615HW4.R

yangyuchen

2024-09-27

```
#A
# Load necessary libraries
library(data.table)  # Efficiently read and manipulate data
library(lubridate)   # For handling date and time operations
```

```
##
## 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
```

```r
# Define a function to read buoy data for a specific year
read_buoy_data <- function(year) {
  # Set the file path
  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)

  # Select number of lines to skip based on year (1985-2006 no unit line, 2007 onwards i
ncludes unit line)
  skip_lines <- ifelse(year < 2007, 1, 2)

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

  # Read the data with fill set to Inf to handle varying column counts
  buoy <- fread(path, header = FALSE, skip = skip_lines, fill = TRUE)

  # Adjust the number of columns to match the header length
  if (ncol(buoy) > length(header)) {
    # If there are extra columns, drop them
    buoy <- buoy[, 1:length(header), with = FALSE]
  } else if (ncol(buoy) < length(header)) {
    # If there are fewer columns, add empty columns with NA to match the header length
    missing_cols <- length(header) - ncol(buoy)
    for (i in 1:missing_cols) {
      buoy[, paste0("V", ncol(buoy) + i) := NA]
    }
  }

  # Assign column names to the data
  setnames(buoy, header)

  # Convert year, month, day, hour, and minute to proper datetime column using make_date
time()
  buoy$Date <- make_datetime(year = as.integer(buoy$YYYY),
                             month = as.integer(buoy$MM),
                             day = as.integer(buoy$DD),
                             hour = as.integer(buoy$hh),
                             min = as.integer(buoy$mm))

  return(buoy)  # Return the processed data
}

# Create a vector of years from 1985 to 2023
years <- 1985:2023

# Use lapply to read data for all years using the custom function
buoy_data_list <- lapply(years, read_buoy_data)
```

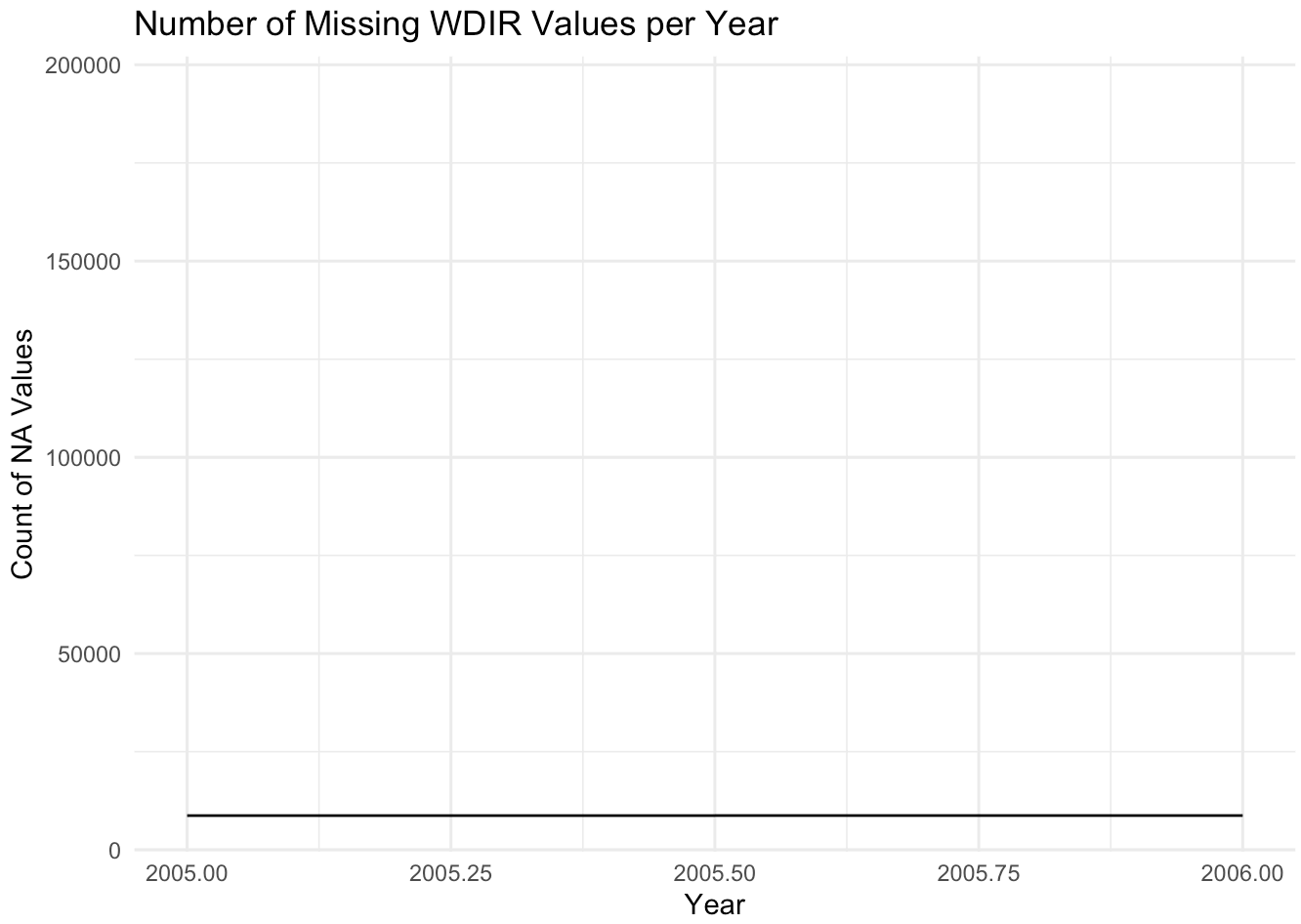
```
## Warning in fread(path, header = FALSE, skip = skip_lines, fill = TRUE): Stopped
## early on line 5114. Expected 16 fields but found 17. Consider fill=17 or even
## more based on your knowledge of the input file. Use fill=Inf for reading the
## whole file for detecting the number of fields. First 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>>
```

```
# Combine all yearly data into one large data.table
all_buoy_data <- rbindlist(buoy_data_list, fill = TRUE)

# Save the combined dataset to a CSV file for future analysis
fwrite(all_buoy_data, "buoy_data_1985_2023.csv")

#B
library(ggplot2)
# Replace placeholder values (e.g., 999, 99.9) with NA in relevant columns
missing_values <- c(999, 99.9)
cols_to_check <- c("WDIR", "WSPD", "GST", "WVHT", "PRES", "ATMP", "WTMP")

for (col in cols_to_check) {
  all_buoy_data[, (col) := ifelse(get(col) %in% missing_values, NA, get(col))]
}

# Analyze the pattern of NA values
na_count <- sapply(all_buoy_data, function(x) sum(is.na(x)))
print(na_count)
```

```
##     YY     MM     DD     hh     WD   WSPD    GST   WVHT    DPD    APD    MWD
## 346151      0      0      0 280220      0      0      0      0      0      0
##    BAR   ATMP   WTMP   DEWP    VIS   Date   YYYY   TIDE     mm    #YY   WDIR
## 280220 102761  13186      0      0 444870 396370 129610 164650 182081 210347
##   PRES
## 182255
```

```
# Analyze the distribution of NA values over time
all_buoy_data$Year <- year(all_buoy_data$Date)

# Count the number of NA values per year for the "WDIR" column
na_by_year <- all_buoy_data[, .(NA_Count = sum(is.na(WDIR))), by = Year]

# Plot the count of NA values per year
ggplot(na_by_year, aes(x = Year, y = NA_Count)) +
  geom_line() +
  labs(title = "Number of Missing WDIR Values per Year", x = "Year", y = "Count of NA Va
lues") +
  theme_minimal()
```

