Assignment 8: Time Series Analysis

Yufan Du

Fall 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
# check working directory
setwd("/home/guest/EDE_Fall2024/Data/Raw/Ozone_TimeSeries")

# Load the required libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(trend)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
# Define a custom ggplot theme
custom_theme <- theme_minimal() +
  theme(
    text = element_text(size = 12, color = "black"),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 12),
    legend.position = "right",
    plot.background = element_rect(fill = "lightblue", color = NA)
  )

# Set this theme as the default for all ggplots
theme_set(custom_theme)
```

2. Import the ten datasets from the Ozone_TimeSeries folder in the Raw data folder. These contain ozone concentrations at Garinger High School in North Carolina from 2010-2019 (the EPA air database only allows downloads for one year at a time). Import these either individually or in bulk and then combine them into a single dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
#1
# List all CSV files in the current working directory
file_list <- list.files(path = here::here("Data", "Raw", "Ozone_TimeSeries"), pattern = "*.csv", full.na

# Load each file and combine them into a single dataframe
GaringerOzone <- file_list %>%
  lapply(read.csv) %>%
  bind_rows()


# Check the result
dim(GaringerOzone)  # Confirm 3589 rows and 20 columns
```

```
## [1] 3589   20
```

```
head(GaringerOzone) # View the first few rows of the combined data
```

```
##         Date Source   Site.ID POC Daily.Max.8.hour.Ozone.Concentration UNITS
## 1 01/01/2010    AQS 371190041   1                                 0.031   ppm
## 2 01/02/2010    AQS 371190041   1                                 0.033   ppm
## 3 01/03/2010    AQS 371190041   1                                 0.035   ppm
## 4 01/04/2010    AQS 371190041   1                                 0.031   ppm
## 5 01/05/2010    AQS 371190041   1                                 0.027   ppm
## 6 01/07/2010    AQS 371190041   1                                 0.033   ppm
##   DAILY_AQI_VALUE           Site.Name DAILY_OBS_COUNT PERCENT_COMPLETE
## 1              29 Garinger High School              17              100
## 2              31 Garinger High School              17              100
## 3              32 Garinger High School              17              100
## 4              29 Garinger High School              17              100
## 5              25 Garinger High School              17              100
## 6              31 Garinger High School              17              100
##   AQS_PARAMETER_CODE AQS_PARAMETER_DESC CBSA_CODE
## 1              44201              Ozone     16740
## 2              44201              Ozone     16740
## 3              44201              Ozone     16740
## 4              44201              Ozone     16740
## 5              44201              Ozone     16740
## 6              44201              Ozone     16740
##                           CBSA_NAME STATE_CODE          STATE COUNTY_CODE
## 1 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 2 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 3 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 4 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 5 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 6 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
##        COUNTY SITE_LATITUDE SITE_LONGITUDE
## 1 Mecklenburg       35.2401      -80.78568
## 2 Mecklenburg       35.2401      -80.78568
## 3 Mecklenburg       35.2401      -80.78568
## 4 Mecklenburg       35.2401      -80.78568
## 5 Mecklenburg       35.2401      -80.78568
## 6 Mecklenburg       35.2401      -80.78568
```

Wrangle

3. Set your date column as a date class.

4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.

5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need to fill in any missing days with NA. Create a new data frame that contains a sequence of dates from 2010-01-01 to 2019-12-31 (hint: `as.data.frame(seq())`). Call this new data frame Days. Rename the column name in Days to "Date".

