MA617HW3_b-d

2024-09-28

a.

```
# Load required libraries
library(data.table)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

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

```
# Define the base URL components for NOAA data
file_root <- "https://www.ndbc.noaa.gov/view_text_file.php?filename=44013h"
tail <- ".txt.gz&dir=data/historical/stdmet/"

# Initialize an empty list to store data for each year
all_years_data <- list()

# Loop through each year from 1985 to 2023
for (year in 1985:2023) {

  # Construct the full URL for the current year
  path <- paste0(file_root, year, tail)

  # Try to process each year's data, and catch errors if they occur
  tryCatch({

    # Read the header line
    header <- scan(path, what = 'character', nlines = 1)

    # Determine how many lines to skip based on the year
    skip_lines <- if (year < 1990) 1 else 2

    # Read the buoy data, filling missing values and skipping the appropriate number of lines
    buoy_data <- fread(path, header = FALSE, skip = skip_lines, fill = TRUE)
```

```
    # Handle cases where the number of columns in data differs from the header
    if (length(header) != ncol(buoy_data)) {
      warning(paste("Mismatch in columns for year", year))

      # Adjust header or data if there is a mismatch
      if (length(header) > ncol(buoy_data)) {
        header <- header[1:ncol(buoy_data)]  # Trim the header
      } else {
        buoy_data <- buoy_data[, 1:length(header)]  # Trim the data
      }
    }

    # Assign column names to the data
    colnames(buoy_data) <- header

    # Validate the date columns (YY, MM, DD, hh) before parsing
    if (all(c("YY", "MM", "DD", "hh") %in% names(buoy_data))) {
      # Convert date components into a single Date column
      buoy_data$Date <- ymd_h(paste(buoy_data$YY, buoy_data$MM, buoy_data$DD, buoy_data$hh, sep = "-"))
    } else {
      warning(paste("Date columns missing for year", year))
      buoy_data$Date <- NA
    }

    # Store the buoy data for this year in the list
    all_years_data[[as.character(year)]] <- buoy_data

  }, error = function(e) {
    # If there's an error, print a message and continue with the next year
    warning(paste("Failed to process data for year", year, ":", e$message))
  })
}
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 1999
```

```
## Warning in fread(path, header = FALSE, skip = skip_lines, fill = TRUE): Stopped
## early on line 5114. Expected 16 fields but found 17. Consider fill=17 or even
## more based on your knowledge of the input file. Use fill=Inf for reading the
## whole file for detecting the number of fields. First discarded non-empty line:
## <<2000 08 01 00 78 4.3 5.1 0.58 8.33 5.36 999 1022.9 17.3 17.5 15.0 99.0
## 99.00>>
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Mismatch in
## columns for year 2000
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2000
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2001
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2002

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2003

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2004

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2005

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2006

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2007

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2008

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2009

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2010

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2011

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2012

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2013

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2014

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2015

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2016

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2017

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2018
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2019

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2020

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2021

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2022

## Warning in doTryCatch(return(expr), name, parentenv, handler): Date columns
## missing for year 2023
```

```
# Combine all the data into a single data.table
combined_data <- rbindlist(all_years_data, use.names = TRUE, fill = TRUE)

# View the first few rows of the combined data
head(combined_data)
```

```
##       YY    MM    DD    hh    WD  WSPD   GST  WVHT   DPD   APD   MWD    BAR
##    <int> <int> <int> <int> <int> <num> <num> <num> <num> <num> <int>  <num>
## 1:    85     1     1     0    60     4     5    99    99    99   999 1030.3
## 2:    85     1     1     1    80     4     5    99    99    99   999 1030.0
## 3:    85     1     1     2   100     4     5    99    99    99   999 1030.1
## 4:    85     1     1     3   100     4     5    99    99    99   999 1029.4
## 5:    85     1     1     4   110     4     5    99    99    99   999 1028.6
## 6:    85     1     1     5    90     4     5    99    99    99   999 1027.8
##     ATMP  WTMP  DEWP   VIS                Date  YYYY  TIDE    mm   #YY  WDIR
##    <num> <num> <num> <num>              <POSc> <int> <num> <int> <int> <int>
## 1:   4.7   6.7   999    99 1985-01-01 00:00:00    NA    NA    NA    NA    NA
## 2:   5.1   6.7   999    99 1985-01-01 01:00:00    NA    NA    NA    NA    NA
## 3:   5.6   6.6   999    99 1985-01-01 02:00:00    NA    NA    NA    NA    NA
## 4:   5.8   6.7   999    99 1985-01-01 03:00:00    NA    NA    NA    NA    NA
## 5:   5.8   6.7   999    99 1985-01-01 04:00:00    NA    NA    NA    NA    NA
## 6:   5.3   6.7   999    99 1985-01-01 05:00:00    NA    NA    NA    NA    NA
##     PRES
##    <num>
## 1:    NA
## 2:    NA
## 3:    NA
## 4:    NA
## 5:    NA
## 6:    NA
```

```
#save the csv file for 1985-2023
write.csv(combined_data, "buoy_44013_data_1985_2023.csv", row.names = FALSE)


nrow(combined_data)
```

```
## [1] 462284
```

b.

```
# Load the CSV file
combined_data <- fread("buoy_44013_data_1985_2023.csv")

# List the columns where '999' or '99' represent missing values
missing_columns <- c("WDIR", "WSPD", "GST", "WVHT", "DPD", "APD", "MWD", "ATMP", "WTMP", "DEWP", "VIS")

# Replace '999' or '99' with NA in the relevant columns
combined_data[, (missing_columns) := lapply(.SD, function(x) fifelse(x == 999 | x == 99, NA, x)), .SDcol

