HW4

Vajinder

2024-09-27

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.4.1
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.1
```

```
##
## 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(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.4.1
```

```
##
## 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)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.1
```

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.4.1
```

```
##
## 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(tibble)
```

```
## Warning: package 'tibble' was built under R version 4.4.1
```

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.4.1
```

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.

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

load_buoy_data <- function(year) {
  path <- paste0(file_root, year, tail)


  if (year < 2007) {
  header <- scan(path, what = 'character', nlines = 1)
  buoy <- read.table(path, fill = TRUE, header = TRUE, sep = "")
  buoy <- add_column(buoy, mm = NA, .after = "hh")
  buoy <- add_column(buoy, TIDE = NA, .after = "VIS")

  } else {
```

```
  header <- scan(path, what = 'character', nlines = 1)
  buoy <- fread(path, header = FALSE, skip = 1, fill = TRUE)

    setnames(buoy, header)
  }

  #return(buoy)
}

all_data <- lapply(1985:2023, load_buoy_data)

combined_data <- rbindlist(all_data, fill = TRUE)
```

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?

Sometimes, we can use mean or average instead of NA and sometimes, Regression models can be used in order to predict nearest possible value instead of using NAs

When might it not be? Sometimes, For example in Buoy, We might get so many NAs that we won't have much of data to get information from. Additionally, Sometimes adding NAs might give us wrong information as well. In other words, sometimes 0 or TRUE/FALSE are better than NA.

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

A lot of NAs in the beginning that almost forces us to see the summary if it is all NA. But it starts showing improvement after 2015s which I believe is due to investment on to resources to get the information more. On enquiring in depth, I got the following info regardnig the trends of funding which matched somewhat with the NA trend we see in the data: From 1985 - Mid 2000s: Funding was stable but not enough as they started looking for more resources and investments to improve technology. Mid to Late 2000s: Funding levels experienced some fluctuations due to budgetary constraints and changing federal priorities. Despite these challenges, there was ongoing investment in advanced technologies, including remote sensing and automated data collection systems, which are crucial for climate monitoring

2010s to Present: The trend has seen increased funding as awareness of climate change has risen. New collaborations with private companies and research institutions have also emerged, aiming to develop innovative marine observation technologies. Projects like the partnership with Saildrone to replace moored buoys indicate a shift towards sustainable and efficient monitoring methods

Government Shutdowns and financial crisis: Notable government shutdowns can significantly affect funding. For example,2008 financial crisis, the 2013 shutdown, lasting 16 days, and the one in late 2018 to early 2019 had substantial impacts on various government agencies, including those involved in climate and ocean data collection. During these periods, funding was halted, leading to delays in projects and a reevaluation of budgets.

```
combined_data <- combined_data %>%
  mutate(
    YY = as.character(YY),
    `#YY` = as.character(`#YY`),
    YYYY = as.character(YYYY)
  )

# Combine year columns safely using coalesce
combined_data <- combined_data %>%
  mutate(YYYY = coalesce(YYYY, `#YY`, YY))
```

```r
combined_data <- combined_data %>%
  mutate(BAR = coalesce(as.numeric(BAR), as.numeric(PRES)),  # Convert BAR and PRES to numeric
    WD = coalesce(as.numeric(WD), as.numeric(WDIR)))
```

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `BAR = coalesce(as.numeric(BAR), as.numeric(PRES))`.
## Caused by warning in `list2()`:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```r
combined_data <- combined_data %>%
  select(-TIDE, -TIDE.1, -mm.1,- WDIR, -PRES,-`#YY`,-YY)

combined_data$datetime <- ymd_h(paste(combined_data$YYYY, combined_data$MM, combined_data$DD, combined_
```

```
## Warning: 17 failed to parse.
```

```r
combined_data <- combined_data %>%
  mutate(across(everything(),
                ~ na_if(as.numeric(as.character(.)), 99) %>%
                na_if(999) %>%
                na_if(9999)))
```