6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined data frame GaringerOzone.

```r
# 3
# Convert the Date column to Date format
GaringerOzone$Date <- mdy(GaringerOzone$Date)
summary(GaringerOzone)
```

```
##       Date               Source             Site.ID               POC
##  Min.   :2010-01-01   Length:3589        Min.   :371190041   Min.   :1
##  1st Qu.:2012-07-03   Class :character   1st Qu.:371190041   1st Qu.:1
##  Median :2015-01-04   Mode  :character   Median :371190041   Median :1
##  Mean   :2015-01-01                      Mean   :371190041   Mean   :1
##  3rd Qu.:2017-07-02                      3rd Qu.:371190041   3rd Qu.:1
##  Max.   :2019-12-31                      Max.   :371190041   Max.   :1
##  Daily.Max.8.hour.Ozone.Concentration    UNITS           DAILY_AQI_VALUE
##  Min.   :0.00200                      Length:3589        Min.   :  2.00
##  1st Qu.:0.03200                      Class :character   1st Qu.: 30.00
##  Median :0.04100                      Mode  :character   Median : 38.00
##  Mean   :0.04163                                         Mean   : 41.57
##  3rd Qu.:0.05100                                         3rd Qu.: 47.00
##  Max.   :0.09300                                         Max.   :169.00
##   Site.Name         DAILY_OBS_COUNT PERCENT_COMPLETE AQS_PARAMETER_CODE
##  Length:3589        Min.   : 6.00   Min.   : 35.0    Min.   :44201
##  Class :character   1st Qu.:17.00   1st Qu.:100.0    1st Qu.:44201
##  Mode  :character   Median :17.00   Median :100.0    Median :44201
##                     Mean   :16.97   Mean   : 99.8    Mean   :44201
##                     3rd Qu.:17.00   3rd Qu.:100.0    3rd Qu.:44201
##                     Max.   :19.00   Max.   :100.0    Max.   :44201
##  AQS_PARAMETER_DESC   CBSA_CODE      CBSA_NAME           STATE_CODE
##  Length:3589        Min.   :16740   Length:3589        Min.   :37
##  Class :character   1st Qu.:16740   Class :character   1st Qu.:37
##  Mode  :character   Median :16740   Mode  :character   Median :37
##                     Mean   :16740                      Mean   :37
##                     3rd Qu.:16740                      3rd Qu.:37
##                     Max.   :16740                      Max.   :37
##     STATE            COUNTY_CODE     COUNTY           SITE_LATITUDE
##  Length:3589        Min.   :119   Length:3589        Min.   :35.24
##  Class :character   1st Qu.:119   Class :character   1st Qu.:35.24
##  Mode  :character   Median :119   Mode  :character   Median :35.24
##                     Mean   :119                      Mean   :35.24
##                     3rd Qu.:119                      3rd Qu.:35.24
##                     Max.   :119                      Max.   :35.24
##  SITE_LONGITUDE
##  Min.   :-80.79
##  1st Qu.:-80.79
##  Median :-80.79
##  Mean   :-80.79
##  3rd Qu.:-80.79
##  Max.   :-80.79
```

```r
# 4
# Select only the necessary columns
GaringerOzone <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
```

```r
summary(GaringerOzone)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-03   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2015-01-04   Median :0.04100                      Median : 38.00
##  Mean   :2015-01-01   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-02   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
```

```r
# 5 Missing data
#Create a sequence of dates from 2010-01-01 to 2019-12-31
Days <- data.frame(Date = seq(ymd("2010-01-01"), ymd("2019-12-31"), by = "day"))

summary(GaringerOzone)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-03   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2015-01-04   Median :0.04100                      Median : 38.00
##  Mean   :2015-01-01   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-02   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
```

```r
# 6 combine
# Merge Days with GaringerOzone, filling in missing dates with NA
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")

# Check the merged dataframe
dim(GaringerOzone)
```

```
## [1] 3652    3
```

```r
summary(GaringerOzone)
```

