limits = c(0, 1000000), # 0 to $1,000,000  
 labels = scales::label\_dollar()  
 ) +  
 labs(  
 title = "County Median Home Prices Q3 2024",  
 subtitle = "Southeastern & Midwestern United States",  
 caption = "Data Source: National Association of Realtors"  
 ) +  
 guides(fill = guide\_colorbar(  
 label.theme = element\_text(size = 24),  
 barheight = unit(10, "cm"),  
 barwidth = unit(2, "cm"),  
 title.position = "top"  
 )) +  
 theme\_linedraw(base\_size = 16) +  
 theme(  
 plot.title = element\_text(size = 40, hjust = 0.5),  
 plot.subtitle = element\_text(size = 34, hjust = 0.5),  
 plot.caption = element\_text(size = 28),  
 legend.position = "right",  
 axis.text = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.title = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank()  
 ) +  
 coord\_sf(expand = TRUE, clip = "off")  
  
print(ggSEMWPrices)



### Southeastern and Midwestern United States

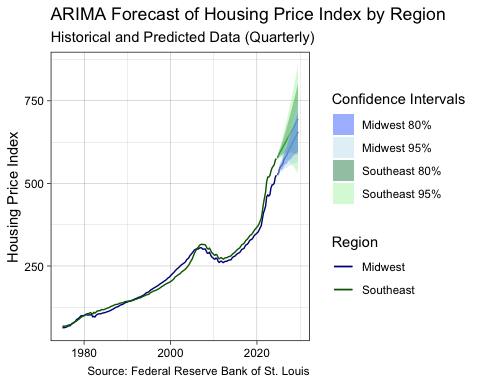
### Historical HPI Comparison

library(tidyverse)  
library(lubridate)  
library(readxl)  
library(readr)  
  
# Set the file path and read the Excel data from sheet 2.  
fredHPIData\_file\_path <- "/Users/cmacbook/Documents/Data Analytics Capstone/fredHPIData.xlsx"  
fredHPIData <- read\_xlsx(fredHPIData\_file\_path, sheet = 2)  
  
# Check column names  
# print(names(fredHPIData))  
# Rename the date column (adjust if needed)  
fredHPIData <- fredHPIData %>%  
 rename(date = observation\_date)  
  
# Pivot from wide to long format, selecting all columns that end with "STHPI"  
fredHPIData\_long <- fredHPIData %>%  
 pivot\_longer(  
 cols = ends\_with("STHPI"),  
 names\_to = "state\_abbr",  
 values\_to = "hpi"  
 ) %>%  
 # Remove the "STHPI" suffix so that state\_abbr contains just the abbreviation (e.g., "AL")  
 mutate(state\_abbr = str\_remove(state\_abbr, "STHPI"))  
  
# Define state abbreviations for each region.  
southeast\_abbr <- c("AR", "LA", "KY", "TN", "MS", "AL", "GA", "FL", "NC", "SC", "VA", "WV")  
midwest\_abbr <- c("MN", "WI", "MI", "OH", "IN", "IL", "IA", "MO")  
  
# Ensure state abbreviations are uppercase and assign a region.  
fredHPIData\_long <- fredHPIData\_long %>%  
 mutate(  
 state\_abbr = toupper(state\_abbr),  
 region = case\_when(  
 state\_abbr %in% southeast\_abbr ~ "Southeast",  
 state\_abbr %in% midwest\_abbr ~ "Midwest",  
 TRUE ~ NA\_character\_  
 )  
 ) %>%  
 filter(!is.na(region)) # Keep only states in our defined regions  
  
# Group the data by region and date to calculate the average HPI for each region.  
fredHPIData\_regional <- fredHPIData\_long %>%  
 group\_by(region, date) %>%  
 summarise(avg\_hpi = mean(hpi, na.rm = TRUE), .groups = "drop")  
  
# Inspect the aggregated time series data  
# print(fredHPIData\_regional)  
  
ggplot(fredHPIData\_regional, aes(x = date, y = avg\_hpi, color = region)) +  
 geom\_line() +  
 scale\_color\_manual(values = c("Midwest" = "blue2", "Southeast" = "darkgreen")) +  
 labs(title = "Average Housing Price Index by Region",  
 caption = "Data Source: Federal Reserve Bank of St. Louis",  
 color = "Region",  
 x = NULL,  
 y = "Average HPI") +  
 theme\_linedraw()



