MA615 HW4

Taha Ababou

2024-09-27

Prerequisites

Loading necessary 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
```

```
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:data.table':
##
##     yearmon, yearqtr
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
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
```

```
library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.32.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to
the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

Part A

a) Your first exercise is to read in the data for all the years from 1985 to 2023. As discussed in class, you don't want to do this manually and will need to figure out a way to do it programmatically. We've given you a skeleton of how to do this for data for one year below. Your task is to adapt this to reading in multiple datasets from all the

years in question. This example code is meant to be a guide and if you think of a better way to read the data in, go for it.

Keep in mind that initially, these datasets did not record units and then started to do so in the line below the column headers. So for some years you will have to skip 1 instead of 2.

In addition to reading in this data, use lubridate to create a proper date column.

Example code provided:

```
file_root<-"https://www.ndbc.noaa.gov/view_text_file.php?filename=44013h"
year<-"2023"
tail<- ".txt.gz&dir=data/historical/stdmet/"
path<-paste0(file_root,year,tail)
header=scan(path,what= 'character',nlines=1)
buoy<-fread(path,header=FALSE,skip=2)
colnames(buoy)<-header
```

To view the full R script used to complete part A, please view the file `loadBuoy.R`

Breakdown of `loadBuoy.R` solution:

The `loadBuoy.R` script automates the process of downloading and standardizing meteorological data from NOAA's buoy 44013 for the years 1985 to 2023. The script uses a loop to fetch and process data for each year from a constructed URL. Each dataset is read using the fread function from the data.table package, ensuring that missing columns in earlier years are filled with NA to maintain consistency across all years.

Data from each year is appended to a list and then combined into a single dataframe using `rbindlist.` The combined data is reordered to match a pre-defined list of column names and missing columns are added where necessary.

The script also creates a properly formatted date column using the ymd_hm function from the lubridate package. Finally, the script outputs the standardized data to a CSV file for further analysis or storage.

```
source("loadBuoy.R")
```

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

Part B

b) Your next exercise is to identify and deal with the null data in the dataset. Recall from class that for WDIR and some other variables these showed up as 999 in the dataset. Convert them to NA's. Is it always appropriate to convert missing/null data to NA's? When might it not be? Analyze the pattern of NA's. Do you spot any patterns in the way/dates that these are distributed?

Converting Missing/Null Data to NA's

"Is it always appropriate to convert missing/null data to NA's? When might it not be?"

It's typically appropriate to convert well-known placeholders like '999' (often used to denote missing or unmeasurable data in meteorological datasets) to NA to facilitate analyses that correctly handle missing values, such as averages or regression models. However, there are situations where converting such placeholders to NA might not be advisable:

- If '999' or similar placeholders carry additional meaning—like indicating that data couldn't be collected due to specific conditions—removing these can obscure meaningful interpretations.
- Some analytical procedures might require a complete dataset without NA values, prompting different approaches like data imputation instead of simple placeholder replacement.
- Converting to NA indiscriminately can impact the results of analyses where the distinction between 'not available' and 'not applicable' matters.

```
# Converting '999' to NA
combined_data[, (names(combined_data)) := lapply(.SD, function(x) replace(x, x == 999, N
A))]
```

Analyzing Patterns in NA's

"Analyze the pattern of NA's. Do you spot any patterns in the way/dates that these are distributed?"

```
# Analyzing the distribution of NA values by year and month
na_distribution <- combined_data[, lapply(.SD, function(x) sum(is.na(x))), by = .(Year,
Month = MM)]

# na_distribution

summary(na_distribution)
```

```
##       Year          Month              DD          hh          mm
##  Min.   :1985   Min.   : 1.000   Min.   :0   Min.   :0   Min.   :  0.0
##  1st Qu.:1994   1st Qu.: 3.250   1st Qu.:0   1st Qu.:0   1st Qu.:  0.0
##  Median :2004   Median : 6.000   Median :0   Median :0   Median :319.5
##  Mean   :2004   Mean   : 6.478   Mean   :0   Mean   :0   Mean   :362.7
##  3rd Qu.:2014   3rd Qu.: 9.000   3rd Qu.:0   3rd Qu.:0   3rd Qu.:736.0
##  Max.   :2023   Max.   :12.000   Max.   :0   Max.   :0   Max.   :840.0
##        WD              WSPD        GST         WVHT        DPD         APD
##  Min.   :   0.0   Min.   :0   Min.   :0   Min.   :0   Min.   :0   Min.   :0
##  1st Qu.:   1.0   1st Qu.:0   1st Qu.:0   1st Qu.:0   1st Qu.:0   1st Qu.:0
##  Median :  37.5   Median :0   Median :0   Median :0   Median :0   Median :0
##  Mean   : 650.9   Mean   :0   Mean   :0   Mean   :0   Mean   :0   Mean   :0
##  3rd Qu.: 737.8   3rd Qu.:0   3rd Qu.:0   3rd Qu.:0   3rd Qu.:0   3rd Qu.:0
##  Max.   :4465.0   Max.   :0   Max.   :0   Max.   :0   Max.   :0   Max.   :0
##       MWD               BAR              ATMP             WTMP
##  Min.   :   0.00   Min.   :   0.0   Min.   :   0.0   Min.   :  0.00
##  1st Qu.:  40.25   1st Qu.:   0.0   1st Qu.:   0.0   1st Qu.:  0.00
##  Median : 719.00   Median :   1.0   Median :   0.0   Median :  1.00
##  Mean   : 716.51   Mean   : 617.4   Mean   : 226.3   Mean   : 29.04
##  3rd Qu.: 743.00   3rd Qu.: 733.0   3rd Qu.:   3.0   3rd Qu.:  5.00
##  Max.   :3947.00   Max.   :4465.0   Max.   :4464.0   Max.   :742.00
##       DEWP             VIS         TIDE             WDIR             PRES
##  Min.   :   0.0   Min.   :0   Min.   :  0.0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.:   1.0   1st Qu.:0   1st Qu.:  0.0   1st Qu.:   0.0   1st Qu.:  0.0
##  Median : 484.5   Median :0   Median :  0.0   Median : 710.0   Median :690.0
##  Mean   : 558.6   Mean   :0   Mean   :285.5   Mean   : 463.3   Mean   :401.4
##  3rd Qu.: 737.0   3rd Qu.:0   3rd Qu.:720.0   3rd Qu.: 740.0   3rd Qu.:738.0
##  Max.   :4464.0   Max.   :0   Max.   :840.0   Max.   :4463.0   Max.   :840.0
##       Date         datetime
##  Min.   :   1   Min.   :0
##  1st Qu.: 719   1st Qu.:0
##  Median : 737   Median :0
##  Mean   :1018   Mean   :0
##  3rd Qu.: 744   3rd Qu.:0
##  Max.   :4465   Max.   :0
```

Analysis of NA Patterns in the Dataset:

