MA615_HW4_JF

2024-09-25

Question a

```
# Load 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(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
```

```
# Create a function to read buoy data for a given year
read_buoy_data <- function(year) {
  file_root <- "https://www.ndbc.noaa.gov/view_text_file.php?filename=44013h"
  tail <- ".txt.gz&dir=data/historical/stdmet/"
  path <- paste0(file_root, year, tail)

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

  # Determine number of lines to skip
  skip_lines <- if(year >= 2007) 2 else 1

  # Read data
  buoy <- tryCatch({
    fread(path, header = FALSE, skip = skip_lines, fill = TRUE)
  }, error = function(e) {
    message("Error reading data for year ", year, ": ", e$message)
    return(NULL)
  })

  if (is.null(buoy)) return(NULL)

  # Ensure consistent number of columns
  expected_cols <- length(header)
  if (ncol(buoy) > expected_cols) {
    buoy <- buoy[, 1:expected_cols, with = FALSE]
  } else if (ncol(buoy) < expected_cols) {
    for (i in (ncol(buoy) + 1):expected_cols) {
      buoy[[paste0("V", i)]] <- NA
    }
  }

  # Set column names
  setnames(buoy, header)

  # Add date column
  buoy$DATE <- ymd(paste(buoy$YY, buoy$MM, buoy$DD, sep = "-"))

  return(buoy)
}

# Read data for all years from 1985 to 2023
all_buoy_data <- lapply(1985:2023, read_buoy_data)
```

```
## 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>>
```

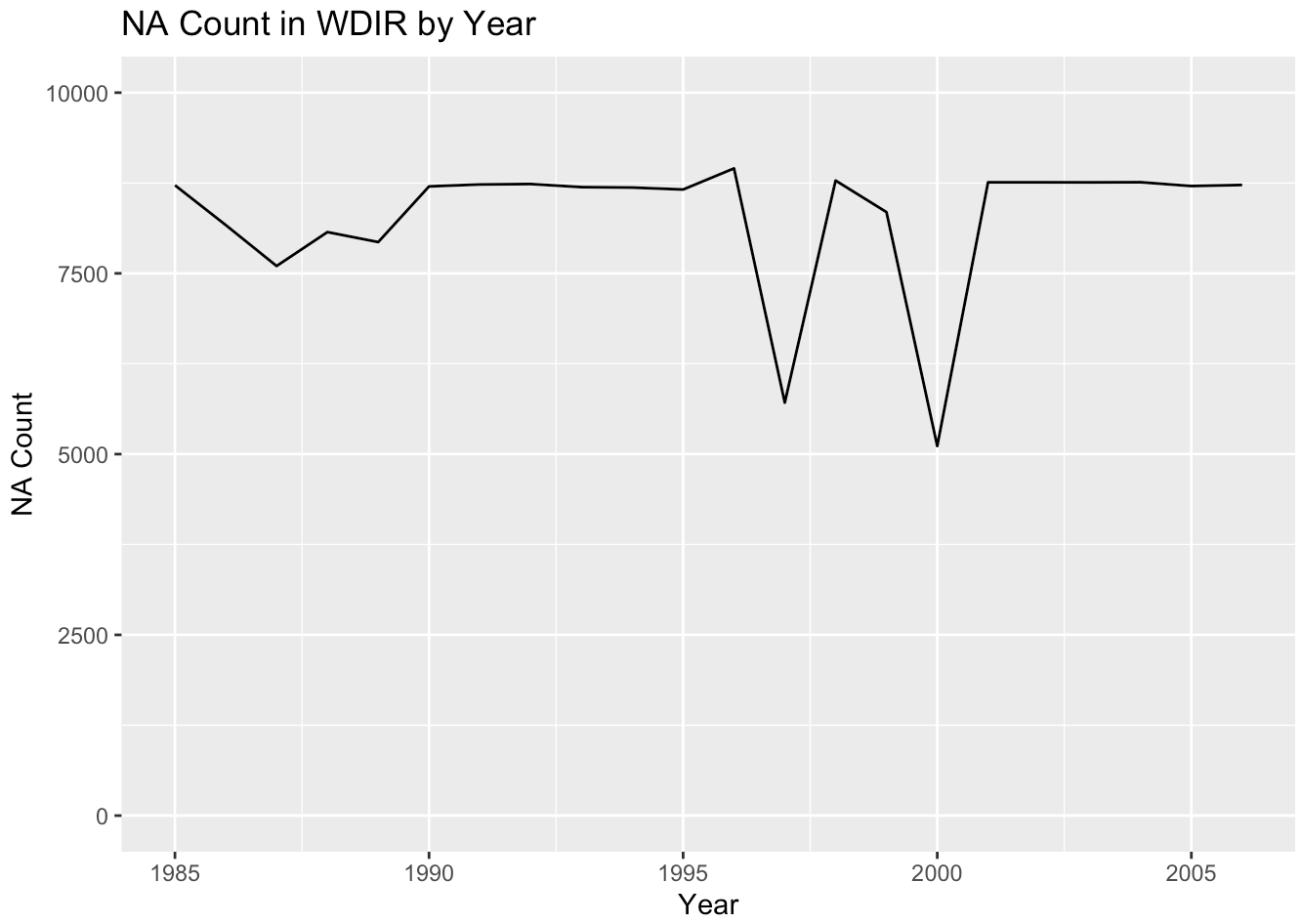
```
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
```

```
# Remove any NULL entries (years where reading failed)
all_buoy_data <- all_buoy_data[!sapply(all_buoy_data, is.null)]

# Combine all years into one dataset
combined_buoy_data <- rbindlist(all_buoy_data, fill = TRUE)
```