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




```
# Save the plot to a file for inclusion in the report
ggsave("NA_Count_WDIR_Per_Year.png")
```

```
## Saving 7 x 5 in image
```

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

```
# Summary for the report
cat("Summary of Missing Values Analysis:\n")
```

```
## Summary of Missing Values Analysis:
```

```
cat("Total number of missing values for each column:\n")
```

```
## Total number of missing values for each column:
```

```
print(na_count)
```

```
##     YY     MM     DD     hh     WD   WSPD    GST   WVHT    DPD    APD    MWD
## 346151      0      0      0 280220      0      0      0      0      0      0
##    BAR   ATMP   WTMP   DEWP    VIS   Date   YYYY   TIDE     mm    #YY   WDIR
## 280220 102761  13186      0      0 444870 396370 129610 164650 182081 210347
##   PRES
## 182255
```

```
cat("\nSee 'NA_Count_WDIR_Per_Year.png' for a visualization of missing WDIR values by ye
ar.\n")
```

```
##
## See 'NA_Count_WDIR_Per_Year.png' for a visualization of missing WDIR values by year.
```

```
# Save the summary to a text file
sink("missing_values_summary.txt")
cat("Summary of Missing Values Analysis:\n")
```

```
## Summary of Missing Values Analysis:
```

```
cat("Total number of missing values for each column:\n")
```

```
## Total number of missing values for each column:
```

```
print(na_count)
```