### Southeastern and Midwestern United States ARIMA Forecasting

library(forecast)  
library(lubridate)  
library(ggplot2)  
  
# Suppose you have regional aggregated data in a data frame called 'fredHPIData\_regional'  
# with columns: region, date, and avg\_hpi  
  
# Split the data by region:  
southeast\_ts <- fredHPIData\_regional %>%  
 filter(region == "Southeast") %>%  
 arrange(date)  
  
midwest\_ts <- fredHPIData\_regional %>%  
 filter(region == "Midwest") %>%  
 arrange(date)  
  
# Create time series objects (adjust the start and frequency as needed)  
# Here we assume quarterly data starting in 1975  
southeast\_ts\_obj <- ts(southeast\_ts$avg\_hpi, start = c(year(min(southeast\_ts$date)), quarter(min(southeast\_ts$date))), frequency = 4)  
midwest\_ts\_obj <- ts(midwest\_ts$avg\_hpi, start = c(year(min(midwest\_ts$date)), quarter(min(midwest\_ts$date))), frequency = 4)  
  
# Fit ARIMA models  
arima\_model\_se <- auto.arima(southeast\_ts\_obj)  
arima\_model\_mw <- auto.arima(midwest\_ts\_obj)  
  
# Forecast for the same horizon (for example, forecast for 20 quarters)  
horizon <- 20  
forecast\_se <- forecast(arima\_model\_se, h = horizon, level = c(80, 95))  
forecast\_mw <- forecast(arima\_model\_mw, h = horizon, level = c(80, 95))  
  
# Convert forecasts to data frames and add a region label and dates:  
forecast\_se\_df <- as.data.frame(forecast\_se) %>%  
 mutate(Date = seq(from = max(southeast\_ts$date) + months(3),  
 by = "3 months", length.out = nrow(.)),  
 Region = "Southeast")  
  
forecast\_mw\_df <- as.data.frame(forecast\_mw) %>%  
 mutate(Date = seq(from = max(midwest\_ts$date) + months(3),  
 by = "3 months", length.out = nrow(.)),  
 Region = "Midwest")  
  
# Combine the forecast data  
forecast\_combined <- bind\_rows(forecast\_se\_df, forecast\_mw\_df)  
  
# Also combine historical data for plotting  
historical\_combined <- fredHPIData\_regional %>%  
 rename(HPI = avg\_hpi)  
  
# Plot with ggplot2  
ggplot() +  
 # Historical data  
 geom\_line(data = historical\_combined, aes(x = date, y = HPI, color = region), size = 0.5) +  
 # Forecast lines  
 geom\_line(data = forecast\_combined, aes(x = Date, y = `Point Forecast`, color = Region), size = .5) +  
 # Confidence intervals for Southeast  
 geom\_ribbon(data = forecast\_se\_df, aes(x = Date, ymin = `Lo 80`, ymax = `Hi 80`, fill = "Southeast 80%"), alpha = 0.5) +  
 geom\_ribbon(data = forecast\_se\_df, aes(x = Date, ymin = `Lo 95`, ymax = `Hi 95`, fill = "Southeast 95%"), alpha = 0.35) +  
 # Confidence intervals for Midwest  
 geom\_ribbon(data = forecast\_mw\_df, aes(x = Date, ymin = `Lo 80`, ymax = `Hi 80`, fill = "Midwest 80%"), alpha = 0.5) +  
 geom\_ribbon(data = forecast\_mw\_df, aes(x = Date, ymin = `Lo 95`, ymax = `Hi 95`, fill = "Midwest 95%"), alpha = 0.35) +  
 scale\_color\_manual(values = c("Southeast" = "darkgreen", "Midwest" = "darkblue")) +  
 scale\_fill\_manual(values = c("Southeast 80%" = "seagreen", "Southeast 95%" = "lightgreen",   
 "Midwest 80%" = "royalblue1", "Midwest 95%" = "lightblue")) +  
 labs(title = "ARIMA Forecast of Housing Price Index by Region",  
 subtitle = "Historical and Predicted Data (Quarterly)",  
 x = NULL,  
 y = "Housing Price Index",  
 caption = "Source: Federal Reserve Bank of St. Louis",  
 color = "Region",  
 fill = "Confidence Intervals") +  
 theme\_linedraw()



library(tidyverse)  
library(lubridate)  
library(forecast)  
library(readxl)  
library(gt)  
  