Question b

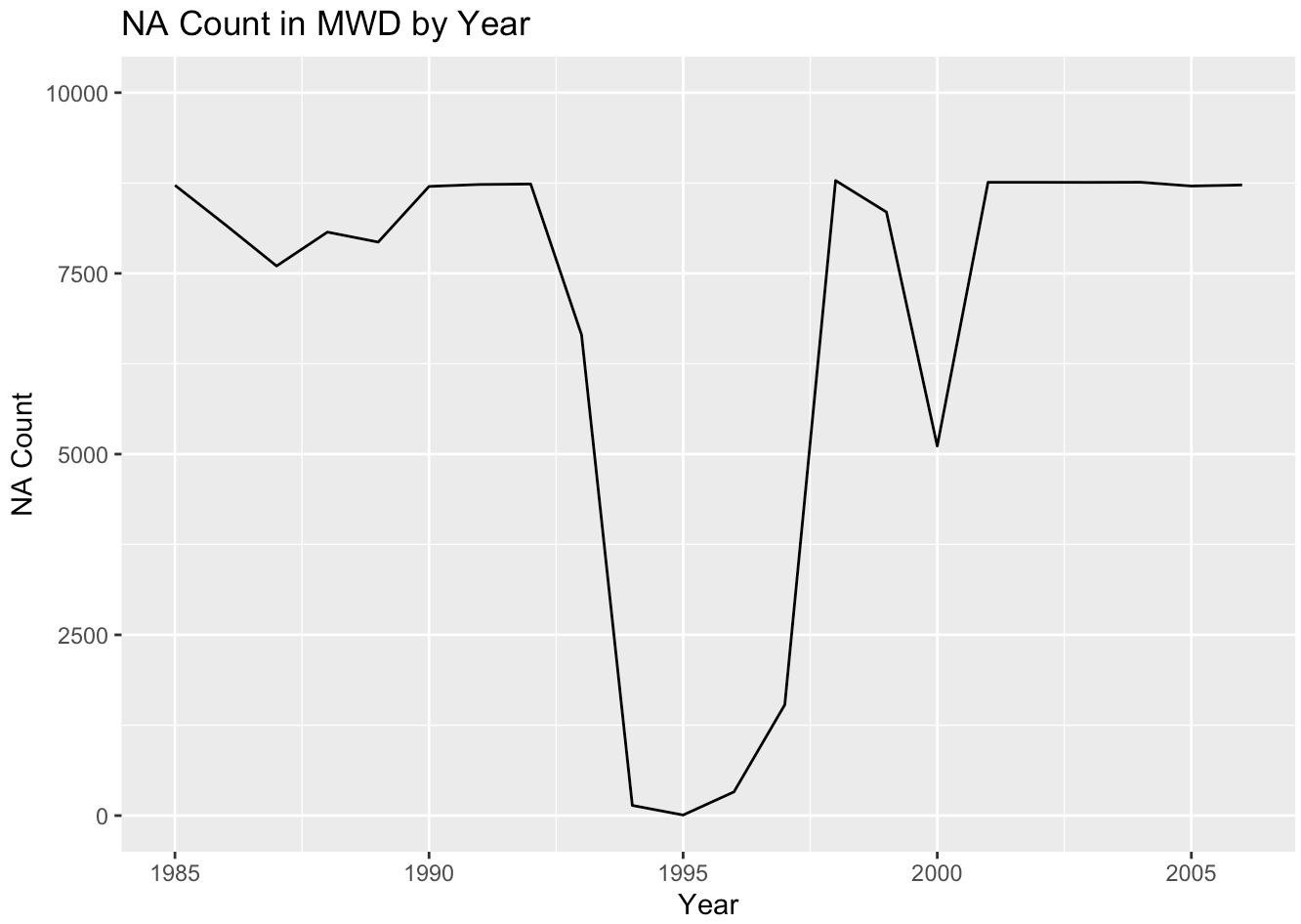
```
# Load libraries
library(ggplot2)

# Convert 999 to NA for WDIR, MWD, and DEWP
combined_buoy_data$WDIR[combined_buoy_data$WDIR == 999] <- NA
combined_buoy_data$MWD[combined_buoy_data$MWD == 999] <- NA
combined_buoy_data$DEWP[combined_buoy_data$DEWP == 999] <- NA

# Analyze NA patterns
na_summary <- combined_buoy_data %>%
  group_by(year(DATE)) %>%
  summarise(across(everything(), ~sum(is.na(.))))

# Visualize NA patterns
ggplot(na_summary, aes(x = `year(DATE)`, y = WDIR)) +
  geom_line() +
  labs(title = "NA Count in WDIR by Year", x = "Year", y = "NA Count") +
  ylim(0, 10000)
```

