615HW4

2024-09-25

HW introduction

(a)

As we discussed in lecture on Wednesday, the National Oceanic and Atmospheric Administration(NOAA) keeps track of meteorological data from a number of buoys. For this exercise, we are interested in Buoy number 44013 located sixteen nautical miles east of Boston Harbour and the questions will deal with data from that buoy. This is the link to the National Data Buoy Center: https://www.ndbc.noaa.gov/ This is the link to the buoy of interest: https://www.ndbc.noaa.gov/station__page.php?station=44013 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.

```r
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
```

Save your code in an R Script with an appropriate name which you must include in your Github submission. Keep in mind that if we clone your repository, this script must run without errors for you to get credit. Remember to comment your code for readability. For the following questions (b through d), you will need to put your code, its output, and your written answers into a pdf. One way to do this is to use R Markdown or Quarto and knit into a pdf. Include the rmd/qmd file as well the pdf in the github repository you submit.

```r
file_root <- "https://www.ndbc.noaa.gov/view_text_file.php?filename=44013h"
tail <- ".txt.gz&dir=data/historical/stdmet/"

# Empty list to store data for each year
all_years_data <- list()

# Loop through the years from 1985 to 2023
for (year in 1985:2023) {
  # Construct the file path for the current year
  path <- paste0(file_root, year, tail)

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

  # Read the buoy data, skipping the appropriate number of rows (2 for data + units row)
  buoy <- fread(path, header = FALSE, skip = 2, fill = TRUE)

  # Adjust header if it has more columns than the data
  if (length(header) > ncol(buoy)) {
    header <- header[1:ncol(buoy)]  # Truncate the header to match the data
  } else if (ncol(buoy) > length(header)) {
    buoy[, extra_col := NA]  # Add a placeholder column if data has more columns
  }

  # Set the column names
  colnames(buoy) <- header

  # Combine year (YY or YYYY) into a single 'YEAR' column
  # Logic: For years after 2006, #YY represents years starting from 2007
  buoy$YEAR <- ifelse(year >= 2007, as.numeric(buoy$`#YY`),
                      ifelse(as.numeric(buoy$YY) < 100, as.numeric(buoy$YY) + 1900, as.numeric(buoy$YY))

  # Add a proper date column using lubridate
  buoy$date <- ymd_hms(paste(buoy$YEAR, buoy$MM, buoy$DD, buoy$hh, buoy$mm))

  # Append the data to the list for all years
  all_years_data[[as.character(year)]] <- buoy
}

# Combine all years' data into one data.table and head of it
all_data <- rbindlist(all_years_data, fill = TRUE)

head(all_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     1    80     4     5    99    99    99   999 1030.0
## 2:     85     1     1     2   100     4     5    99    99    99   999 1030.1
## 3:     85     1     1     3   100     4     5    99    99    99   999 1029.4
## 4:     85     1     1     4   110     4     5    99    99    99   999 1028.6
## 5:     85     1     1     5    90     4     5    99    99    99   999 1027.8
## 6:     85     1     1     6    60     4     6    99    99    99   999 1027.7
##      ATMP  WTMP  DEWP   VIS  YEAR   date  YYYY  TIDE    mm   #YY  WDIR  PRES
##     <num> <num> <num> <num> <num> <POSc> <int> <num> <int> <int> <int> <num>
## 1:    5.1   6.7   999    99  1985   <NA>    NA    NA    NA    NA    NA    NA
## 2:    5.6   6.6   999    99  1985   <NA>    NA    NA    NA    NA    NA    NA
## 3:    5.8   6.7   999    99  1985   <NA>    NA    NA    NA    NA    NA    NA
## 4:    5.8   6.7   999    99  1985   <NA>    NA    NA    NA    NA    NA    NA
## 5:    5.3   6.7   999    99  1985   <NA>    NA    NA    NA    NA    NA    NA
## 6:    5.5   6.7   999    99  1985   <NA>    NA    NA    NA    NA    NA    NA
```

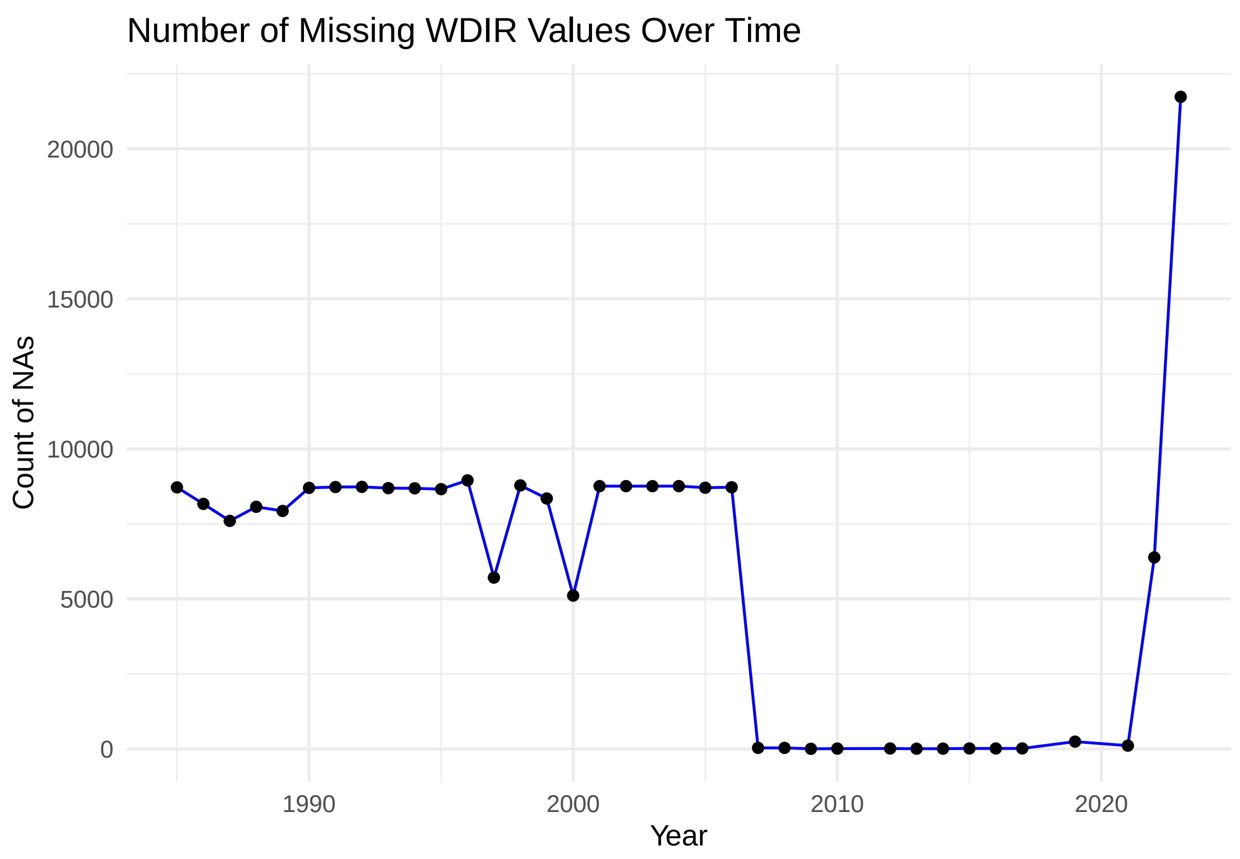
(b)