# Load the Housing Price Index (HPI) dataset  
fredHPIData\_file\_path <- "/Users/cmacbook/Documents/Data Analytics Capstone/fredHPIData.xlsx"  
fredHPIData <- read\_xlsx(fredHPIData\_file\_path, sheet = 2)  
  
# Rename the date column if needed  
fredHPIData <- fredHPIData %>%  
 rename(date = observation\_date) %>%  
 mutate(date = as.Date(date)) # Ensure it's in Date format  
  
# Convert dataset from wide to long format  
fredHPIData\_long <- fredHPIData %>%  
 pivot\_longer(  
 cols = ends\_with("STHPI"),  
 names\_to = "state\_abbr",  
 values\_to = "hpi"  
 ) %>%  
 mutate(  
 state\_abbr = str\_remove(state\_abbr, "STHPI") # Remove suffix from state abbreviations  
 )  
  
# Define state groups  
southeast\_abbr <- c("AR", "LA", "KY", "TN", "MS", "AL", "GA", "FL", "NC", "SC", "VA", "WV")  
midwest\_abbr <- c("MN", "WI", "MI", "OH", "IN", "IL", "IA", "MO")  
  
# Assign region labels  
fredHPIData\_long <- fredHPIData\_long %>%  
 mutate(  
 state\_abbr = toupper(state\_abbr),  
 region = case\_when(  
 state\_abbr %in% southeast\_abbr ~ "Southeast",  
 state\_abbr %in% midwest\_abbr ~ "Midwest",  
 TRUE ~ NA\_character\_  
 )  
 ) %>%  
 filter(!is.na(region)) # Keep only relevant states  
  
# Aggregate by region and date (average HPI per region)  
fredHPIData\_regional <- fredHPIData\_long %>%  
 group\_by(region, date) %>%  
 summarise(avg\_hpi = mean(hpi, na.rm = TRUE), .groups = "drop")  
  
# Convert to time series format (Quarterly data)  
fredHPIData\_regional\_ts <- fredHPIData\_regional %>%  
 pivot\_wider(names\_from = region, values\_from = avg\_hpi)   
 # Removed the filter to include all years (starting from 1975)  
  
# Create time series objects  
midwest\_ts <- ts(fredHPIData\_regional\_ts$Midwest, start = c(1975, 1), frequency = 4)  
southeast\_ts <- ts(fredHPIData\_regional\_ts$Southeast, start = c(1975, 1), frequency = 4)  
  
# Fit ARIMA models  
arima\_model\_midwest <- auto.arima(midwest\_ts)  
arima\_model\_southeast <- auto.arima(southeast\_ts)  
  
quarters\_to\_forecast <- 8  
forecast\_midwest <- forecast(arima\_model\_midwest, h = quarters\_to\_forecast, level = c(80, 95))  
forecast\_southeast <- forecast(arima\_model\_southeast, h = quarters\_to\_forecast, level = c(80, 95))  
  
# Convert forecasts to data frames  
forecast\_midwest\_df <- as.data.frame(forecast\_midwest) %>%  
 mutate(  
 Date = seq(from = max(fredHPIData\_regional\_ts$date) + months(3),   
 by = "3 months",   
 length.out = nrow(.)),  
 Region = "Midwest"  
 ) %>%  
 rename(  
 `Forecast HPI` = `Point Forecast`,  
 `Lower 80%` = `Lo 80`,  
 `Upper 80%` = `Hi 80`,  
 `Lower 95%` = `Lo 95`,  
 `Upper 95%` = `Hi 95`  
 )  
  
forecast\_southeast\_df <- as.data.frame(forecast\_southeast) %>%  
 mutate(  
 Date = seq(from = max(fredHPIData\_regional\_ts$date) + months(3),   
 by = "3 months",   
 length.out = nrow(.)),  
 Region = "Southeast"  
 ) %>%  
 rename(  
 `Forecast HPI` = `Point Forecast`,  
 `Lower 80%` = `Lo 80`,  
 `Upper 80%` = `Hi 80`,  
 `Lower 95%` = `Lo 95`,  
 `Upper 95%` = `Hi 95`  
 )  
  
