R Notebook

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

plot(cars)



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

library(readxl)   
# Load the readxl package to read Excel files  
library(ggplot2)  
library(maps)   
# Load maps for U.S. state boundary data  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Load dplyr for data manipulation  
library(tidyr)   
# Load tidyr for unnesting data  
# Read the Excel file

# reads the file from file path and loads to sales\_data variable  
sales\_data <- read\_excel("/Users/ajaykommineni/Downloads/Sales Reps.xlsx")

setwd("/Users/ajaykommineni/Downloads")  
states\_data <- read\_excel("Sales Reps.xlsx", sheet = "States")  
#Reads data from an Excel file. Here, it reads the "States" and "Reps" sheets.  
# Read the "States" sheet  
# Read the "Reps" sheet  
reps\_data <- read\_excel("Sales Reps.xlsx", sheet = "Reps")  
  
# Check column names  
colnames(states\_data)

## [1] "U.S. States and Territories" "Abbreviations"   
## [3] "Sales Rep"

# Should be: "U.S. States and Territories", "Abbreviations", "Sales Rep"  
# Print column names of states\_data and reps\_data  
colnames(reps\_data)

## [1] "1" "Andrew Francis"

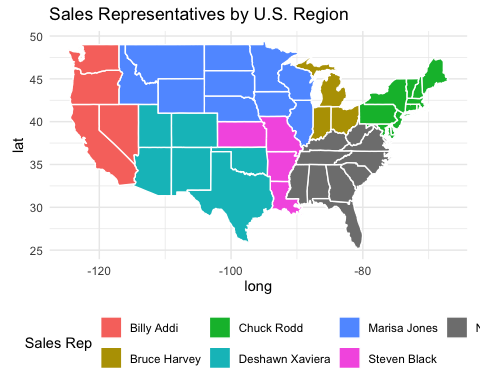
# Rename columns to "Number" and "Name"  
#There is no names given in the data  
#This will help to add column names   
colnames(reps\_data) <- c("Number", "Name")  
colnames(states\_data) <- c("State", "Abbreviation", "Sales\_Rep")  
# Rename columns to "State", "Abbreviation", "Sales\_Rep"  
# Filter out Canadian provinces and territories

# To get only Us we need to Filter out Canadian provinces and territories  
canadian\_abbreviations <- c("AB", "BC", "MB", "NB", "NL", "NT", "NS", "NU", "ON", "PE", "QC", "SK", "YT") #List of Canadian abbreviations  
states\_data <- states\_data %>%  
 filter(!Abbreviation %in% canadian\_abbreviations)  
# Remove rows where Abbreviation is in the Canadian list  
# Clean and merge data  
# Clean and prepare the Sales\_Rep column  
states\_data <- states\_data %>%  
 mutate(Sales\_Rep = strsplit(as.character(Sales\_Rep), " and ")) %>%   
 # Split "Sales\_Rep" by " and "  
 unnest(Sales\_Rep) %>%   
 # Unnest the list into separate rows (e.g., "1 and 2" becomes two rows: 1 and 2)  
 mutate(Sales\_Rep = as.numeric(Sales\_Rep))   
# Convert Sales\_Rep to numeric   
# Merge states\_data with reps\_data to get sales rep names  
combined\_data <- states\_data %>%  
 left\_join(reps\_data, by = c("Sales\_Rep" = "Number")) # Join on Sales\_Rep (states\_data) and Number (reps\_data)

# Prepare U.S. map data  
us\_map <- map\_data("state")  
# Get U.S. state boundary data from the maps package  
combined\_data <- combined\_data %>%  
 mutate(State = tolower(State))  
# Convert state names to lowercase to match us\_map's "region" column  
# Merge U.S. map data with sales rep data  
us\_map <- us\_map %>%  
 left\_join(combined\_data, by = c("region" = "State"))

## Warning in left\_join(., combined\_data, by = c(region = "State")): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 3895 of `x` matches multiple rows in `y`.  
## ℹ Row 1 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