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.

```r
# Define missing value placeholders for relevant columns
missing_values <- c(999, 999.0, 9999, 99.0, 99.00)

# Columns to check for missing values
columns_to_check <- setdiff(colnames(all_data), "YEAR")  # All except 'YEAR'

# Loop through columns and replace 99, 999, 9999 with NA
for (col in columns_to_check) {
  all_data[get(col) %in% missing_values, (col) := NA]
}

# Ensure that NA's were applied correctly
summary(all_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      Min.   : 0.0    Min.   : 0.00   Min.   :0.00
##  1st Qu.:125      1st Qu.: 3.5    1st Qu.: 4.20   1st Qu.:0.41
##  Median :210      Median : 5.3    Median : 6.50   Median :0.66
##  Mean   :197      Mean   : 5.9    Mean   : 7.29   Mean   :0.87
##  3rd Qu.:280      3rd Qu.: 7.9    3rd Qu.: 9.70   3rd Qu.:1.06
##  Max.   :360      Max.   :25.7    Max.   :32.40   Max.   :9.10
##  NA's   :295742   NA's   :33183   NA's   :33485   NA's   :144267
##       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.:1010.2
##  Median : 7.69    Median : 4.70    Median : 94.0    Median :1015.8
##  Mean   : 7.39    Mean   : 4.96    Mean   :124.4    Mean   :1015.6
##  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.   :1045.8
##  NA's   :147959   NA's   :144267   NA's   :327149   NA's   :281345
##       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   :253597   NA's   :443042
##       YEAR            date                              YYYY
##  Min.   :1985    Min.   :2020-08-01 01:00:00.00    Min.   :1999
##  1st Qu.:1998    1st Qu.:2020-08-09 01:14:50.00    1st Qu.:2001
##  Median :2012    Median :2020-09-10 08:16:50.00    Median :2003
##  Mean   :2009    Mean   :2020-09-25 23:38:42.30    Mean   :2003
##  3rd Qu.:2021    3rd Qu.:2020-12-02 08:15:50.00    3rd Qu.:2005
##  Max.   :2023    Max.   :2020-12-13 00:23:50.00    Max.   :2006
##                  NA's   :425482                   NA's   :396356
##       TIDE              mm              #YY              WDIR
##  Min.   : NA      Min.   : 0.00    Min.   :2007     Min.   :  0.0
##  1st Qu.: NA      1st Qu.:10.00    1st Qu.:2015     1st Qu.:131.0
##  Median : NA      Median :40.00    Median :2021     Median :205.0
##  Mean   :NaN      Mean   :31.58    Mean   :2018     Mean   :197.3
##  3rd Qu.: NA      3rd Qu.:50.00    3rd Qu.:2022     3rd Qu.:280.0
##  Max.   : NA      Max.   :50.00    Max.   :2023     Max.   :360.0
##  NA's   :462279   NA's   :164630   NA's   :182059   NA's   :210712
##       PRES
##  Min.   : 970
##  1st Qu.:1011
##  Median :1016
##  Mean   :1016
##  3rd Qu.:1021
##  Max.   :1046
##  NA's   :187754
```

```
#Summarize the number of NAs for WDIR by year and plot the number of NAs over time
na_summary_by_year <- all_data %>%
  group_by(YEAR) %>%
  summarise(WDIR_NA_count = sum(is.na(WDIR)),
            total_count = n())

# Plot the number of NAs over time
ggplot(na_summary_by_year, aes(x = YEAR, y = WDIR_NA_count)) +
  geom_line(color = "blue") +
  geom_point() +
  labs(title = "Number of Missing WDIR Values Over Time",
       x = "Year",
       y = "Count of NAs") +
  theme_minimal()
```



(b.1) Is it always appropriate to convert missing/null data to NA's? When might it not.

When it is missing data, converting placeholders like 999 or 99.0 to NA is appropriate. This makes it clear that the data is absent and allows statistical functions to handle the missing values correctly. When the placeholder 999 or similar values have specific meanings other than "missing" (e.g., outliers, extreme conditions), converting these to NA might lose important information.

(b.2) Analyze the pattern of NA's. Do you spot any patterns in the way/dates that these are distributed?