# Combine both forecasts  
forecast\_table <- bind\_rows(forecast\_midwest\_df, forecast\_southeast\_df) %>%  
 select(Region, Date, `Forecast HPI`, `Lower 80%`, `Upper 80%`, `Lower 95%`, `Upper 95%`)  
  
# Generate the table using gt()  
forecast\_table %>%  
 gt() %>%  
 tab\_header(  
 title = "ARIMA Forecast of Housing Price Index (1975–2030)",  
 subtitle = "Forecasted HPI Values with 80% and 95% Confidence Intervals"  
 ) %>%  
 fmt\_number(  
 columns = vars(`Forecast HPI`, `Lower 80%`, `Upper 80%`, `Lower 95%`, `Upper 95%`),  
 decimals = 2  
 ) %>%  
 cols\_label(  
 Region = "Region",  
 Date = "Forecast Date",  
 `Forecast HPI` = "HPI Forecast",  
 `Lower 80%` = "Lower 80%",  
 `Upper 80%` = "Upper 80%",  
 `Lower 95%` = "Lower 95%",  
 `Upper 95%` = "Upper 95%"  
 ) %>%  
 tab\_source\_note(  
 source\_note = "Source: Federal Reserve Bank of St. Louis (FRED)"  
 ) %>%  
 tab\_options(  
 table.font.size = "small",  
 column\_labels.font.weight = "bold",  
 table.border.top.width = px(2),  
 table.border.bottom.width = px(2),  
 heading.align = "center")

Table 1: ARIMA Forecast of Housing Price Index (1975–2030)

Forecasted HPI Values with 80% and 95% Confidence Intervals

| Region | Forecast Date | HPI Forecast | Lower 80% | Upper 80% | Lower 95% | Upper 95% |
| --- | --- | --- | --- | --- | --- | --- |
| Midwest | 2024-10-01 | 526.02 | 522.72 | 529.31 | 520.98 | 531.05 |
| Midwest | 2025-01-01 | 531.37 | 525.32 | 537.42 | 522.12 | 540.62 |
| Midwest | 2025-04-01 | 545.25 | 537.21 | 553.28 | 532.96 | 557.53 |
| Midwest | 2025-07-01 | 551.37 | 541.64 | 561.11 | 536.48 | 566.26 |
| Midwest | 2025-10-01 | 554.46 | 541.98 | 566.95 | 535.37 | 573.56 |
| Midwest | 2026-01-01 | 560.24 | 544.74 | 575.73 | 536.54 | 583.93 |
| Midwest | 2026-04-01 | 571.46 | 553.28 | 589.65 | 543.65 | 599.27 |
| Midwest | 2026-07-01 | 577.73 | 557.04 | 598.42 | 546.09 | 609.37 |
| Southeast | 2024-10-01 | 577.28 | 573.90 | 580.67 | 572.11 | 582.46 |
| Southeast | 2025-01-01 | 584.84 | 577.93 | 591.75 | 574.27 | 595.41 |
| Southeast | 2025-04-01 | 592.85 | 583.07 | 602.63 | 577.90 | 607.80 |
| Southeast | 2025-07-01 | 597.62 | 584.59 | 610.65 | 577.69 | 617.55 |
| Southeast | 2025-10-01 | 602.85 | 585.72 | 619.99 | 576.65 | 629.06 |
| Southeast | 2026-01-01 | 609.81 | 588.30 | 631.31 | 576.92 | 642.70 |
| Southeast | 2026-04-01 | 616.12 | 590.16 | 642.09 | 576.41 | 655.84 |
| Southeast | 2026-07-01 | 621.61 | 590.82 | 652.39 | 574.53 | 668.69 |
| Source: Federal Reserve Bank of St. Louis (FRED) | | | | | | |

library(gt)  
library(dplyr)  
library(forecast)  
library(lubridate)  
  
