Data Cleaning and Aggregation Process

1. **Combining Observations**: Grouping observations within the same hour by station.
2. **Output Observation Count**: Count the number of observations by station.
3. **Station Filtering**: Remove stations with fewer than 7008 (80%) of the data points.

After Cleaning:

- **LA_temp_2023_cleaned.csv**
  - Original stations: 13
  - Stations retained: 10
- **LA_temp_2022_cleaned.csv**
  - Original stations: 13
  - Stations retained: 10
- **LA_temp_2021_cleaned.csv**
  - Original stations: 13
  - Stations retained: 10
- **LA_temp_2020_cleaned.csv**
  - Original stations: 13
  - Stations retained: 9

Question: For daily temperature summaries, should we: - Take the average of the maximum temperatures recorded across all stations? - Or take the highest temperature recorded from any of the stations?

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(readr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

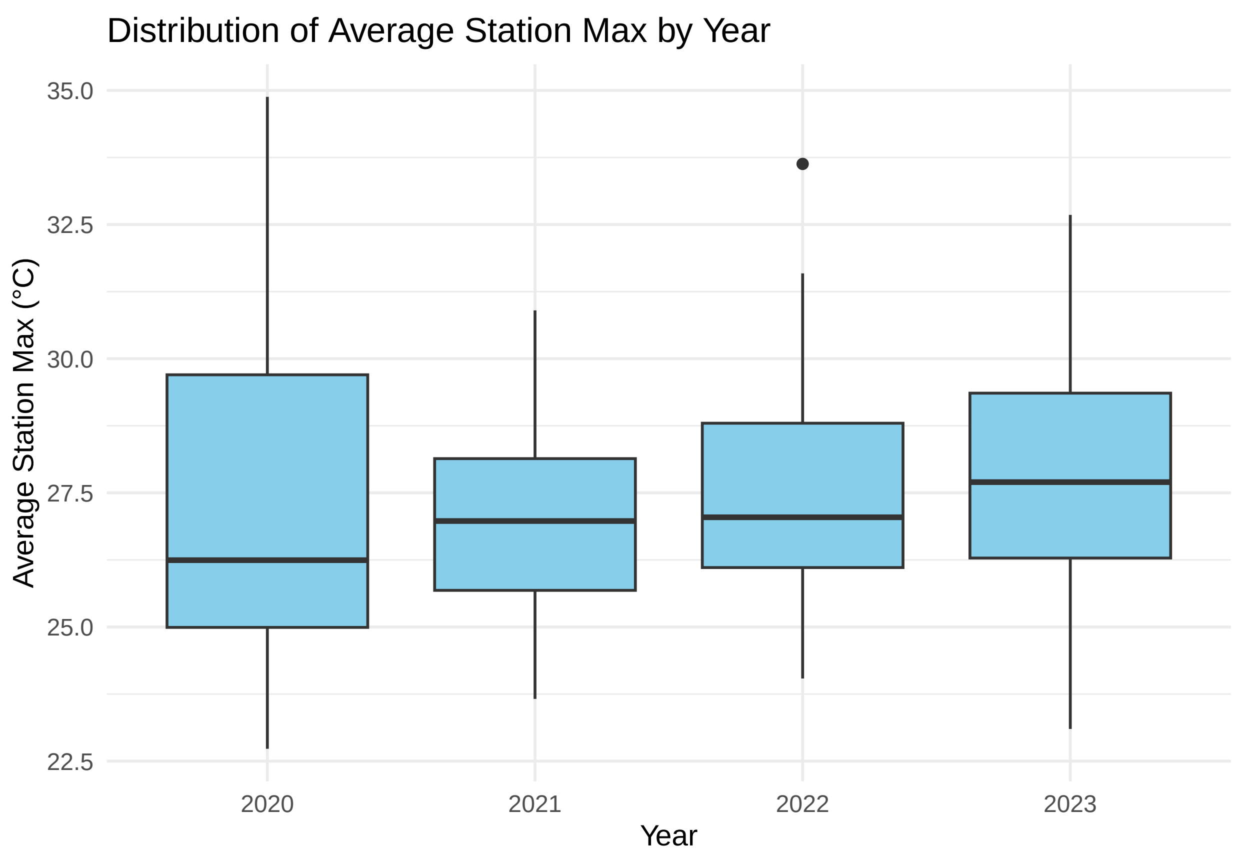
```r
library(ggplot2)

setwd("./Station_New")
files <- list.files(pattern = "*_daily_max.csv")


combined_data <- data.frame()
for (file in files) {
  print(paste("Reading:", file))
  data <- read_csv(file)
  data$DATE <- as.Date(data$DATE)
  combined_data <- bind_rows(combined_data, data)
}
```

```
## [1] "Reading: LA_temp_2020_cleaned_daily_max.csv"

## Rows: 366 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (2): avg_station_max, overall_max
## date (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.

## [1] "Reading: LA_temp_2021_cleaned_daily_max.csv"

## Rows: 365 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (2): avg_station_max, overall_max
## date (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.

## [1] "Reading: LA_temp_2022_cleaned_daily_max.csv"

## Rows: 365 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (2): avg_station_max, overall_max
## date (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.

## [1] "Reading: LA_temp_2023_cleaned_daily_max.csv"
## Rows: 365 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (2): avg_station_max, overall_max
## date (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.
```