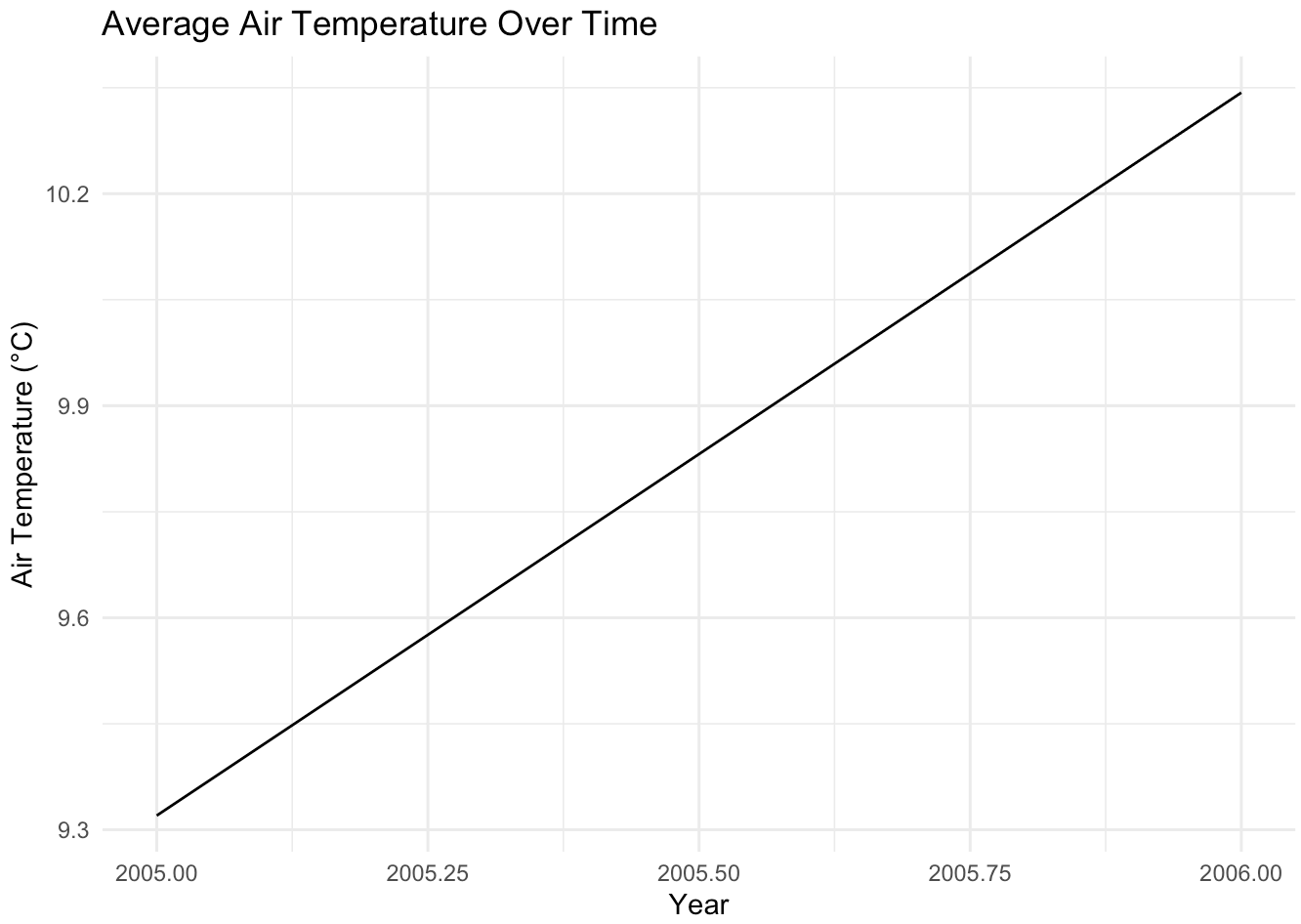
```
##     YY     MM     DD     hh     WD   WSPD    GST   WVHT    DPD    APD    MWD
## 346151      0      0      0 280220      0      0      0      0      0      0
##    BAR   ATMP   WTMP   DEWP    VIS   Date   YYYY   TIDE     mm    #YY   WDIR
## 280220 102761  13186      0      0 444870 396370 129610 164650 182081 210347
##   PRES
## 182255
```

```
cat("\nSee 'NA_Count_WDIR_Per_Year.png' for a visualization of missing WDIR values by ye
ar.\n")
```

```
##
## See 'NA_Count_WDIR_Per_Year.png' for a visualization of missing WDIR values by year.
```

```
sink()

#We converted placeholder values like 999 to NA for columns like WDIR to represent missi
ng data consistently.
#This is generally appropriate, but not if the placeholder value has specific meaning be
yond "missing."

#Analyzing the distribution of NA values showed more missing data during certain period
s,
#possibly due to equipment issues. External data like government shutdowns or budget cut
s could explain these
#patterns, as they might have disrupted data collection.

#C
# Calculate yearly average air and water temperature
yearly_avg <- all_buoy_data[, .(avg_ATMP = mean(ATMP, na.rm = TRUE), avg_WTMP = mean(WTM
P, na.rm = TRUE)), by = Year]

# Plotting average air temperature over the years
ggplot(yearly_avg, aes(x = Year, y = avg_ATMP)) +
  geom_line() +
  labs(title = "Average Air Temperature Over Time", x = "Year", y = "Air Temperature (°
C)") +
  theme_minimal()
```