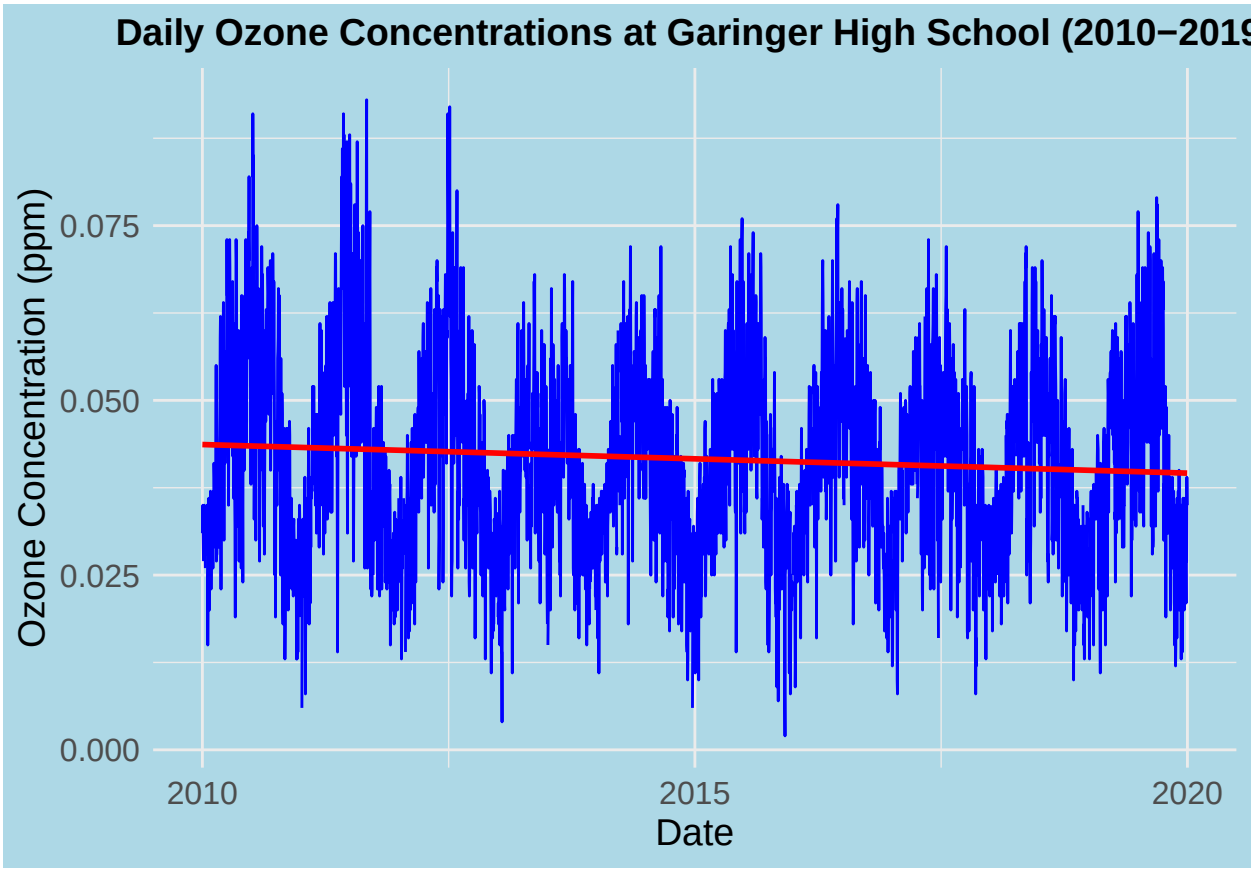
```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
```

Visualize

7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```r
#7
# Plot daily ozone concentrations over time with a trend line
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(color = "blue", size = 0.5) +  # Line for daily concentrations
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Linear trend line
  labs(
    title = "Daily Ozone Concentrations at Garinger High School (2010-2019)",
    x = "Date",
    y = "Ozone Concentration (ppm)"
  ) +
  custom_theme
```

```
## 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.

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

## Warning: Removed 63 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```



Answer:The plot suggests a slight decline in ozone concentrations over time, as indicated by the downward-sloping red trend line. Although the change is subtle, it could indicate a gradual

decrease in ozone levels at this location over the decade. This downward trend exists within a context of strong seasonal fluctuations, with higher concentrations in warmer months and lower concentrations in cooler months.

Time Series Analysis

Study question: Have ozone concentrations changed over the 2010s at this station?

8. Use a linear interpolation to fill in missing daily data for ozone concentration. Why didn't we use a piecewise constant or spline interpolation?

```
#8
#  Check for missing values
sum(is.na(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration))  # Initial count of NAs
```

```
## [1] 63
```

```
# S Perform linear interpolation to fill missing values
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Co

#  Verify that missing values have been filled
sum(is.na(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration))  # Should now be 0
```

```
## [1] 0
```

Answer: I used linear interpolation to fill in missing daily ozone concentration values in the Daily.Max.8.hour.Ozone.Concentration column. Initially, there were 63 missing values.The linear interpolation estimates each missing value by linearly connecting the preceding and following known values. This method is appropriate because it provides a straightforward and realistic approximation for time-series data, avoiding the abrupt transitions that could arise from methods like piecewise constant interpolation or the over-smoothing that spline interpolation might introduce. After interpolation, no missing values remain in the dataset.

