Assignment 8: Time Series Analysis

Mason Ibrahim

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
getwd()
```

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

```
#Load Packages
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(ggthemes)
library(here)
```

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

```
#Set ggplot theme
my_theme <- theme_base() +
  theme(
    rect = element_rect(color = "darkblue"),
    text = element_text(color = "blue"),
    plot.title = element_text(
      size = 16,
      face = "bold",
      color = "blue"
    ),
    axis.title.x = element_text(size = 12, color = "blue"),
    axis.title.y = element_text(size = 12, color = "blue"),
    axis.text = element_text(size = 12, color = "lightblue"),

    axis.ticks = element_line(color = "lightblue"),
    panel.grid.major = element_line(color = "lightblue"),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "white", color = NA),
    legend.key = element_rect(fill = "white", color = "blue"),
    legend.background = element_rect(fill = "lightblue", color = "blue")
  )

complete = TRUE
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
#Import datasets
Ozone_TS_2010 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)
```

```
Ozone_TS_2011 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)
Ozone_TS_2012 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                          stringsAsFactors = TRUE)
Ozone_TS_2013 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                          stringsAsFactors = TRUE)
Ozone_TS_2014 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                          stringsAsFactors = TRUE)
Ozone_TS_2015 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                          stringsAsFactors = TRUE)
Ozone_TS_2016 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                          stringsAsFactors = TRUE)
Ozone_TS_2017 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                          stringsAsFactors = TRUE)
Ozone_TS_2018 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                          stringsAsFactors = TRUE)
Ozone_TS_2019 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                          stringsAsFactors = TRUE)

#Combine into single dataframe
Ozone_TS_complete <-
  bind_rows(Ozone_TS_2010,
            Ozone_TS_2011,
            Ozone_TS_2012,
            Ozone_TS_2013,
            Ozone_TS_2014,
            Ozone_TS_2015,
            Ozone_TS_2016,
            Ozone_TS_2017,
            Ozone_TS_2018,
            Ozone_TS_2019)
```

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
#Set date
Ozone_TS_complete$Date <- mdy(Ozone_TS_complete$Date)

# 4
#Wrangle dataset
Ozone_TS_complete <- Ozone_TS_complete %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
# 5
#Create Days data frame
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"),
                          by = "day"))

# Rename the column to "Date"
colnames(Days) <- "Date"

# 6

#Combine Days and Ozone data frames
GaringerOzone <- left_join(Days, Ozone_TS_complete)
```

```
## Joining with ‘by = join_by(Date)‘
```