# Check if the replacement worked
summary(combined_data)
```

```
##        YY               MM               DD               hh
##  Min.   :85.0     Min.   : 1.000   Min.   : 1.00   Min.   : 0.0
##  1st Qu.:88.0     1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 6.0
##  Median :92.0     Median : 7.000   Median :16.00   Median :11.0
##  Mean   :91.5     Mean   : 6.593   Mean   :15.73   Mean   :11.5
##  3rd Qu.:95.0     3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:17.0
##  Max.   :98.0     Max.   :12.000   Max.   :31.00   Max.   :23.0
##  NA's   :346143
##        WD              WSPD            GST             WVHT
##  Min.   :  0.0    Min.   : 0.0    Min.   : 0.00   Min.   :0.00
##  1st Qu.:134.0    1st Qu.: 3.5    1st Qu.: 4.20   1st Qu.:0.41
##  Median :222.0    Median : 5.3    Median : 6.50   Median :0.66
##  Mean   :264.2    Mean   : 5.9    Mean   : 7.29   Mean   :0.87
##  3rd Qu.:297.0    3rd Qu.: 7.9    3rd Qu.: 9.70   3rd Qu.:1.06
##  Max.   :999.0    Max.   :25.7    Max.   :32.40   Max.   :9.10
##  NA's   :280220   NA's   :33183   NA's   :33485   NA's   :144269
##       DPD              APD              MWD              BAR
##  Min.   : 0.00    Min.   : 0.00    Min.   :  0.0    Min.   : 964.6
##  1st Qu.: 4.55    1st Qu.: 3.85    1st Qu.: 77.0    1st Qu.:1010.3
##  Median : 7.69    Median : 4.70    Median : 94.0    Median :1015.8
##  Mean   : 7.39    Mean   : 4.96    Mean   :124.4    Mean   :1066.8
##  3rd Qu.:10.00    3rd Qu.: 5.85    3rd Qu.:130.0    3rd Qu.:1021.2
##  Max.   :25.00    Max.   :12.10    Max.   :360.0    Max.   :9999.0
##  NA's   :147961   NA's   :144269   NA's   :327154   NA's   :280220
##       ATMP             WTMP             DEWP             VIS
##  Min.   :-19.70   Min.   :-1.80   Min.   :-24.9    Min.   : 0.0
##  1st Qu.:  3.90   1st Qu.: 5.80   1st Qu.: -0.2    1st Qu.: 8.1
##  Median :  9.70   Median :10.50   Median :  7.1    Median : 9.4
##  Mean   :  9.86   Mean   :11.04   Mean   :  6.6    Mean   :12.5
##  3rd Qu.: 16.70   3rd Qu.:16.20   3rd Qu.: 14.7    3rd Qu.:11.6
##  Max.   : 32.10   Max.   :27.80   Max.   : 26.1    Max.   :36.0
##  NA's   :102761   NA's   :13185   NA's   :253602   NA's   :443047
##       Date                                YYYY             TIDE
##  Min.   :1985-01-01 00:00:00.000   Min.   :1999     Min.   :99
##  1st Qu.:1988-07-09 02:00:00.000   1st Qu.:2001     1st Qu.:99
##  Median :1992-01-07 07:00:00.000   Median :2003     Median :99
##  Mean   :1991-12-20 10:59:45.585   Mean   :2003     Mean   :99
```

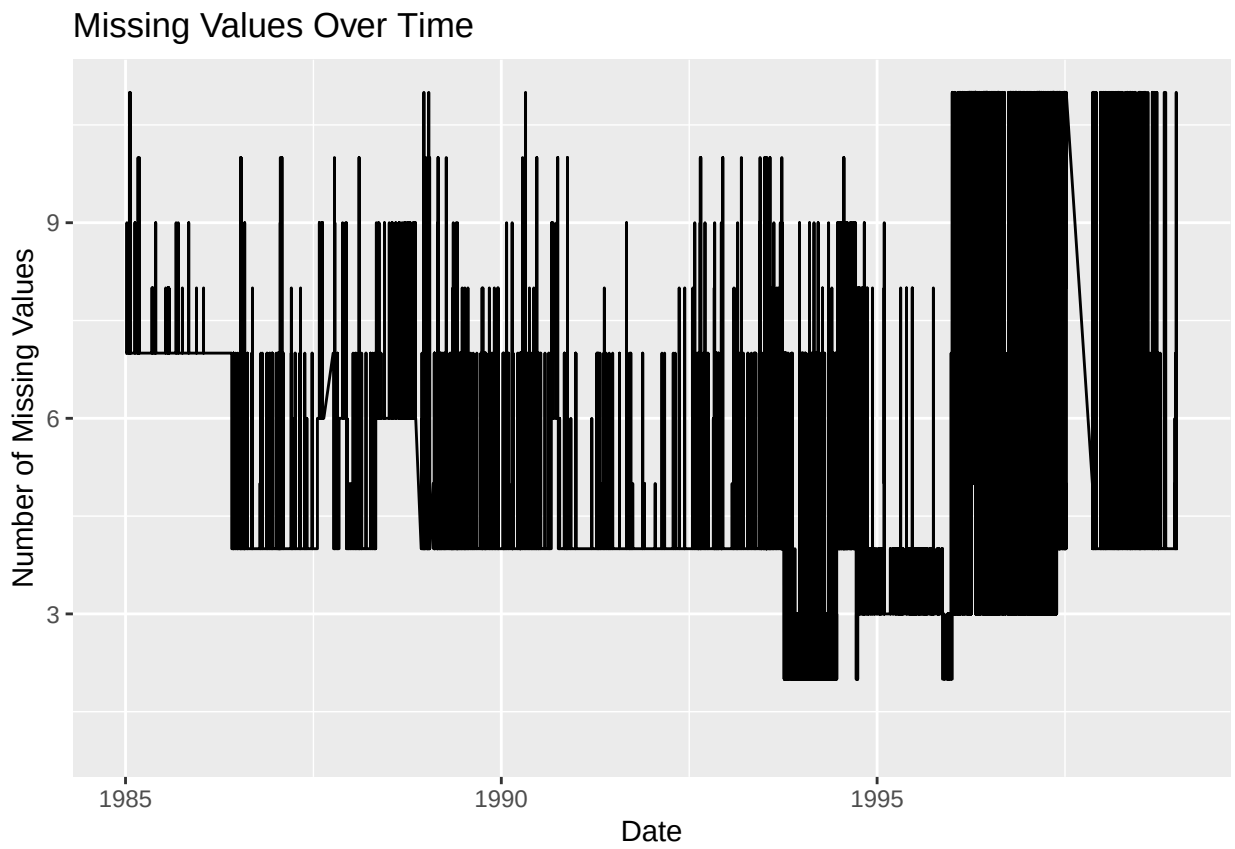
```
##  3rd Qu.:1995-05-09 23:00:00.000   3rd Qu.:2005     3rd Qu.:99
##  Max.   :1998-12-31 23:00:00.000   Max.   :2006     Max.   :99
##  NA's   :346143                    NA's   :396361   NA's   :129599
##        mm              #YY              WDIR             PRES
##  Min.   : 0.00    Min.   :2007     Min.   :  0.0    Min.   : 970
##  1st Qu.:10.00    1st Qu.:2015     1st Qu.:131.0    1st Qu.:1011
##  Median :40.00    Median :2021     Median :205.0    Median :1016
##  Mean   :31.58    Mean   :2018     Mean   :197.3    Mean   :1193
##  3rd Qu.:50.00    3rd Qu.:2022     3rd Qu.:280.0    3rd Qu.:1022
##  Max.   :50.00    Max.   :2023     Max.   :360.0    Max.   :9999
##  NA's   :164635   NA's   :182064   NA's   :210717   NA's   :182064
```

```
# Create a summary of NA counts for each column
na_counts <- sapply(combined_data, function(x) sum(is.na(x)))
print(na_counts)
```

```
##     YY     MM     DD     hh     WD   WSPD    GST   WVHT    DPD    APD    MWD
## 346143      0      0      0 280220  33183  33485 144269 147961 144269 327154
##    BAR   ATMP   WTMP   DEWP    VIS   Date   YYYY   TIDE     mm    #YY   WDIR
## 280220 102761  13185 253602 443047 346143 396361 129599 164635 182064 210717
##   PRES
## 182064
```

```
# Check for NA values by Date (if you want to analyze them over time)
na_by_date <- combined_data[, .(NA_count = rowSums(is.na(.SD))), .SDcols = missing_columns, by = Date]

# Visualize the pattern of missing values over time using ggplot2
library(ggplot2)

ggplot(na_by_date, aes(x = Date, y = NA_count)) +
  geom_line() +
  labs(title = "Missing Values Over Time", x = "Date", y = "Number of Missing Values")
```

```
## Warning: Removed 346143 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



From the missing data graph, we see a high density of values unavailable. The range seems to fluctuate and has spikes showing there might have been external issues such as equipment malfunctioning and maintenance. However, from $1990-1995$ there seems to be less of the missing values, which means that the malfunctioning was less frequent. From the summary that we see, WDIR, BAR, VIS, MWD seems to have a very high missing value, while VIS seems to have much lower missing values. This shows that some of the systems for specific criteria has been malfunctioning. Replacing missing values with NA might be effective to see the malfunctioning of systems, but it could be better for us to find other dataset that has specific criteria that we have more missing values on. There are some patterns shown in the graph, such as higher frequency of missing values in 1985 to 1990, which could be due to technological limitations. Also, there has been a significant increase in missing data from 1995 to 2023, showing possible operational changes or the end of dataset coverage.

