Assignment 8: Time Series Analysis

Sujay Dhanagare

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

```
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(ggplot2)
library(trend)
library(zoo)
```

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

```
library(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(here)
```

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

```
here
```

```
## function (...)
## {
##     .root_env$root$f(...)
## }
## <bytecode: 0x608267f1d010>
## <environment: namespace:here>
```

```
# Set Theme
my_theme <- theme_classic() +
  theme(
    line = element_line(
      color = '#000080',  # Navy blue for the Ashoka Chakra
      size = 2,
      linetype = 'solid'
    ),
    rect = element_rect(
      fill = 'white',  # White background as in the flag's middle band
      colour = 'black'
    ),
    text = element_text(
      face = 'plain',
      colour = '#000080',  # Navy blue text
      size = 16
    ),

    # Customize Plot Title
    plot.title = element_text(
      face = "bold",
      size = 20,
      color = "#FF9933",  # Saffron color for the plot title
      hjust = 0.5
    ),
```

```
    # Axis Titles are blank
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),

    # Customize Axis Ticks
    axis.ticks = element_line(
      color = "#138808"  # Green color for the ticks
    ),

    # Customize Major Grid Lines
    panel.grid.major = element_line(
      color = "#E5E5E5",
      size = 0.5
    ),

    # Remove Minor Grid Lines
    panel.grid.minor = element_blank(),

    # Customize Plot Background
    plot.background = element_rect(
      fill = "#FFFFFF",  # White background
      colour = NA
    ),

    # Customize Panel Background
    panel.background = element_rect(
      fill = "#FFFFFF",
      colour = NA
    ),

    # Customize Legend Key
    legend.key = element_rect(
      fill = "#FFFFFF",
      colour = "#FF9933"  # Saffron border for legend keys
    ),

    # Set Legend Position
    legend.position = "right",

    # Ensure theme completeness
    complete = TRUE
  )
```

```
## Warning: The ‘size‘ argument of ‘element_line()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Set this theme as default
theme_set(my_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 folder
file_list <- list.files(path = "Data/Raw/Ozone_TimeSeries", pattern = "*.csv", full.names = TRUE)

# Import all datasets and combine them into a single dataframe
GaringerOzone <- lapply(file_list, function(file) {
  # Read each file
  data <- read_csv(file)

  # Convert Date column from m/d/y format to Date class using lubridate's mdy()
  data$Date <- mdy(data$Date)

  return(data)
}) %>%
  bind_rows()

dim(GaringerOzone)
```

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

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.

```
# 3
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%y")
# 4

GaringerOzone_Processed <- GaringerOzone %>%
                              select("Date", "Daily Max 8-hour Ozone Concentration", "DAILY_AQI_VALUE")
# 5
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"),
                          by = "day"))

colnames(Days) <- "Date"
```

```r
# 6
GaringerOzone_Daily <- left_join(Days, GaringerOzone_Processed, by ="Date")
```