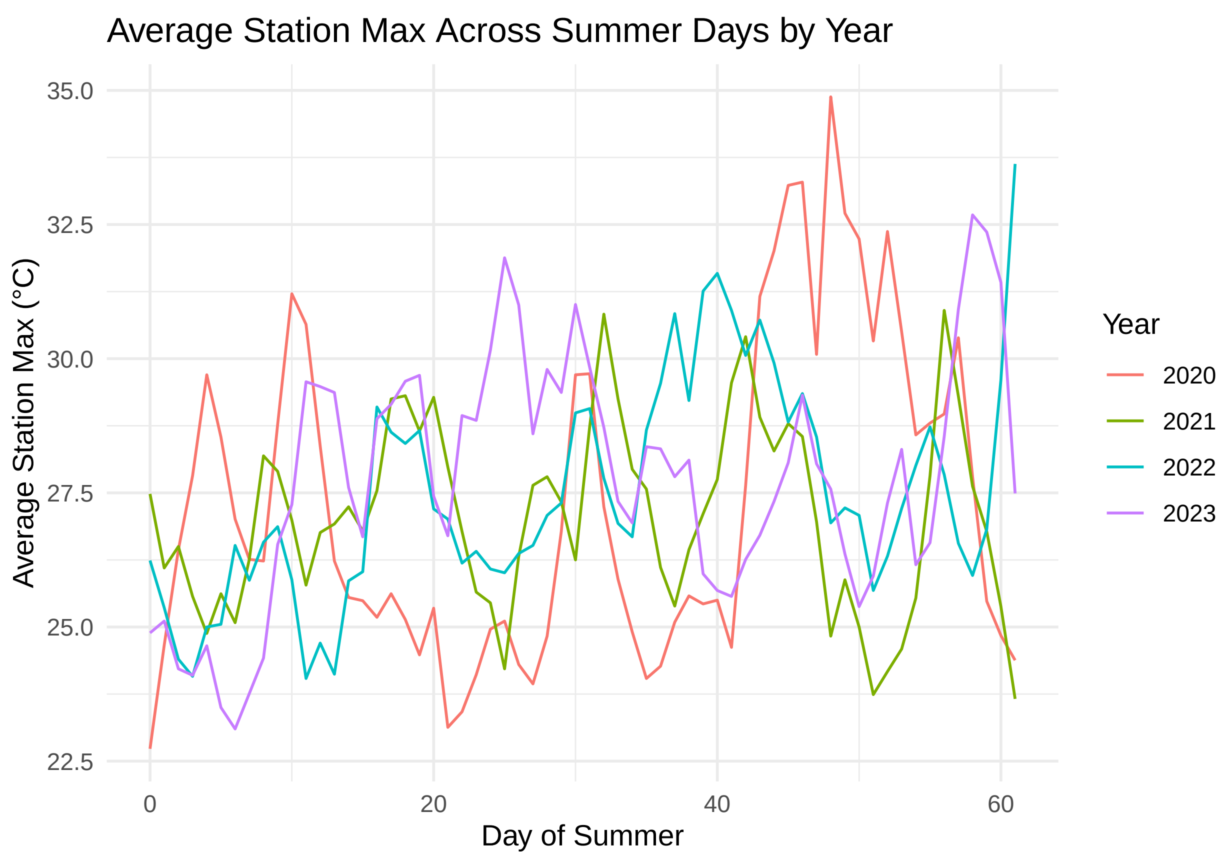
```
summer_data <- combined_data %>%
  mutate(
    year = year(DATE),
    month = month(DATE)
  ) %>%
  filter(month %in% c(7, 8))


summer_summary <- summer_data %>%
  group_by(year) %>%
  summarize(
    avg_daily_station_max = round(mean(avg_station_max, na.rm = TRUE), 2),
    avg_daily_overall_max = round(mean(overall_max, na.rm = TRUE), 2),
    n_days = n()
  ) %>%
  arrange(desc(avg_daily_station_max))

summer_summary
```

```
## # A tibble: 4 x 4
##    year avg_daily_station_max avg_daily_overall_max n_days
##   <dbl>                 <dbl>                 <dbl>  <int>
## 1  2023                  27.8                  35.0     62
## 2  2022                  27.5                  34.5     62
## 3  2020                  27.3                  34.4     62
## 4  2021                  27.0                  34.0     62
```

```
ggplot(summer_data, aes(x = factor(year), y = avg_station_max)) +
  geom_boxplot(fill = "skyblue") +
  labs(
    title = "Distribution of Average Station Max by Year",
    x = "Year",
    y = "Average Station Max (°C)"
  ) +
  theme_minimal()
```



```r
summer_data <- summer_data %>%
  mutate(
    day_of_year = yday(DATE) - yday(as.Date(paste0(year, "-07-01")))
  )

ggplot(summer_data, aes(x = day_of_year, y = avg_station_max, color = factor(year))) +
  geom_line() +
  labs(
    title = "Average Station Max Across Summer Days by Year",
    x = "Day of Summer",
    y = "Average Station Max (°C)",
    color = "Year"
  ) +
  theme_minimal()
```