9. Create a new data frame called `GaringerOzone.monthly` that contains aggregated data: mean ozone concentrations for each month. In your pipe, you will need to first add columns for year and month to form the groupings. In a separate line of code, create a new Date column with each month-year combination being set as the first day of the month (this is for graphing purposes only)

```
#9
# Extract year and month
GaringerOzone <- GaringerOzone %>%
  mutate(year = year(Date), month = month(Date))

# Calculate monthly averages
GaringerOzone.monthly <- GaringerOzone %>%
  group_by(year, month) %>%
  summarize(monthly_avg_ozone = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```r
# Create a new Date column for monthly data
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = as.Date(paste(year, month, "01", sep = "-")))

# Check the resulting data frame
head(GaringerOzone.monthly)
```

```
## # A tibble: 6 x 4
##    year month monthly_avg_ozone Date
##   <dbl> <dbl>              <dbl> <date>
## 1  2010     1             0.0305 2010-01-01
## 2  2010     2             0.0345 2010-02-01
## 3  2010     3             0.0446 2010-03-01
## 4  2010     4             0.0556 2010-04-01
## 5  2010     5             0.0466 2010-05-01
## 6  2010     6             0.0576 2010-06-01
```

10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```r
#10
# Create the daily time series
GaringerOzone.daily.ts <- ts(
  GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
  start = c(2010, 1),
  frequency = 365
)

# Create the monthly time series
GaringerOzone.monthly.ts <- ts(
  GaringerOzone.monthly$monthly_avg_ozone,
  start = c(2010, 1),
  frequency = 12
)

# Check the first few values of each time series
head(GaringerOzone.daily.ts)
```

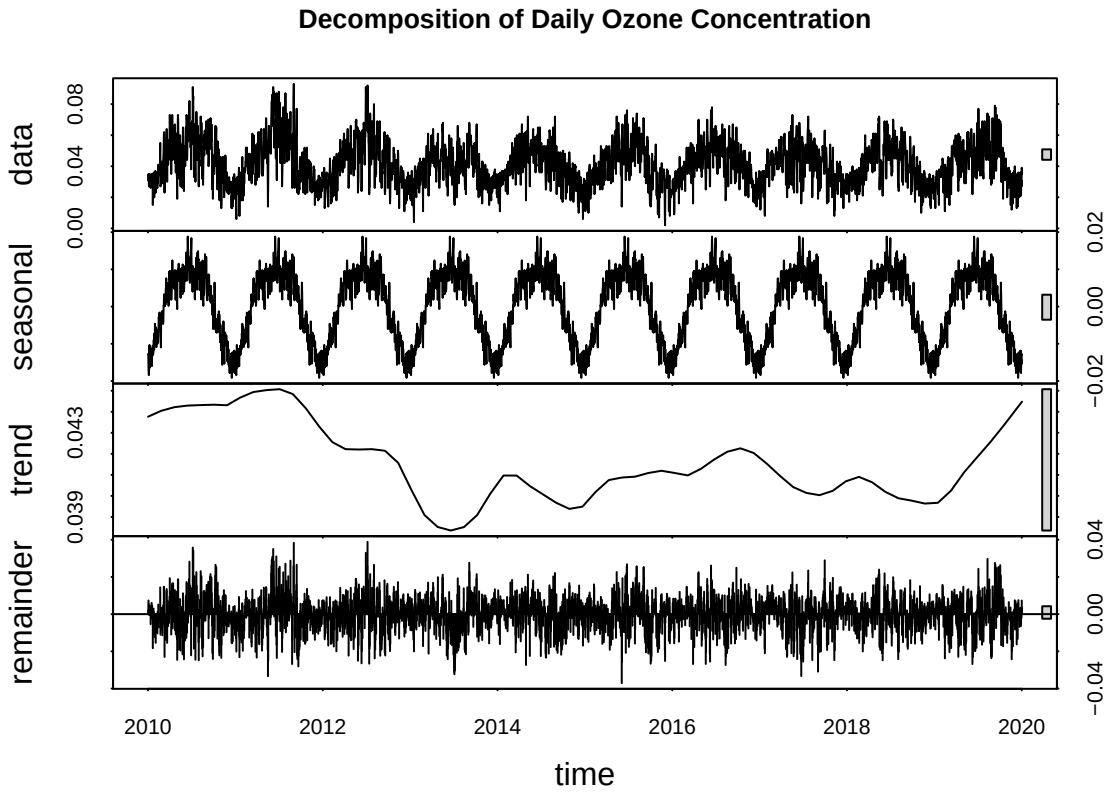
```
## [1] 0.031 0.033 0.035 0.031 0.027 0.030
```