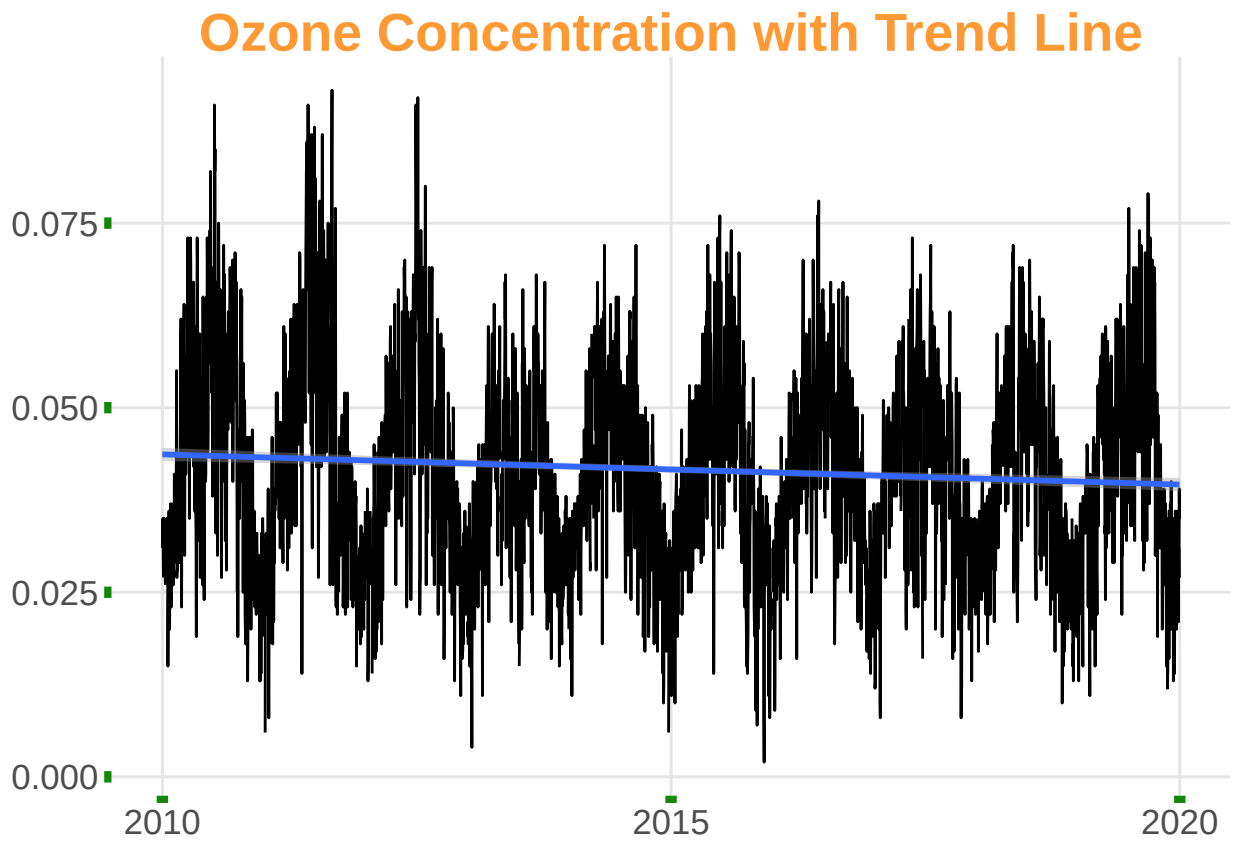
Visualize

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
ggplot(GaringerOzone_Daily, aes(x = Date, y = `Daily Max 8-hour Ozone Concentration` ))+
  geom_line()+
  geom_smooth(method = "lm") +
  labs(title = "Ozone Concentration with Trend Line",
       x = "Date",
       y = "Daily Max 8-hour Ozone Concentration (ppm)") +
  my_theme
```

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

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



Answer: Yes, the plot does suggest a slight decreasing trend in ozone concentration over time (2010-2020), as indicated by the downward slope of the blue linear trend line.

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
GaringerOzone_Daily$`Daily Max 8-hour Ozone Concentration` <- na.approx(GaringerOzone_Daily$`Daily Max 8
```

Answer: Linear interpolation was chosen because it provides a simple yet effective way to estimate missing values in time series data while preserving trends and avoiding over-smoothing. Piecewise constant interpolation would have introduced artificial steps, and spline interpolation might have smoothed out important variations 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

#Add Year and Month columns
GaringerOzone_Processed <- GaringerOzone_Processed %>%
  mutate(Year = year(Date),
         Month = month(Date))