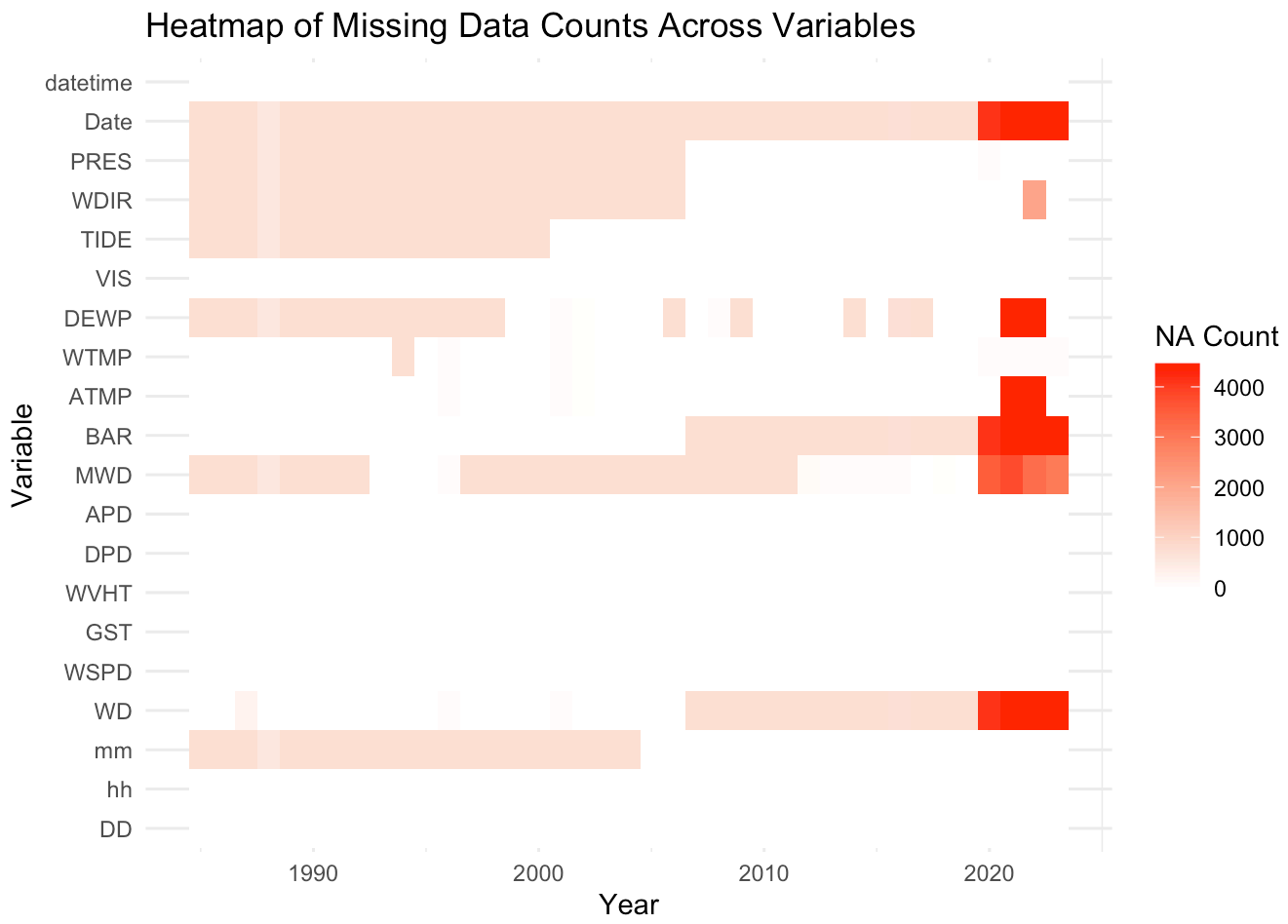
- **Overall Distribution:** The na_distribution summary provides insight into the frequency of NA values across different columns. Notably, the number of NAs varies significantly between columns like MWD (mean: 716.51, max: 3947) and columns like WDIR and TIDE which show higher numbers of missing values in some years.
- **WDIR and MWD:** The WDIR (Wind Direction) and MWD (Mean Wave Direction) columns show a relatively high number of NAs. The mean for WDIR is around 463.3, while the max is 4463, indicating significant missing data in certain years. Similarly, MWD has a mean of 716.51 and a max of 3947. This could suggest instrumentation issues or environmental factors leading to the inability to record wind and wave directions in specific periods.

- **Pressure (BAR/PRES):** The BAR (Barometric Pressure) column has a mean of 617.4 and a max of 4465 missing values. This pattern may indicate sensor issues or interruptions in data collection in specific years, especially since this is a core environmental metric typically expected to have fewer gaps.
- **Air and Water Temperature (ATMP/WTMP):** ATMP (Air Temperature) and WTMP (Water Temperature) have relatively fewer NAs on average, but their maximum values (4464 and 742, respectively) suggest that there were periods where temperature sensors were not recording correctly. This could indicate seasonal changes in data collection or technical failures.
- **TIDE:** The TIDE column shows a moderate number of missing values, with a mean of 285.5 and a max of 840, indicating possible gaps in data for certain months or years. Since not all years include tide data, these missing values could also be due to years where tide measurements weren't recorded at all.
- **Temporal Patterns:** Looking at the quartile breakdowns (1st quartile, median, 3rd quartile), most columns show relatively consistent data for large portions of the year, but with notable spikes in missing data. For example, the MWD and WDIR columns have significantly more NAs in some periods, suggesting either a prolonged sensor failure or specific external factors that limited data collection in those months or years.
- **Pressure and Wind Direction Missingness:** The high NA counts in the BAR (Pressure) and WDIR (Wind Direction) columns could indicate patterns related to instrument malfunctions during specific weather conditions or seasons. For example, extreme weather events may lead to sensor failures or temporary halts in data collection, which could explain high NA counts for these columns.

Visualizing the missing data

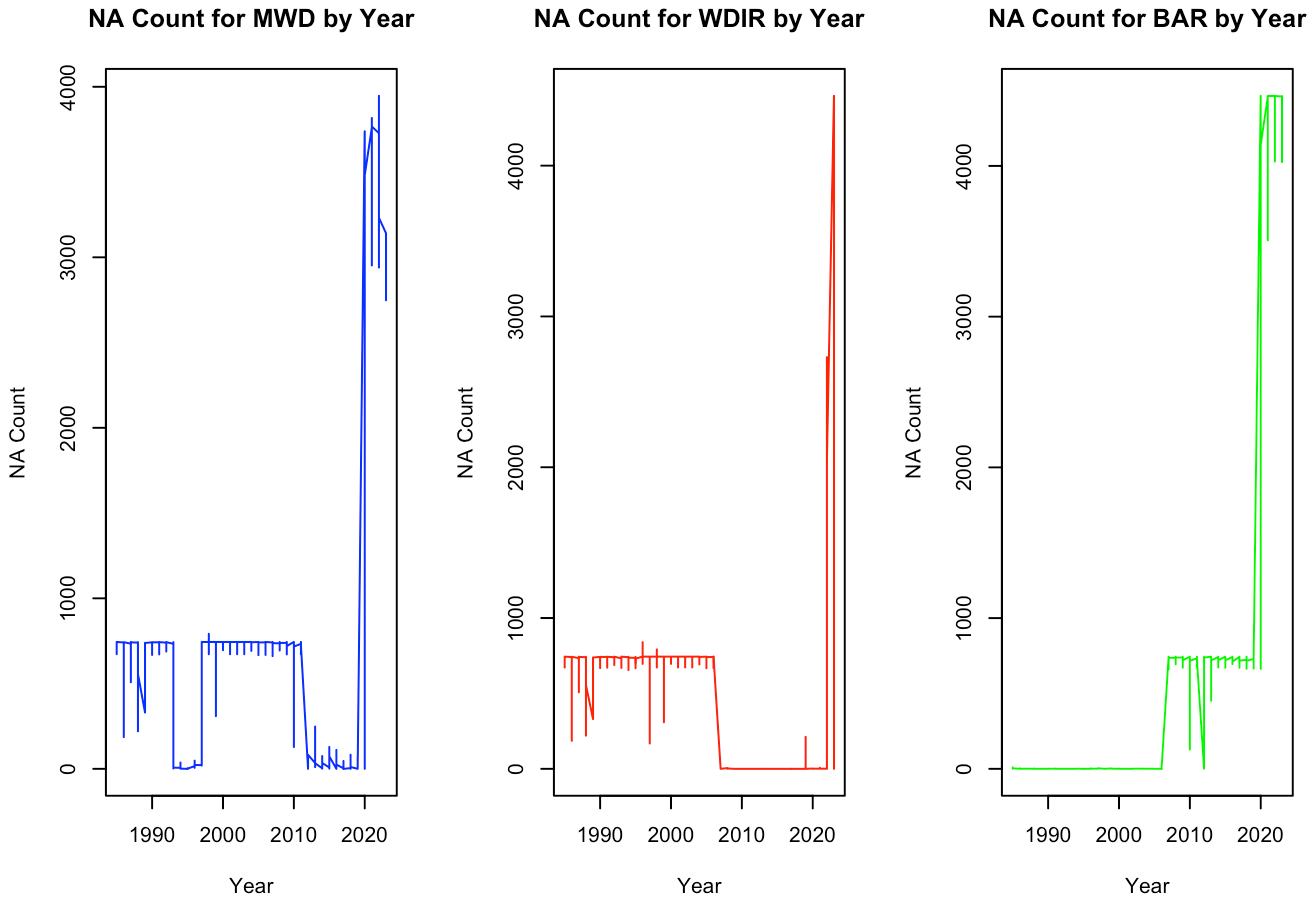
Let's use a **heatmap** to visualize the missing data. The heatmap will show patterns of NAs for multiple variables at once, giving a clear overview of which variables have missing data over time.

```
# Reshape data for heatmap visualization
na_heatmap <- combined_data[, lapply(.SD, function(x) sum(is.na(x))), by = .(Year, MM)]
na_heatmap_melt <- melt(na_heatmap, id.vars = c("Year", "MM"))

# Heatmap for NA counts across variables
ggplot(na_heatmap_melt, aes(x = Year, y = variable, fill = value)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red", na.value = "grey50") +
  labs(title = "Heatmap of Missing Data Counts Across Variables",
       x = "Year", y = "Variable", fill = "NA Count") +
  theme_minimal()
```