```r
head(GaringerOzone.monthly.ts)
```

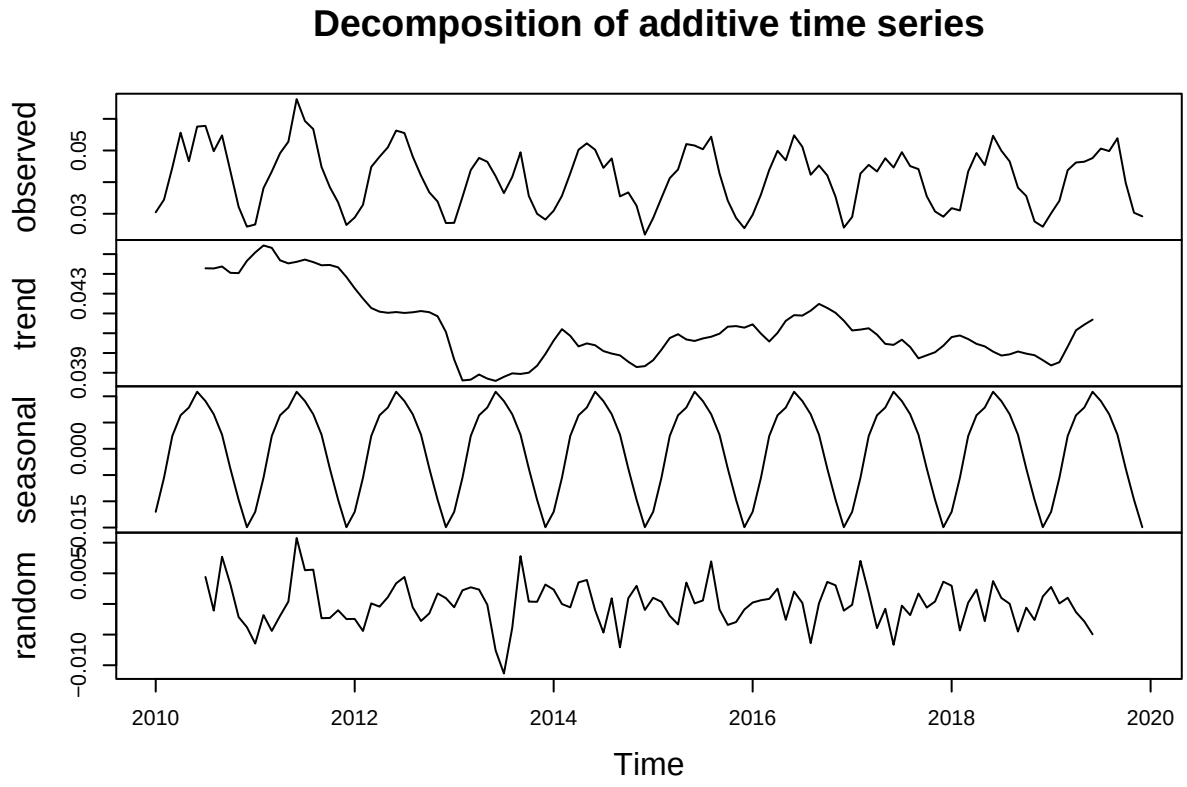
```
## [1] 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
```

11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```r
#11
# Decompose the daily time series
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.decomp, main = "Decomposition of Daily Ozone Concentration")
```




```r