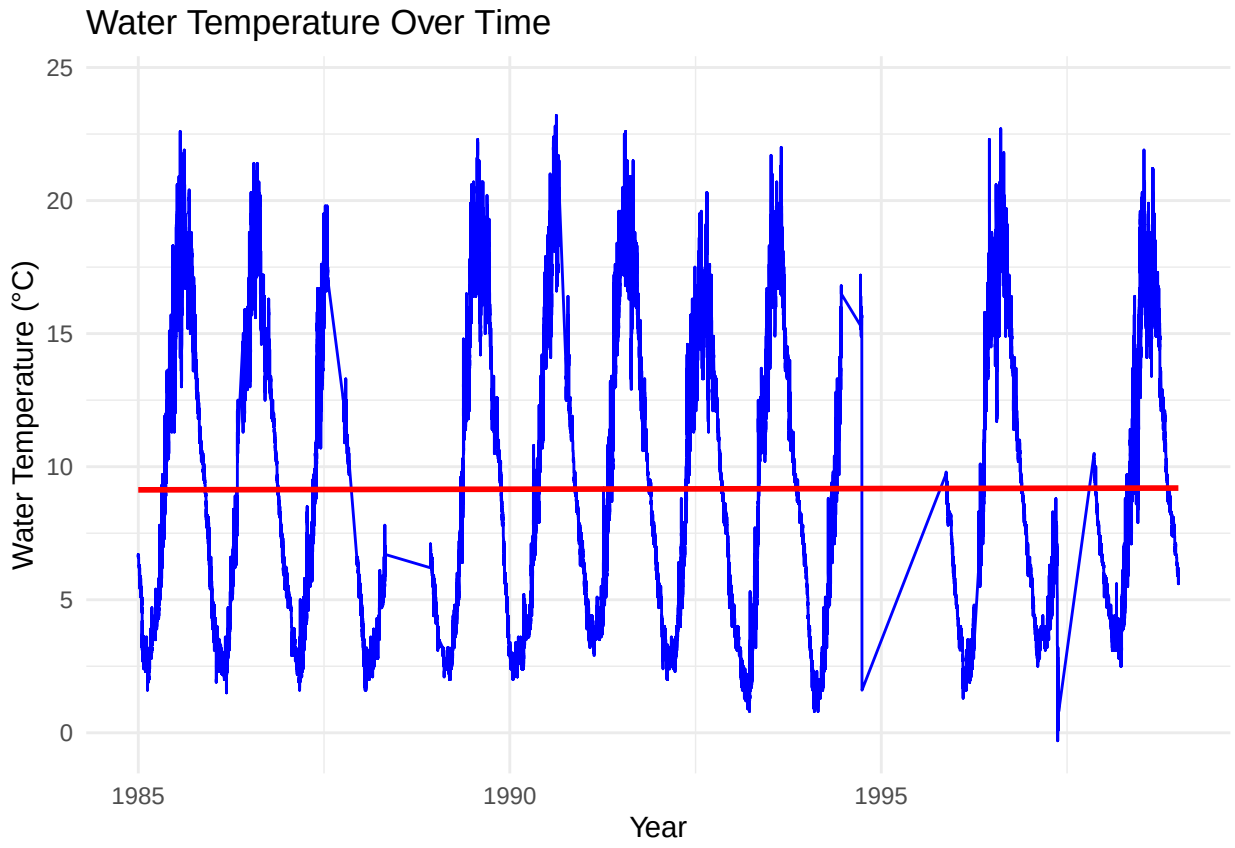
```r
# Load required libraries
library(data.table)
library(ggplot2)

# Filter out rows with missing values in key climate variables
climate_data <- combined_data[, .(Date, WTMP, ATMP, BAR, WSPD)]
climate_data <- climate_data[!is.na(WTMP) & !is.na(ATMP) & !is.na(BAR) & !is.na(WSPD)]

# Ensure the date column is properly formatted
climate_data$Date <- as.Date(climate_data$Date)

# Plot Water Temperature over Time
ggplot(climate_data, aes(x = Date, y = WTMP)) +
  geom_line(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") + # Trend line
```

```
  labs(title = "Water Temperature Over Time", x = "Year", y = "Water Temperature (°C)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 65470 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

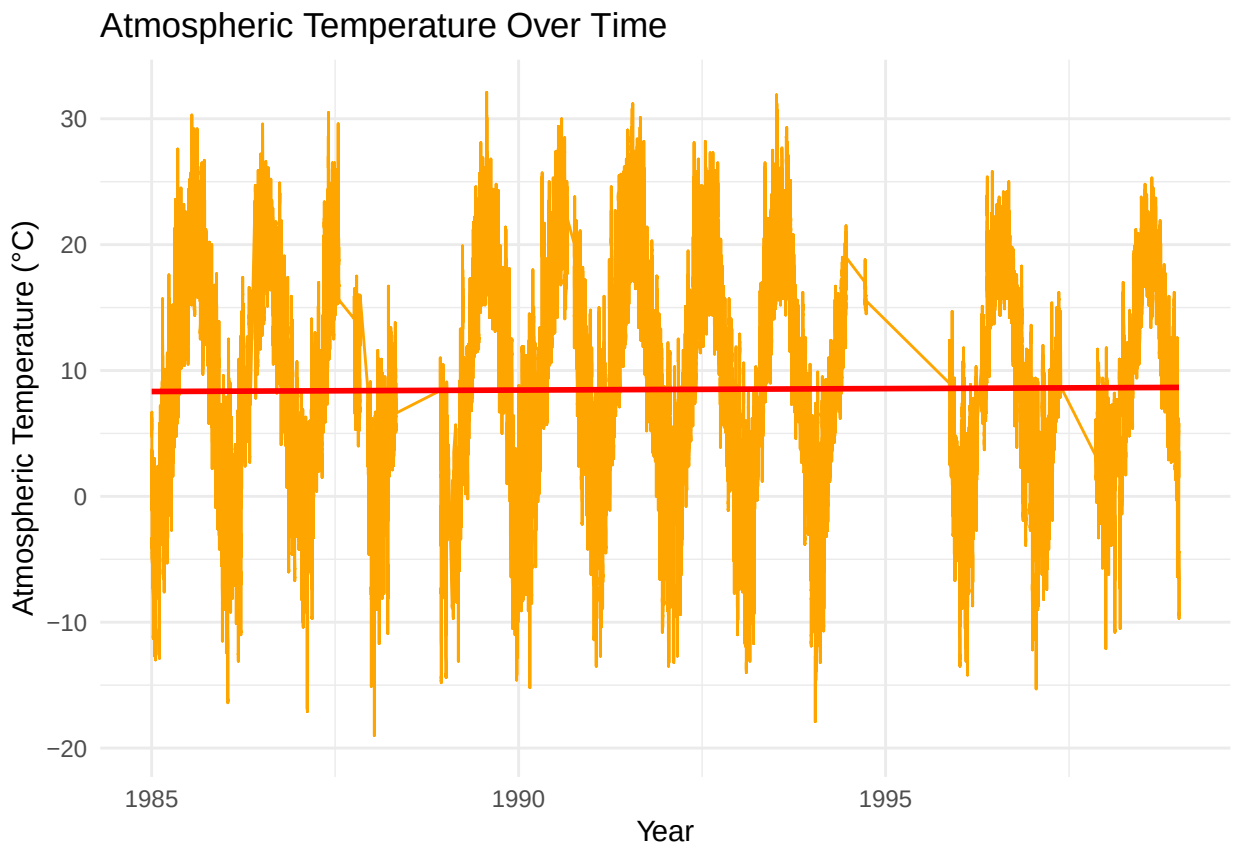
```
## Warning: Removed 65470 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Plot Atmospheric Temperature over Time
ggplot(climate_data, aes(x = Date, y = ATMP)) +
  geom_line(color = "orange") +
  geom_smooth(method = "lm", se = FALSE, color = "red") + # Trend line
  labs(title = "Atmospheric Temperature Over Time", x = "Year", y = "Atmospheric Temperature (°C)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