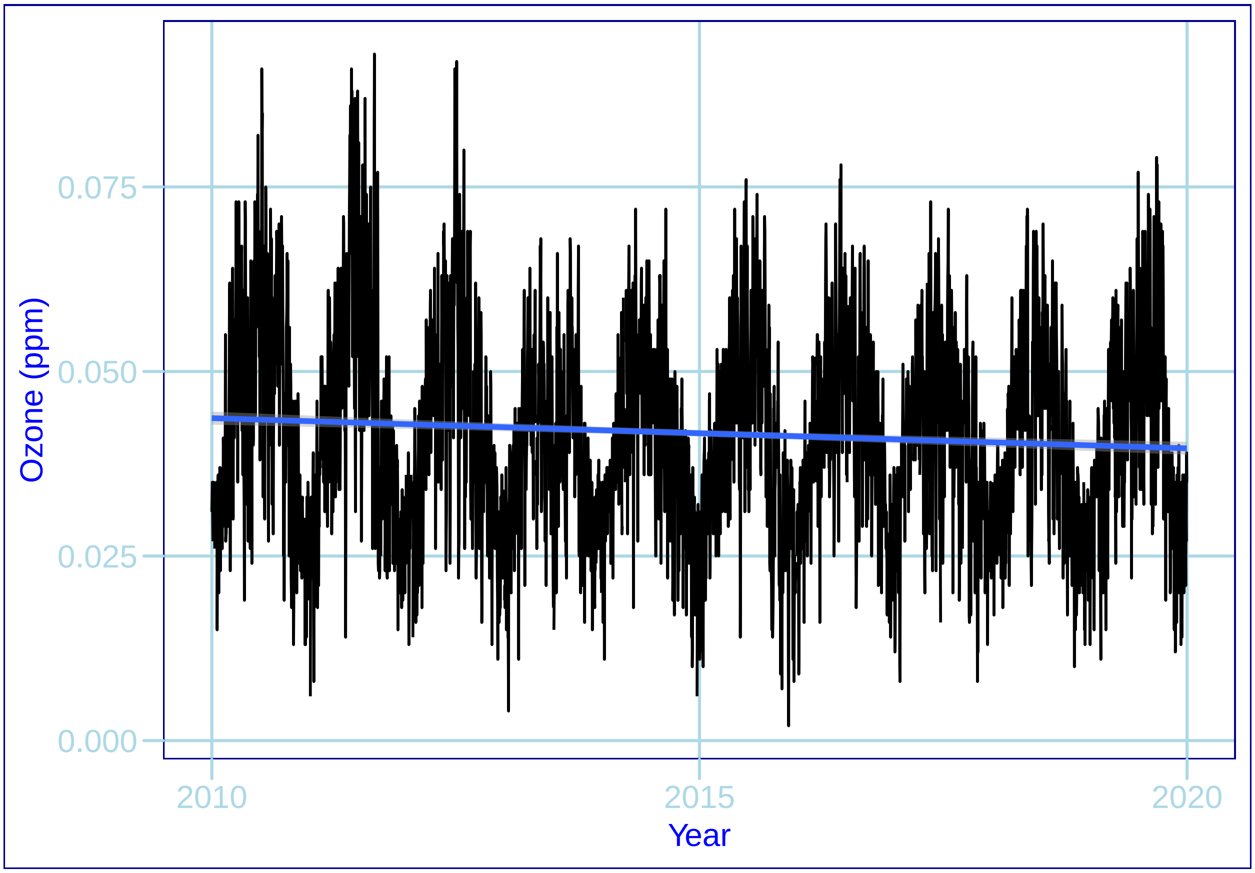
Visualize

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
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm")+
  labs(x = "Year", y = "Ozone (ppm)")
```

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

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



Answer: While there is a slight negaative trend in Ozone over time, it is very faint, and may not be significant.

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_clean <-
  GaringerOzone %>%
  mutate( Ozone_concentration_clean = zoo::na.approx
          (Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Linear interpolation assumes missing data falls between known measurements along a straight line, making it well-suited for steadily changing environmental data trends, unlike piecewise constant methods that rely on the nearest neighbor, or spline interpolation that uses quadratic functions.

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

GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(Year = format(Date, "%Y"),
         Month = format(Date, "%m")) %>%
  group_by(Year, Month) %>%
  summarize(Mean_Ozone = mean(Ozone_concentration_clean, na.rm = TRUE)) %>%
  ungroup()
```

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

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate( Date = my(paste0(Month,"-",Year)))
```

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.

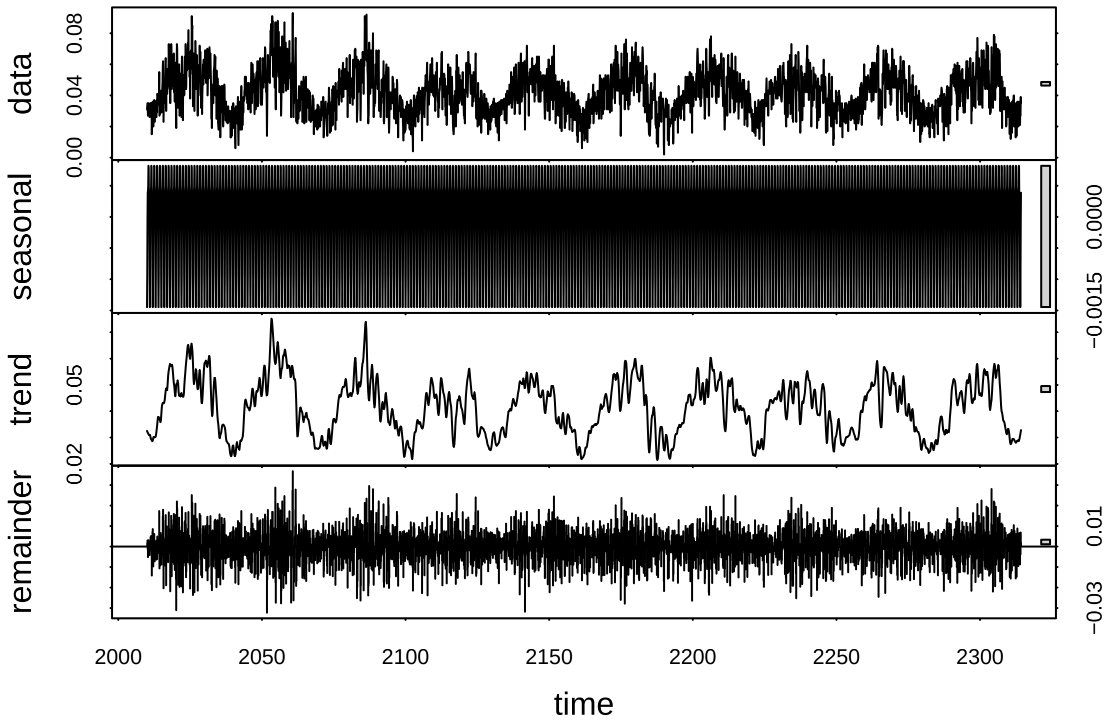
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Ozone_concentration_clean,
                   start=c(2010,01),
                   frequency=12)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone,
                   start=c(2010,01),
                   frequency=12)
```