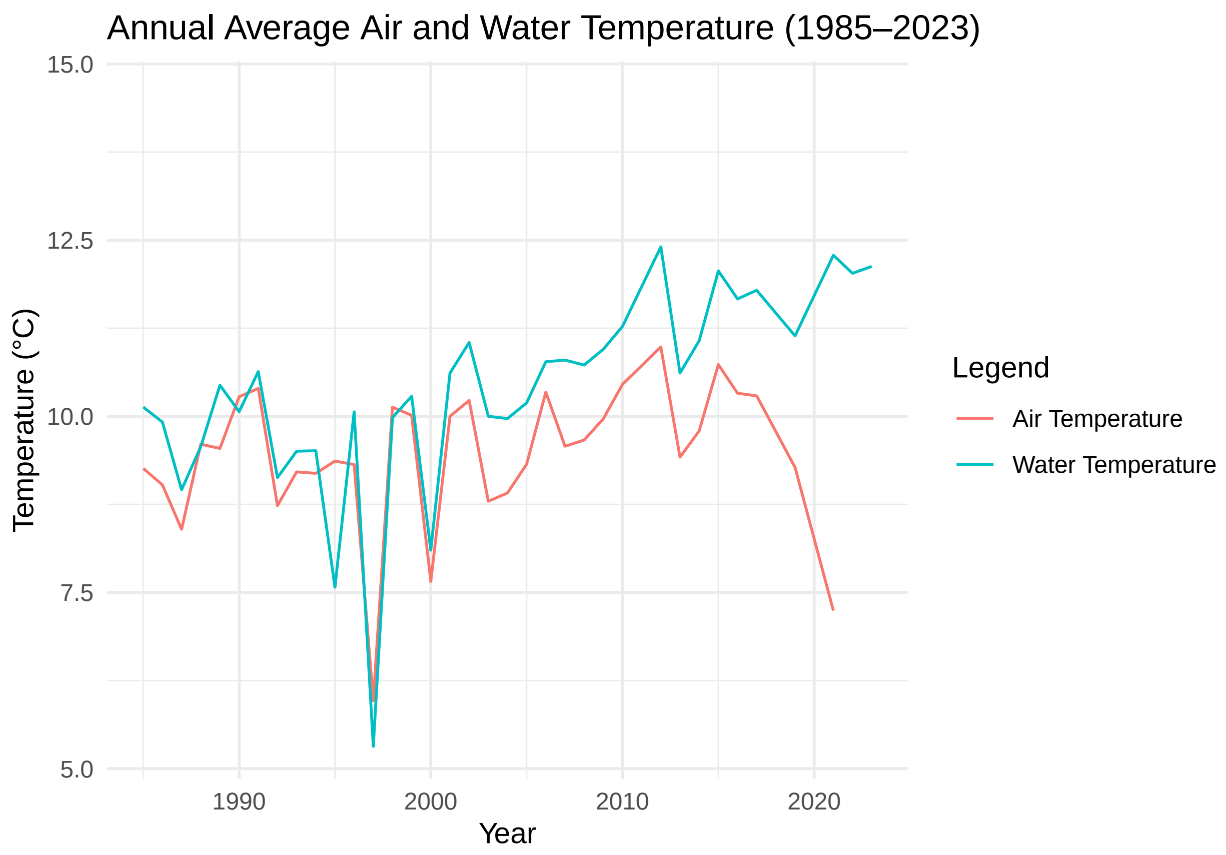
Between 1985 and 2005, missing data remained stable with minor fluctuations, suggesting routine operational and maintenance activities. A notable improvement occurred from 2006 to 2015, with missing values nearly eliminated, likely due to advancements in sensor technology and better monitoring systems. However, from 2016 onwards, especially after 2020, there was a sharp increase in missing data, reaching over 20,000 instances by 2023. This recent surge could be attributed to equipment failures, decreased maintenance capabilities due to external factors like COVID-19, and possible budgetary constraints affecting the regular upkeep of data collection systems.

(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.

key climate indicators (ATMP: Air Temperature, WTMP: Water Temperature)

```r
# Summarize the annual average air and water temperatures
temp_summary_by_year <- all_data %>%
  group_by(YEAR) %>%
  summarise(
    avg_air_temp = mean(ATMP, na.rm = TRUE),
    avg_water_temp = mean(WTMP, na.rm = TRUE)
  )

# Plot the trends over time for both air and water temperatures
ggplot(temp_summary_by_year, aes(x = YEAR)) +
  geom_line(aes(y = avg_air_temp, color = "Air Temperature")) +
  geom_line(aes(y = avg_water_temp, color = "Water Temperature")) +
  labs(title = "Annual Average Air and Water Temperature (1985-2023)",
       x = "Year",
       y = "Temperature (°C)",
       color = "Legend") +
  theme_minimal()
```



```r
#  Fit linear regression models to check for temperature trends over time
# Air Temperature
air_temp_lm <- lm(avg_air_temp ~ YEAR, data = temp_summary_by_year)
summary(air_temp_lm)
```

```
##
```