# Decompose the monthly time series
GaringerOzone.monthly.decomp <- decompose(GaringerOzone.monthly.ts)
plot(GaringerOzone.monthly.decomp)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

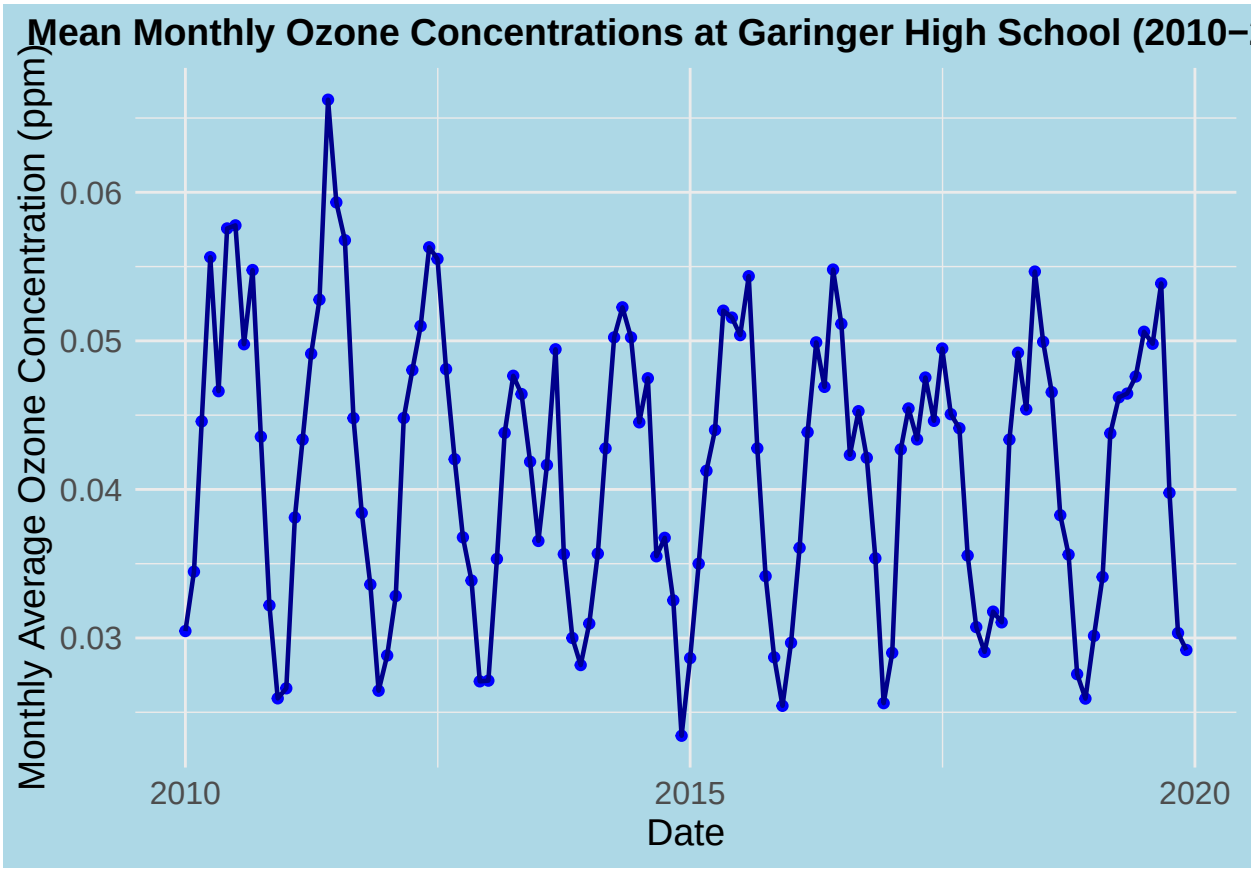
```
#12
# Create a time series object for monthly ozone concentrations
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$monthly_avg_ozone,
                               start = c(2010, 1),
                               frequency = 12)