```
## Warning: There were 16 warnings in `mutate()`.
## The first warning was:
## i In argument: `across(...)`.
## Caused by warning in `vec_cast()`:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 15 remaining warnings.
```

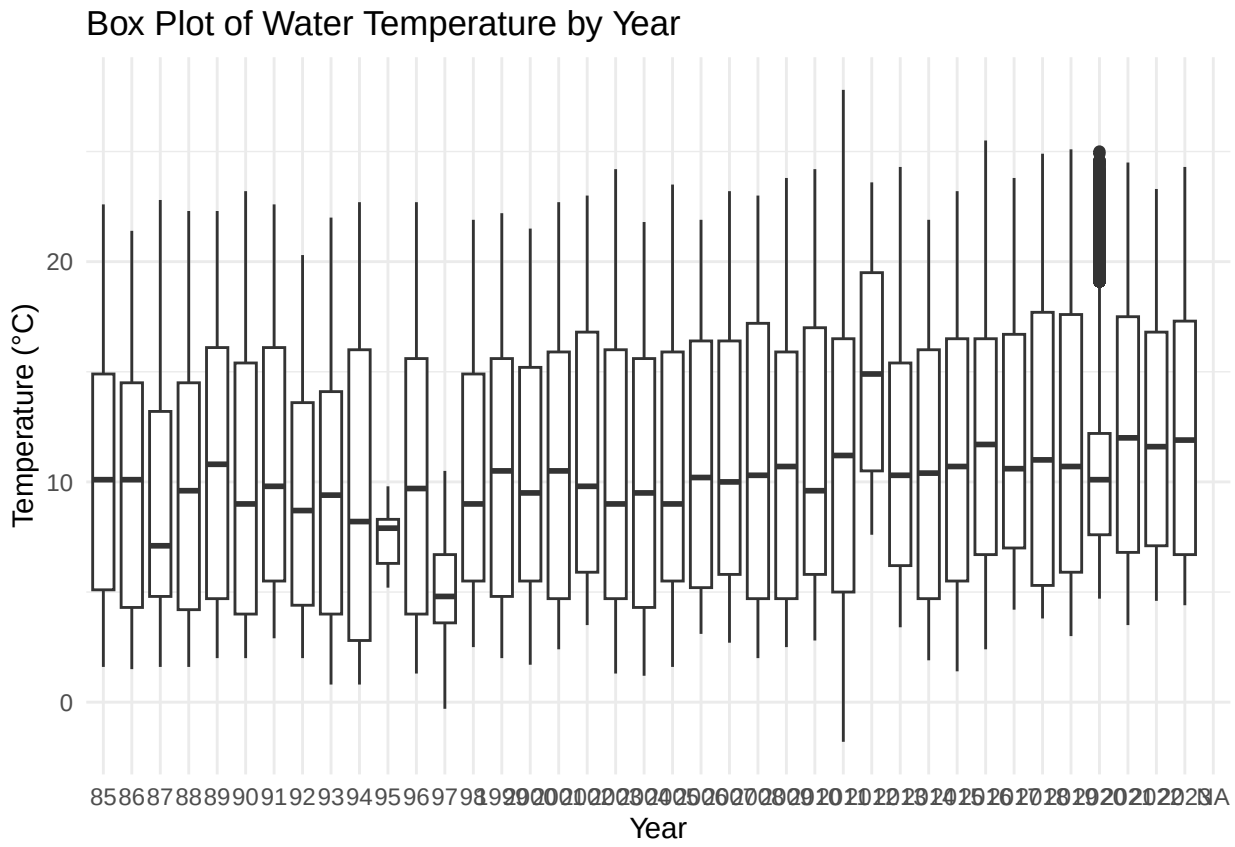
```r
#summary(combined_data)
#str(combined_data)
#str(combined_data$datetime)
if (!inherits(combined_data$datetime, "POSIXct")) {
  combined_data$datetime <- ymd_h(paste(combined_data$YYYY, combined_data$MM, combined_data$DD, combine
}
```