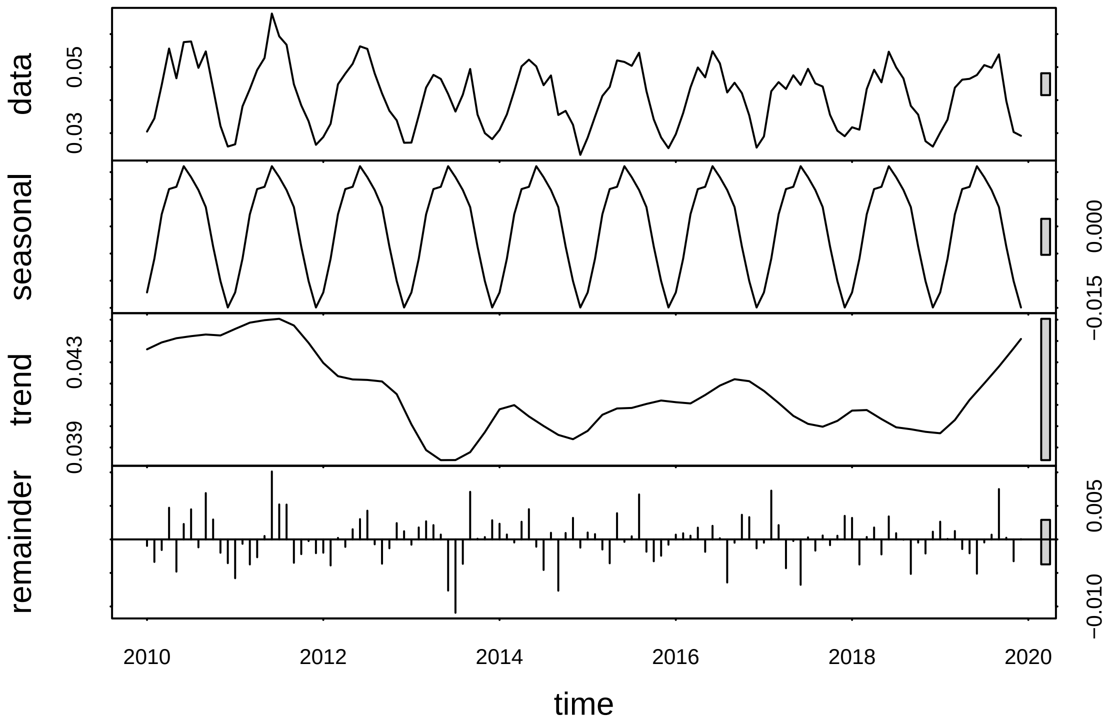
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11
GaringerOzone.daily.Decomposed <-
  stl(GaringerOzone.daily.ts, s.window = "periodic")
GaringerOzone.monthly.Decomposed <-
  stl(GaringerOzone.monthly.ts, s.window = "periodic")