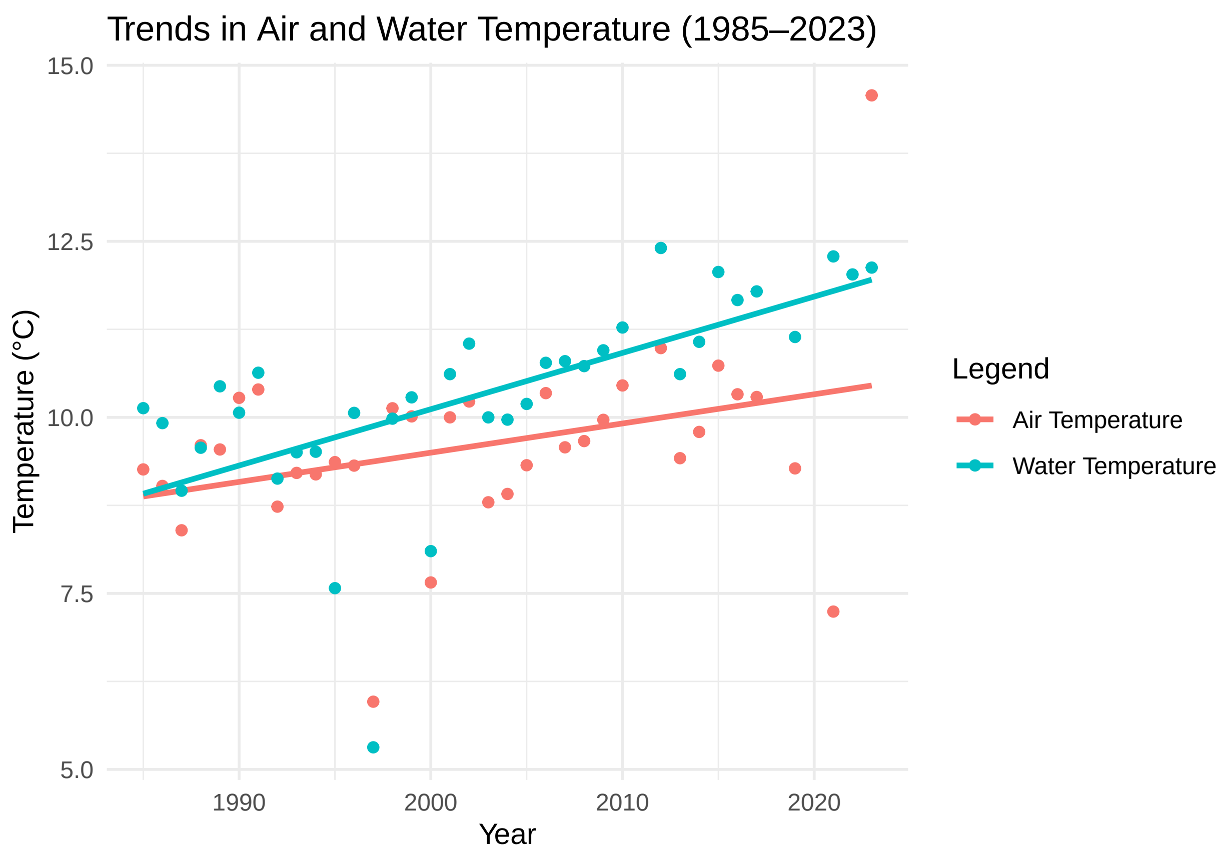
```
## Call:
## lm(formula = avg_air_temp ~ YEAR, data = temp_summary_by_year)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4118 -0.4101  0.0858  0.5770  4.1206
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -73.45662   39.82298  -1.845   0.0741 .
## YEAR          0.04148    0.01989   2.086   0.0448 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.262 on 33 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1165, Adjusted R-squared:  0.08969
## F-statistic:  4.35 on 1 and 33 DF,  p-value: 0.04482
```

```
# Water Temperature
water_temp_lm <- lm(avg_water_temp ~ YEAR, data = temp_summary_by_year)
summary(water_temp_lm)
```

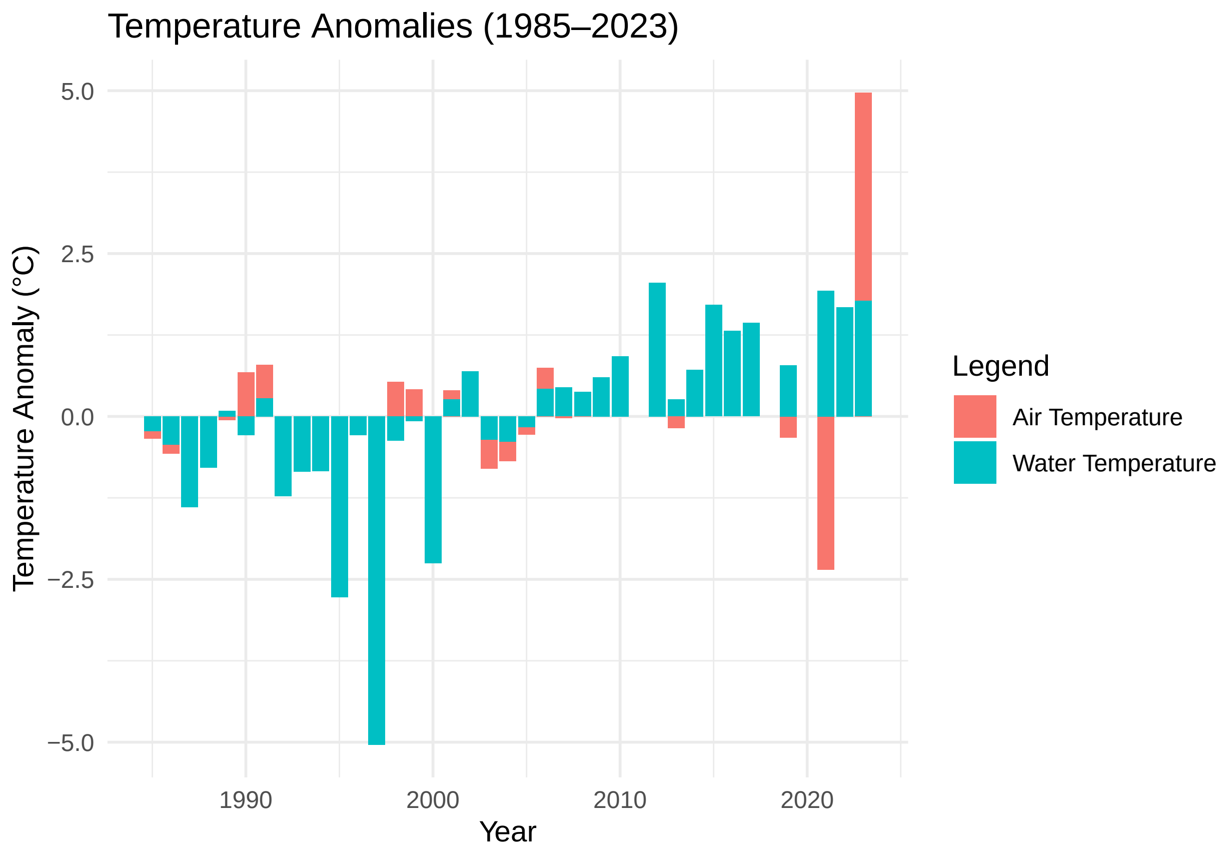
```
##
## Call:
## lm(formula = avg_water_temp ~ YEAR, data = temp_summary_by_year)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5615 -0.2034  0.1631  0.4362  1.3298
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -149.84305   32.97621  -4.544 6.65e-05 ***
## YEAR           0.07998    0.01646   4.858 2.62e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.092 on 34 degrees of freedom
## Multiple R-squared:  0.4097, Adjusted R-squared:  0.3924
## F-statistic:  23.6 on 1 and 34 DF,  p-value: 2.622e-05
```

```
# Display regression lines on the plots
ggplot(temp_summary_by_year, aes(x = YEAR)) +
  geom_point(aes(y = avg_air_temp, color = "Air Temperature")) +
  geom_smooth(aes(y = avg_air_temp, color = "Air Temperature"), method = "lm", se = FALSE) +
  geom_point(aes(y = avg_water_temp, color = "Water Temperature")) +
  geom_smooth(aes(y = avg_water_temp, color = "Water Temperature"), method = "lm", se = FALSE) +
  labs(title = "Trends in Air and Water Temperature (1985-2023)",
       x = "Year",
       y = "Temperature (°C)",
       color = "Legend") +
  theme_minimal()
```