```
## Warning: 17 failed to parse.
```

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.

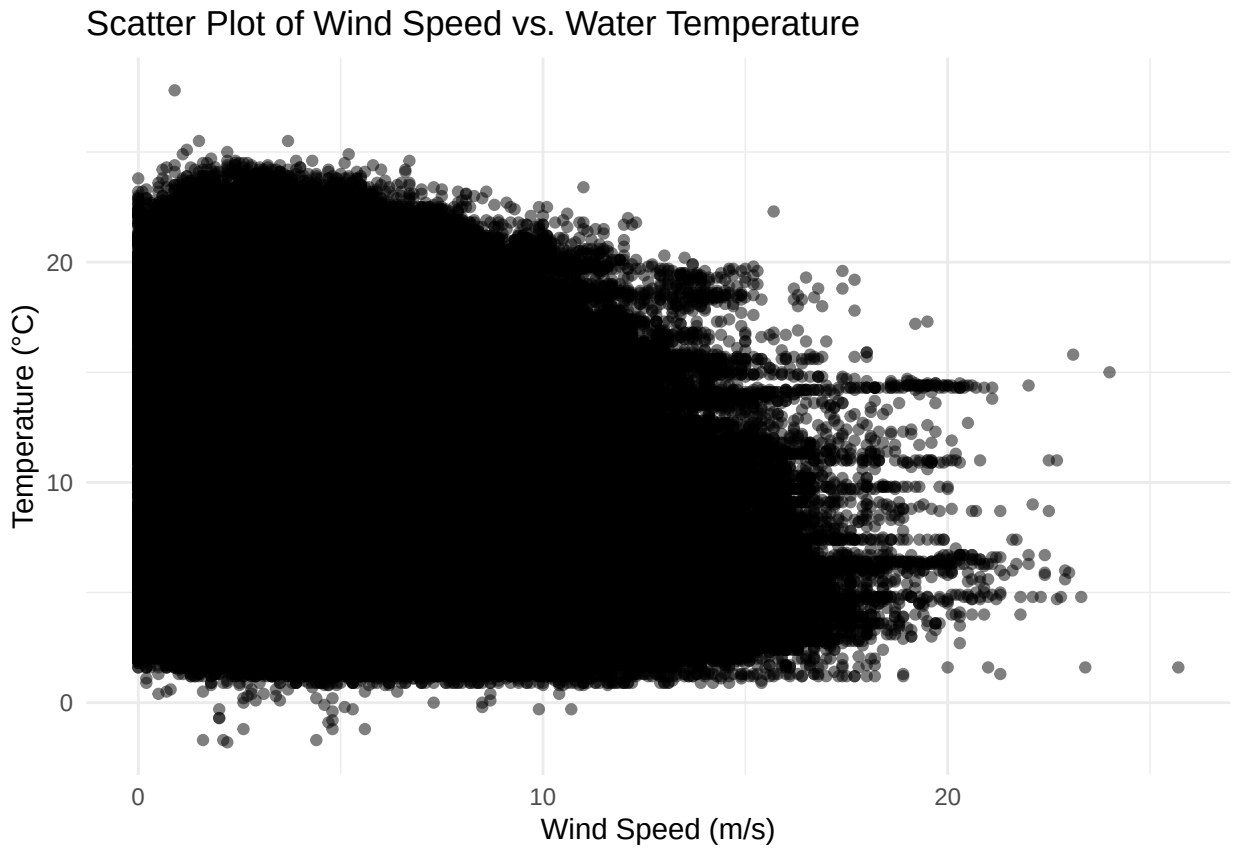
```r
# Box Plot
ggplot(combined_data, aes(x = factor(YYYY), y = WTMP)) +
  geom_boxplot() +
  labs(title = "Box Plot of Water Temperature by Year", x = "Year", y = "Temperature (°C)") +
  theme_minimal()
```

```
## Warning: Removed 13214 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
# Scatter Plot
ggplot(combined_data, aes(x = WSPD, y = WTMP)) +
  geom_point(alpha = 0.5) +
  labs(title = "Scatter Plot of Wind Speed vs. Water Temperature", x = "Wind Speed (m/s)", y = "Tempera
  theme_minimal()
```