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



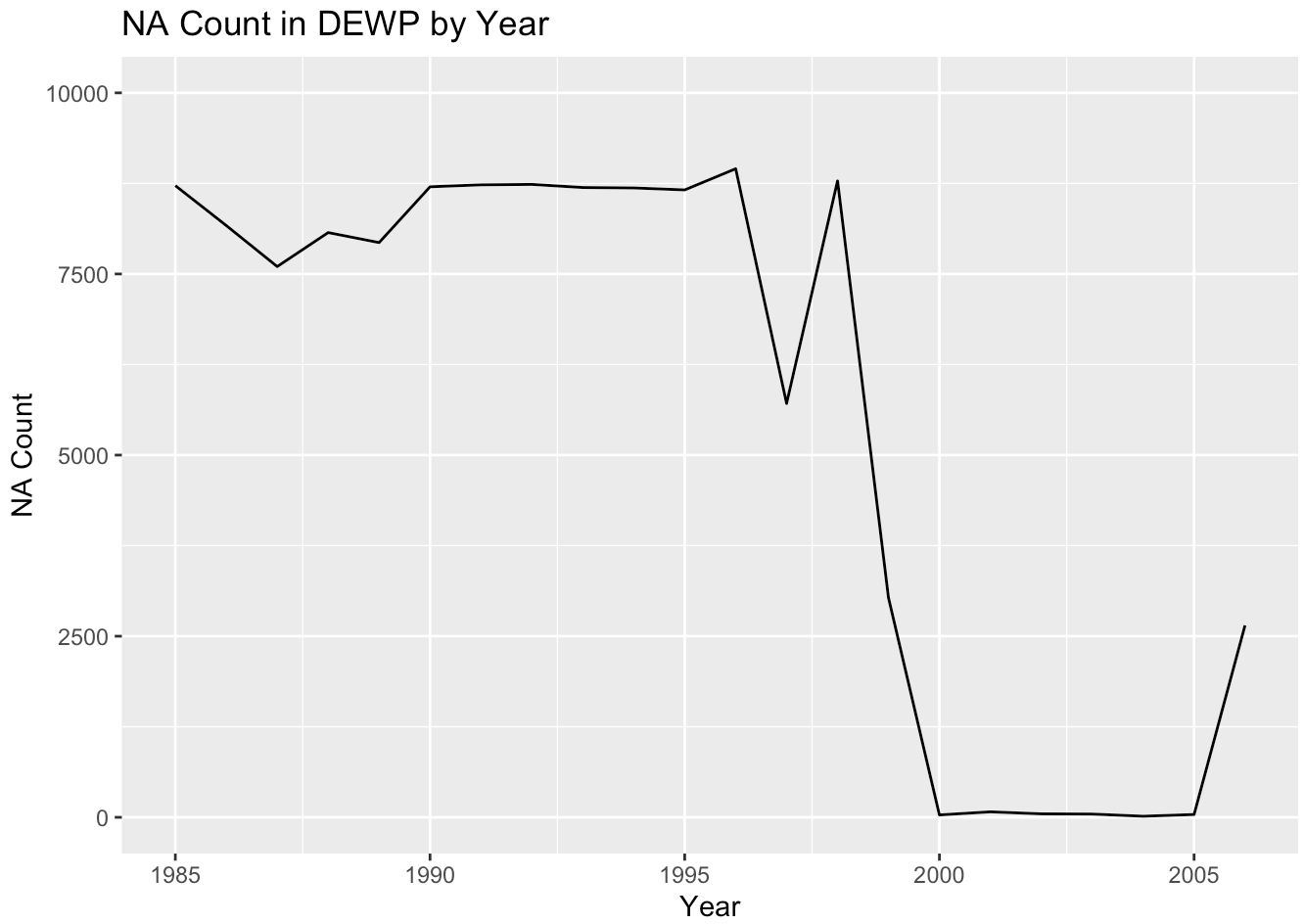

```
ggplot(na_summary, aes(x = `year(DATE)`, y = MWD)) +
  geom_line() +
  labs(title = "NA Count in MWD by Year", x = "Year", y = "NA Count") +
  ylim(0, 10000)
```

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




```
ggplot(na_summary, aes(x = `year(DATE)`, y = DEWP)) +
  geom_line() +
  labs(title = "NA Count in DEWP by Year", x = "Year", y = "NA Count") +
  ylim(0, 10000)
```

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




```
# Save as CSV file
write.csv(combined_buoy_data, "combined_buoy_cleaned_data.csv")
```

1. It is not always appropriate to convert missing or null data to NA's. For example, when working with certain statistical methods, some methods handle missing data differently from NA values.
2. The bar plot was generated to show the number of NA values for variables 'WDIR', 'MWD', and 'DEWP'. For WDIR, the number of missing values fluctuates between 7500 and 10000 over the year. There is a sharp decline around 1999 and 2001, with the NA count dropping below 5000. For MWD, the NA count starts similarly to WDIR, but shows more dramatic fluctuations. A significant decline is seen around the mid-1990s, where the NA count drops to nearly zero by 1995. After 1995, there is a sharp rise back to around 10000 NAs, followed by a pattern of fluctuation similar to the WDIR graph, with another dip around 2001. For DEWP, from 1985 to 1999, the missing value count fluctuates between 7500 and 10000, similar to the other variables (WDIR and MWD). Around 2000, there is a sharp and consistent drop in missing values, decreasing rapidly from 1999 to around 2000. By 2000, the number of missing values reaches nearly zero. From 2001 to 2005, after the sharp decline, the missing values remain very low, fluctuating around 1000 to 2500 until around 2005, when there is a small uptick in missing values.

Bonus. Some additional data sources such as weather event data, maintenance logs, technology upgrade information, government shutdown data, and NOAA budget data can be added. The observed pattern of missing data in DEWP, WDIR, and MWD from 1985 to 2005 can be explained by NOAA budget fluctuations and government shutdowns. Periods of reduced budget or shutdowns correlate with increases in missing data, while increases in funding and modernization efforts led to a reduction in missing values, particularly after 2000.

Question c

```
# Load libraries
# Install.packages('Kendall')
library(ggplot2)
library(dplyr)
library(lubridate)
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(Kendall)

combined_buoy_cleaned_data <- read.csv("combined_buoy_cleaned_data.csv")

# Prepare the data
combined_buoy_cleaned_data$WSPD[combined_buoy_cleaned_data$WSPD == 99] <- NA

climate_data <- combined_buoy_cleaned_data %>%
  mutate(
    Year = year(DATE),
    Month = month(DATE)
  ) %>%
  group_by(Year, Month) %>%
  summarise(
    AvgWindSpeed = mean(WSPD, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  mutate(Date = as.Date(paste(Year, Month, "01", sep = "-")))

# Wind Speed Trend
wind_speed_plot <- ggplot(climate_data, aes(x = Date, y = AvgWindSpeed)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "green") +
  labs(title = "Average Monthly Wind Speed Over Time",
       x = "Year",
       y = "Average Wind Speed (m/s)") +
  theme_minimal()

# Statistical Analysis
wind_speed_model <- lm(AvgWindSpeed ~ Date, data = climate_data)

# Mann-Kendall test for trend
wind_speed_mk <- MannKendall(climate_data$AvgWindSpeed)

# Print results
print(summary(wind_speed_model))
```

```
##
## Call:
## lm(formula = AvgWindSpeed ~ Date, data = climate_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73472 -1.28436  0.07646  1.09742  2.87085
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.385e+00  3.811e-01  16.755   <2e-16 ***
## Date        -3.733e-05  3.876e-05  -0.963    0.336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.413 on 242 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.003819,   Adjusted R-squared:  -0.0002977
## F-statistic: 0.9277 on 1 and 242 DF,  p-value: 0.3364
```

```
print(wind_speed_mk)
```