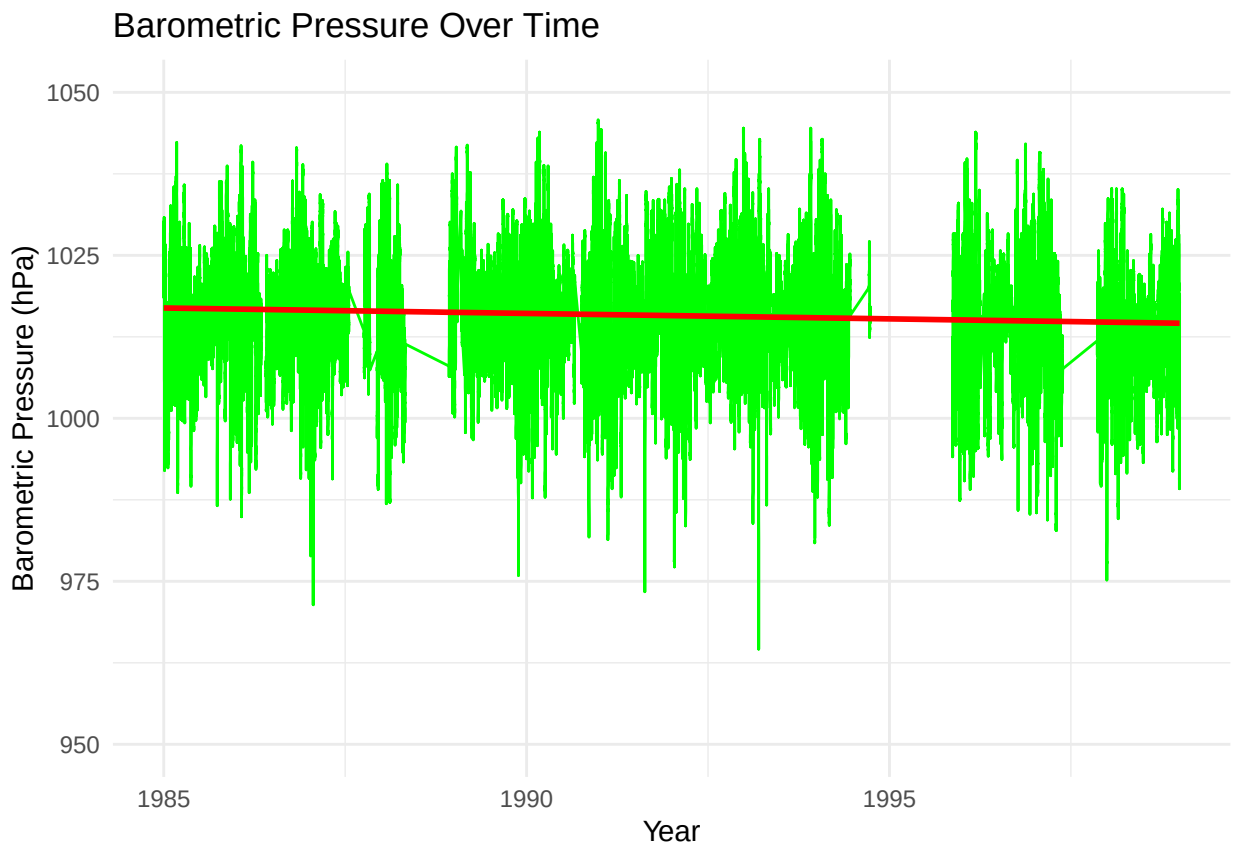
```
## Warning: Removed 65470 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 65470 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Plot Barometric Pressure over Time with zoomed-in scale
ggplot(climate_data, aes(x = Date, y = BAR)) +
  geom_line(color = "green") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +  # Trend line
  labs(title = "Barometric Pressure Over Time", x = "Year", y = "Barometric Pressure (hPa)") +
  scale_y_continuous(limits = c(950, 1050)) +  # Set realistic limits for barometric pressure
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

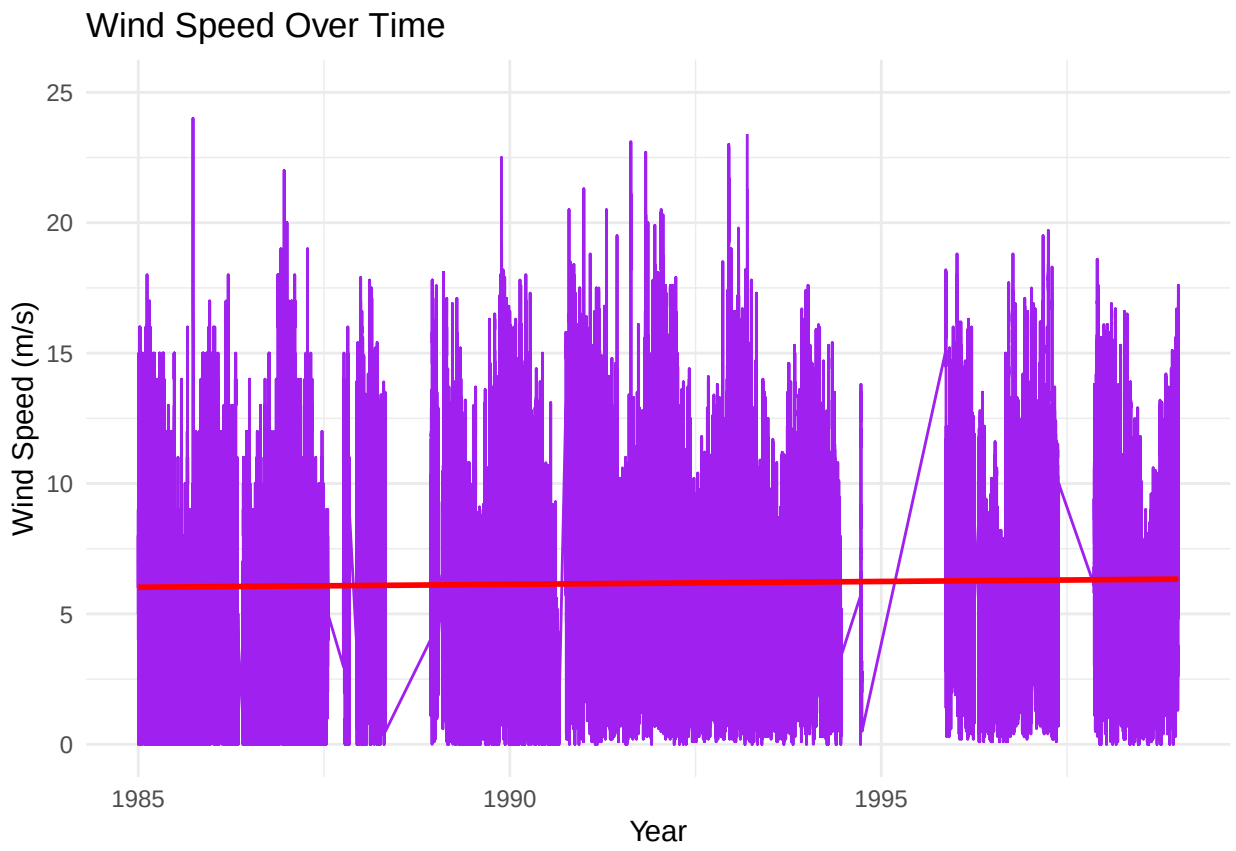
```
## Warning: Removed 65516 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 65470 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Plot Wind Speed over Time with adjusted scale
ggplot(climate_data, aes(x = Date, y = WSPD)) +
  geom_line(color = "purple") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +  # Trend line
  labs(title = "Wind Speed Over Time", x = "Year", y = "Wind Speed (m/s)") +
  scale_y_continuous(limits = c(0, 25)) +  # Set realistic limits for wind speed
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 65471 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 65470 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Convert Date to a numeric format (number of days since origin)
climate_data$Date_numeric <- as.numeric(climate_data$Date)