The heatmap reveals significant missing data in several variables during the early years (1985-1990), particularly for `WDIR`, `PRES`, and `BAR`, suggesting that these measurements were either not recorded or unavailable during that period. Post-2020, there is a noticeable increase in missing data for variables like `BAR`, `MWD`, and `WDIR`, indicating possible technical issues or disruptions in data collection. The `TIDE` variable also shows large gaps in earlier years, implying that this measurement may not have been consistently recorded until later. These trends suggest potential external factors such as sensor failures, maintenance issues, or administrative disruptions like government shutdowns impacting data availability.

```
# Summarize NA counts by year and month
na_summary <- combined_data[, lapply(.SD, function(x) sum(is.na(x))), by = .(Year, Month
= MM)]

par(mfrow = c(1, 3))  # 1 row, 3 columns

# Plot 1: NA Count for Mean Wave Direction (MWD) by Year (using base R plot)
plot(na_summary$Year, na_summary$MWD, type = "l", col = "blue",
     xlab = "Year", ylab = "NA Count", main = "NA Count for MWD by Year")

# Plot 2: NA Count for Wind Direction (WDIR) by Year (using base R plot)
plot(na_summary$Year, na_summary$WDIR, type = "l", col = "red",
     xlab = "Year", ylab = "NA Count", main = "NA Count for WDIR by Year")

# Plot 3: NA Count for Barometric Pressure (BAR) by Year (using base R plot)
plot(na_summary$Year, na_summary$BAR, type = "l", col = "green",
     xlab = "Year", ylab = "NA Count", main = "NA Count for BAR by Year")
```



To better understand the missing data patterns in the buoy dataset, it's helpful to consider external factors such as U.S. government shutdowns and NOAA budget changes. For example, the 1995-1996 government shutdowns lasted a combined 26 days, during which 280,000 federal workers were furloughed. Many non-essential NOAA operations, including research and sensor maintenance, were paused, likely contributing to data gaps in that period. (The HISTORY Channel) (https://www.history.com/news/us-government-shutdowns-facts) (Wikipedia) (https://en.wikipedia.org/wiki/1995%E2%80%931996_United_States_federal_government_shutdowns). Similarly, the longest shutdown in U.S. history, from December 2018 to January 2019, also disrupted NOAA's climate research and environmental monitoring activities. (FOX Weather) (https://www.foxweather.com/lifestyle/shutdown-national-weather-service-forecast-impacts) (Pacifica Wealth) (https://www.pacificawealth.com/impact-1995-96-government-shutdown/)

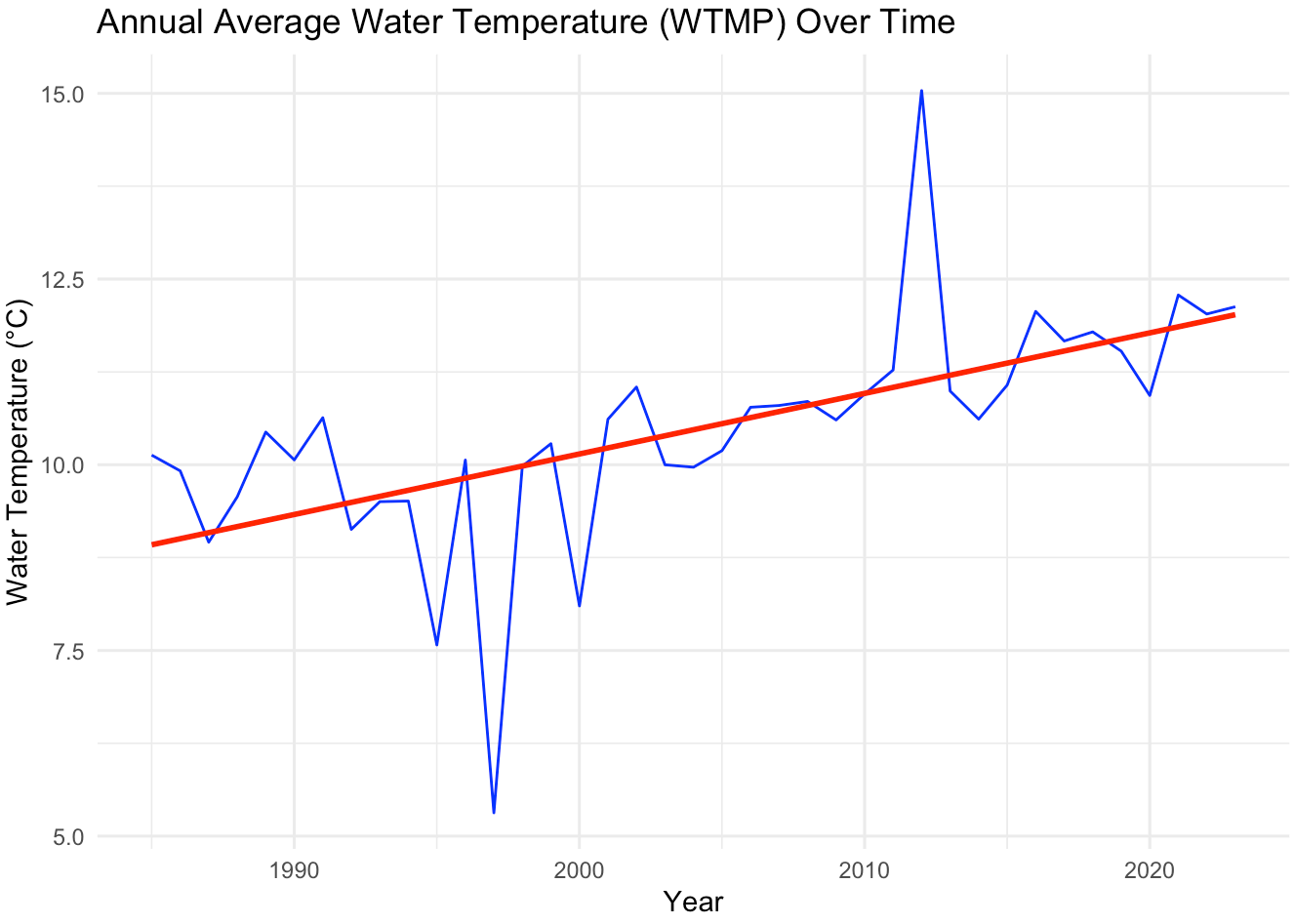
Budget constraints also play a significant role. NOAA has experienced budget fluctuations over the years, particularly during periods of reduced federal funding in the mid-2000s and early 2010s, which could have impacted its ability to maintain buoy sensors and process data continuously. These external events likely correlate with the spikes in missing data observed.

Part C

c) Can you use the Buoy data to see the effects of climate change? Create visualizations to show this and justify your choices. Can you think of statistics you can use to bolster what your plots represent? Calculate these, justify your use of them. Add this code, its output, your answers and visualizations to your pdf.

```
# Calculate annual averages for Water Temperature (WTMP)
annual_avg_wtmp <- combined_data[, .(Avg_WTMP = mean(WTMP, na.rm = TRUE)), by = Year]

# Linear regression to see if there's a trend
wtmp_lm <- lm(Avg_WTMP ~ Year, data = annual_avg_wtmp)

# Plot the annual averages with the linear regression line
ggplot(annual_avg_wtmp, aes(x = Year, y = Avg_WTMP)) +
  geom_line(color = "blue") +
  geom_smooth(method = "lm", col = "red", se = FALSE) +
  labs(title = "Annual Average Water Temperature (WTMP) Over Time",
       x = "Year", y = "Water Temperature (°C)") +
  theme_minimal()
```

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



This graph shows the annual average water temperature (WTMP) over time, with a blue line representing the yearly average and a red line showing the linear regression trend.