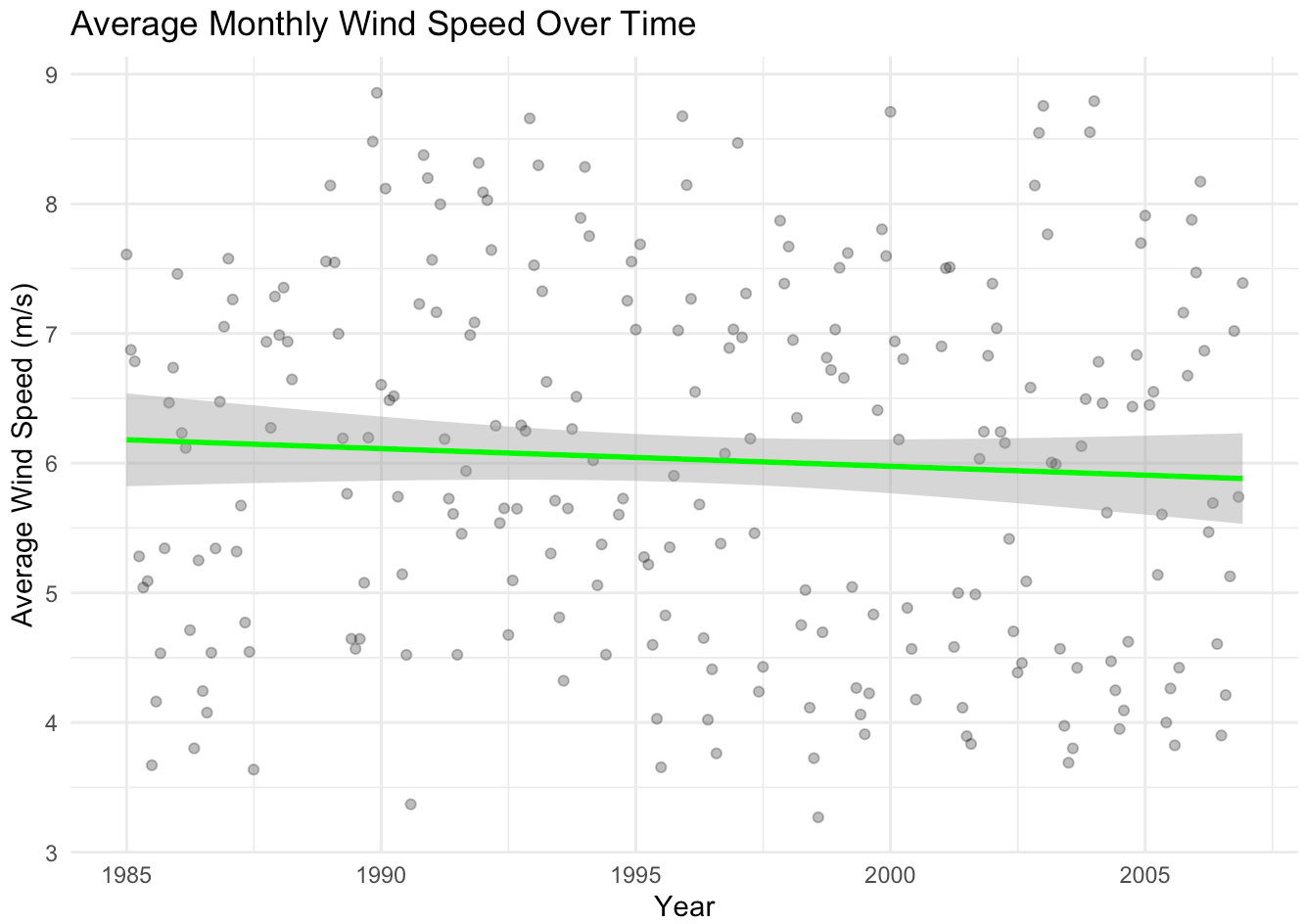
```
## tau = -0.0498, 2-sided pvalue =0.24615
```

```
print(wind_speed_plot)
```

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

```
## Warning: Removed 12 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



The scatter plot shows average monthly wind speeds from 1985 to 2005. There's a slight downward trend visible in the green line, suggesting a minor decrease in average wind speeds over this period.

The coefficient of linear regression for Date is -3.733e-05, indicating a very slight decrease in average wind speed over time. This decrease is NOT statistically significant (p-value = 0.3364), which is more than the common threshold of 0.05. However, the R-squared value is very low (0.003819), meaning only about 0.3819% of the variation in wind speed is explained by time. This suggests a very weak relationship.

The Mann-Kendall test shows a tau value of -0.0498, indicating a weak negative trend. However, the p-value (0.24615) is not significant at the 0.05 level, suggesting we can't reject the null hypothesis of no trend.

Question d

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

# Load the datasets
rainfall <- read.csv("Rainfall.csv")
buoy <- read.csv("combined_buoy_cleaned_data.csv")

# Summarize Rainfall Data
rainfall_summary <- rainfall %>%
  summarize(
    total_measurements = n(),
    mean_precipitation = mean(HPCP, na.rm = TRUE),
    median_precipitation = median(HPCP, na.rm = TRUE),
    max_precipitation = max(HPCP, na.rm = TRUE)
  )

# Summarize Buoy Data
buoy_summary <- buoy %>%
  summarize(
    mean_wind_speed = mean(WSPD, na.rm = TRUE),
    mean_wave_height = mean(WVHT, na.rm = TRUE),
    mean_air_temp = mean(ATMP, na.rm = TRUE),
    mean_water_temp = mean(WTMP, na.rm = TRUE)
  )

# Print summaries
rainfall_summary
```

total_measurements <int>	mean_precipitation <dbl>	median_precipitation <dbl>	max_precipitation <dbl>
31714	0.0387485	0.01	2.03