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



```
# Calculate the annual temperature anomalies (difference from the mean) for further analysis
# This can help highlight any abnormal years or extreme events
temp_summary_by_year <- temp_summary_by_year %>%
  mutate(
    air_temp_anomaly = avg_air_temp - mean(avg_air_temp, na.rm = TRUE),
    water_temp_anomaly = avg_water_temp - mean(avg_water_temp, na.rm = TRUE)
  )

# Plot the anomalies over time
ggplot(temp_summary_by_year, aes(x = YEAR)) +
  geom_bar(aes(y = air_temp_anomaly, fill = "Air Temperature"), stat = "identity", position = "dodge") +
  geom_bar(aes(y = water_temp_anomaly, fill = "Water Temperature"), stat = "identity", position = "dodg
  labs(title = "Temperature Anomalies (1985-2023)",
       x = "Year",
       y = "Temperature Anomaly (°C)",
       fill = "Legend") +
  theme_minimal()
```



(c.1)

Plot1: Annual Average Air and Water Temperature (1985–2023): This plot shows the average annual air temperature (red line) and water temperature (blue line) over time. Both air and water temperatures show an upward trend over the years. Water temperature appears to have more variability but rises more significantly compared to air temperature, especially after 2000.

(c.2)

Plot2: Trends in Air and Water Temperature (1985–2023): This scatter plot with trend lines highlights the gradual increase in both air and water temperatures. Water temperature (cyan points) shows a steeper upward trend, suggesting that ocean warming is occurring at a faster rate compared to atmospheric warming.

These increasing trends are consistent with the global effects of climate change, where warming oceans and rising air temperatures are major indicators.

(c.3)

Plot3: Temperature Anomalies (1985–2023): This bar plot illustrates temperature anomalies, showing deviations from the long-term average for both air and water temperatures. There is a noticeable increase in temperature anomalies (both positive and negative) after the year 2000, with more frequent positive anomalies, especially in water temperatures after 2010.

This suggests that both air and water temperatures are increasing more frequently beyond their historical norms, a clear indicator of climate variability and the impact of climate change.

(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?

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

head(rainfall_data$DATE)
```

```
## [1] "19850101 01:00" "19850101 09:00" "19850101 10:00" "19850101 11:00"
## [5] "19850101 12:00" "19850101 13:00"
```

```r
# Step 1: Parse the DATE column using ymd_hm
rainfall_data$DATE <- ymd_hm(rainfall_data$DATE)
```

```r
# Step 2: Extract the year from the parsed DATE column
rainfall_data$YEAR <- year(rainfall_data$DATE)
```

```r
# Step 3: Summarize total rainfall by year
rainfall_by_year <- rainfall_data %>%
  group_by(YEAR) %>%
  summarise(total_rainfall = sum(HPCP, na.rm = TRUE))  # Sum the total rainfall for each year

print(head(rainfall_by_year))
```

```
## # A tibble: 6 x 2
##    YEAR total_rainfall
##   <dbl>          <dbl>
## 1  1985           36.6
## 2  1986           44.3
## 3  1987           45.5
## 4  1988           34.8
## 5  1989           42.4
## 6  1990           46.5
```

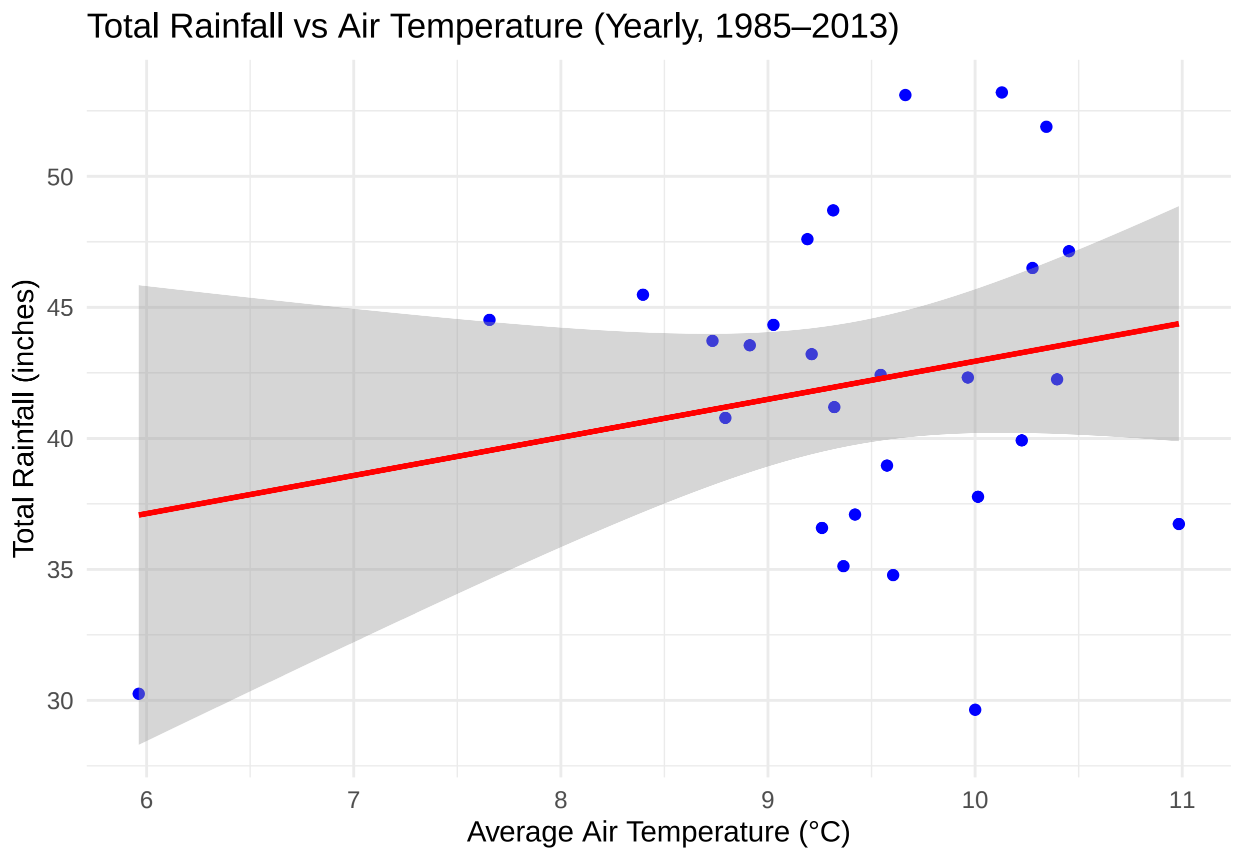
```r
buoy_data_by_year <- all_data %>%
  filter(YEAR >= 1985 & YEAR <= 2013) %>%  # Filter buoy data to match rainfall data's year range
  group_by(YEAR) %>%
  summarise(
    avg_air_temp = mean(ATMP, na.rm = TRUE),       # Average Air Temperature
    avg_water_temp = mean(WTMP, na.rm = TRUE),     # Average Water Temperature
    avg_wind_speed = mean(WSPD, na.rm = TRUE)      # Average Wind Speed
  )

