Assignment 8: Time Series Analysis

Ana Andino

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

```
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
```

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
#As individual datasets:
data1 <- read.csv(
  "~/EDE_Fall2024/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
  stringsAsFactors = TRUE)
#GaringerOzone <- rbind (data1, data2...)

#upload together:
folder.path <- "~/EDE_Fall2024/Data/Raw/Ozone_TimeSeries"
file.list <- list.files(path = folder.path, pattern = "*.csv",
                        full.names = TRUE)

GaringerOzone <- file.list %>%
  lapply(read.csv, stringsAsFactors = TRUE) %>%
  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 <- GaringerOzone %>%
  rename(Max.Ozone.Daily= Daily.Max.8.hour.Ozone.Concentration)

GaringerOzone <- GaringerOzone %>%
  select(Date, Max.Ozone.Daily, DAILY_AQI_VALUE)

# 5
summary(GaringerOzone$Date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2010-01-01" "2012-07-03" "2015-01-04" "2015-01-01" "2017-07-02" "2019-12-31"
```

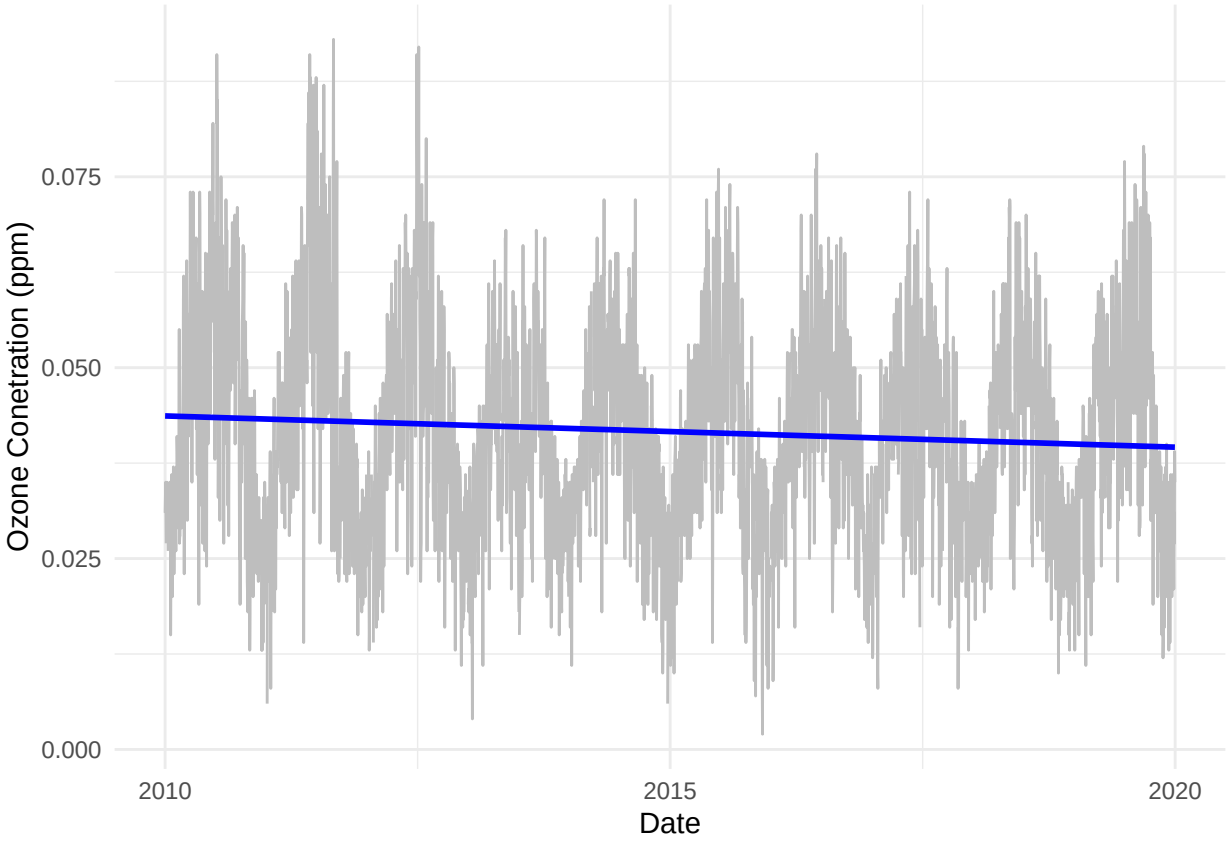
```
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"),
                          by = "day"))
colnames(Days) <- "Date"