# Fit linear models for each variable
lm_wind_speed <- lm(WSPD ~ Date_numeric, data = climate_data)
lm_wtmp <- lm(WTMP ~ Date_numeric, data = climate_data)
lm_atmp <- lm(ATMP ~ Date_numeric, data = climate_data)
lm_bar <- lm(BAR ~ Date_numeric, data = climate_data)

# Extract slopes (coefficients for Date_numeric)
slope_wind_speed <- coef(lm_wind_speed)["Date_numeric"]
slope_wtmp <- coef(lm_wtmp)["Date_numeric"]
slope_atmp <- coef(lm_atmp)["Date_numeric"]
slope_bar <- coef(lm_bar)["Date_numeric"]

# Print the slopes
cat("Slope of Atmospheric Temperature Trend:", slope_atmp, "\n")
```

```
## Slope of Atmospheric Temperature Trend: 6.327675e-05
```

```
cat("Slope of Water Temperature Trend:", slope_wtmp, "\n")
```

```
## Slope of Water Temperature Trend: 1.308733e-05
```

```r
cat("Slope of Barometric Pressure Trend:", slope_bar, "\n")
```

```
## Slope of Barometric Pressure Trend: -0.002285527
```

```r
cat("Slope of Wind Speed Trend:", slope_wind_speed, "\n")
```

```
## Slope of Wind Speed Trend: 6.097378e-05
```

To see the effects of climate change, I used four different variables which were Atmospheric temperature, water temperature, barometric pressure, and wind speed. These were variables that I thought would represent climate changes, and hence plotted a time series plot. I also drew a trend line for each graphs to see if there were any increase or decrease in total. As you can see from the slopes, the atmospheric trend as well as water temperature and wind speed tredn tends to increase by a small amount. However, what looks like a small amount of increase can actually be a very big increase in the climate in general. Hence, I tried to divide the data into two, first half and second half, and see the differences in the mean.

```r
library(data.table)
library(ggplot2)

# Filter out rows with missing values in key climate variables
climate_data <- combined_data[, .(Date, WTMP, ATMP, BAR, WSPD)]
climate_data <- climate_data[!is.na(WTMP) & !is.na(ATMP) & !is.na(BAR) & !is.na(WSPD)]

# Ensure the date column is properly formatted
climate_data$Date <- as.Date(climate_data$Date)

# Split the data into two halves
mid_date <- climate_data$Date[round(nrow(climate_data) / 2)]

first_half <- climate_data[Date <= mid_date]
second_half <- climate_data[Date > mid_date]

# Calculate the mean for each variable in both halves
means_first_half <- first_half[, lapply(.SD, mean, na.rm = TRUE), .SDcols = c("WTMP", "ATMP", "BAR", "WS
means_second_half <- second_half[, lapply(.SD, mean, na.rm = TRUE), .SDcols = c("WTMP", "ATMP", "BAR",

# Combine the results for comparison
mean_comparison <- data.table(
  Variable = c("Water Temperature (WTMP)", "Atmospheric Temperature (ATMP)", "Barometric Pressure (BAR)
  First_Half_Mean = as.numeric(means_first_half),
  Second_Half_Mean = as.numeric(means_second_half),
  Difference = as.numeric(means_second_half) - as.numeric(means_first_half)
)

# Display the comparison
print(mean_comparison)
```

```
##                          Variable First_Half_Mean Second_Half_Mean  Difference
##                            <char>           <num>            <num>       <num>
## 1:       Water Temperature (WTMP)        9.294068         8.433903 -0.86016433
## 2: Atmospheric Temperature (ATMP)        8.630688         7.680865 -0.94982312
## 3:      Barometric Pressure (BAR)     1021.091825      1015.135839 -5.95598538
## 4:              Wind Speed (WSPD)        6.164029         6.206083  0.04205404
```

As you can see from the mean, there has been some significant changes in the temperature. For both water and atmospheric temperature, there has been almost 1 degree difference. However, unlike the slope showing an increase, there has been a decrease in the temperatures. This means that the mean could have been affected by in what weather there were missing data. If there were more missing data during the summer for temperatures, the temperatures could be decreasing while the slope still tends to increase.

d.

```
# Load necessary libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```
# Load the rainfall dataset
rainfall_data <- read.csv("Rainfall.csv")

# Check the structure of the dataset to verify column names
str(rainfall_data)
```

```
## 'data.frame':    31714 obs. of  6 variables:
##  $ STATION         : chr  "COOP:190770" "COOP:190770" "COOP:190770" "COOP:190770" ...
##  $ STATION_NAME    : chr  "BOSTON LOGAN INTERNATIONAL AIRPORT MA US" "BOSTON LOGAN INTERNATIONAL AIR
##  $ DATE            : chr  "19850101 01:00" "19850101 09:00" "19850101 10:00" "19850101 11:00" ...
##  $ HPCP            : num  0 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ Measurement.Flag: chr  "g" " " " " " " " ...
##  $ Quality.Flag    : logi  NA NA NA NA NA NA ...
```

```
# Ensure the DATE column exists and is in proper format
# If DATE is in a different format, adjust the format string accordingly
rainfall_data$DATE <- as.Date(rainfall_data$DATE, format = "%Y-%m-%d")

# Summarize rainfall data: Total, Average, and Maximum Rainfall
rainfall_summary <- rainfall_data %>%
  summarise(
    Total_Rainfall = sum(HPCP, na.rm = TRUE),    # Sum of rainfall
    Average_Rainfall = mean(HPCP, na.rm = TRUE), # Average rainfall
    Max_Rainfall = max(HPCP, na.rm = TRUE)       # Maximum rainfall
  )

# Print the rainfall summary
print(rainfall_summary)
```

```
##   Total_Rainfall Average_Rainfall Max_Rainfall
## 1        1228.87        0.0387485         2.03
```

From the data frame, the data on total rainfall, average rainfall, and max rainfall is shown.

```r
# Load necessary libraries
library(dplyr)
library(ggplot2)

# Load the dataset (make sure to replace with your actual file path)
rainfall_data <- read.csv("Rainfall.csv")

# Convert DATE to proper datetime format (assuming format is YYYYMMDD HH:MM)
rainfall_data$DATE <- as.POSIXct(rainfall_data$DATE, format = "%Y%m%d %H:%M")

# Summarize the dataset to check for missing or incorrect values
summary(rainfall_data)
```

```
##    STATION          STATION_NAME           DATE
##  Length:31714       Length:31714       Min.   :1985-01-01 01:00:00.000
##  Class :character   Class :character   1st Qu.:1995-12-16 21:45:00.000
##  Mode  :character   Mode  :character   Median :2002-04-30 01:30:00.000
##                                        Mean   :2001-06-17 14:15:59.378
##                                        3rd Qu.:2008-02-12 23:15:00.000
##                                        Max.   :2013-12-31 21:00:00.000
##                                        NA's   :6
##       HPCP         Measurement.Flag   Quality.Flag
##  Min.   :0.00000   Length:31714       Mode:logical
##  1st Qu.:0.00000   Class :character   NA's:31714
##  Median :0.01000   Mode  :character
##  Mean   :0.03875
##  3rd Qu.:0.04000
##  Max.   :2.03000
##
```

```r
# Check for negative or unusually large values in the rainfall data (HPCP)
rainfall_data <- rainfall_data %>%
  filter(HPCP >= 0 & HPCP <= 500)  # Assuming rainfall should be between 0 and 500 mm

# Aggregate total rainfall by day, month, and year for analysis
rainfall_data_daily <- rainfall_data %>%
  group_by(Day = as.Date(DATE)) %>%
  summarise(Daily_Rainfall = sum(HPCP, na.rm = TRUE))

rainfall_data_monthly <- rainfall_data %>%
  group_by(Month = format(DATE, "%Y-%m")) %>%
  summarise(Monthly_Rainfall = sum(HPCP, na.rm = TRUE))