plot(GaringerOzone.daily.Decomposed)
```



```
plot(GaringerOzone.monthly.Decomposed)
```



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

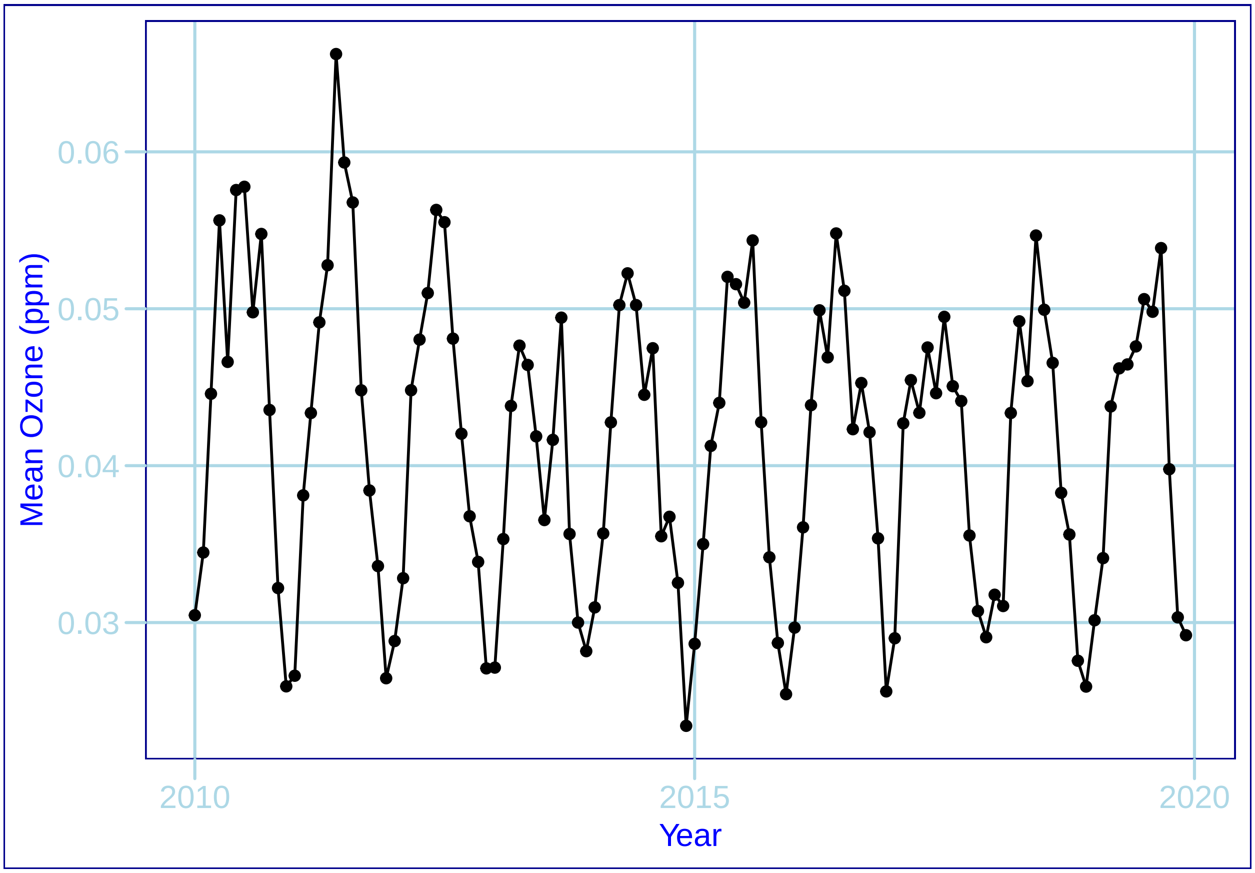
```
#12
monthly.Ozone.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(monthly.Ozone.trend)
```

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

Answer: The Seasonal Mann-Kendall test is good for the monthly ozone series because it accounts for seasonality, which allows detecting of long-term trends within each season despite regular fluctuations. Its non-parametric nature also makes it good for environmental data, which may not follow a normal distribution.

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_point()+
  geom_line()+
  labs(x= "Year", y = "Mean Ozone (ppm)")
```



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: There is a slight negative trend in ozone concentrations over the 2010s, with a Kendall's tau value of -0.143. This suggests a decrease in ozone levels during this period. The trend is statistically significant (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.

```r
#15
seasonal_component <- GaringerOzone.monthly.Decomposed$time.series[, "seasonal"]
GaringerOzone.monthly.non_seasonal <- GaringerOzone.monthly.ts - seasonal_component


#16
non_seasonal_trend <- Kendall::MannKendall(GaringerOzone.monthly.non_seasonal)
summary(non_seasonal_trend)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The tau value is -0.165, a slightly higher negative correction than -0.143, with a p-value of 0.00754. The p-value suggests that the negative trend in ozone concentrations is more statistically significan than the results obtained with the Seasonal Mann Kendall.