```
## Warning: Removed 45389 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```




```r
combined_data$WTMP <- as.numeric(combined_data$WTMP)

yearly_temp_stats <- combined_data %>%
  group_by(year = year(datetime)) %>%
  summarise(
    min_temp = min(WTMP, na.rm = TRUE),
    max_temp = max(WTMP, na.rm = TRUE)
  )
```

```
## Warning: There were 2 warnings in `summarise()`.
## The first warning was:
## i In argument: `min_temp = min(WTMP, na.rm = TRUE)`.
## i In group 40: `year = NA`.
## Caused by warning in `min()`:
## ! no non-missing arguments to min; returning Inf
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```r
ggplot(yearly_temp_stats, aes(x = year)) +
  geom_point(aes(y = min_temp), color = "blue") +   # Scatter plot for min temp
  geom_line(aes(y = min_temp), color = "blue", size = 1) + # Line for min temp
  geom_point(aes(y = max_temp), color = "red") +     # Scatter plot for max temp
  geom_line(aes(y = max_temp), color = "red", size = 1) + # Line for max temp
  labs(title = "Yearly Minimum and Maximum Temperatures from 1985-2023",
       x = "Year",
       y = "Temperature (°C)",
```

```
      color = "Legend") +
  theme_minimal() +
  scale_color_manual(values = c("Min Temperature" = "blue", "Max Temperature" = "red")) +
  scale_y_continuous(breaks = seq(min(yearly_temp_stats$min_temp, na.rm = TRUE),
                                  max(yearly_temp_stats$max_temp, na.rm = TRUE),
                                  by = 5))
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

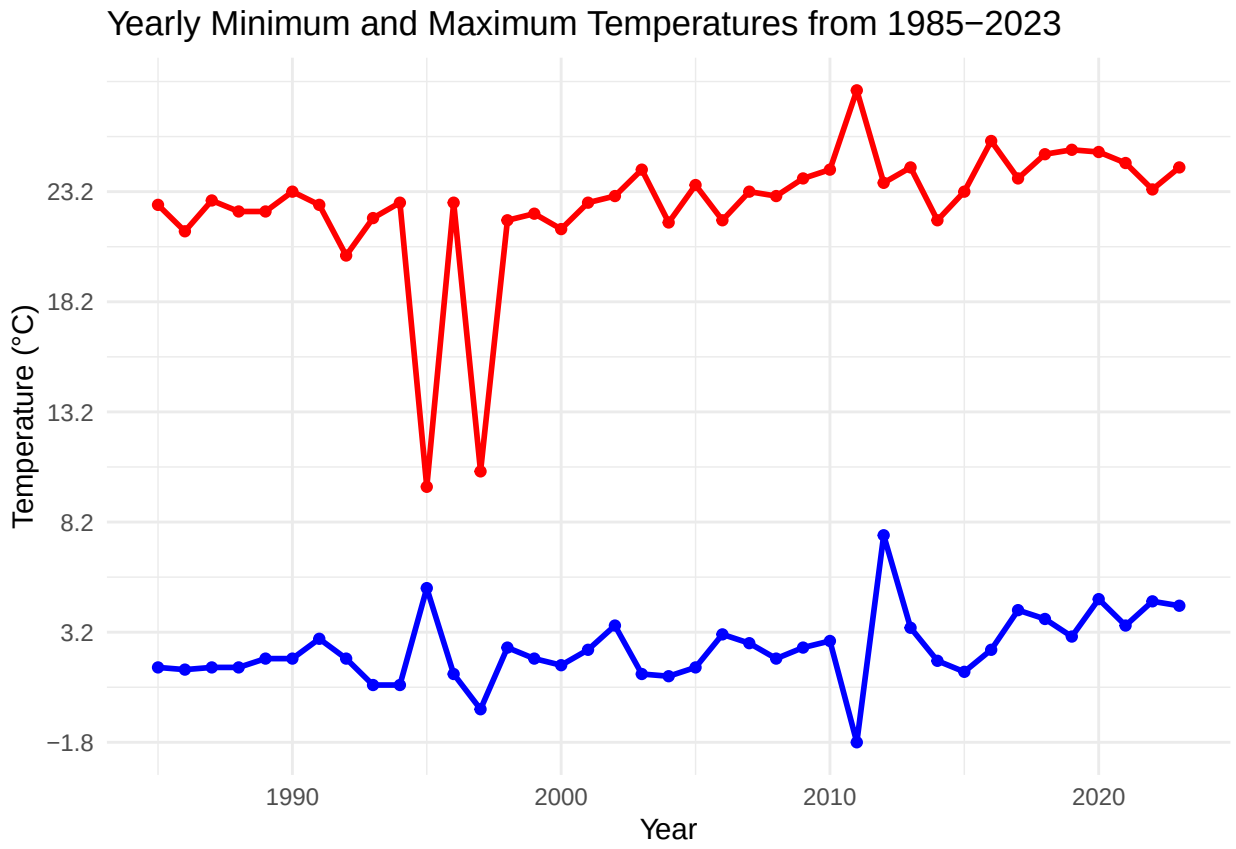
```
## Warning: No shared levels found between ‘names(values)‘ of the manual scale and the
## data’s colour values.
```

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

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

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

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




```
yearly_temp_stats <- combined_data %>%
  group_by(year = year(datetime)) %>%
  summarise(
    min_temp = min(WTMP, na.rm = TRUE),
    max_temp = max(WTMP, na.rm = TRUE),
    mean_temp = mean(WTMP, na.rm = TRUE),
    sd_temp = sd(WTMP, na.rm = TRUE),
    p25_temp = quantile(WTMP, 0.25, na.rm = TRUE),
    p75_temp = quantile(WTMP, 0.75, na.rm = TRUE)
  )
```