# 6
GaringerOzone <- left_join(Days, GaringerOzone, 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?

```
#7

ozone.plot <- ggplot(GaringerOzone,
                     aes(x= Date, y= Max.Ozone.Daily))+
                       geom_line(color = "grey") +
                       geom_smooth(method = "lm", color = "blue", se = FALSE) +
                       labs(Title = " Daily Ozone Concentration (2010-2019)",
                            x = "Date",
                            y = "Ozone Conetration (ppm)") +
  theme_minimal()

plot(ozone.plot)
```

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

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



Answer: It appears like a slighlty negative trend over time for the daily ozonce concentration

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
library(zoo)
GaringerOzone$Max.Ozone.Daily <-
  na.approx(GaringerOzone$Max.Ozone.Daily)
```

Answer: linear interpolation assumes a steady rate of change where there is no data thus, providing a smooth transition while piecewise functions fils the NAs with the prvious known value which could introduce sudden jumps or downs in the data. This applies even more to daily datasets.

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)

```r
#9


# Load necessary library
library(dplyr)

GaringerOzone.monthly <- GaringerOzone %>%
  mutate(year = as.numeric(format(Date, "%Y")),
         month = as.numeric(format(Date, "%m"))) %>%
  group_by(year, month) %>%
  summarize(mean_ozone = mean(Max.Ozone.Daily, na.rm = TRUE)) %>%
  ungroup()
```

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

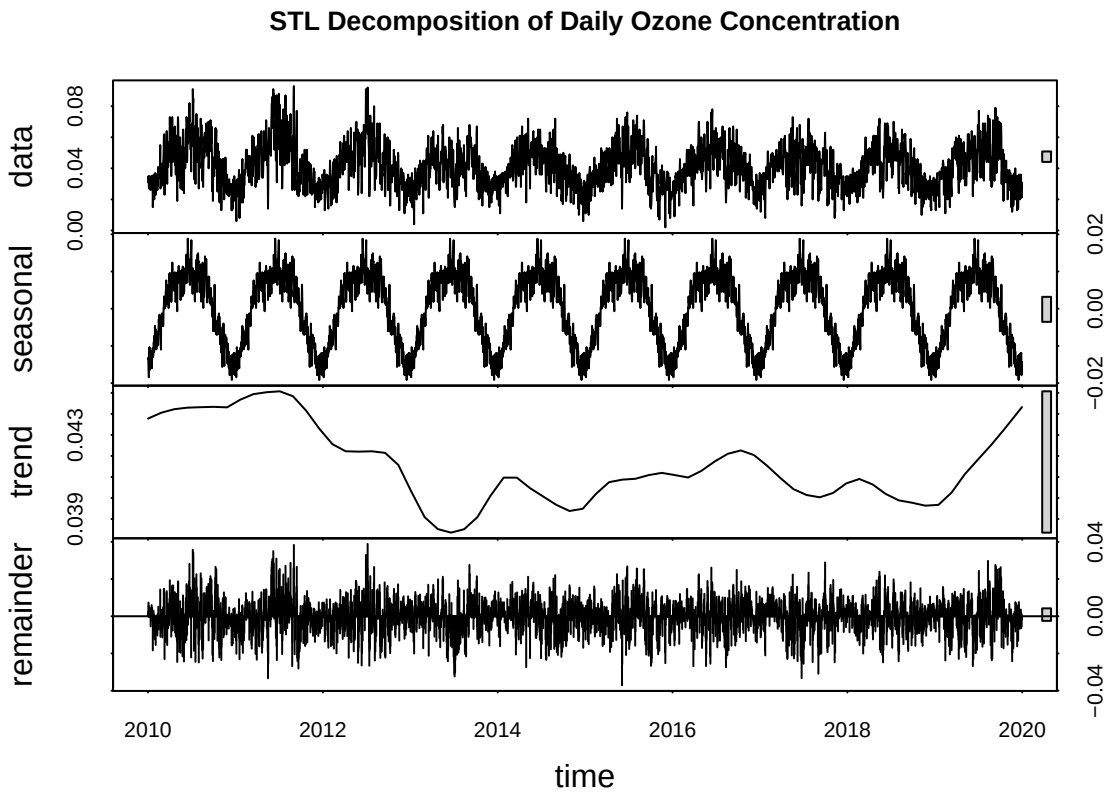
```r
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = as.Date(paste(year, month, "01", sep = "-")))
```

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.

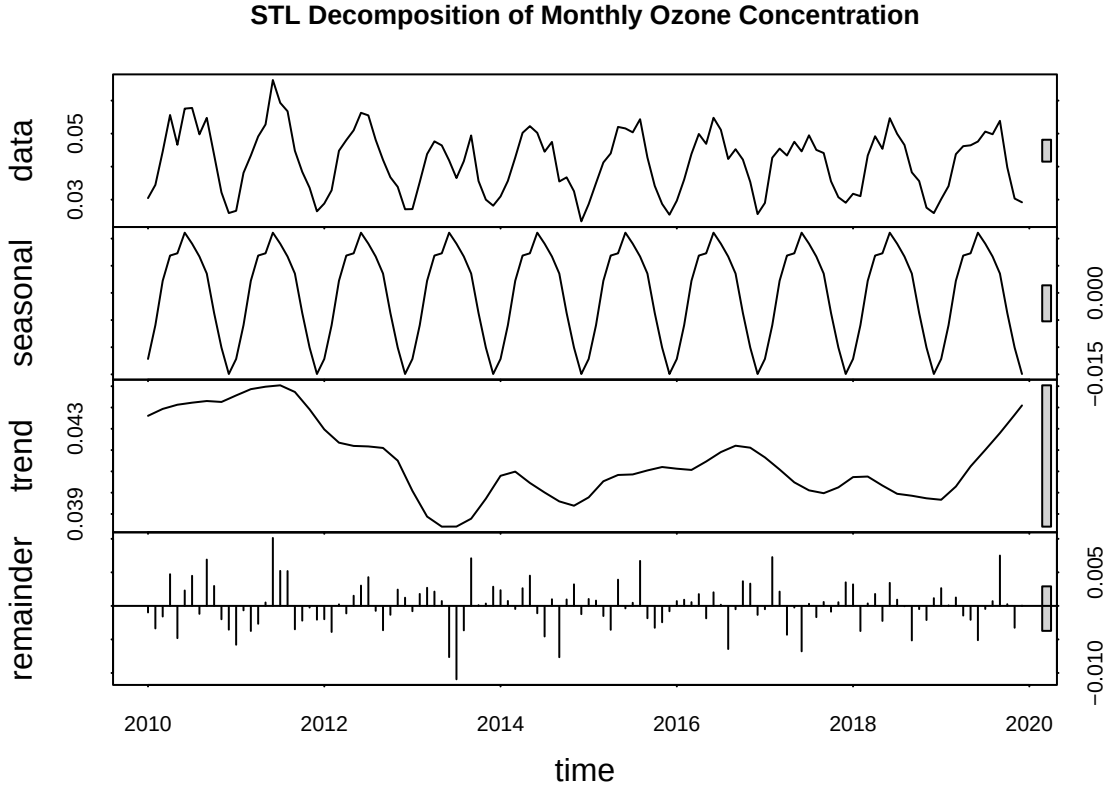
```r
#10