1 row

```
buoy_summary
```

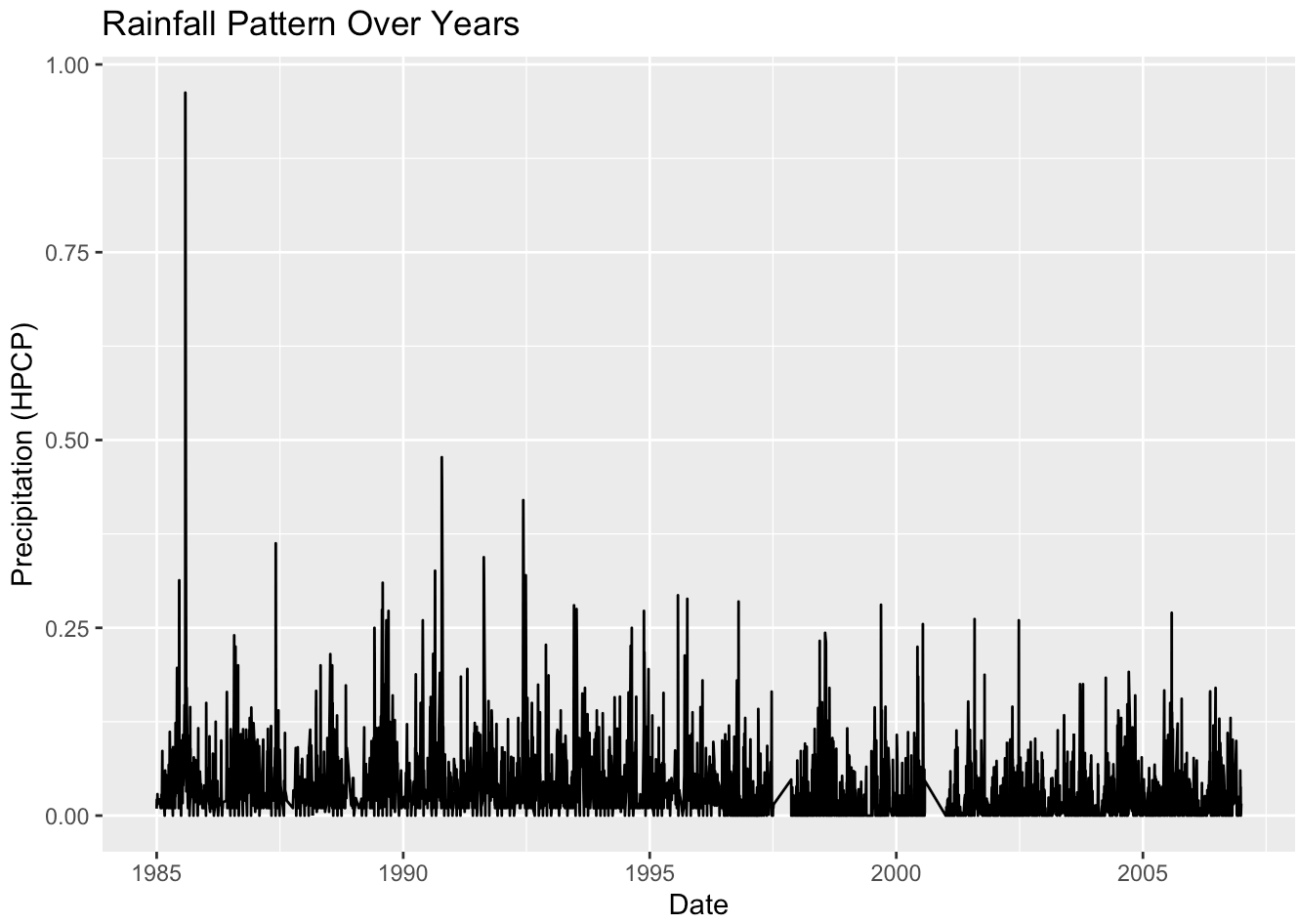
mean_wind_speed <dbl>	mean_wave_height <dbl>	mean_air_temp <dbl>	mean_water_temp <dbl>
12.58519	31.49111	229.7276	39.22387

