Lab 7 Making maps

Chandana Gadela

2024-10-12

# Load required libraries  
library(tidyverse) # A collection of R packages for data manipulation and visualization

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(maps) # Provides functions for maps and geographical data

##   
## Attaching package: 'maps'  
##   
## The following object is masked from 'package:purrr':  
##   
## map

library(mapdata) # Extends the 'maps' package with additional data  
library(readxl) # Allows reading Excel files

loads the libraries that will be used for working with data and creating maps. Each library serves a specific purpose: manipulating data, working with maps, and reading Excel files.

# Load the datasets from the specified Excel file  
territories\_data <- read\_excel("/Users/jessy/Desktop/Info Visualization/wk7/Sales Reps.xlsx", sheet = 1)  
sales\_reps\_data <- read\_excel("/Users/jessy/Desktop/Info Visualization/wk7/Sales Reps.xlsx", sheet = 2, col\_names = FALSE)

## New names:  
## • `` -> `...1`  
## • `` -> `...2`

reads data from an Excel file. The first sheet contains information about the U.S. states and territories, while the second sheet contains information about sales representatives.

# Clean territories data  
territories <- territories\_data %>%  
 select(`U.S. States and Territories`, Abbreviations, `Sales Rep`) %>% # Select relevant columns  
 mutate(  
 state = tolower(`U.S. States and Territories`), # Convert state names to lowercase for consistency  
 abbrev = Abbreviations, # Create a column for state abbreviations  
 rep\_id = as.character(`Sales Rep`) # Convert Sales Rep ID to character for consistent merging  
 )

clean up the data for U.S. states and territories. select the relevant columns (state name, abbreviation, and sales rep ID), convert the state names to lowercase for consistency, and ensure the sales rep ID is treated as text.

# Add Canadian provinces and their sales representatives  
canadian\_provinces <- tribble( # Create a tibble for Canadian provinces  
 ~state, ~abbrev, ~rep\_id,  
 "alberta", "AB", "7",  
 "british columbia", "BC", "9",  
 "manitoba", "MB", "7",  
 "new brunswick", "NB", "3",  
 "newfoundland and labrador", "NL", "3",  
 "northwest territories", "NT", "7",  
 "nova scotia", "NS", "3",  
 "nunavut", "NU", NA\_character\_, # No representative assigned  
 "ontario", "ON", "3",  
 "prince edward island", "PE", "3",  
 "quebec", "QC", "3",  
 "saskatchewan", "SK", "7",  
 "yukon", "YT", "7"  
)

This part manually adds data for Canadian provinces and their corresponding sales representatives. Each row includes the province name, its abbreviation, and the sales rep ID.

# Combine US territories with Canadian provinces  
territories <- bind\_rows(territories, canadian\_provinces) # Bind the two datasets together

we combine the U.S. state data with the Canadian province data into one dataset, so we can later display both on the same map.

# Clean sales representatives data  
sales\_reps <- sales\_reps\_data %>%  
 rename( # Rename columns for easier access  
 rep\_id = ...1, # First column as rep\_id  
 rep\_name = ...2 # Second column as rep\_name  
 ) %>%  
 mutate(  
 rep\_id = as.character(rep\_id) # Ensure rep\_id is character for consistency  
 )

chunk cleans up the sales representatives’ data. It renames the columns to make them easier to work with and ensures the sales rep IDs are treated as text.

# Define a color palette for each representative  
rep\_colors <- c(  
 "1" = "#E41A1C", # Red  
 "2" = "#377EB8", # Blue  
 "3" = "#4DAF4A", # Green  
 "4" = "#984EA3", # Purple  
 "5" = "#FF7F00", # Orange  
 "6" = "#FFFF33", # Yellow  
 "7" = "#A65628", # Brown  
 "8" = "#F781BF", # Pink  
 "9" = "#999999" # Gray  
)  
  
# Get map data for US states  
us\_states <- map\_data("state") # Retrieves data for US states  
  
# Get map data for Canadian provinces and territories  
canada\_map <- map\_data("world") %>%  
 filter(region %in% c("Canada", "alberta", "british columbia", "manitoba",   
 "new brunswick", "newfoundland and labrador",   
 "northwest territories", "nova scotia", "nunavut",   
 "ontario", "prince edward island", "quebec",   
 "saskatchewan", "yukon")) # Filters for Canadian provinces  
  
# Combine both maps (US and Canada)  
us\_canada\_map <- bind\_rows(  
 us\_states %>% mutate(country = "USA"), # Label US states  
 canada\_map %>% mutate(country = "Canada") # Label Canadian provinces  
)  
  