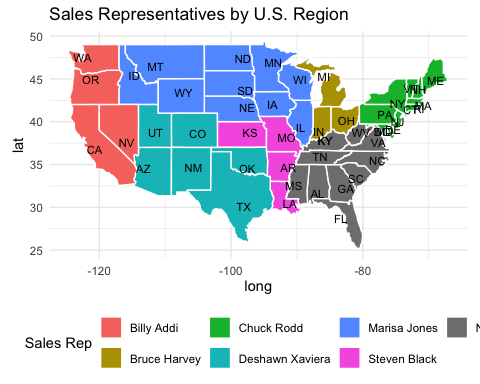
# Join on region (us\_map) and State (combined\_data)  
# Plot the map (U.S. only)  
ggplot() +  
#ploting a map of the U.S., with each state colored by its assigned sales rep.  
 geom\_polygon(data = us\_map, aes(x = long, y = lat, group = group, fill = Name), color = "white") +  
#Draws polygons (e.g., U.S. states) on the map, colored by sales rep.  
 coord\_fixed(1.3) +   
 #Ensures the map has a fixed aspect ratio to prevent distortion  
 theme\_minimal() +  
 labs(title = "Sales Representatives by U.S. Region", fill = "Sales Rep") +  
 # Adding title and legend label  
 theme(legend.position = "bottom")



# Move the legend to the bottom of the plot

# Plot the map (U.S. only) with state abbreviations

#Calculate the center of each state for placing abbreviations  
state\_centers <- us\_map %>%  
 group\_by(region) %>%  
 summarize(  
 long = mean(long),   
 # Calculate the mean longitude (center of the state)  
 lat = mean(lat)   
 # Calculate the mean latitude (center of the state)  
 ) %>%  
 left\_join(combined\_data, by = c("region" = "State"))   
# Add abbreviations to the centers  
  
# Plot the map (U.S. only) with state abbreviations  
ggplot() +  
 geom\_polygon(data = us\_map, aes(x = long, y = lat, group = group, fill = Name), color = "white") + # Draw U.S. states, colored by sales rep  
 geom\_text(data = state\_centers, aes(x = long, y = lat, label = Abbreviation), color = "black", size = 3) +   
 # Add state abbreviations  
 coord\_fixed(1.3) +   
 theme\_minimal() +   
 labs(title = "Sales Representatives by U.S. Region", fill = "Sales Rep") +   
 theme(legend.position = "bottom")



# Load libraries  
library(ggplot2)  
library(maps)  
library(dplyr)  
  
# Filter Kentucky data  
kentucky\_data <- states\_data %>%  
 filter(State == "Kentucky") %>%  
 mutate(Sales\_Rep = ifelse(Abbreviation %in% c("Boone", "Campbell", "Kenton", "Pendleton", "Gant", "Gallatin", "Carroll", "Owen", "Henry", "Trimble", "Oldham", "Jefferson", "Shelby", "Franklin", "Scott", "Woodford", "Fayette", "Jessmine"), 2, 1))  
  
# Get Kentucky county map data  
kentucky\_map <- map\_data("county") %>%  
 filter(region == "kentucky") %>%  
 left\_join(kentucky\_data, by = c("subregion" = "Abbreviation"))  
  
# Plot the map  
ggplot() +  
 geom\_polygon(data = kentucky\_map, aes(x = long, y = lat, group = group, fill = factor(Sales\_Rep)), color = "white") +  
 scale\_fill\_manual(values = c("1" = "green", "2" = "red")) + # Green for Rep 1, Red for Rep 2  
 coord\_fixed(1.3) + # Fix aspect ratio  
 theme\_minimal() + # Minimal theme  
 labs(title = "Kentucky Sales Representatives by County", fill = "Sales Rep") + # Title and legend  
 theme(legend.position = "bottom") # Legend at the bottom