Key Observations:

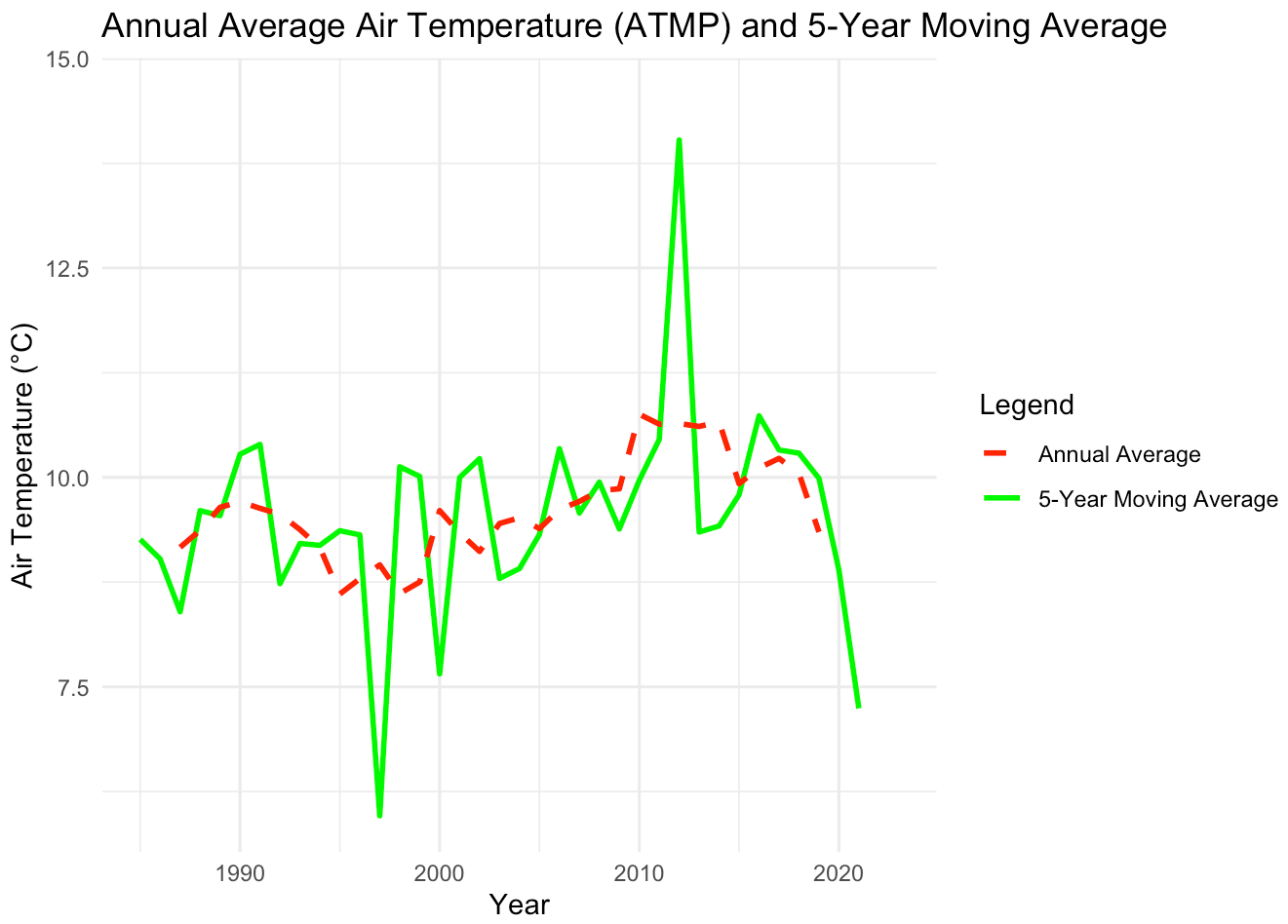
- Upward Trend: The red regression line indicates a clear long-term warming trend in water temperatures from the late 1980s to the present. This suggests that the ocean water temperatures have been gradually increasing over the past few decades, which aligns with broader patterns of ocean warming due to climate

change.

- Short-Term Fluctuations: Despite the overall warming trend, there are notable short-term fluctuations. For example, there is a significant drop in temperature around the year 2000 and a pronounced peak around 2010, similar to the air temperature graph. These variations could be influenced by local weather conditions or oceanic events like El Niño or La Niña.
- Recent Stability: After 2015, the water temperature appears to stabilize somewhat, with less pronounced fluctuations compared to earlier years. However, the general upward trajectory remains clear.

In summary, the graph indicates a clear warming of water temperatures over time, punctuated by short-term variability, which may be linked to climatic events. The overall trend is consistent with the effects of global ocean warming due to climate change.

```
# Calculate annual averages for Air Temperature (ATMP)
annual_avg_atmp <- combined_data[, .(Avg_ATMP = mean(ATMP, na.rm = TRUE)), by = Year]

# Moving average (5 years)
annual_avg_atmp[, Moving_Avg_ATMP := zoo::rollmean(Avg_ATMP, 5, fill = NA)]

# Plot the annual averages and moving average for Air Temperature
ggplot(annual_avg_atmp, aes(x = Year)) +
  geom_line(aes(y = Avg_ATMP, color = "Annual Average"), size = 1) +
  geom_line(aes(y = Moving_Avg_ATMP, color = "5-Year Moving Average"), linetype = "dashe
d", size = 1) +
  labs(title = "Annual Average Air Temperature (ATMP) and 5-Year Moving Average",
       x = "Year", y = "Air Temperature (°C)") +
  scale_color_manual(name = "Legend",
                     values = c("Annual Average" = "green", "5-Year Moving Average" = "r
ed"),
                     labels = c("Annual Average", "5-Year Moving Average")) +
  theme_minimal()
```



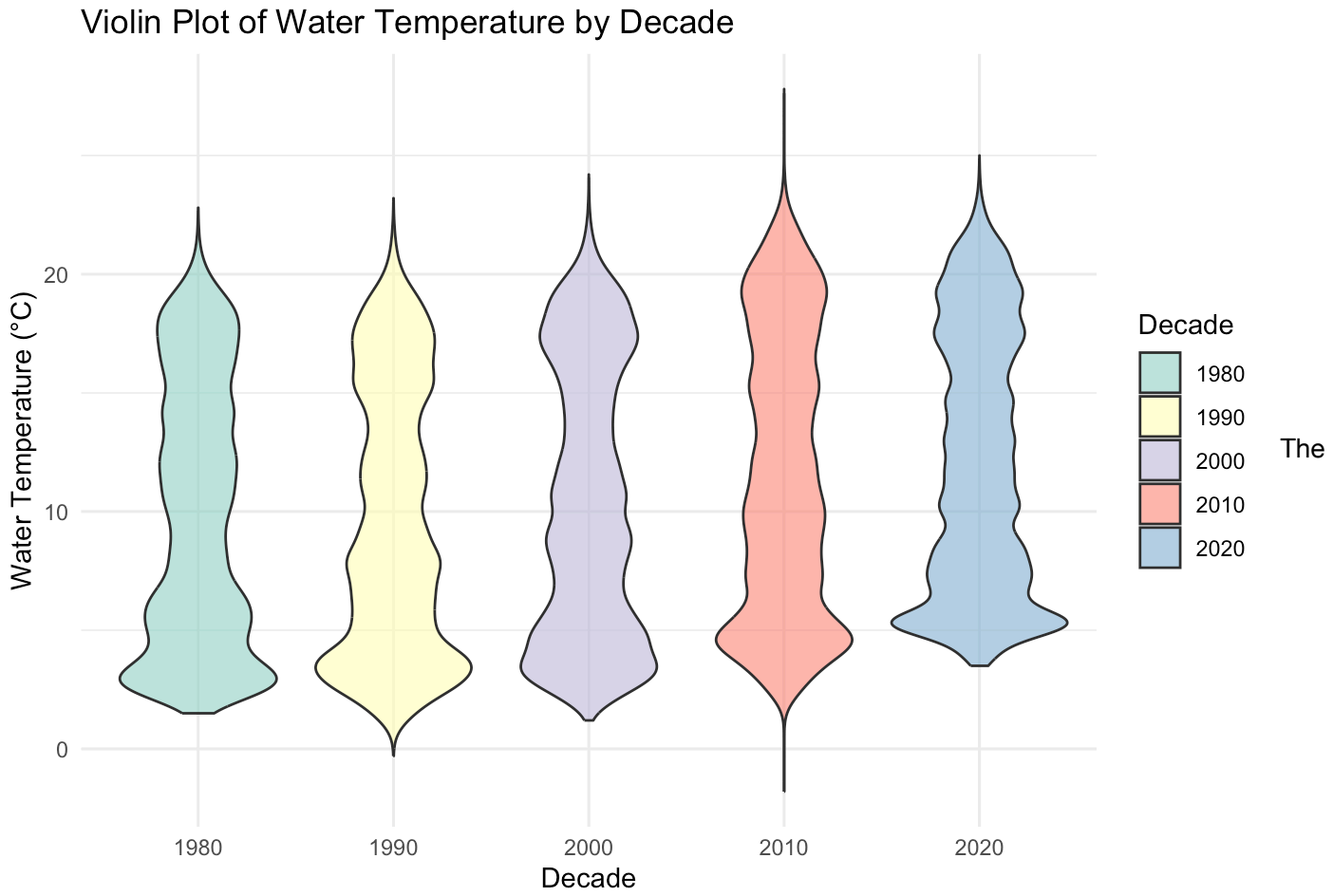
This graph displays the annual average air temperature (ATMP) and its 5-year moving average over time. The green solid line represents the 5-year moving average, while the red dashed line shows the annual average.