combined_data <- left_join(rainfall_by_year, buoy_data_by_year, by = "YEAR")
```

```r
head(combined_data)
```

```
## # A tibble: 6 x 5
##    YEAR total_rainfall avg_air_temp avg_water_temp avg_wind_speed
##   <dbl>          <dbl>        <dbl>          <dbl>          <dbl>
## 1  1985           36.6         9.26          10.1            5.63
## 2  1986           44.3         9.03           9.92           5.55
## 3  1987           45.5         8.40           8.96           5.89
## 4  1988           34.8         9.60           9.57           7.07
## 5  1989           42.4         9.54          10.4            6.30
## 6  1990           46.5        10.3           10.1            6.36
```

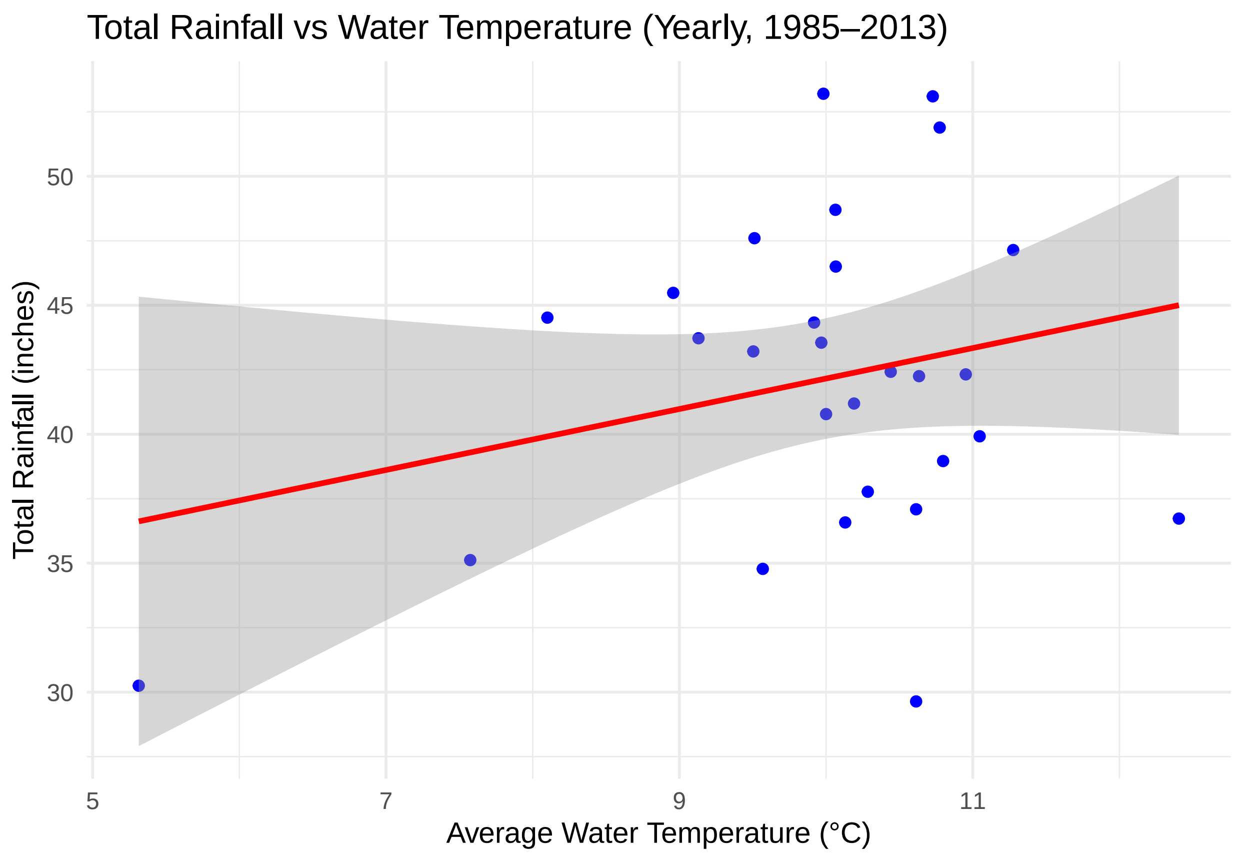
```r
# Plot 1: Total Rainfall vs Air Temperature (Yearly Data)
ggplot(combined_data, aes(x = avg_air_temp, y = total_rainfall)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Total Rainfall vs Air Temperature (Yearly, 1985-2013)",
       x = "Average Air Temperature (°C)",
       y = "Total Rainfall (inches)") +
  theme_minimal()
```

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



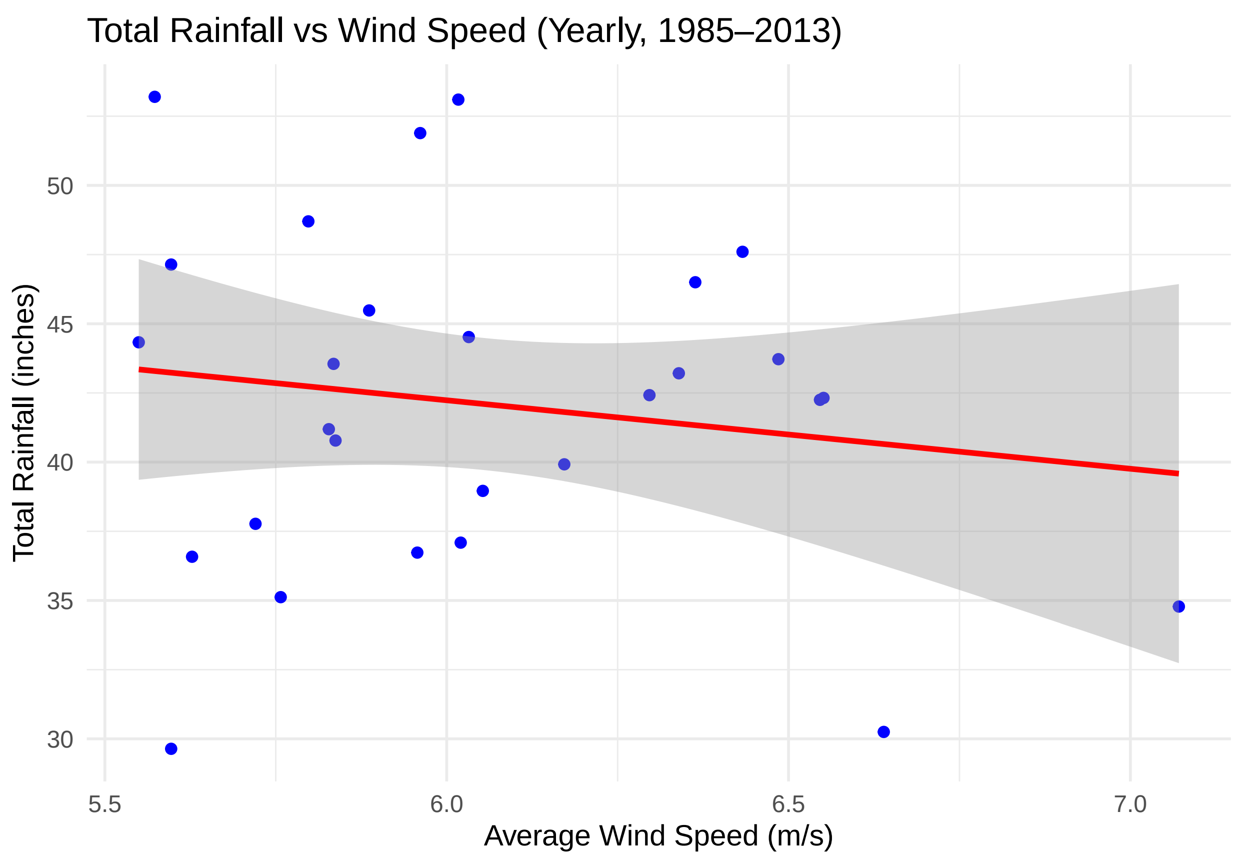
```r
# Plot 2: Total Rainfall vs Water Temperature (Yearly Data)
ggplot(combined_data, aes(x = avg_water_temp, y = total_rainfall)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Total Rainfall vs Water Temperature (Yearly, 1985-2013)",
       x = "Average Water Temperature (°C)",
       y = "Total Rainfall (inches)") +
  theme_minimal()
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