# Step 2: Aggregate data to calculate mean ozone concentrations for each month
GaringerOzone.monthly <- GaringerOzone_Processed %>%
  group_by(Year, Month) %>%
  summarise(Mean_Ozone_Concentration = mean(`Daily Max 8-hour Ozone Concentration`, na.rm = TRUE)) %>%
  ungroup()
```

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

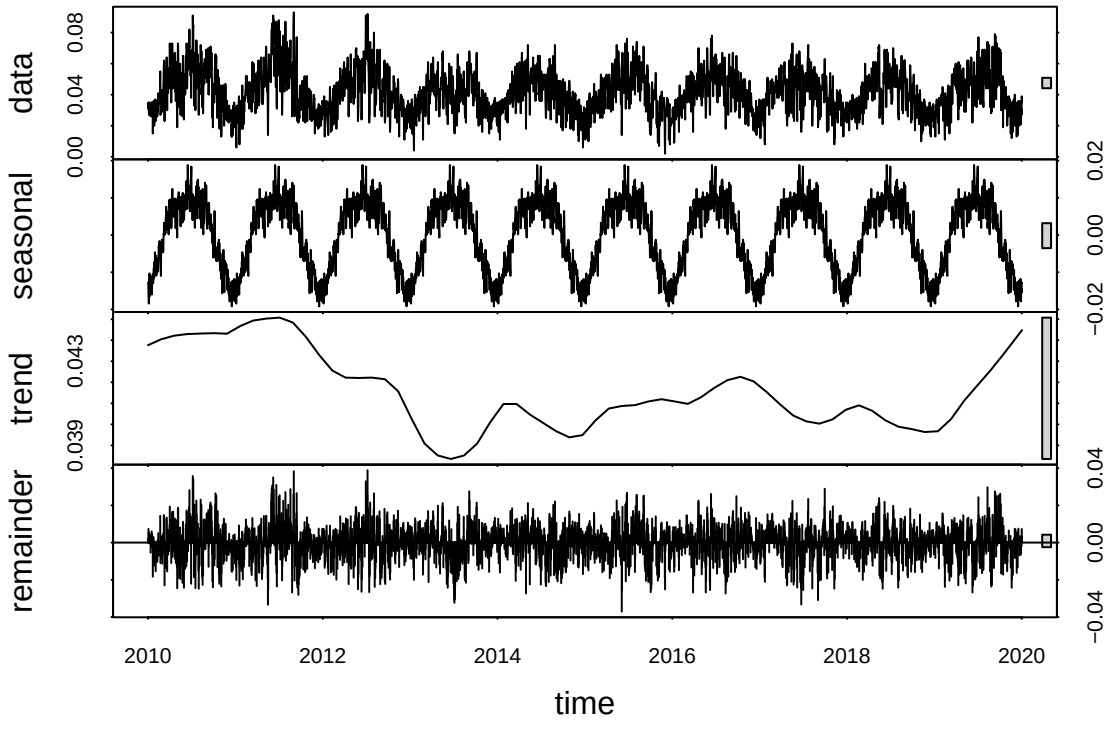
```
# Step 3: Create a new Date column with the first day of each month for graphing purposes
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(Year, Month, 1))
```

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.

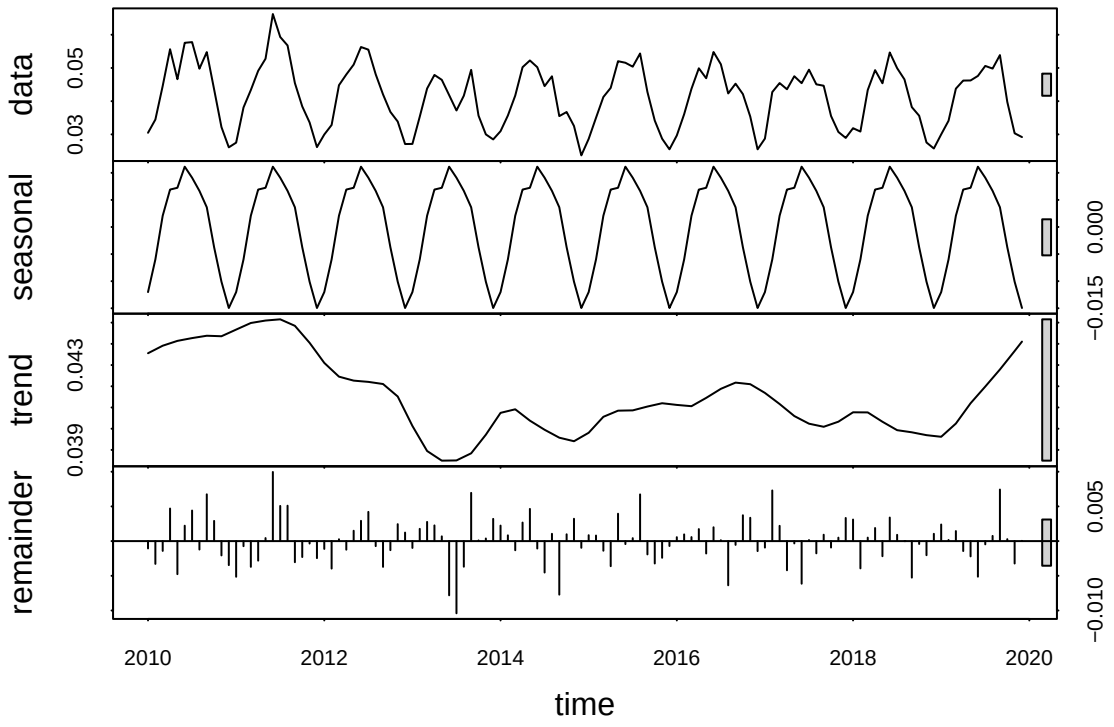
```r
#10
GaringerOzone.daily.ts <- ts(GaringerOzone_Daily$`Daily Max 8-hour Ozone Concentration`,
                             start = c(2010, 1),
                             frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone_Concentration,
                               start = c(2010, 1),
                               frequency = 12)
```

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

```r
#11

daily_decompose <- stl(GaringerOzone.daily.ts, s.window = "periodic")
monthly_decompose <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