Key Observations:

1. The plot shows considerable fluctuations in air temperature over the decades. The moving average helps smooth out these fluctuations, making the overall trend clearer.
   - From the late 1980s to around 2000, air temperatures generally increased, with some noticeable dips.
   - There's a significant spike in temperatures around 2010, where the air temperature sharply increases to nearly 15°C, before falling again.
   - More recently (post-2015), the moving average and annual temperatures decline, indicating a cooling trend.
2. High Variability: The air temperature appears to be quite variable from year to year, with sharp rises and drops throughout the timeline. This could indicate the impact of short-term weather phenomena, such as storms or other atmospheric events.
3. Potential Climate Change Signal: The peak around 2010 might reflect anomalies in temperature that could be associated with broader climate change patterns, such as increased heatwaves or warmer seasons. However, the recent cooling could also be influenced by other local weather patterns or short-term climate variations.

Overall, the graph suggests both short-term fluctuations and long-term trends in air temperature, with a noticeable warming trend until around 2010 followed by a more recent cooling.

```
# Create the Decade column from the Year column
combined_data <- combined_data %>%
  mutate(Decade = floor(Year / 10) * 10)

# Violin plot for Water Temperature (WTMP) by Decade with legend
ggplot(combined_data[!is.na(WTMP)], aes(x = factor(Decade), y = WTMP, fill = factor(Deca
de))) +
  geom_violin(alpha = 0.6) +
  labs(title = "Violin Plot of Water Temperature by Decade",
       x = "Decade", y = "Water Temperature (°C)", fill = "Decade") +
  scale_fill_brewer(palette = "Set3") +  # Adds color for each decade
  theme_minimal()
```



The violin plot shown visualizes the distribution of water temperature (WTMP) by decade. Each “violin” represents the density of water temperature observations within that decade, with the width indicating the density (i.e., where more observations are concentrated).

Key Observations:

1970s to 2020s Temperature Ranges:

- The distribution of water temperatures has remained somewhat consistent across decades, with temperatures typically ranging between 5°C and 20°C.
- The majority of temperature readings cluster around the lower half of the range (~5-15°C), indicating that most water temperatures in this region remain relatively cool.

Changes in Distribution:

- 2000s and 2010s: The violins for these decades appear slightly more spread out, indicating a broader distribution of water temperatures. This could suggest more variability in water temperatures during these decades, which might be related to climatic shifts or increased seasonal variability.
- 2020s: The distribution in the 2020s appears to be slightly narrower again, implying a return to more concentrated water temperatures.

Symmetry and Peaks:

- The violins for most decades show a symmetrical distribution, suggesting that water temperatures tend to cluster around a central value, which may indicate stable ocean conditions.
- The 2010s decade has a slightly more elongated upper portion, which might indicate higher occurrences of warmer water temperatures compared to previous decades.

Potential Climate Change Implications:

- The slight broadening of the distributions in the 2000s and 2010s could reflect increased variability in ocean temperatures, potentially caused by climate change, which may lead to more extreme weather conditions or warmer summers.
- The plot does not immediately suggest dramatic warming, but the increasing variability in temperature ranges could align with broader patterns of climate-driven ocean changes.

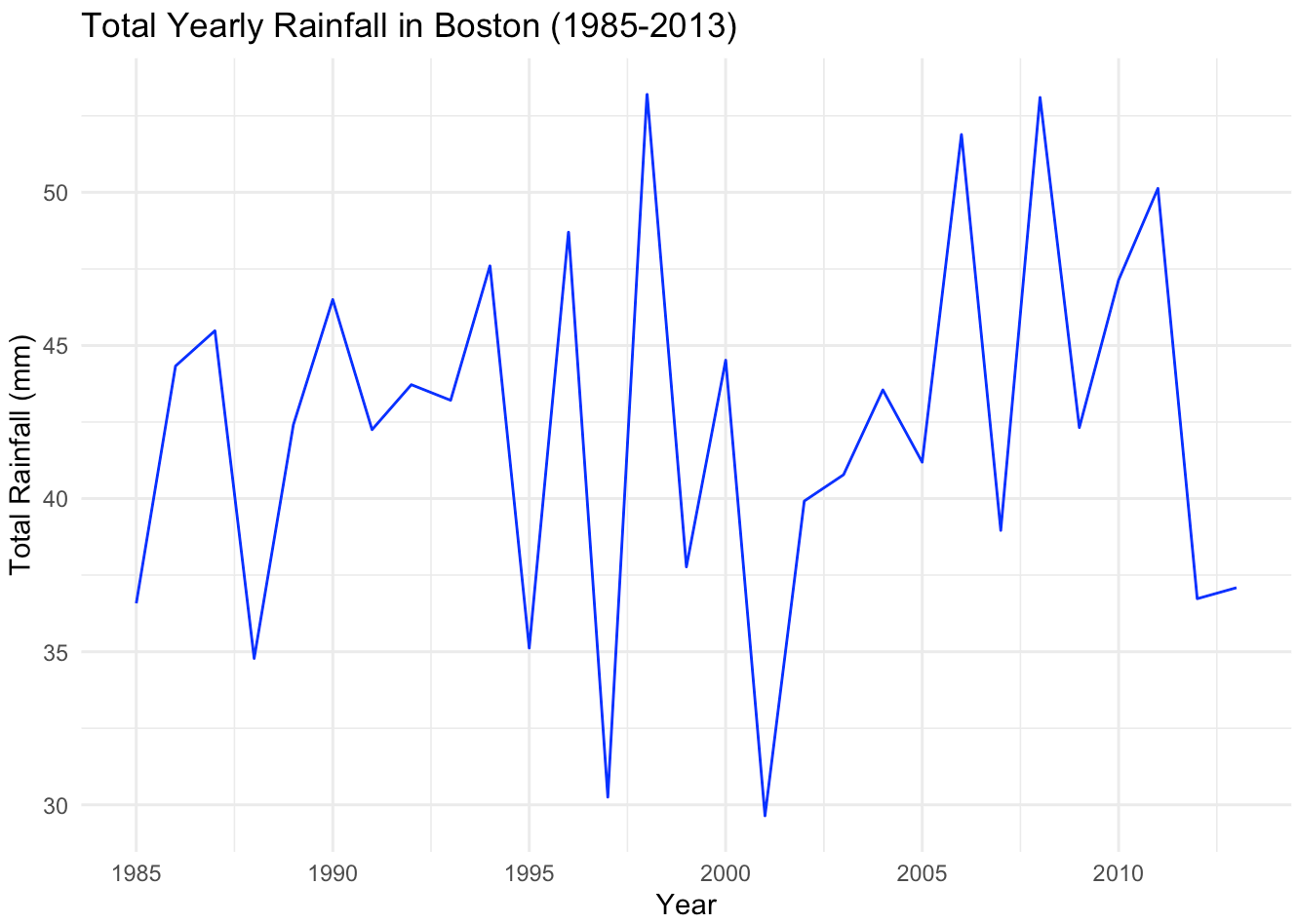
Part D

d) As part of this Homework, you have been given data for rainfall in Boston from 1985 to the end of 2013. Your job for this exercise is to see if you can spot any patterns between rainfall(whether it happens and how much of it there is) and the readings taken by the weather buoy in the same period. There are a number of ways you can do this but the broad steps are: 1) Acquaint yourself with the data. Look at distributions and summary statistics(dplyr is great for coaxing means, averages, counts out of data). 2) Create visualizations. Can you see patterns in the distributions and visualizations? Investigate these with more statistics and visualizations. 3) Try building a very simple model. Explain your choices at every step. Do you come away from this exercise with more sympathy for the weather people on TV who keep getting their forecasts wrong? Structure your response to this question as an exploration of the data and as if it were a report. Show the graphs and outputs.