GaringerOzone.daily.ts <- ts(GaringerOzone$Max.Ozone.Daily,
                             start = c(2010,1),
                             end = c(2019, 365),
                             frequency =  365)

# Create a monthly time series object for GaringerOzone.monthly.ts
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                               start = c(2010, 1),
                               end = c(2019, 12),
                               frequency = 12)
```

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

```r
#11

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

plot(GaringerOzone.daily.stl, main =
       "STL Decomposition of Daily Ozone Concentration")
```




```
GaringerOzone.monthly.stl <- stl(GaringerOzone.monthly.ts,
                                 s.window = "periodic")

plot(GaringerOzone.monthly.stl, main =
       "STL Decomposition of Monthly Ozone Concentration")
```



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

library(Kendall)
smk_analysis <- SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(smk_analysis)
```

```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

```
smk.trend.monthly <- trend::smk.test(GaringerOzone.monthly.ts)
smk.trend.monthly
```

```
##
##  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
```

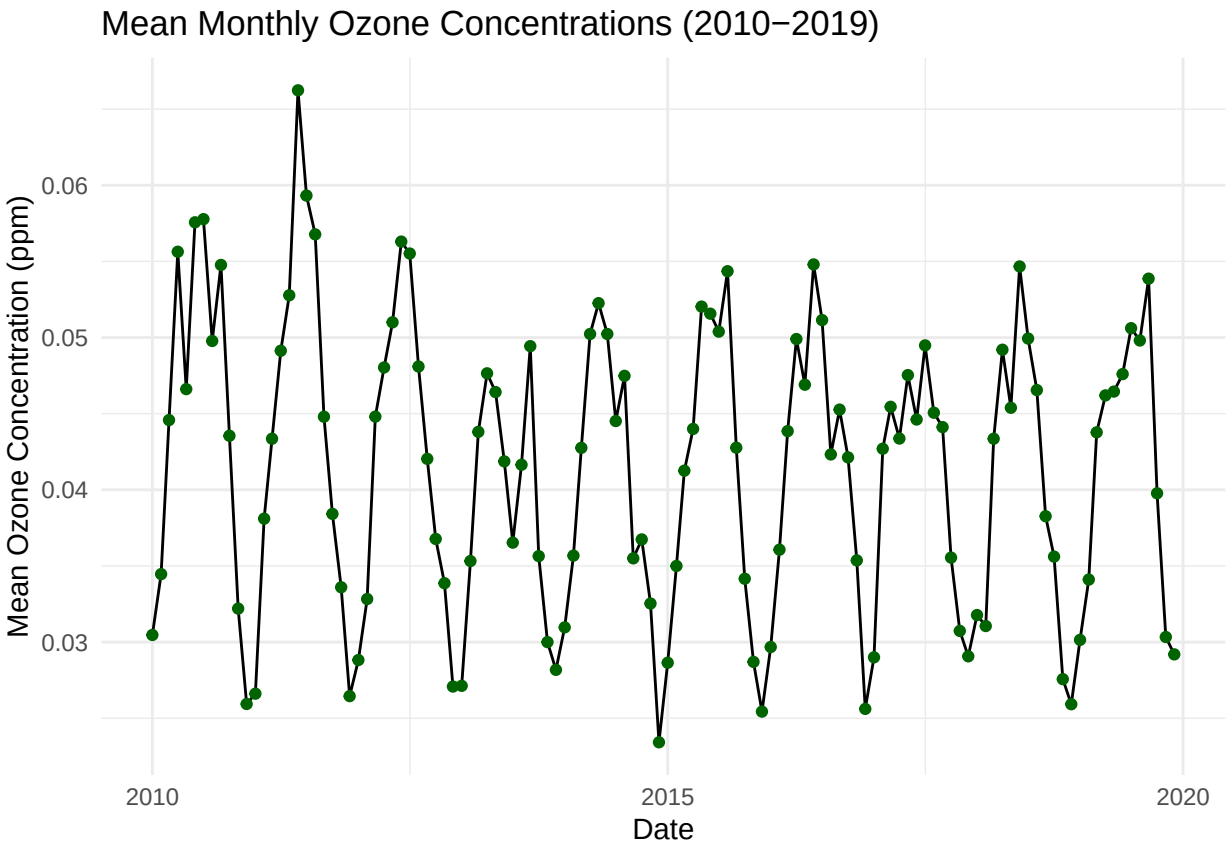
```
summary(smk.trend.monthly)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 12:   S = 0  11  125  0.244  0.894  0.37109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The season Mann-Kendall can detect trends in data with seasonal cycles and it specifically checks for a monotonic trend while allowing seasonality. It has a no seasonality, non-parametric, missing data allowed component. Also, the data has already been wrnagled for the format needed.

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

ggplot(GaringerOzone.monthly,
       aes(x = Date, y = mean_ozone)) +
  geom_line() +
  geom_point(color = "darkgreen") +
  labs(title = "Mean Monthly Ozone Concentrations (2010-2019)",
       x = "Date",
       y = "Mean Ozone Concentration (ppm)") +
  theme_minimal()
```



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 graph displays noticeable seasonal variability, with peaks and troughs that correspond to expected environmental cycles. Over the observed period, a general downward trend in ozone concentration is suggested thus the ozone concentrations have changed since 2010s.The Seasonal Mann-Kendall test supports this observation, indicating a statistically significant monotonic decrease in monthly ozone levels (Score = -77, Var(Score) = 1499, tau = -0.143, p*** = 0.047).

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

seasonal_component <- GaringerOzone.monthly.stl$time.series[, "seasonal"]
GaringerOzone.monthly_noseason <- GaringerOzone.monthly.ts - seasonal_component
head(GaringerOzone.monthly_noseason)
```

```
## [1] 0.04263190 0.04041003 0.04234881 0.04875492 0.03932081 0.04647348
```

```r
#16

mk_nonseasonal <- MannKendall(GaringerOzone.monthly_noseason)
print(mk_nonseasonal)
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```r
#for easy comparison:
print(smk_analysis)
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The tau value is slightly more negative in the deseasonalized series, indicating a somewhat stronger downward trend when seasonality is removed.The p-value for the deseasonalized series is lower (0.0075 while the other is 0.0467), indicating a stronger statistical significance for the downward trend in the absence of seasonal fluctuations which was already statistically significant. In summary, the results indicate that the downward trend in ozone concentrations over time is more pronounced and statistically significant once seasonal effects are removed.