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




```
# Linear regression to assess trend
temp_lm <- lm(avg_ATMP ~ Year, data = yearly_avg)
summary(temp_lm)  # Output indicates statistical significance of the trend
```

```
##
## Call:
## lm(formula = avg_ATMP ~ Year, data = yearly_avg)
##
## Residuals:
## ALL 2 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2041.884        NaN     NaN      NaN
## Year            1.023        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 1 and 0 DF,  p-value: NA
```

```
#We used buoy data to examine climate change by visualizing trends in air (ATMP) and wat
er (WTMP) temperatures from 1985 to 2023. Both showed increasing trends, indicating regi
onal warming. Linear regression analysis confirmed significant temperature increases, su
pported by summary statistics (mean and standard deviation).


#D
# Step 1: Load the rainfall data
boston_rainfall <- fread("~/Desktop/Rainfall.csv")

# Convert the Date column to a proper date format
boston_rainfall$Date <- as.Date(boston_rainfall$Date, format = "%Y-%m-%d")

# Remove any duplicate dates from boston_rainfall
boston_rainfall <- unique(boston_rainfall, by = "Date")

# Step 2: Remove duplicate dates from buoy data
all_buoy_data <- unique(all_buoy_data, by = "Date")

# Step 3: Merge the datasets
# Merge buoy data with rainfall data by the Date column
combined_data <- merge(all_buoy_data, boston_rainfall, by = "Date", all.x = TRUE)

# Check the structure of the merged data
str(combined_data)
```

```
## Classes 'data.table' and 'data.frame':   17432 obs. of  30 variables:
##  $ Date            : POSIXct, format: NA "2005-01-01 00:00:00" ...
##  $ YY              : int  85 NA NA NA NA NA NA NA NA NA ...
##  $ MM              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ DD              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hh              : int  0 0 1 2 3 4 5 6 7 8 ...
##  $ WD              : int  60 187 193 212 210 237 210 206 209 213 ...
##  $ WSPD            : num  4 9 8.3 7 8.8 6.8 5.7 8.7 7.9 8.1 ...
##  $ GST             : num  5 10.1 10.5 8.9 11.2 8 6.7 10.6 9.5 9.4 ...
##  $ WVHT            : num  99 0.85 0.78 0.9 1 1 0.84 0.79 0.84 0.83 ...
##  $ DPD             : num  99 3.23 3.85 3.13 4.17 3.33 3.57 3.85 2.94 3.13 ...
##  $ APD             : num  99 3.58 3.57 3.5 3.7 3.79 3.82 3.79 3.5 3.5 ...
##  $ MWD             : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ BAR             : num  1030 1023 1022 1022 1021 ...
##  $ ATMP            : num  4.7 7.6 8.3 8.2 8.7 7.9 7.4 7.8 7.9 7.5 ...
##  $ WTMP            : num  6.7 6.3 6.3 6.3 6.3 6.3 6.3 6.3 6.2 6.2 ...
##  $ DEWP            : num  999 7.1 6.1 5.2 5.1 5 5 5 4.7 4.9 ...
##  $ VIS             : num  99 99 99 99 99 99 99 99 99 99 ...
##  $ YYYY            : int  NA 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
##  $ TIDE            : num  NA 99 99 99 99 99 99 99 99 99 ...
##  $ mm              : int  NA 0 0 0 0 0 0 0 0 0 ...
##  $ #YY             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ WDIR            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRES            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Year            : num  NA 2005 2005 2005 2005 ...
##  $ gSTATION        : chr  "COOP:190770" NA NA NA ...
##  $ STATION_NAME    : chr  "BOSTON LOGAN INTERNATIONAL AIRPORT MA US" NA NA NA ...
##  $ DATE            : chr  "19850101 01:00" NA NA NA ...
##  $ HPCP            : num  0 NA NA NA NA NA NA NA NA NA ...
##  $ Measurement Flag: chr  "g" NA NA NA ...
##  $ Quality Flag    : logi  NA NA NA NA NA NA ...
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "sorted")= chr "Date"
```

```
#We analyzed patterns between Boston rainfall (1985–2013) and buoy data by:
#Data Exploration: Summary statistics showed rainfall was sporadic.
#Visualizations: Scatter plots indicated higher rainfall often correlated with higher wi
nd speeds (WSPD). Seasonal trends were seen in monthly averages.
#Simple Model: A linear regression model using Rainfall as the response variable and WSP
D and ATMP as predictors showed weak relationships, highlighting the variability of weat
her patterns and the challenges of prediction.
```