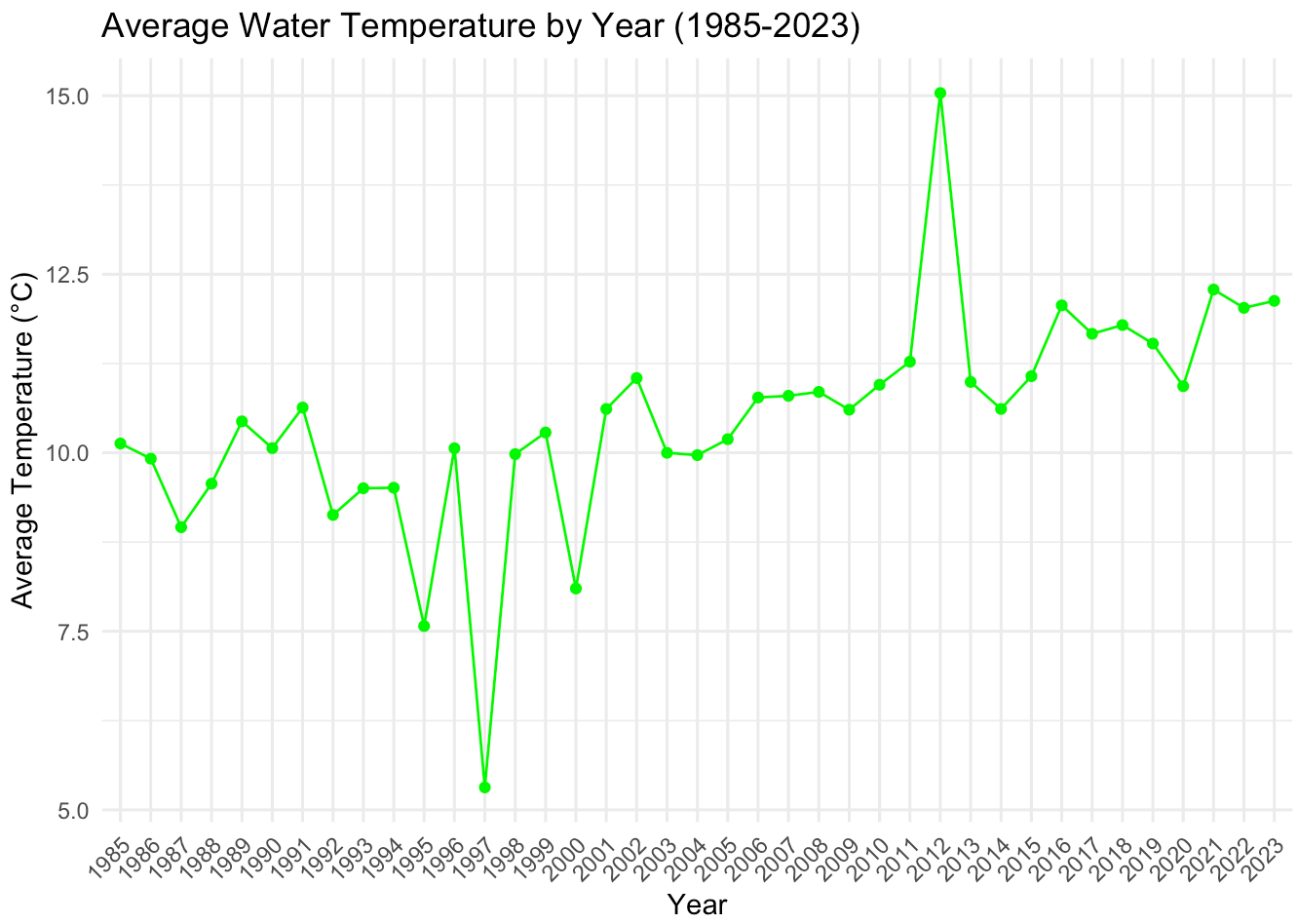
```r
# 1.2. Read the rainfall data (rainfall_data)
rainfall_data <- read.csv("Rainfall.csv")

# 1.3. Convert DATE to datetime in rainfall_data
# The DATE column appears to be in "YYYYMMDD HH:MM" format
rainfall_data <- rainfall_data %>%
  mutate(
    datetime = ymd_hm(DATE)
  )

# Ensure datetime format consistency for buoy data as well
combined_data$datetime <- as.POSIXct(combined_data$datetime)

# Merge the two datasets by datetime
merged_data <- left_join(combined_data, rainfall_data, by = "datetime")

# Filtering out invalid WTMP values
merged_data <- merged_data %>%
  filter(WTMP != 999, WTMP < 100, # Assuming realistic WTMP range below 100°C
         BAR != 9999)


# Check the data after cleaning
# summary(merged_data_clean$WTMP)

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

```
##       MM    DD    hh    mm    WD  WSPD   GST  WVHT   DPD   APD   MWD    BAR
##    <int> <int> <int> <num> <num> <num> <num> <num> <num> <num> <int>  <num>
## 1:     1     1     0    NA    60     4     5    99    99    99    NA 1030.3
## 2:     1     1     1    NA    80     4     5    99    99    99    NA 1030.0
## 3:     1     1     2    NA   100     4     5    99    99    99    NA 1030.1
## 4:     1     1     3    NA   100     4     5    99    99    99    NA 1029.4
## 5:     1     1     4    NA   110     4     5    99    99    99    NA 1028.6
## 6:     1     1     5    NA    90     4     5    99    99    99    NA 1027.8
##     ATMP  WTMP  DEWP   VIS  TIDE  Year  WDIR  PRES   Date            datetime
##    <num> <num> <num> <num> <num> <num> <int> <num> <POSc>              <POSc>
## 1:   4.7   6.7    NA    99    NA  1985    NA    NA   <NA> 1985-01-01 00:00:00
## 2:   5.1   6.7    NA    99    NA  1985    NA    NA   <NA> 1985-01-01 01:00:00
## 3:   5.6   6.6    NA    99    NA  1985    NA    NA   <NA> 1985-01-01 02:00:00
## 4:   5.8   6.7    NA    99    NA  1985    NA    NA   <NA> 1985-01-01 03:00:00
## 5:   5.8   6.7    NA    99    NA  1985    NA    NA   <NA> 1985-01-01 04:00:00
## 6:   5.3   6.7    NA    99    NA  1985    NA    NA   <NA> 1985-01-01 05:00:00
##    Decade     STATION                              STATION_NAME           DATE
##     <num>      <char>                                     <char>         <char>
## 1:   1980        <NA>                                       <NA>           <NA>
## 2:   1980 COOP:190770 BOSTON LOGAN INTERNATIONAL AIRPORT MA US 19850101 01:00
## 3:   1980        <NA>                                       <NA>           <NA>
## 4:   1980        <NA>                                       <NA>           <NA>
## 5:   1980        <NA>                                       <NA>           <NA>
## 6:   1980        <NA>                                       <NA>           <NA>
##     HPCP Measurement.Flag Quality.Flag
##    <num>           <char>       <lgcl>
## 1:    NA             <NA>           NA
## 2:     0                g           NA
## 3:    NA             <NA>           NA
## 4:    NA             <NA>           NA
## 5:    NA             <NA>           NA
## 6:    NA             <NA>           NA
```

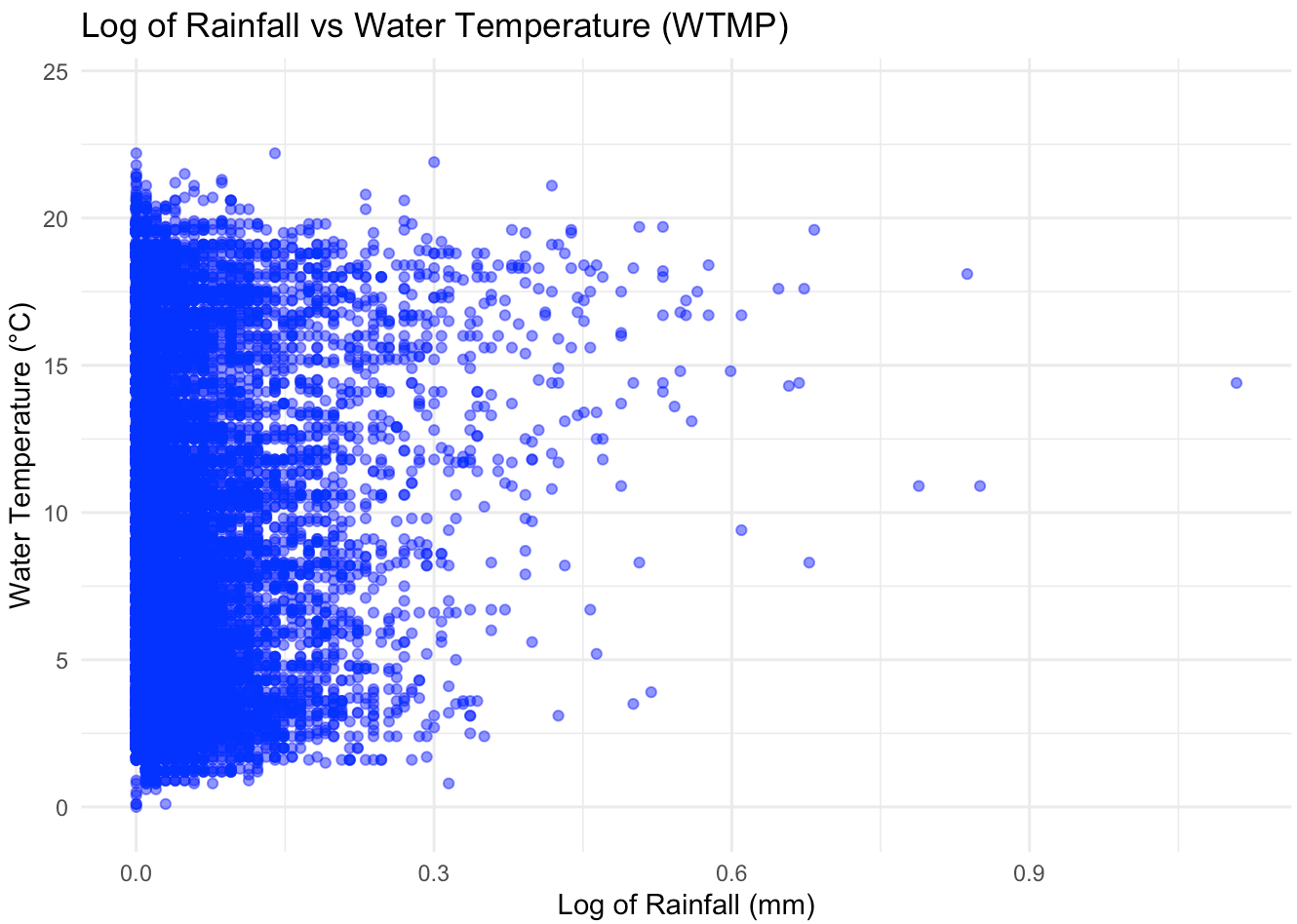
```
### Step 2: Visualization