# Now perform the Seasonal Mann-Kendall test
smk_result <- smk.test(GaringerOzone.monthly.ts)

# Display the result
smk_result
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

Answer: The results of the Seasonal Mann-Kendall test indicate a statistically significant trend in the monthly average ozone concentrations at Garinger High School over the period from 2010 to 2019. The test yielded a Z-value of -1.963 and a p-value of 0.04965, which is just below the significance threshold of 0.05. This suggests that the observed trend is unlikely to be due to random chance. The negative Z-value indicates a downward trend in ozone concentrations, meaning that ozone levels have slightly decreased over the years studied. The test statistic S= -77 and variance S=1499 support this conclusion, highlighting a small but significant decline in ozone concentration levels over time.

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13
# Plot mean monthly ozone concentrations with points and lines
ggplot(GaringerOzone.monthly, aes(x = Date, y = monthly_avg_ozone)) +
  geom_point(color = "blue", size = 1.5) +       # Points for monthly averages
  geom_line(color = "darkblue", size = 0.8) +    # Line connecting monthly points
  labs(
    title = "Mean Monthly Ozone Concentrations at Garinger High School (2010-2019)",
    x = "Date",
    y = "Monthly Average Ozone Concentration (ppm)"
  ) +
  custom_theme
```



14. To accompany your graph, summarize your results in context of the research question. Include output

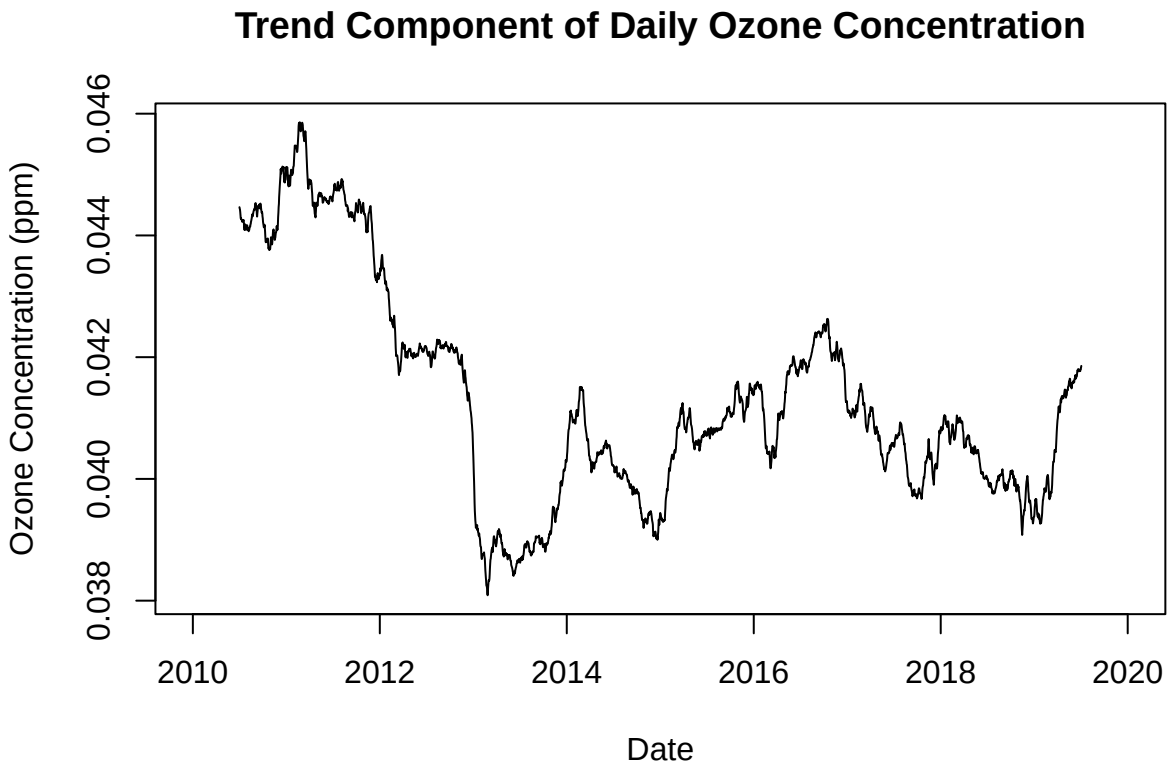
from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

Answer: In the decomposition plot from question 11, the "seasonal" panel provides a visualization of the seasonal component of daily ozone concentration. This component highlights the repeating seasonal pattern in ozone levels across each year, showing higher concentrations during certain times and lower concentrations at other times. The seasonal variations are consistent year over year, suggesting a recurring seasonal pattern in ozone concentrations, likely influenced by factors such as temperature, sunlight, and weather conditions, which can vary seasonally.

15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
#15
# Re-create the time series object
GaringerOzone.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                       start = c(2010, 1), frequency = 365)

# Perform time series decomposition
decomposition <- decompose(GaringerOzone.ts, type = "multiplicative")

# Plot only the trend component of the decomposition
plot(decomposition$trend, main = "Trend Component of Daily Ozone Concentration",
     ylab = "Ozone Concentration (ppm)", xlab = "Date")
```



```r
#16
# Ensure the monthly data is in a time series format
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$monthly_avg_ozone, start = c(2010, 1), frequency =

# Run the Seasonal Mann-Kendall Test
smk_result <- smk.test(GaringerOzone.monthly.ts)

# Print the test result
print(smk_result)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

Answer: The trend component indicates a noticeable initial decline in ozone concentrations from 2010 through 2013, reaching a low point around 2014. Afterward, there is an upward trend from 2014 to 2017, with some additional fluctuations observed towards the end of the period, particularly in 2018 and 2019. This trend highlights periods of both decrease and increase in ozone levels, suggesting that while ozone concentrations initially declined, there were subsequent variations influenced by factors that may require further analysis (e.g., seasonal or regulatory changes). This trend component helps isolate the longer-term changes in ozone concentration apart from seasonal variations and irregular fluctuations, providing insight into the overall direction of ozone concentration changes at Garinger High School over this period.