# Fix the Month format for ggplot (convert to date format with first day of the month)
rainfall_data_monthly$Month <- as.Date(paste0(rainfall_data_monthly$Month, "-01"), format = "%Y-%m-%d")

rainfall_data_yearly <- rainfall_data %>%
  group_by(Year = format(DATE, "%Y")) %>%
```

```r
  summarise(Yearly_Rainfall = sum(HPCP, na.rm = TRUE))

# View the summarized data
print(head(rainfall_data_daily))
```

```
## # A tibble: 6 x 2
##   Day        Daily_Rainfall
##   <date>              <dbl>
## 1 1985-01-01           0.1
## 2 1985-01-02           0.19
## 3 1985-01-05           0.09
## 4 1985-01-07           0.03
## 5 1985-01-08           0.34
## 6 1985-01-17           0.1
```

```r
print(head(rainfall_data_monthly))
```

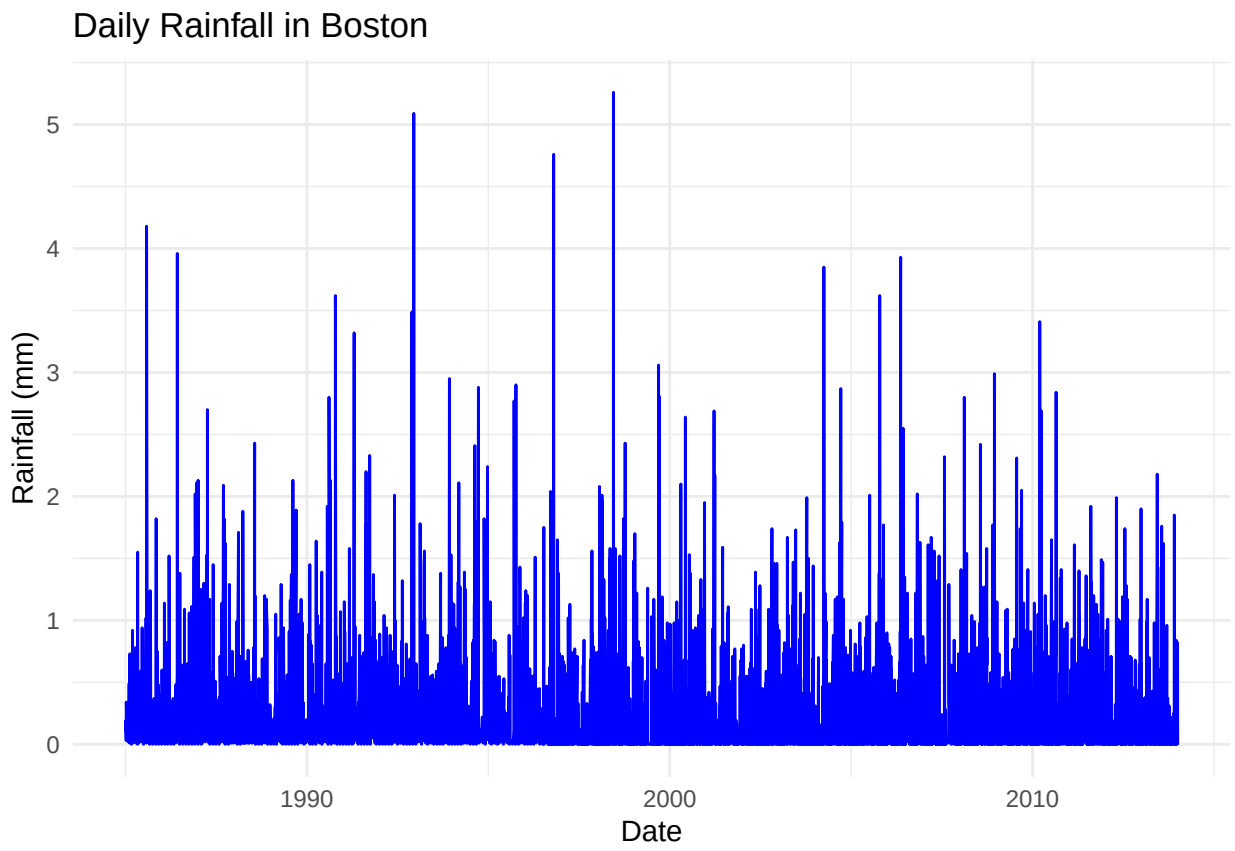
```
## # A tibble: 6 x 2
##   Month      Monthly_Rainfall
##   <date>                <dbl>
## 1 1985-01-01             1.12
## 2 1985-02-01             1.83
## 3 1985-03-01             2.26
## 4 1985-04-01             1.65
## 5 1985-05-01             3.36
## 6 1985-06-01             3.94
```