plot(daily_decompose)
```




```r
plot(monthly_decompose)
```



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

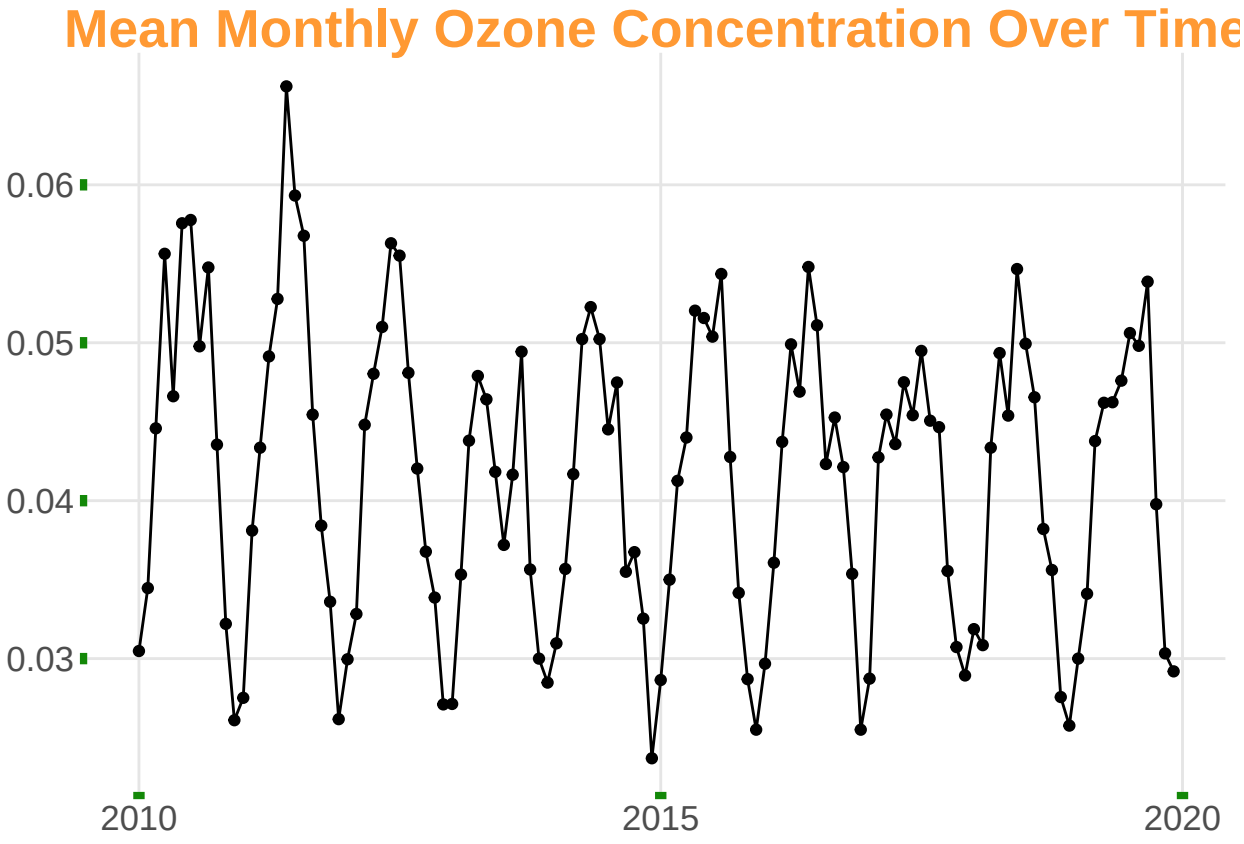
```
#12

smk_result <- smk.test(GaringerOzone.monthly.ts)

print(smk_result)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -2.2478, p-value = 0.02459
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -88 1498
```

Answer:

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.

```r
# 13

ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone_Concentration)) +
  geom_line() +
  geom_point() +
  labs(title = "Mean Monthly Ozone Concentration Over Time",
       x = "Date",
       y = "Mean Ozone Concentration (ppm)") +
my_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: The Mann-Kendall trend test yielded a z-value of -2.2478 and a p-value of 0.02459, indicating that the trend is statistically significant at the 5% significance level. This suggests that there is a significant decreasing trend in ozone concentrations over the study period (2010-2020). The negative S value (-88) further supports the presence of a downward trend in ozone concentrations, indicating that ozone levels have been gradually declining over time. This result is consistent with the visual inspection of the plot, where both the monthly mean ozone concentrations and the daily observations show a slight downward trend, despite clear seasonal fluctuations. The linear regression line (in blue) in the second plot also suggests a slight decrease in ozone concentrations. In conclusion, ozone concentrations at Garinger High School have significantly decreased over the 2010s, as confirmed by both the visual trend and the Seasonal Mann-Kendall test ($z = -2.2478$, $p = 0.02459$).

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

# Extract the seasonal component
seasonal_component <- monthly_decompose$time.series[, "seasonal"]

# Subtract the seasonal component
detrended_series <- GaringerOzone.monthly.ts - seasonal_component

#16

# Perform Mann-Kendall test on the detrended series
mk_non_seasonal <- MannKendall(detrended_series)

print(mk_non_seasonal)
```

```
## tau = -0.179, 2-sided pvalue =0.0037728
```

Answer: Both tests indicate a significant downward trend in ozone concentrations over time at Garinger High School from 2010 to 2020. However, after removing seasonality, the trend appears to be stronger, suggesting that while seasonal fluctuations are important, there is an underlying long-term decrease in ozone levels.