```
## Warning: There were 2 warnings in `summarise()`.
## The first warning was:
## i In argument: `min_temp = min(WTMP, na.rm = TRUE)`.
## i In group 40: `year = NA`.
## Caused by warning in `min()`:
## ! no non-missing arguments to min; returning Inf
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
trend_model <- lm(mean_temp ~ year, data = yearly_temp_stats)
summary(trend_model)
```

```
##
## Call:
## lm(formula = mean_temp ~ year, data = yearly_temp_stats)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6460 -0.2824  0.0376  0.3300  3.8758
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -149.74208   33.30107  -4.497 6.60e-05 ***
## year           0.07997    0.01662   4.813 2.51e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.168 on 37 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.385,  Adjusted R-squared:  0.3684
## F-statistic: 23.16 on 1 and 37 DF,  p-value: 2.513e-05
```

Even though the model tells about 38% variability explained by year on the mean temperature but it still gives us small evidence of global warming over the time. Min temprature can be seen having increase over the time from the graph as well.

```
rainfall_data <- read_csv("rainfall.csv")
```

```
## Rows: 31714 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): STATION, STATION_NAME, Measurement Flag
## dbl  (1): HPCP
## lgl  (1): Quality Flag
## dttm (1): DATE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(rainfall_data)
```

```
## Rows: 31,714
## Columns: 6
## $ STATION            <chr> "COOP:190770", "COOP:190770", "COOP:190770", "COOP:~
## $ STATION_NAME       <chr> "BOSTON LOGAN INTERNATIONAL AIRPORT MA US", "BOSTON~
## $ DATE               <dttm> 1985-01-01 01:00:00, 1985-01-01 09:00:00, 1985-01-~
## $ HPCP               <dbl> 0.00, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.0~
## $ `Measurement Flag` <chr> "g", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ `Quality Flag`     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