# Add rep\_id and abbreviation to US states data  
us\_canada\_map <- us\_canada\_map %>%  
 left\_join( # Join the territories data to map data  
 territories %>%  
 select(state, rep\_id, abbrev) %>% # Select relevant columns for joining  
 mutate(state = tolower(state)), # Convert state names to lowercase  
 by = c("region" = "state") # Match based on region and state  
 )  
  
# Calculate centers for labeling states and provinces  
region\_centers <- us\_canada\_map %>%  
 group\_by(region) %>% # Group by region  
 summarize(  
 long = mean(range(long)), # Calculate average longitude for positioning  
 lat = mean(range(lat)), # Calculate average latitude for positioning  
 rep\_id = first(rep\_id), # Take the first representative ID in the group  
 abbrev = first(abbrev) # Take the first abbreviation in the group  
 )

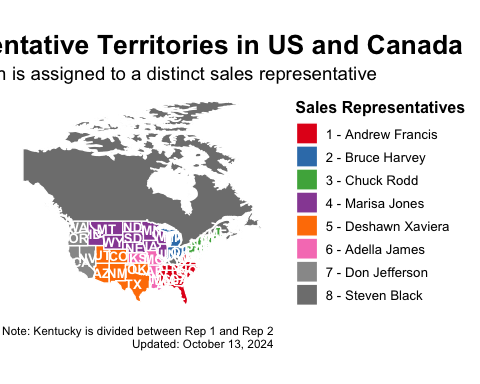
# Create the main map with both US and Canadian regions  
main\_map <- ggplot() +  
 # Plot regions with geom\_polygon  
 geom\_polygon(data = us\_canada\_map,   
 aes(x = long, y = lat, group = group, fill = factor(rep\_id)), # Map aesthetics  
 color = "white", size = 0.5) + # White border for clarity  
 # Add labels for states and provinces  
 geom\_text(data = region\_centers,  
 aes(x = long, y = lat, label = abbrev), # Position labels at calculated centers  
 size = 3.5, fontface = "bold", color = "white") + # Formatting for labels  
 # Use the custom color palette for filling regions  
 scale\_fill\_manual(  
 values = rep\_colors, # Assign colors to reps  
 name = "Sales Representatives", # Legend title  
 labels = paste(  
 sales\_reps$rep\_id,  
 "-", sales\_reps$rep\_name # Create labels for sales reps  
 )  
 ) +  
 # Set coordinate system for better aspect ratio  
 coord\_fixed(ratio = 1.3) + # Adjust ratio to improve map appearance  
 # Customize the appearance of the map  
 theme\_minimal() + # Minimal theme for a clean look  
 theme(  
 plot.title = element\_text(size = 20, face = "bold", hjust = 0.5), # Title formatting  
 plot.subtitle = element\_text(size = 14, hjust = 0.5), # Subtitle formatting  
 legend.position = "right", # Position the legend  
 legend.title = element\_text(size = 12, face = "bold"), # Legend title formatting  
 legend.text = element\_text(size = 10), # Legend text formatting  
 panel.grid = element\_blank(), # Remove grid lines  
 axis.text = element\_blank(), # Remove axis text  
 axis.title = element\_blank() # Remove axis titles  
 ) +  
 # Add titles and captions to the plot  
 labs(  
 title = "Sales Representative Territories in US and Canada", # Main title  
 subtitle = "Each region is assigned to a distinct sales representative", # Subtitle  
 caption = paste0( # Caption with additional information  
 "Note: Kentucky is divided between Rep 1 and Rep 2\n", # Note about Kentucky  
 "Updated: ", format(Sys.Date(), "%B %d, %Y") # Date of the update  
 )  
 )

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

map that shows both U.S. states and Canadian provinces along with their assigned sales representatives, you would merge the cleaned data with map data, assign colors to different sales reps, and label the states and provinces on the map. The rest of the code uses the combined data to generate a clear, labeled map showing the territories.

# Print the main map to the console  
print(main\_map)

## Warning: Removed 1 row containing missing values or values outside the scale range  
## (`geom\_text()`).



# Save the combined map as a PNG file with specified dimensions and resolution  
ggsave("combined\_us\_canada\_sales\_map.png", main\_map, width = 30, height = 15, dpi = 400) # Increased dimensions for better clarity

## Warning: Removed 1 row containing missing values or values outside the scale range  
## (`geom\_text()`).