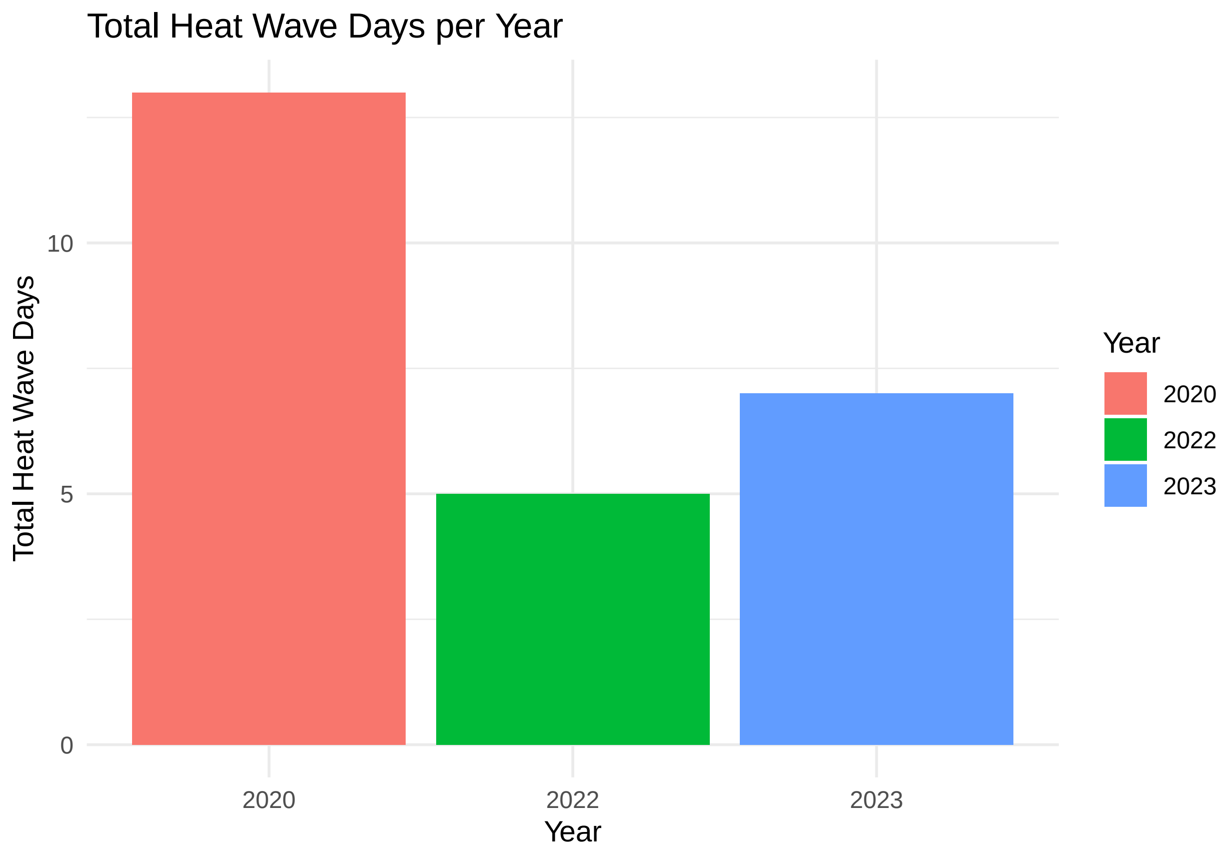


```
summer_data <- summer_data %>%
  mutate(
    is_heat_wave = avg_station_max > 30
  )

summer_data <- summer_data %>%
  mutate(
    heat_wave_group = cumsum(!is_heat_wave)
  ) %>%
  group_by(heat_wave_group) %>%
  mutate(
    heat_wave_id = ifelse(is_heat_wave, cur_group_id(), NA_integer_)
  ) %>%
  ungroup()


heat_wave_data <- summer_data %>%
  filter(is_heat_wave) %>%
  arrange(heat_wave_id, DATE)


heat_wave_summary <- heat_wave_data %>%
  group_by(heat_wave_id, year) %>%
  summarize(
    start_date = min(DATE),
    end_date = max(DATE),
```

```r
    duration_days = as.integer(end_date - start_date) + 1,
    .groups = 'drop'
  ) %>%
  filter(duration_days >= 2) %>%  # **Exclude** heat waves lasting only one day
  arrange(year, desc(duration_days)) %>%
  ungroup()

heat_wave_summary
```

```
## # A tibble: 5 x 5
##   heat_wave_id  year start_date end_date   duration_days
##          <int> <dbl> <date>     <date>             <dbl>
## 1           41  2020 2020-08-13 2020-08-23            11
## 2           10  2020 2020-07-11 2020-07-12             2
## 3          145  2022 2022-08-09 2022-08-13             5
## 4          215  2023 2023-08-27 2023-08-30             4
## 5          186  2023 2023-07-25 2023-07-27             3
```

```r
heat_wave_count <- heat_wave_summary %>%
  group_by(year) %>%
  summarize(
    total_heat_waves = n(),
    average_duration = round(mean(duration_days), 2),
    max_duration = max(duration_days),
    .groups = 'drop'
  ) %>%
  arrange(year)

heat_wave_days_per_year <- heat_wave_summary %>%
  group_by(year) %>%
  summarize(
    total_heat_wave_days = sum(duration_days),
    .groups = 'drop'
  ) %>%
  arrange(year)


ggplot(heat_wave_days_per_year, aes(x = factor(year), y = total_heat_wave_days, fill = factor(year))) +
  geom_bar(stat = "identity") +
  labs(
    title = "Total Heat Wave Days per Year",
    x = "Year",
    y = "Total Heat Wave Days",
    fill = "Year"
  ) +
  theme_minimal()
```


Total Heat Wave Days per Year
Total Heat Wave Days
0
5
10
2020
2022
2023
Year
Year
2020
2022
2023