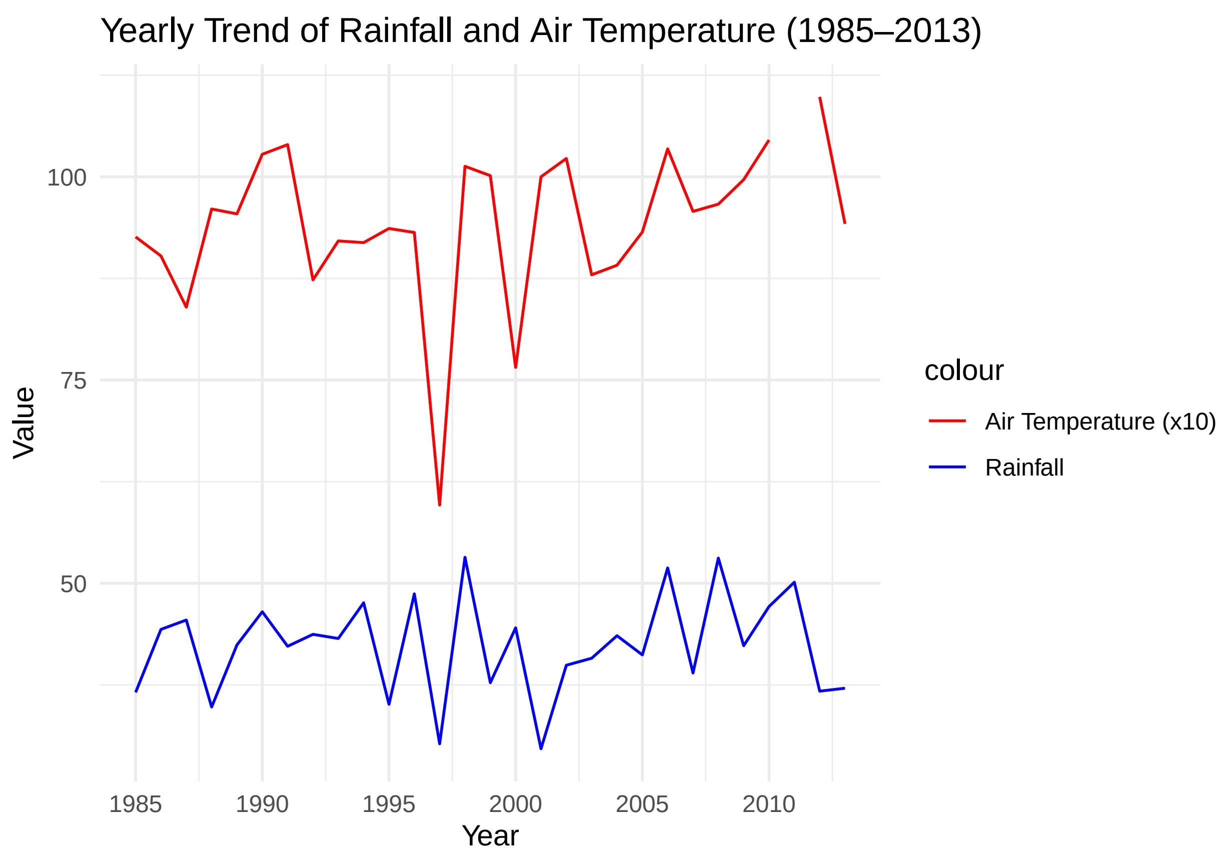


```r
# Plot 3: Total Rainfall vs Wind Speed (Yearly Data)
ggplot(combined_data, aes(x = avg_wind_speed, y = total_rainfall)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Total Rainfall vs Wind Speed (Yearly, 1985-2013)",
       x = "Average Wind Speed (m/s)",
       y = "Total Rainfall (inches)") +
  theme_minimal()
```