```r
print(head(rainfall_data_yearly))
```

```
## # A tibble: 6 x 2
##   Year  Yearly_Rainfall
##   <chr>           <dbl>
## 1 1985             36.6
## 2 1986             44.3
## 3 1987             45.2
## 4 1988             34.8
## 5 1989             42.4
## 6 1990             46.5
```

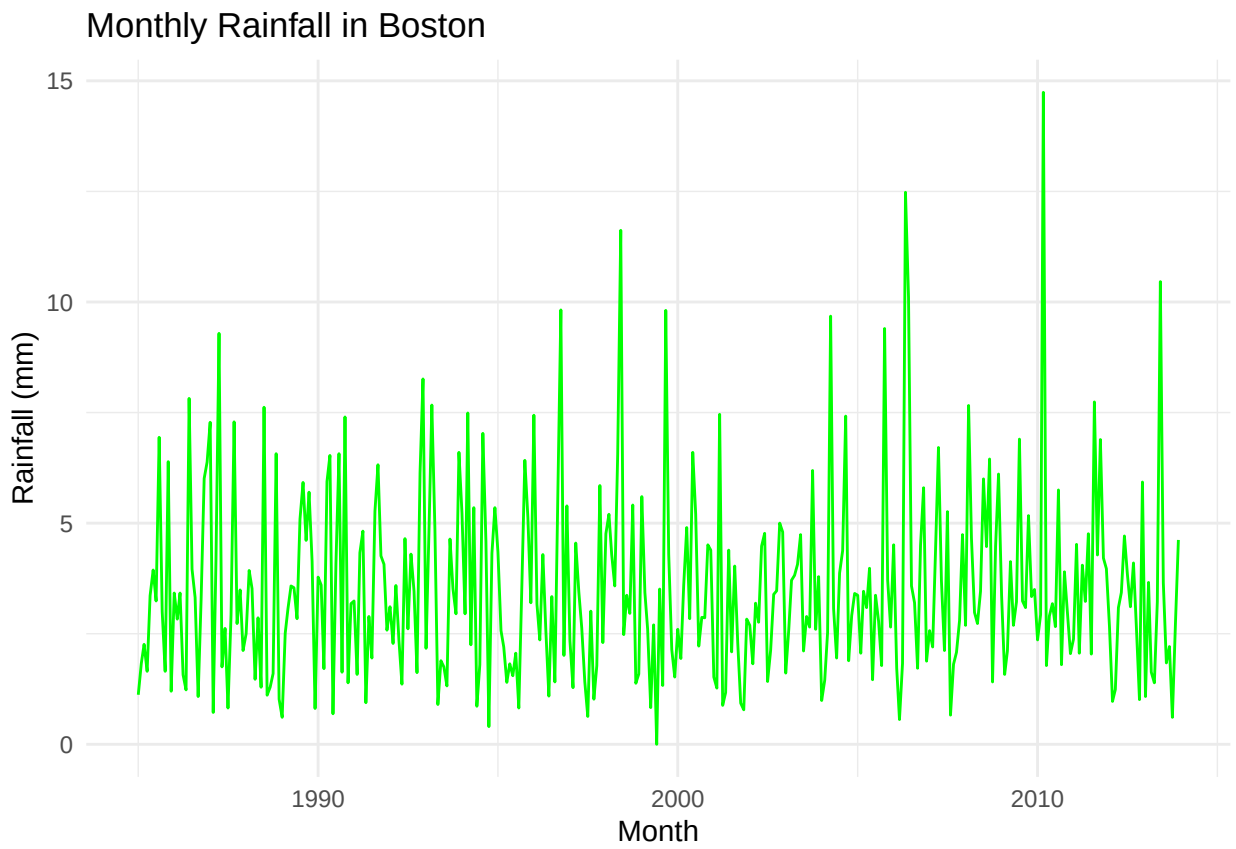
```r
# Plot rainfall patterns over time
# Daily Rainfall
ggplot(rainfall_data_daily, aes(x = Day, y = Daily_Rainfall)) +
  geom_line(color = "blue") +
  labs(title = "Daily Rainfall in Boston", x = "Date", y = "Rainfall (mm)") +
  theme_minimal()
```

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



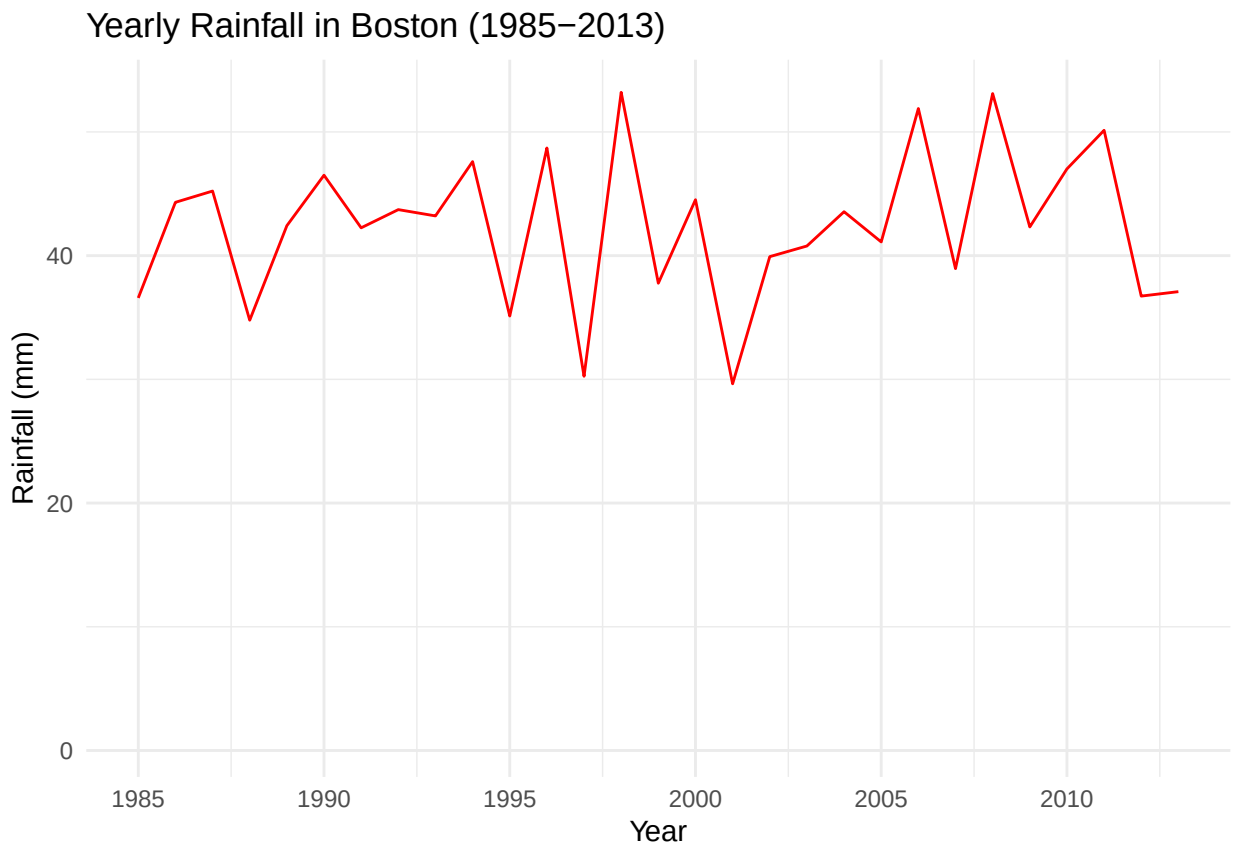

```
# Monthly Rainfall (now with proper Date format)
ggplot(rainfall_data_monthly, aes(x = Month, y = Monthly_Rainfall)) +
  geom_line(color = "green") +
  labs(title = "Monthly Rainfall in Boston", x = "Month", y = "Rainfall (mm)") +
  theme_minimal()
```

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




```
# Yearly Rainfall
ggplot(rainfall_data_yearly, aes(x = as.numeric(Year), y = Yearly_Rainfall)) +
  geom_line(color = "red") +
  labs(title = "Yearly Rainfall in Boston (1985-2013)", x = "Year", y = "Rainfall (mm)") +
  theme_minimal()
```

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



With the analysis above, I chose to find the daily, monthly, and yearly rainfall. This shows a vague understanding of what the rain looks like in Boston over the years. I have also created a chart to see the numeric values as well as the graph for the trend. As you can see, here has been increasing rainfall in boston, when looking at the graph.

```
library(data.table)

# Load rainfall data
rainfall_data <- fread("Rainfall.csv")

# Load weather data
weather_data <- fread("buoy_44013_data_1985_2023.csv")

# Convert the DATE column in rainfall data to proper date format
rainfall_data$DATE <- as.Date(rainfall_data$DATE, format = "%Y%m%d")

# Ensure the Date column in weather data is in date format
weather_data$Date <- as.Date(weather_data$Date)

# Aggregating buoy data by Date
daily_weather <- weather_data[, .(
  Avg_Barometric_Pressure = mean(BAR, na.rm = TRUE),
  Avg_Wind_Speed = mean(WSPD, na.rm = TRUE),
  Avg_Atmospheric_Temperature = mean(ATMP, na.rm = TRUE)
), by = .(Date)]

# Merge the rainfall data with the weather data by the DATE column
```

```r
merged_data <- merge(rainfall_data, daily_weather, by.x = "DATE", by.y = "Date", all.x = TRUE)

# Check the merged data
head(merged_data)
```

```
## Key: <DATE>
##          DATE     STATION                              STATION_NAME  HPCP
##        <Date>      <char>                                    <char> <num>
## 1: 1985-01-01 COOP:190770 BOSTON LOGAN INTERNATIONAL AIRPORT MA US  0.00
## 2: 1985-01-01 COOP:190770 BOSTON LOGAN INTERNATIONAL AIRPORT MA US  0.01
## 3: 1985-01-01 COOP:190770 BOSTON LOGAN INTERNATIONAL AIRPORT MA US  0.01
## 4: 1985-01-01 COOP:190770 BOSTON LOGAN INTERNATIONAL AIRPORT MA US  0.01
## 5: 1985-01-01 COOP:190770 BOSTON LOGAN INTERNATIONAL AIRPORT MA US  0.01
## 6: 1985-01-01 COOP:190770 BOSTON LOGAN INTERNATIONAL AIRPORT MA US  0.01
##    Measurement Flag Quality Flag Avg_Barometric_Pressure Avg_Wind_Speed
##              <char>       <lgcl>                   <num>          <num>
## 1:                g           NA                1026.054       5.916667
## 2:                            NA                1026.054       5.916667
## 3:                            NA                1026.054       5.916667
## 4:                            NA                1026.054       5.916667
## 5:                            NA                1026.054       5.916667
## 6:                            NA                1026.054       5.916667
##    Avg_Atmospheric_Temperature
##                          <num>
## 1:                    5.820833
## 2:                    5.820833
## 3:                    5.820833
## 4:                    5.820833
## 5:                    5.820833
## 6:                    5.820833
```