1 row

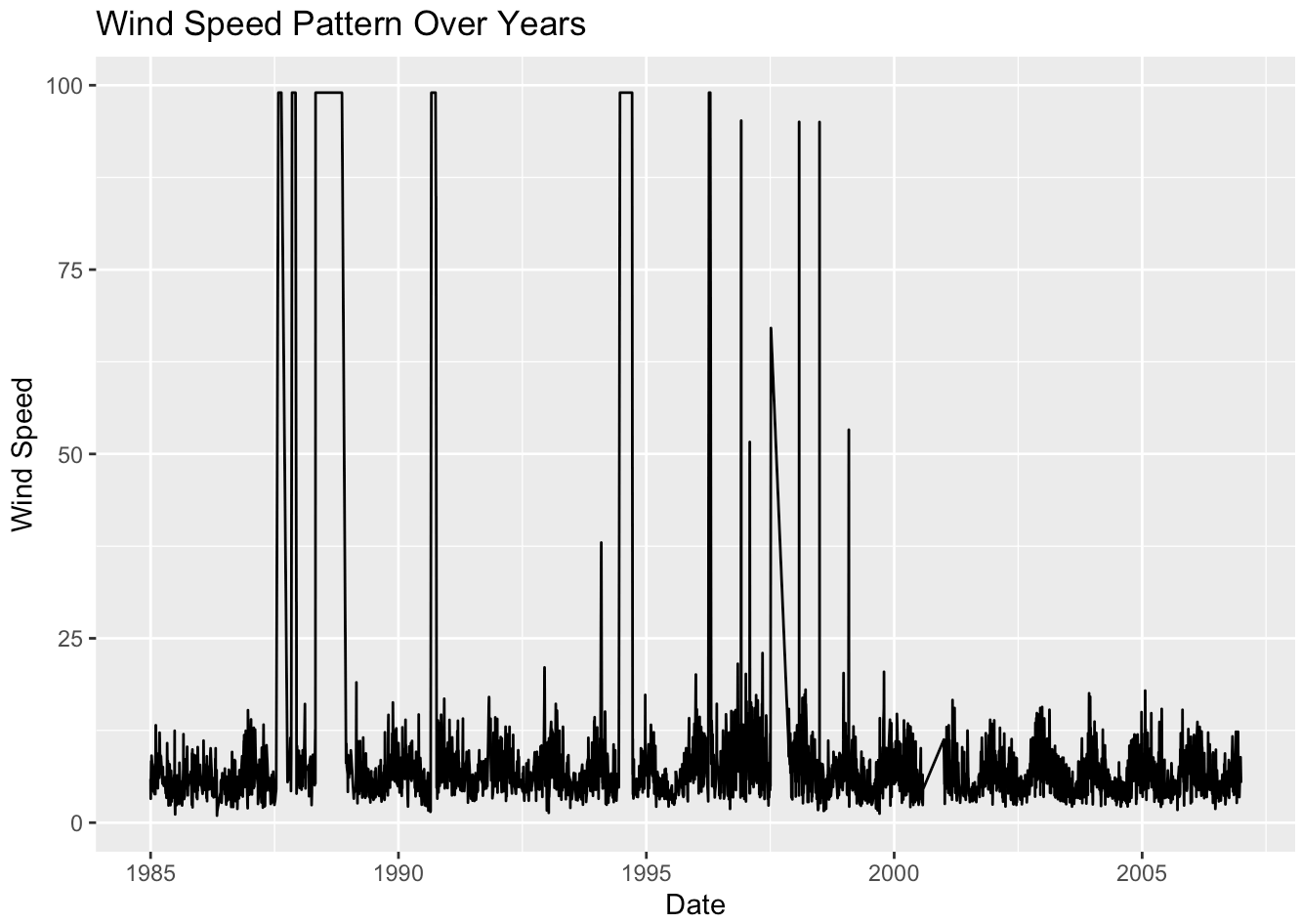
```
# Convert to proper date format for rainfall data
rainfall$DATE <- as.Date(substr(rainfall$DATE, 1, 8), format = "%Y%m%d")

# Group by DATE and calculate the daily mean for rainfall
rainfall_daily_mean <- rainfall %>%
  group_by(DATE) %>%
  summarize(across(.cols = where(is.numeric), .fns = mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarize()`.
## ℹ In argument: `across(.cols = where(is.numeric), .fns = mean, na.rm = TRUE)`.
## ℹ In group 1: `DATE = 1985-01-01`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

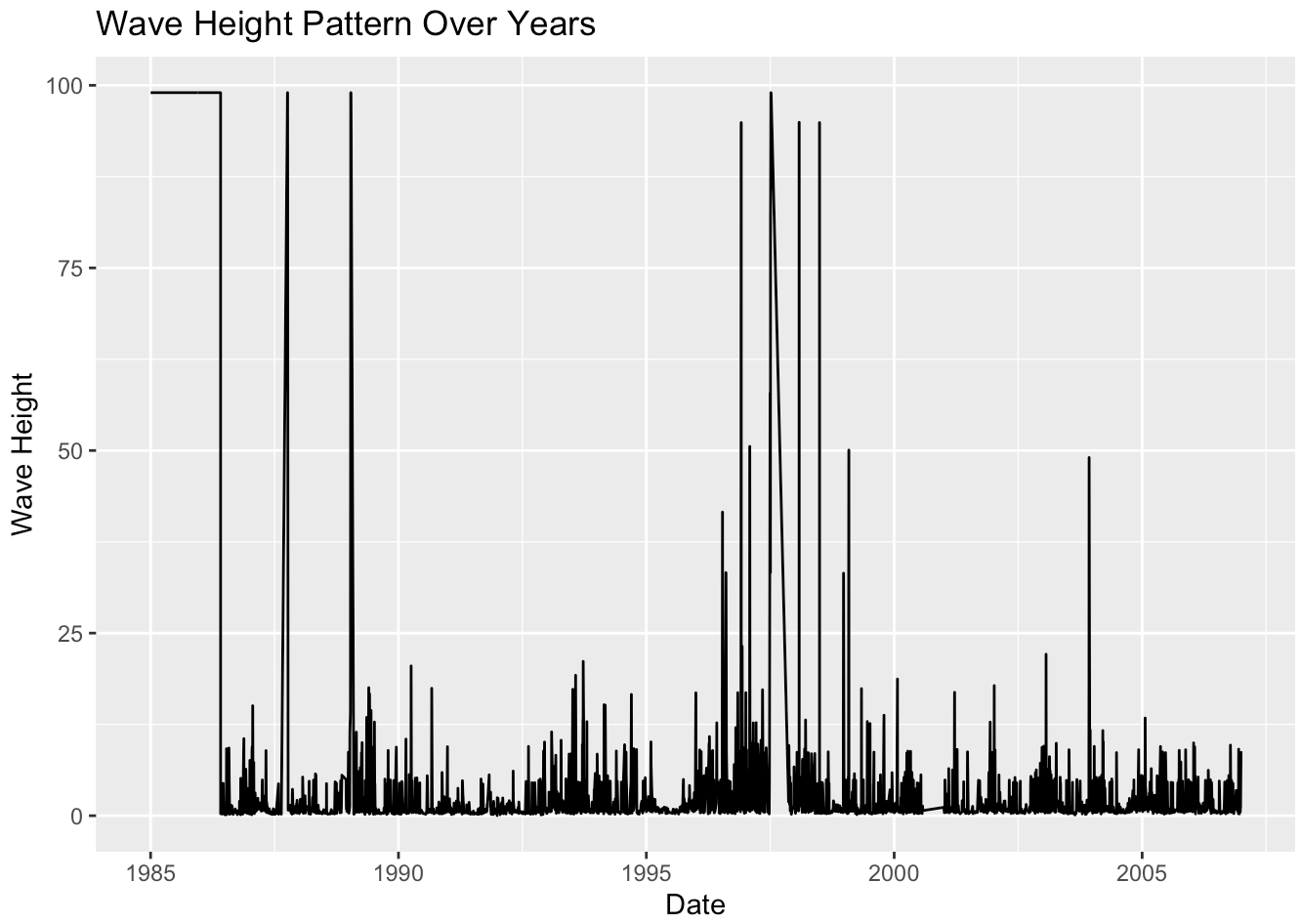
```
# Group by DATE and calculate the daily mean for buoy data
buoy_daily_mean <- buoy %>%
  group_by(DATE) %>%
  summarize(across(.cols = where(is.numeric), .fns = mean, na.rm = TRUE))