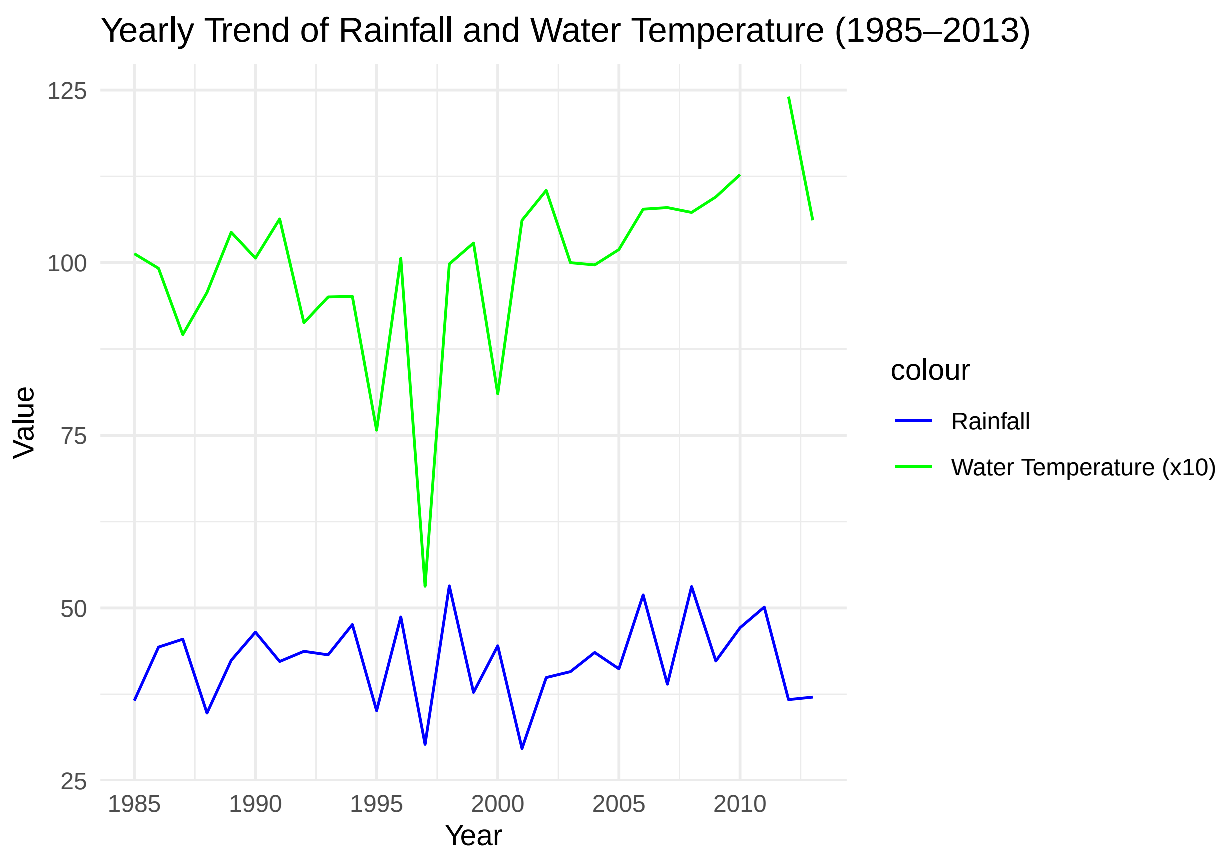
```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```r
# Trend of Rainfall and Air Temperature over Years
ggplot(combined_data, aes(x = YEAR)) +
  geom_line(aes(y = total_rainfall, color = "Rainfall")) +
  geom_line(aes(y = avg_air_temp * 10, color = "Air Temperature (x10)")) + # Scaling air temp for compa
  labs(title = "Yearly Trend of Rainfall and Air Temperature (1985-2013)",
       x = "Year",
       y = "Value") +
  scale_color_manual(values = c("Rainfall" = "blue", "Air Temperature (x10)" = "red")) +
  theme_minimal()
```



```r
# Trend of Rainfall and Water Temperature over Years
ggplot(combined_data, aes(x = YEAR)) +
  geom_line(aes(y = total_rainfall, color = "Rainfall")) +
  geom_line(aes(y = avg_water_temp * 10, color = "Water Temperature (x10)")) + # Scaling water temp for
  labs(title = "Yearly Trend of Rainfall and Water Temperature (1985-2013)",
       x = "Year",
       y = "Value") +
  scale_color_manual(values = c("Rainfall" = "blue", "Water Temperature (x10)" = "green")) +
  theme_minimal()
```



Although Rainfall remains relatively stable, the ATMP & WTMP are in an increasing trend.