# Extract only the first three numbers from the ARIMA model order  
arima\_model\_midwest\_order <- arimaorder(arima\_model\_midwest)[1:3]  
arima\_model\_southeast\_order <- arimaorder(arima\_model\_southeast)[1:3]  
  
# Latest Historical HPI values  
latest\_hpi\_midwest <- tail(fredHPIData\_regional %>% filter(region == "Midwest"), 1)$avg\_hpi  
latest\_hpi\_southeast <- tail(fredHPIData\_regional %>% filter(region == "Southeast"), 1)$avg\_hpi  
  
# Forecast start & end dates  
forecast\_start <- max(fredHPIData\_regional\_ts$date) + months(3) # First forecasted quarter  
forecast\_end <- forecast\_start + months((quarters\_to\_forecast - 1) \* 3) # Last forecasted quarter  
  
# Extract summary statistics from forecasts  
mean\_forecast\_midwest <- mean(forecast\_midwest$mean)  
mean\_forecast\_southeast <- mean(forecast\_southeast$mean)  
  
# 80% Confidence Interval  
ci\_80\_lower\_midwest <- mean(forecast\_midwest$lower[,1]) # Lower bound  
ci\_80\_upper\_midwest <- mean(forecast\_midwest$upper[,1]) # Upper bound  
ci\_80\_lower\_southeast <- mean(forecast\_southeast$lower[,1])  
ci\_80\_upper\_southeast <- mean(forecast\_southeast$upper[,1])  
  
# 95% Confidence Interval  
ci\_95\_lower\_midwest <- mean(forecast\_midwest$lower[,2]) # Lower bound  
ci\_95\_upper\_midwest <- mean(forecast\_midwest$upper[,2]) # Upper bound  
ci\_95\_lower\_southeast <- mean(forecast\_southeast$lower[,2])  
ci\_95\_upper\_southeast <- mean(forecast\_southeast$upper[,2])  
  
# Create summary table  
arima\_summary\_table <- tibble(  
 Region = c("Midwest", "Southeast"),  
 `ARIMA Model` = c(  
 paste0("ARIMA(", paste(arima\_model\_midwest\_order, collapse = ","), ")"),  
 paste0("ARIMA(", paste(arima\_model\_southeast\_order, collapse = ","), ")")  
 ),  
 `Latest Historical HPI` = c(latest\_hpi\_midwest, latest\_hpi\_southeast),  
 `Forecast Start Date` = rep(as.character(forecast\_start), 2),  
 `Forecast End Date` = rep(as.character(forecast\_end), 2),  
 `Mean Forecasted HPI` = c(mean\_forecast\_midwest, mean\_forecast\_southeast),  
 `80% CI Lower` = c(ci\_80\_lower\_midwest, ci\_80\_lower\_southeast),  
 `80% CI Upper` = c(ci\_80\_upper\_midwest, ci\_80\_upper\_southeast),  
 `95% CI Lower` = c(ci\_95\_lower\_midwest, ci\_95\_lower\_southeast),  
 `95% CI Upper` = c(ci\_95\_upper\_midwest, ci\_95\_upper\_southeast)  
)  
  
# Generate gt() table  
arima\_summary\_table %>%  
 gt() %>%  
 tab\_header(  
 title = "Summary of Housing Price Index Forecasts (1975–2030)",  
 subtitle = "Southeastern and Midwestern Regions"  
 ) %>%  
 fmt\_number(  
 columns = vars(`Latest Historical HPI`, `Mean Forecasted HPI`, `80% CI Lower`, `80% CI Upper`, `95% CI Lower`, `95% CI Upper`),  
 decimals = 2  
 ) %>%  
 cols\_label(  
 Region = "Region",  
 `ARIMA Model` = "ARIMA Model",  
 `Latest Historical HPI` = "Latest Historical HPI",  
 `Forecast Start Date` = "Forecast Start Date",  
 `Forecast End Date` = "Forecast End Date",  
 `Mean Forecasted HPI` = "Mean Forecasted HPI",  
 `80% CI Lower` = "80% CI Lower",  
 `80% CI Upper` = "80% CI Upper",  
 `95% CI Lower` = "95% CI Lower",  
 `95% CI Upper` = "95% CI Upper"  
 ) %>%  
 tab\_source\_note(  
 source\_note = "Source: Federal Reserve Bank of St. Louis (FRED)"  
 ) %>%  
 tab\_options(  
 table.font.size = "small",  
 column\_labels.font.weight = "bold",  
 table.border.top.width = px(2),  
 table.border.bottom.width = px(2),  
 heading.align = "center")