```r
# Build a linear model to predict rainfall based on weather conditions
model <- lm(HPCP ~ Avg_Barometric_Pressure + Avg_Wind_Speed + Avg_Atmospheric_Temperature, data = merge

# Summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = HPCP ~ Avg_Barometric_Pressure + Avg_Wind_Speed +
##     Avg_Atmospheric_Temperature, data = merged_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15095 -0.04095 -0.02917  0.00876  1.97720
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  7.625e-02  4.376e-03  17.424  < 2e-16 ***
## Avg_Barometric_Pressure     -2.784e-05  4.478e-06  -6.216 5.27e-10 ***
## Avg_Wind_Speed               1.216e-04  3.652e-05   3.328 0.000877 ***
## Avg_Atmospheric_Temperature  2.240e-04  3.816e-05   5.872 4.44e-09 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0829 on 11012 degrees of freedom
##   (20698 observations deleted due to missingness)
## Multiple R-squared:  0.004508,   Adjusted R-squared:  0.004237
## F-statistic: 16.62 on 3 and 11012 DF,  p-value: 9.004e-11
```

```
# Make predictions (if needed)
predictions <- predict(model, newdata = merged_data)

# Evaluate model performance
library(Metrics)
mae_value <- mae(merged_data$HPCP, predictions)
rmse_value <- rmse(merged_data$HPCP, predictions)

cat("Mean Absolute Error (MAE):", mae_value, "\n")
```

```
## Mean Absolute Error (MAE): NA
```

```
cat("Root Mean Square Error (RMSE):", rmse_value, "\n")
```

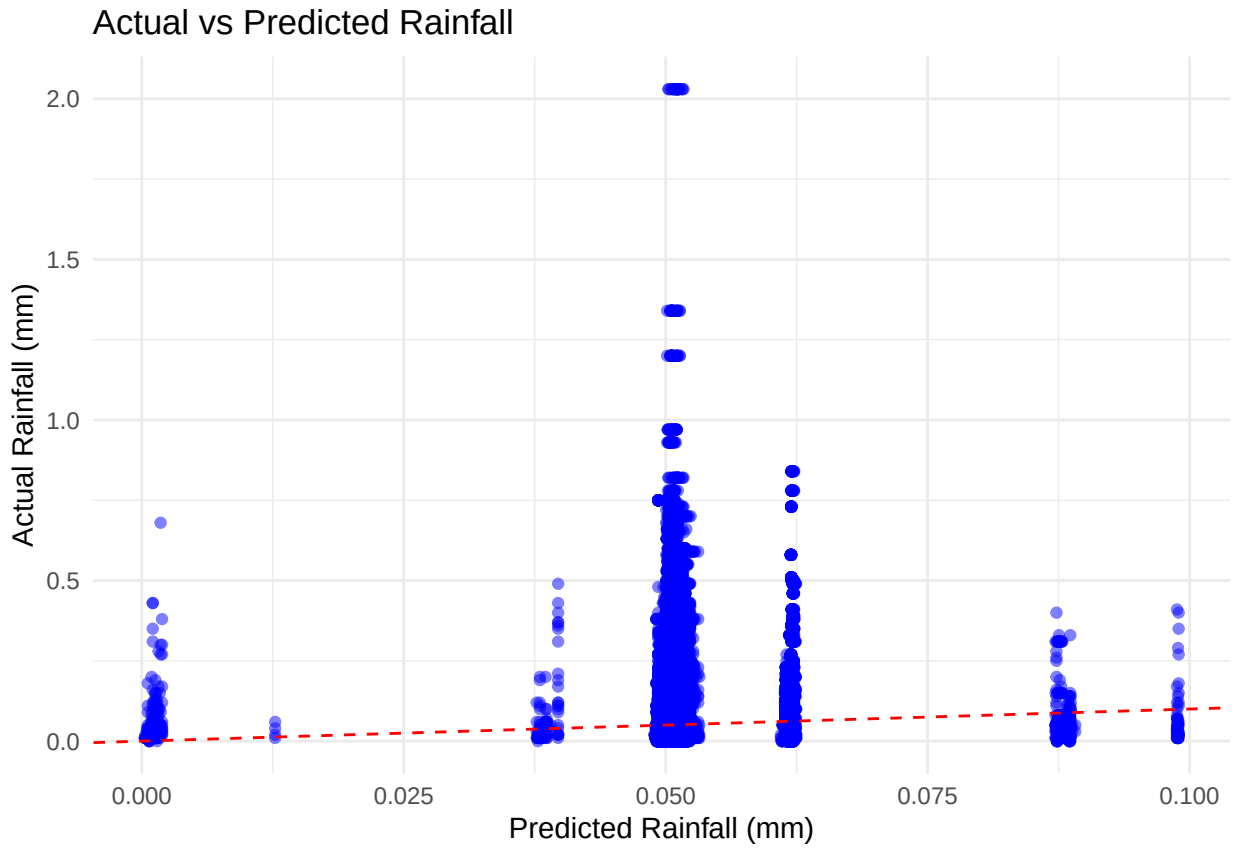
```
## Root Mean Square Error (RMSE): NA
```

```
# Ensure weather_data and rainfall_data are correctly loaded

# Merge the two datasets by Date if the date columns match in both datasets
weather_data$Date <- as.Date(weather_data$Date)  # Convert to Date if necessary
rainfall_data$DATE <- as.Date(rainfall_data$DATE, format = "%Y-%m-%d")

# Merge the data
merged_data <- merge(weather_data, rainfall_data, by.x = "Date", by.y = "DATE")

# Simple regression model using weather variables to predict Rainfall
model <- lm(HPCP ~ BAR + WSPD + ATMP, data = merged_data)

# Summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = HPCP ~ BAR + WSPD + ATMP, data = merged_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.08896 -0.04064 -0.03017  0.00917  1.97974
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.499e-02  4.184e-04  131.43   <2e-16 ***
## BAR         -5.517e-06  3.966e-07  -13.91   <2e-16 ***
## WSPD         1.209e-04  7.333e-06   16.48   <2e-16 ***
```

```
## ATMP         3.754e-05  3.450e-06   10.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08297 on 262780 degrees of freedom
## Multiple R-squared:  0.001679,   Adjusted R-squared:  0.001668
## F-statistic: 147.3 on 3 and 262780 DF,  p-value: < 2.2e-16
```

```r
# Predict rainfall using the model
merged_data$predicted_rainfall <- predict(model, newdata = merged_data)

# Plot actual vs predicted rainfall
library(ggplot2)

ggplot(merged_data, aes(x = predicted_rainfall, y = HPCP)) +
  geom_point(color = "blue", alpha = 0.5) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +  # Line y=x
  labs(title = "Actual vs Predicted Rainfall", x = "Predicted Rainfall (mm)", y = "Actual Rainfall (mm)
  theme_minimal()
```



For this model, I used pressure, wind speed, and atmospheric temperature to give out an assumption on how much rain will fall. Even though it does not seem to show a statistically significant outcome, it was a journey for me. I used the data from both of the csv files, using buoy data as my input variables, and rainfall as my output variable. I assumed that there will be some correlation between the rainfall and the changes in pressure, wind speed, and temperature. However, since weather is not easily predictable, the r^2 value or the actual and predicted rainfall did not show a significant result. I splitted the data to use some as predicted and as actual data, which shows the low accuracy of the model. However, as someone

who feels like there should be more interesting areas in weather forecasting, this helped me to think about what could actually make people laugh. Giving a unprecise model, and guessing together with the people watching the weather forcast could be like a lottery!