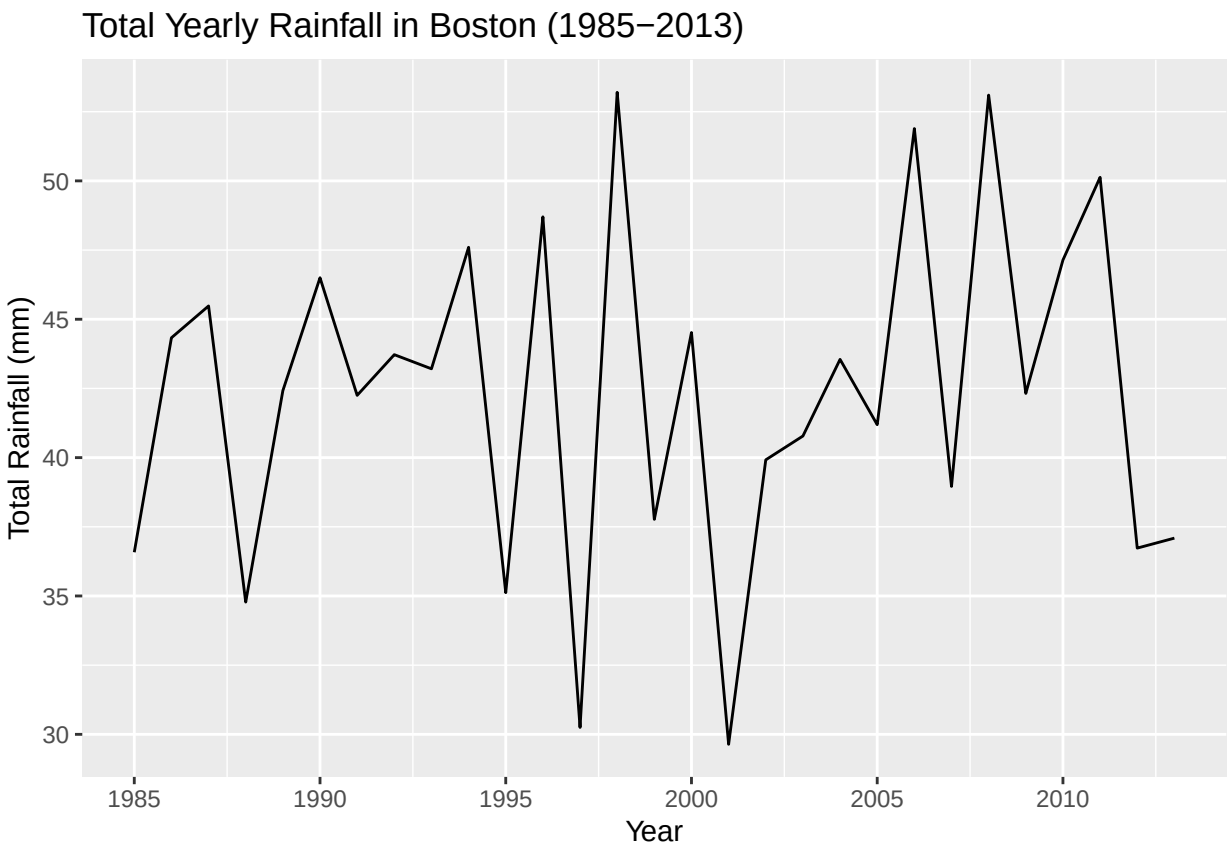
```
glimpse(combined_data)
```

```
## Rows: 465,990
## Columns: 18
## $ MM       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ DD       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ hh       <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,~
```

```
## $ mm       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ WD       <dbl> 60, 80, 100, 100, 110, 90, 60, 30, 40, 40, 50, 60, 70, 70, 70~
## $ WSPD     <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 6, 7, 7, 6, 7, 9, 7, 8, 8, 8, 7, 7, 7~
## $ GST      <dbl> 5, 5, 5, 5, 5, 5, 6, 5, 6, 8, 8, 7, 9, 10, 9, 10, 9, 10, 9, 9~
## $ WVHT     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ DPD      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ APD      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ MWD      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ BAR      <dbl> 1030.3, 1030.0, 1030.1, 1029.4, 1028.6, 1027.8, 1027.7, 1027.~
## $ ATMP     <dbl> 4.7, 5.1, 5.6, 5.8, 5.8, 5.3, 5.5, 5.8, 5.9, 6.2, 6.2, 6.3, 6~
## $ WTMP     <dbl> 6.7, 6.7, 6.6, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.6, 6~
## $ DEWP     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ VIS      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ YYYY     <dbl> 85, 85, 85, 85, 85, 85, 85, 85, 85, 85, 85, 85, 85, 85, 85, 8~
## $ datetime <dttm> 1985-01-01 00:00:00, 1985-01-01 01:00:00, 1985-01-01 02:00:0~
```

```r
rainfall_data <- rainfall_data %>%
  filter(!is.na(HPCP) & !is.na(DATE))
```