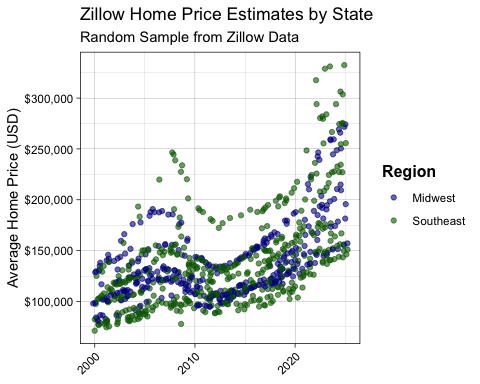
Table 1: Summary of Housing Price Index Forecasts (1975–2030)

Southeastern and Midwestern Regions

| Region | ARIMA Model | Latest Historical HPI | Forecast Start Date | Forecast End Date | Mean Forecasted HPI | 80% CI Lower | 80% CI Upper | 95% CI Lower | 95% CI Upper |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Midwest | ARIMA(0,2,2) | 524.85 | 2024-10-01 | 2026-07-01 | 552.24 | 540.49 | 563.98 | 534.27 | 570.20 |
| Southeast | ARIMA(2,2,0) | 575.28 | 2024-10-01 | 2026-07-01 | 600.37 | 584.31 | 616.44 | 575.81 | 624.94 |
| Source: Federal Reserve Bank of St. Louis (FRED) | | | | | | | | | |

### Zillow Housing Data Scatterplot

library(tidyverse)  
library(lubridate)  
library(readr)  
library(scales)  
  
zillow\_file\_path <- "/Users/cmacbook/Documents/Data Analytics Capstone/zillowCountyData.csv"  
  
zillow\_data <- read\_csv(zillow\_file\_path)  
  
southeast\_states <- c("AR", "LA", "KY", "TN", "MS", "AL", "GA", "FL", "NC", "SC", "VA", "WV")  
midwest\_states <- c("MN", "WI", "MI", "OH", "IN", "IL", "IA", "MO")  
  
# Filter for only Southeastern and Midwestern counties  
zillow\_filtered <- zillow\_data %>%  
 filter(State %in% c(southeast\_states, midwest\_states))  
  
# Convert data from wide to long format (Date format)  
zillow\_long <- zillow\_filtered %>%  
 pivot\_longer(  
 cols = matches("^\\d{4}-\\d{2}-\\d{2}$"), # Select date columns  
 names\_to = "Date",  
 values\_to = "Home\_Price"  
 ) %>%  
 mutate(  
 Date = as.Date(Date),  
 Quarter = floor\_date(Date, "quarter"), # Group into quarterly periods  
 Region = case\_when(  
 State %in% southeast\_states ~ "Southeast",  
 State %in% midwest\_states ~ "Midwest"  
 )  
 )  
  
# Compute the average home price per state per quarter  
zillow\_quarterly\_avg <- zillow\_long %>%  
 group\_by(State, Quarter, Region) %>%  
 summarise(Average\_Home\_Price = mean(Home\_Price, na.rm = TRUE), .groups = "drop")  
  
# Randomly sample points to prevent overcrowding  
set.seed(123) # For reproducibility  
zillow\_sample <- zillow\_quarterly\_avg %>%  
 sample\_frac(0.40)  
  