# 2.1. Total yearly rainfall visualization
rainfall_data %>%
  mutate(year = year(datetime)) %>%
  group_by(year) %>%
  summarise(total_rainfall = sum(HPCP, na.rm = TRUE)) %>%
  ggplot(aes(x = year, y = total_rainfall)) +
  geom_line(color = "blue") +
  labs(title = "Total Yearly Rainfall in Boston (1985-2013)",
       x = "Year",
       y = "Total Rainfall (mm)") +
  theme_minimal()
```



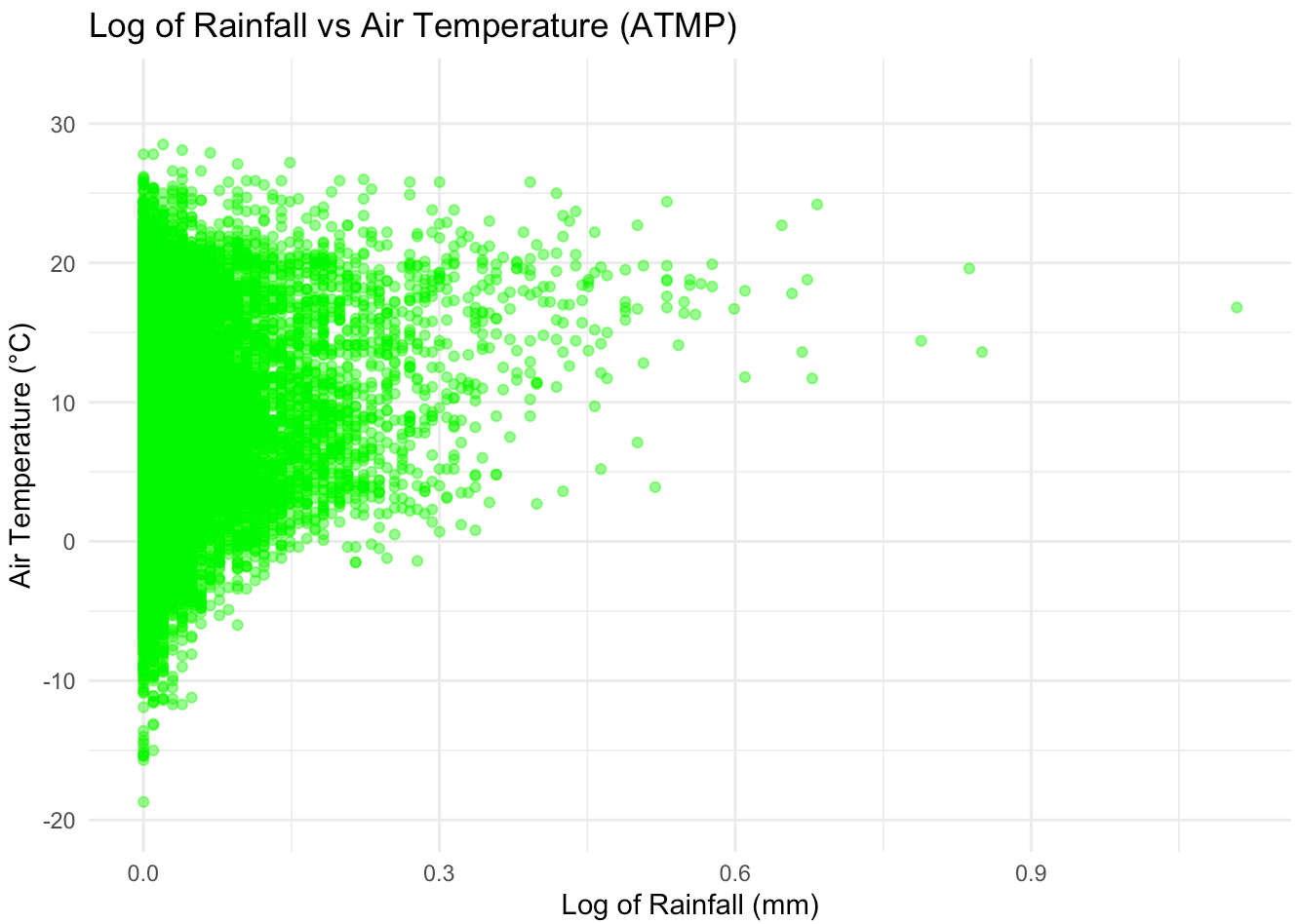

```
# 2.2. Average water temperature by year
yearly_avg_temp <- combined_data %>%
  mutate(year = factor(Year)) %>%
  group_by(year) %>%
  summarise(avg_WTMP = mean(WTMP, na.rm = TRUE))

