Final Project

Sean Lydon, Preet Lodha-Jain, Peter Kim

Overview

We were trying to identify the relationship between climate change the rise of city temperatures across the United States.

Repo Structure

In our repo, you will find both the pdf and the qmd files of our final report, under Project.pdf and Project.qmd. Any and all code we developed will be under the projectCode.R file, which we used to run our code before creating out final report. The current file, the README file, is a brief overview of the repo itself. The Project_Guidelines files is the guidelines we followed to complete this project.

Data Sources and Acknowledgements

Our data set, titled "Climate Change: Earth Surface Temperature Data", came from kaggle.com.

link: https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data?resource=download

Temperature Analysis of Major US Cities

To visualize the temperature trends in major US cities over time, we analyzed the "Climate Change: Earth Surface Temperature Data" dataset. We focused on six major cities: New York, San Francisco, Los Angeles, Houston, Philadelphia, and Chicago. The analysis involved filtering the data for these cities within the United States, ensuring we had temperature readings from January 1, 1913, onwards. We then calculated the yearly average temperature in Fahrenheit for each city.

The following plots illustrate the average yearly temperature for each of the selected cities, along with a smoothed trend line to highlight the overall temperature progression.

```r
# Load required libraries
library(tidyverse)
library(lubridate)
library(ggplot2)
library(dplyr)

# Load the dataset
# The file is too big to link in the Git
file_path <- "~/Downloads/archive/GlobalLandTemperaturesByCity.csv"
GlobalLandTemperaturesByCity <- read_csv(file_path)

# Define the cities of interest
target_cities <- c("New York", "San Francisco", "Los Angeles",
                   "Houston", "Philadelphia", "Chicago")

# Filter and preprocess the data
majorCity_globalTemp <- GlobalLandTemperaturesByCity %>%
  filter(City %in% target_cities,
         Country == "United States",
         !is.na(AverageTemperature),
         dt >= as.Date("1913-01-01")) %>%
  mutate(
    Year = year(dt),
    AverageTemperatureF = (AverageTemperature * 9/5) + 32
  )

# Compute yearly average temperature
yearly_avg_temp <- majorCity_globalTemp %>%
  group_by(City, Year) %>%
  summarize(AvgTemperatureF = mean(AverageTemperatureF, na.rm = TRUE), .groups = 'drop')

# Fancy plotting function
plot_city_temp <- function(city_name) {
  city_data <- filter(yearly_avg_temp, City == city_name) %>%
    arrange(Year) %>%
    mutate(
      RollingAvg = (AvgTemperatureF +
                      lag(AvgTemperatureF, 1) +
                      lag(AvgTemperatureF, 2) +
                      lag(AvgTemperatureF, 3) +
                      lag(AvgTemperatureF, 4)) / 5
    )

  ggplot(city_data, aes(x = Year)) +
    geom_line(aes(y = AvgTemperatureF, color = "Range of Temperatures"), size = 1.1, alpha =
    geom_smooth(aes(y = AvgTemperatureF, color = "Average Temperature"), method = "loess", s
    scale_color_manual(
      name = "Legend",
      values = c("Range of Temperatures" = "#1f77b4", "Average Temperature" = "#ff7f0e")
    ) +
    labs(
      title = paste(city_name),
      subtitle = "Average Yearly Temperature (°F) from 1913 to Present",
      x = "Year",
      y = "Avg Temperature (°F)"
    ) +
    theme_minimal(base_family = "Arial") +
    theme(
      plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
      plot.subtitle = element_text(size = 12, hjust = 0.5),
      axis.title = element_text(size = 11, face = "bold"),
      axis.text = element_text(color = "gray30"),
      legend.title = element_text(face = "bold"),
      legend.position = "bottom",
      panel.grid.minor = element_blank(),
      panel.grid.major = element_line(color = "gray80", linetype = "dotted"),
      plot.background = element_rect(fill = "#f9f9f9", color = NA),
      panel.background = element_rect(fill = "#f9f9f9", color = NA)
    )
}

# Display plots one after another
for (city in target_cities) {
  print(plot_city_temp(city))
}
```