# Create scatter plot with jittering  
ggplot(zillow\_sample, aes(x = Quarter, y = Average\_Home\_Price, color = Region)) +  
 geom\_point(alpha = 0.6, size = 1.5, position = position\_jitter(width = 100, height = 0)) + # Jitter on x-axis  
 scale\_color\_manual(values = c("Midwest" = "darkblue", "Southeast" = "darkgreen")) +  
 scale\_y\_continuous(labels = label\_dollar()) + # Format y-axis as dollar amounts with commas  
 labs(  
 title = "Zillow Home Price Estimates by State",  
 subtitle = "Random Sample from Zillow Data",  
 x = NULL,  
 y = "Average Home Price (USD)",  
 color = "Region"  
 ) +  
 theme\_linedraw() +  
 theme(  
 legend.position = "right",  
 legend.title = element\_text(size = 12, face = "bold"),  
 axis.text.x = element\_text(angle = 45, hjust = 1)  
 )



library(tidyverse)  
library(lubridate)  
library(forecast)  
library(readr)  
library(scales)  
  
# Define file path  
zillow\_file\_path <- "/Users/cmacbook/Documents/Data Analytics Capstone/zillowCountyData.csv"  
  
# Read in the Zillow data  
zillow\_data <- read\_csv(zillow\_file\_path)  
  
# Define the states for each region  
southeast\_states <- c("AR", "LA", "KY", "TN", "MS", "AL", "GA", "FL", "NC", "SC", "VA", "WV")  
midwest\_states <- c("MN", "WI", "MI", "OH", "IN", "IL", "IA", "MO")  
  
# Filter for only Southeastern and Midwestern counties  
zillow\_filtered <- zillow\_data %>%  
 filter(State %in% c(southeast\_states, midwest\_states))  
  
# Convert data from wide to long format (Date format)  
zillow\_long <- zillow\_filtered %>%  
 pivot\_longer(  
 cols = matches("^\\d{4}-\\d{2}-\\d{2}$"), # Select date columns  
 names\_to = "Date",  
 values\_to = "Home\_Price"  
 ) %>%  
 mutate(  
 Date = as.Date(Date),  
 Quarter = floor\_date(Date, "quarter"), # Group into quarterly periods  
 Region = case\_when(  
 State %in% southeast\_states ~ "Southeast",  
 State %in% midwest\_states ~ "Midwest"  
 )  
 )  
  
# Compute the average home price per region per quarter  
zillow\_quarterly\_avg <- zillow\_long %>%  
 group\_by(Quarter, Region) %>%  
 summarise(Average\_Home\_Price = mean(Home\_Price, na.rm = TRUE), .groups = "drop") %>%  
 pivot\_wider(names\_from = Region, values\_from = Average\_Home\_Price)  
  
# Convert to time series format (Quarterly data)  
zillow\_ts <- zillow\_quarterly\_avg %>%  
 filter(Quarter >= as.Date("2000-01-01")) # Keep only 2000 onward  
  
midwest\_ts <- ts(zillow\_ts$Midwest, start = c(2000, 1), frequency = 4)  
southeast\_ts <- ts(zillow\_ts$Southeast, start = c(2000, 1), frequency = 4)  
  
# Fit ARIMA models  
arima\_model\_midwest <- auto.arima(midwest\_ts)  
arima\_model\_southeast <- auto.arima(southeast\_ts)  
  
# Forecast for the next 10 years (40 quarters)  
quarters\_to\_forecast <- 16  
forecast\_midwest <- forecast(arima\_model\_midwest, h = quarters\_to\_forecast, level = c(80, 95))  
forecast\_southeast <- forecast(arima\_model\_southeast, h = quarters\_to\_forecast, level = c(80, 95))  
  