# Merge the two datasets on the DATE column
daily_mean_data <- merge(rainfall_daily_mean, buoy_daily_mean, by = "DATE")

# Plot the pattern for specific metrics - HPCP
ggplot(daily_mean_data, aes(x = DATE, y = HPCP)) +
  geom_line() +
  labs(title = "Rainfall Pattern Over Years", x = "Date", y = "Precipitation (HPCP)")
```



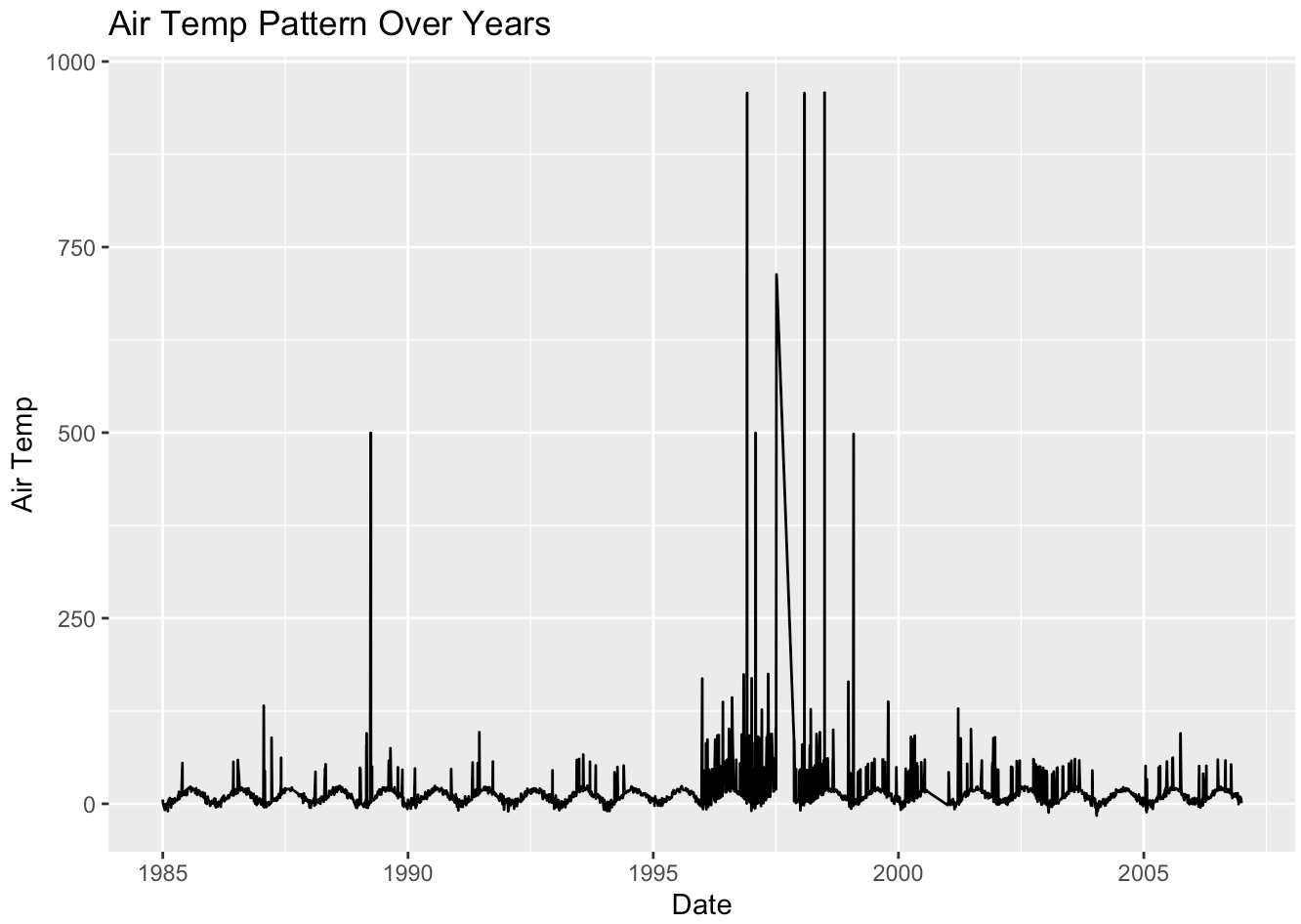

```
# Plot the pattern for specific metrics - WSPD
ggplot(daily_mean_data, aes(x = DATE, y = WSPD)) +
  geom_line() +
  labs(title = "Wind Speed Pattern Over Years", x = "Date", y = "Wind Speed")
```



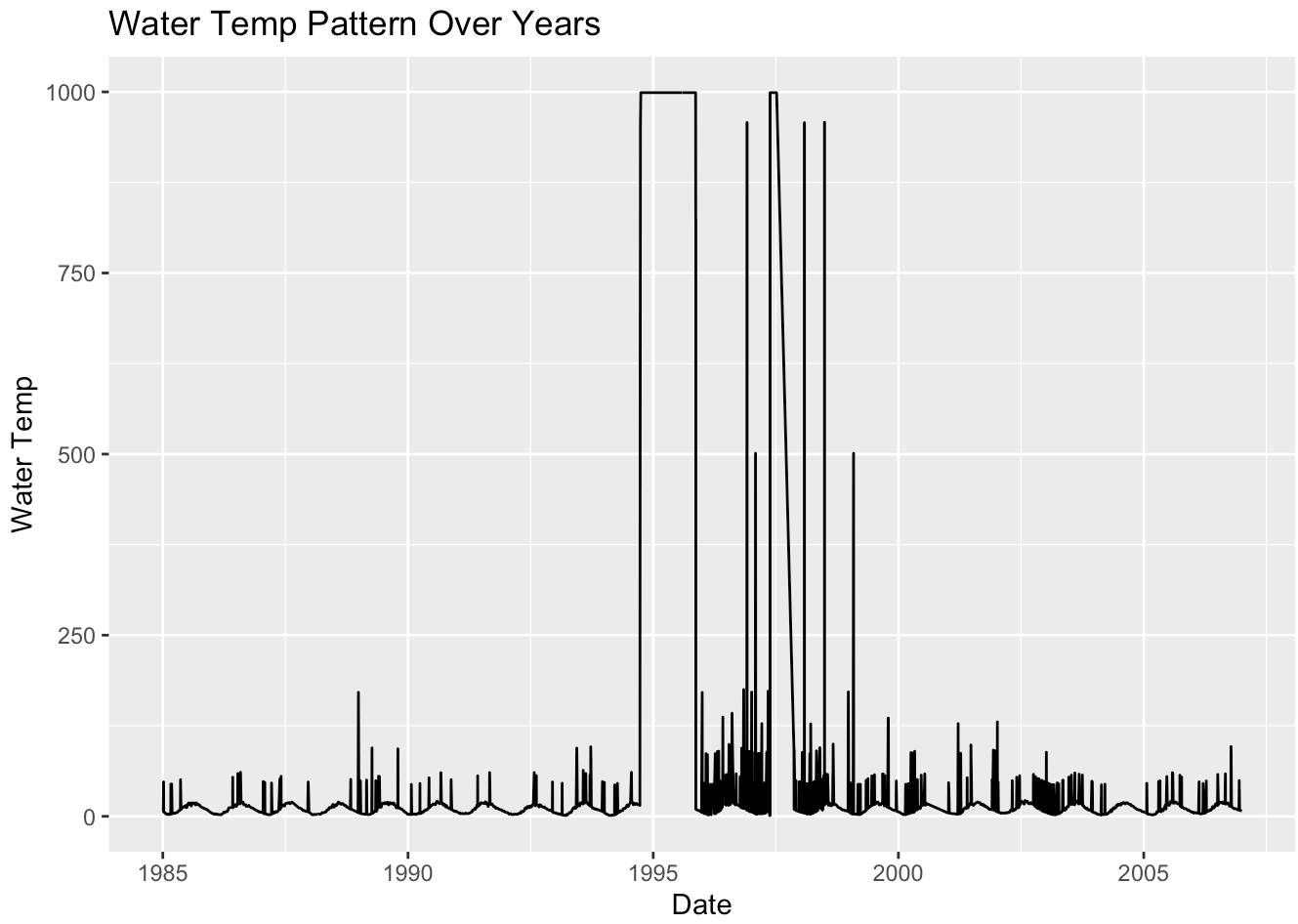