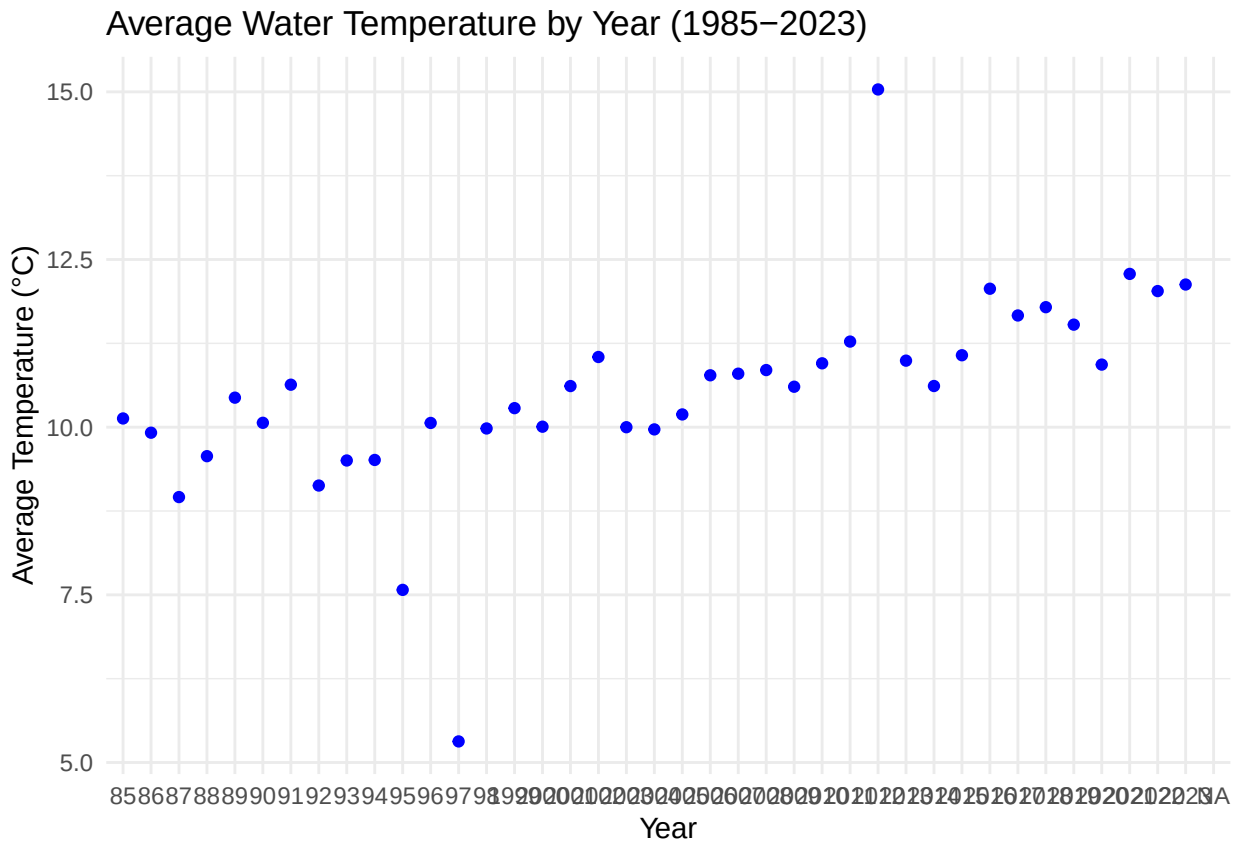
```r
rainfall_data %>%
  mutate(year = year(DATE)) %>%
  group_by(year) %>%
  summarise(total_rainfall = sum(HPCP, na.rm = TRUE)) %>%
  ggplot(aes(x = year, y = total_rainfall)) +
  geom_line() +
  labs(title = "Total Yearly Rainfall in Boston (1985-2013)", x = "Year", y = "Total Rainfall (mm)")
```



```r
yearly_avg_temp <- combined_data %>%
  mutate(year = factor(YYYY)) %>%  # Ensure year is a factor for plotting
  group_by(year) %>%
  summarise(avg_WTMP = mean(WTMP, na.rm = TRUE))

ggplot(yearly_avg_temp, aes(x = year, y = avg_WTMP)) +
  geom_point(color = "blue") +
  labs(title = "Average Water Temperature by Year (1985-2023)",
       x = "Year",
       y = "Average Temperature (°C)") +
  theme_minimal()
```

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



I tried all the three types of model with WTMP, BAR and log model, but neither of them are covering much of variability whereas visualization tells different story as the pattern or rather the line have similar shape if we observe that gave me idea that there might be some relationship between rainfall and Water temperature but model shows something else. This shows that there might be some other variables that can help explain more which I will explore more in depth.

```r
rainfall_data$datetime <- ymd_h(rainfall_data$DATE)
```

```
## Warning: 31201 failed to parse.
```

```r
comb_data <- left_join(combined_data, rainfall_data, by = "datetime")
```

```
## Warning in left_join(combined_data, rainfall_data, by = "datetime"): Detected an unexpected many-to-m
## i Row 185754 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 1 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

```r
model <- lm(HPCP ~ BAR + WTMP + BAR*WTMP, data = comb_data, refresh = 0)
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'refresh' will be disregarded
```

```r
summary(model)
```

```
##
## Call:
## lm(formula = HPCP ~ BAR + WTMP + BAR * WTMP, data = comb_data,
##     refresh = 0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.03738 -0.02855 -0.02448  0.00456  0.50255
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.393e-01  6.900e-01   0.347    0.729
## BAR         -2.117e-04  6.814e-04  -0.311    0.756
## WTMP        -3.278e-02  6.424e-02  -0.510    0.610
## BAR:WTMP     3.273e-05  6.339e-05   0.516    0.606
##
## Residual standard error: 0.05654 on 870 degrees of freedom
##   (995516 observations deleted due to missingness)
## Multiple R-squared:  0.001772,   Adjusted R-squared:  -0.00167
## F-statistic: 0.5149 on 3 and 870 DF,  p-value: 0.6721
```