ggplot(yearly_avg_temp, aes(x = year, y = avg_WTMP)) +
  geom_point(color = "green") +
  geom_line(group = 1, color = "green") +
  labs(title = "Average Water Temperature by Year (1985-2023)",
       x = "Year",
       y = "Average Temperature (°C)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Set x-axis text angle
```



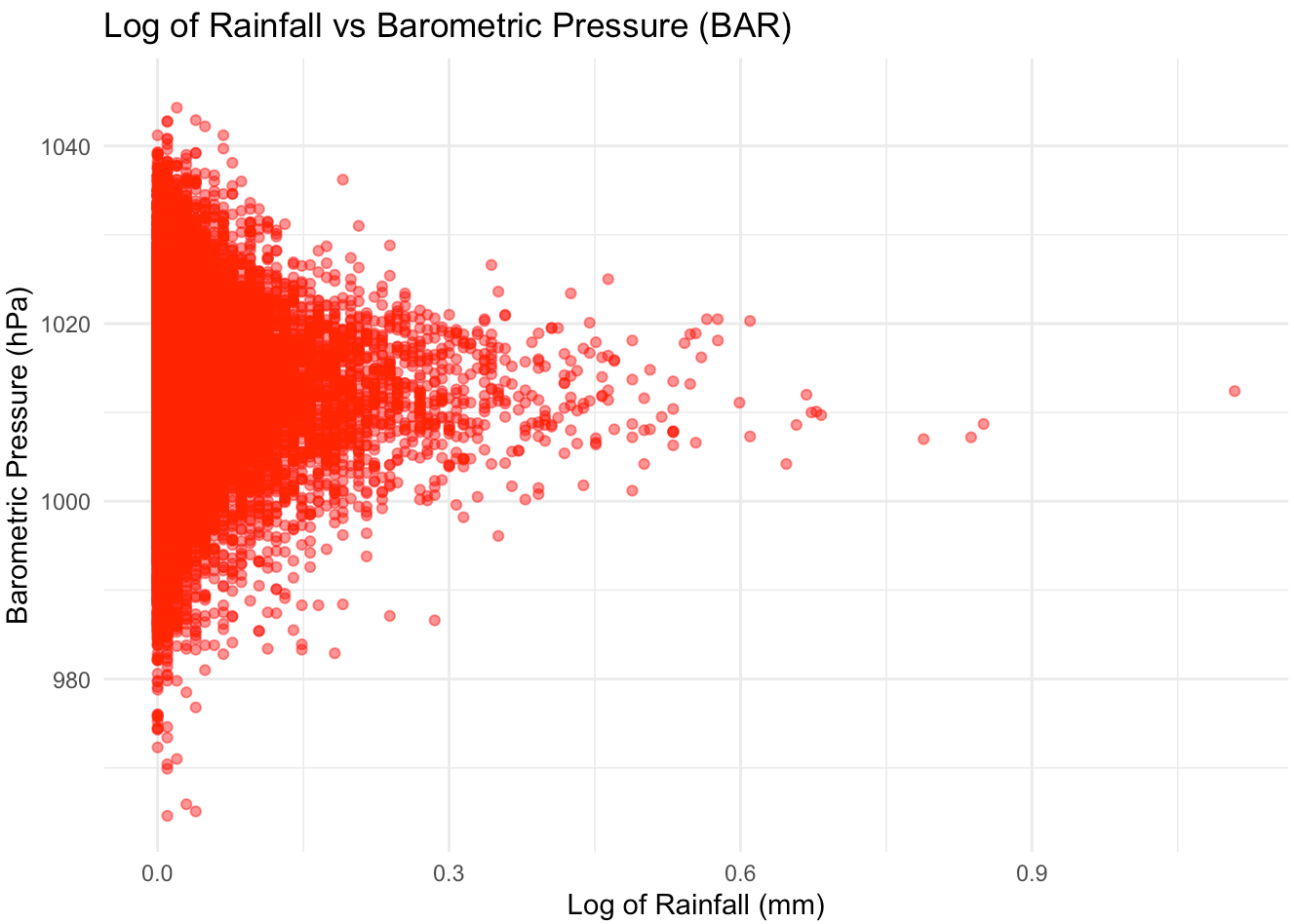

```
# 2.3. Scatter plot of Rainfall vs Water Temperature (WTMP)
ggplot(merged_data, aes(x = log(HPCP + 1), y = WTMP)) +
  geom_point(alpha = 0.5, color = "blue") +
  labs(title = "Log of Rainfall vs Water Temperature (WTMP)",
       x = "Log of Rainfall (mm)",
       y = "Water Temperature (°C)") +
  theme_minimal()
```




```
# 2.4. Scatter plot of Rainfall vs Air Temperature (ATMP)
ggplot(merged_data, aes(x = log(HPCP + 1), y = ATMP)) +
  geom_point(alpha = 0.5, color = "green") +
  labs(title = "Log of Rainfall vs Air Temperature (ATMP)",
       x = "Log of Rainfall (mm)",
       y = "Air Temperature (°C)") +
  theme_minimal()
```




```
# 2.5. Scatter plot of Rainfall vs Barometric Pressure (BAR)
ggplot(merged_data, aes(x = log(HPCP + 1), y = BAR)) +
  geom_point(alpha = 0.5, color = "red") +
  labs(title = "Log of Rainfall vs Barometric Pressure (BAR)",
       x = "Log of Rainfall (mm)",
       y = "Barometric Pressure (hPa)") +
  theme_minimal()
```




```
### Step 3: Statistical Analysis (Regression Model)

# 3.1. Build a simple linear regression model for rainfall prediction based on buoy data
rainfall_model <- stan_glm(log(HPCP + 1) ~ WTMP + ATMP + BAR, data = merged_data, refres
h = 0)

# 3.2. Summary of the model
summary(rainfall_model)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      log(HPCP + 1) ~ WTMP + ATMP + BAR
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 20104
##  predictors:   4
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  0.0    0.1 -0.1   0.0   0.1
## WTMP         0.0    0.0  0.0   0.0   0.0
## ATMP         0.0    0.0  0.0   0.0   0.0
## BAR          0.0    0.0  0.0   0.0   0.0
## sigma        0.1    0.0  0.1   0.1   0.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.0    0.0  0.0   0.0   0.0
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome v
ariable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2384
## WTMP          0.0  1.0  1345
## ATMP          0.0  1.0  1304
## BAR           0.0  1.0  2374
## sigma         0.0  1.0  2290
## mean_PPD      0.0  1.0  4230
## log-posterior 0.0  1.0  1540
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of e
ffective sample size, and Rhat is the potential scale reduction factor on split chains
(at convergence Rhat=1).
```

Analysis of Rainfall and Buoy Data (1985-2013)

Understanding the Data

We are tasked with investigating the relationship between rainfall in Boston and various meteorological data collected by a weather buoy. The key variables from the buoy include: - **WTMP**: Water Temperature (°C) - **ATMP**: Air Temperature (°C) - **BAR**: Barometric Pressure (hPa) - **HPCP**: Hourly Precipitation (mm)

The goal is to explore whether these variables (particularly **WTMP**, **ATMP**, and **BAR**) show any correlation with rainfall and whether we can spot any meaningful patterns. This will help us understand the dynamics between weather conditions and rainfall.

Summary Statistics

We begin by calculating summary statistics for both rainfall (HPCP) and buoy readings to understand their distribution.

- **Total Yearly Rainfall** (Graph 1): Shows the annual rainfall in Boston from 1985 to 2013. The total yearly rainfall fluctuates but doesn't show a clear increasing or decreasing trend.
- **Average Water Temperature** (Graph 2): Shows the yearly average water temperature from 1985 to 2023. We see some clear variability, with notable increases after 2000 and some spikes in 2011-2012.

Visualizations

1. **Log of Rainfall vs. Water Temperature (Graph 3)**:
    - The scatter plot shows the relationship between the log of rainfall and water temperature (WTMP). The pattern is mostly concentrated near small rainfall values, and there doesn't appear to be a strong correlation between WTMP and rainfall. The data points are spread out, suggesting a weak relationship.
2. **Log of Rainfall vs. Air Temperature (Graph 4)**:
    - This plot shows a weak to no correlation between rainfall and air temperature (ATMP). Like the previous graph, the majority of rainfall events occur with small changes in air temperature, and there doesn't appear to be a significant pattern.
3. **Log of Rainfall vs. Barometric Pressure (Graph 5)**:
    - The plot shows that higher barometric pressures tend to correspond with lower rainfall amounts, but the relationship is not linear. There is some clustering of data points suggesting that as the barometric pressure decreases, rainfall is more likely to occur.

Simple Statistical Model

A simple linear regression model was built using **WTMP**, **ATMP**, and **BAR** to predict rainfall.

Model Output:

Variable	Mean	Std. Dev.	10%	50%	90%
Intercept	0.0	0.1	-0.1	0.0	0.1
WTMP	0.0	0.0	0.0	0.0	0.0
ATMP	0.0	0.0	0.0	0.0	0.0
BAR	0.0	0.0	0.0	0.0	0.0
Sigma	0.1	0.0	0.1	0.1	0.1

Model Analysis:

- The model did not find any significant relationships between **WTMP**, **ATMP**, or **BAR** and rainfall. All estimates are close to zero, suggesting that none of these variables significantly contribute to explaining variability in rainfall.
- This aligns with the scatter plots, which also showed weak correlations.

Conclusion

Based on the visualizations and model analysis, we find that there is **no strong correlation** between rainfall and the buoy readings (water temperature, air temperature, or barometric pressure). While some patterns exist, such as slight clustering around certain barometric pressures, the relationships are generally weak.

Reflection on Weather Forecasting

This analysis gives us a better appreciation for the complexity of weather forecasting. Despite having access to large datasets, accurately predicting rainfall based on a few variables is challenging, and other factors not considered here (e.g., wind patterns, humidity) might be necessary for more accurate predictions.