```
# Plot the pattern for specific metrics - WVHT
ggplot(daily_mean_data, aes(x = DATE, y = WVHT)) +
  geom_line() +
  labs(title = "Wave Height Pattern Over Years", x = "Date", y = "Wave Height")
```




```
# Plot the pattern for specific metrics - ATMP
ggplot(daily_mean_data, aes(x = DATE, y = ATMP)) +
  geom_line() +
  labs(title = "Air Temp Pattern Over Years", x = "Date", y = "Air Temp")
```




```
# Plot the pattern for specific metrics - WTMP
ggplot(daily_mean_data, aes(x = DATE, y = WTMP)) +
  geom_line() +
  labs(title = "Water Temp Pattern Over Years", x = "Date", y = "Water Temp")
```




```
# Build a simple linear model to predict rainfall from four metrics
model1 <- lm(HPCP ~ WSPD, data = daily_mean_data)
model2 <- lm(HPCP ~ WVHT, data = daily_mean_data)
model3 <- lm(HPCP ~ ATMP, data = daily_mean_data)
model4 <- lm(HPCP ~ WTMP, data = daily_mean_data)

# Summary of the model
summary(model1)
```

```
##
## Call:
## lm(formula = HPCP ~ WSPD, data = daily_mean_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05120 -0.03002 -0.01537  0.01115  0.92901
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.191e-02  9.680e-04  32.961  < 2e-16 ***
## WSPD        1.948e-04  4.422e-05   4.406 1.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04825 on 3270 degrees of freedom
## Multiple R-squared:  0.005901,   Adjusted R-squared:  0.005597
## F-statistic: 19.41 on 1 and 3270 DF,  p-value: 1.089e-05
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = HPCP ~ WVHT, data = daily_mean_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.04268 -0.03002 -0.01589  0.01092  0.91982
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.326e-02  8.919e-04  37.296  < 2e-16 ***
## WVHT        9.510e-05  3.683e-05   2.582  0.00986 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04835 on 3270 degrees of freedom
## Multiple R-squared:  0.002035,   Adjusted R-squared:  0.00173
## F-statistic: 6.667 on 1 and 3270 DF,  p-value: 0.009862
```

```
summary(model3)
```

```
##
## Call:
## lm(formula = HPCP ~ ATMP, data = daily_mean_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06741 -0.02974 -0.01531  0.01121  0.92845
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.346e-02  9.077e-04  36.869   <2e-16 ***
## ATMP        3.545e-05  2.185e-05   1.622    0.105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04838 on 3270 degrees of freedom
## Multiple R-squared:  0.000804,   Adjusted R-squared:  0.0004984
## F-statistic: 2.631 on 1 and 3270 DF,  p-value: 0.1049
```

```
summary(model4)
```

```
##
## Call:
## lm(formula = HPCP ~ WTMP, data = daily_mean_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.04119 -0.03023 -0.01536  0.01121  0.92889
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.350e-02  8.824e-04  37.967   <2e-16 ***
## WTMP        7.696e-06  3.898e-06   1.975   0.0484 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04837 on 3270 degrees of freedom
## Multiple R-squared:  0.001191,   Adjusted R-squared:  0.0008855
## F-statistic: 3.899 on 1 and 3270 DF,  p-value: 0.0484
```

Model1: Wind speed has a statistically significant relationship with rainfall, but the effect size is very small. The model explains only 0.59% of the variance in rainfall, indicating that wind speed alone is not a strong predictor of rainfall.

Model2: Wave height also has a significant relationship with rainfall, but like wind speed, the model explains only 0.2% of the variance. This suggests that while wave height has an effect on rainfall, it's minimal.

Model3: Air temperature does not have a significant relationship with rainfall. The model explains only 0.08% of the variance, and the p-value suggests that any relationship between air temperature and rainfall is likely due to chance.

Model4: Water temperature has a statistically significant but very small effect on rainfall. The model explains only 0.12% of the variance, which is still negligible.

Yes, this exercise definitely highlights why weather forecasting can be so challenging. While building these models, we saw that even key weather metrics like wind speed, wave height, air temperature, and water temperature explain only a tiny fraction of the variability in rainfall. Despite being significant in some cases, the effect sizes are extremely small, meaning that these factors alone aren't enough to reliably predict rainfall.

```
# rmarkdown::render("MA615_HW4_JF.Rmd", output_format = "pdf_document")
```