# Convert forecasts to data frames  
forecast\_midwest\_df <- as.data.frame(forecast\_midwest) %>%  
 mutate(  
 Date = seq(from = max(zillow\_ts$Quarter) + months(3),   
 by = "3 months",   
 length.out = nrow(.)),  
 Region = "Midwest"  
 ) %>%  
 rename(  
 `Forecast HPI` = `Point Forecast`,  
 `Lower 80%` = `Lo 80`,  
 `Upper 80%` = `Hi 80`,  
 `Lower 95%` = `Lo 95`,  
 `Upper 95%` = `Hi 95`  
 )  
  
forecast\_southeast\_df <- as.data.frame(forecast\_southeast) %>%  
 mutate(  
 Date = seq(from = max(zillow\_ts$Quarter) + months(3),   
 by = "3 months",   
 length.out = nrow(.)),  
 Region = "Southeast"  
 ) %>%  
 rename(  
 `Forecast HPI` = `Point Forecast`,  
 `Lower 80%` = `Lo 80`,  
 `Upper 80%` = `Hi 80`,  
 `Lower 95%` = `Lo 95`,  
 `Upper 95%` = `Hi 95`  
 )  
  
# Historical data for plotting  
historical\_midwest <- zillow\_ts %>%  
 select(Quarter, Midwest) %>%  
 rename(Date = Quarter, HPI = Midwest) %>%  
 mutate(Region = "Midwest")  
  
historical\_southeast <- zillow\_ts %>%  
 select(Quarter, Southeast) %>%  
 rename(Date = Quarter, HPI = Southeast) %>%  
 mutate(Region = "Southeast")  
  
# Combine historical and forecasted data  
forecast\_combined <- bind\_rows(  
 forecast\_midwest\_df %>% rename(HPI = `Forecast HPI`),  
 forecast\_southeast\_df %>% rename(HPI = `Forecast HPI`),  
 historical\_midwest,  
 historical\_southeast  
) %>%  
 mutate(Date = as.Date(Date)) # Ensure Date format

### Zillow Housing Data ARIMA Forecast

library(tidyverse)  
library(lubridate)  
library(forecast)  
library(readr)  
library(scales)  
  
# Plot the ARIMA forecast  
ggplot(data = forecast\_combined, aes(x = Date, y = HPI, color = Region)) +  
 geom\_line(data = filter(forecast\_combined, Date <= max(historical\_midwest$Date)),   
 aes(color = Region), size = 0.5, linetype = "solid") + # Historical data  
 geom\_line(size = 0.5) +  
 geom\_ribbon(data = forecast\_midwest\_df, aes(x = Date, ymin = `Lower 80%`, ymax = `Upper 80%`, fill = "Midwest 80%"),   
 alpha = 0.5, inherit.aes = FALSE) + # 80% Confidence interval for Midwest  
 geom\_ribbon(data = forecast\_midwest\_df, aes(x = Date, ymin = `Lower 95%`, ymax = `Upper 95%`, fill = "Midwest 95%"),   
 alpha = 0.35, inherit.aes = FALSE) + # 95% Confidence interval for Midwest  
 geom\_ribbon(data = forecast\_southeast\_df, aes(x = Date, ymin = `Lower 80%`, ymax = `Upper 80%`, fill = "Southeast 80%"),   
 alpha = 0.5, inherit.aes = FALSE) + # 80% Confidence interval for Southeast  
 geom\_ribbon(data = forecast\_southeast\_df, aes(x = Date, ymin = `Lower 95%`, ymax = `Upper 95%`, fill = "Southeast 95%"),   
 alpha = 0.35, inherit.aes = FALSE) + # 95% Confidence interval for Southeast  
 scale\_color\_manual(values = c("Midwest" = "darkblue", "Southeast" = "darkgreen")) +  
 scale\_fill\_manual(  
 values = c("Midwest 80%" = "royalblue1", "Midwest 95%" = "lightblue3",   
 "Southeast 80%" = "palegreen4", "Southeast 95%" = "palegreen"),  
 name = "Confidence Intervals") +  
 scale\_y\_continuous(labels = label\_dollar()) +  
 scale\_x\_date(  
 breaks = seq(as.Date("2000-01-01"), as.Date("2035-01-01"), by = "5 years"),  
 labels = scales::label\_date(format = "%Y")  
 ) +  
 labs(  
 title = "ARIMA Forecast of Zillow Home Price Estimates",  
 subtitle = "Historical and Predicted Data (Quarterly)",  
 x = NULL,  
 y = "Average Home Price (USD)",  
 color = "Region"  
 ) +  
 theme\_linedraw()

