Assignment 8: Time Series Analysis

Jazmine Pritchett

Fall 2023

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

```
#Setting up session
getwd()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
library(tidyverse)
library(lubridate)
install.packages("trend")
library(trend)
install.packages("zoo")
library(zoo)
library(here)
here()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

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 data ser for each year individually
NC2010 <- read.csv(file = here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
stringsAsFactors = TRUE)

NC2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
stringsAsFactors = TRUE)

NC2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
stringsAsFactors = TRUE)

NC2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
stringsAsFactors = TRUE)

NC2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
stringsAsFactors = TRUE)

NC2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
stringsAsFactors = TRUE)

NC2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
stringsAsFactors = TRUE)

NC2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
stringsAsFactors = TRUE)

NC2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
stringsAsFactors = TRUE)

NC2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
stringsAsFactors = TRUE)

#combining data set into 'GraingerOzone' dataframe
GaringerOzone <- rbind(NC2010, NC2011, NC2012, NC2013,
NC2014, NC2015 ,NC2016, NC2017, NC2018, NC2019)
```

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".

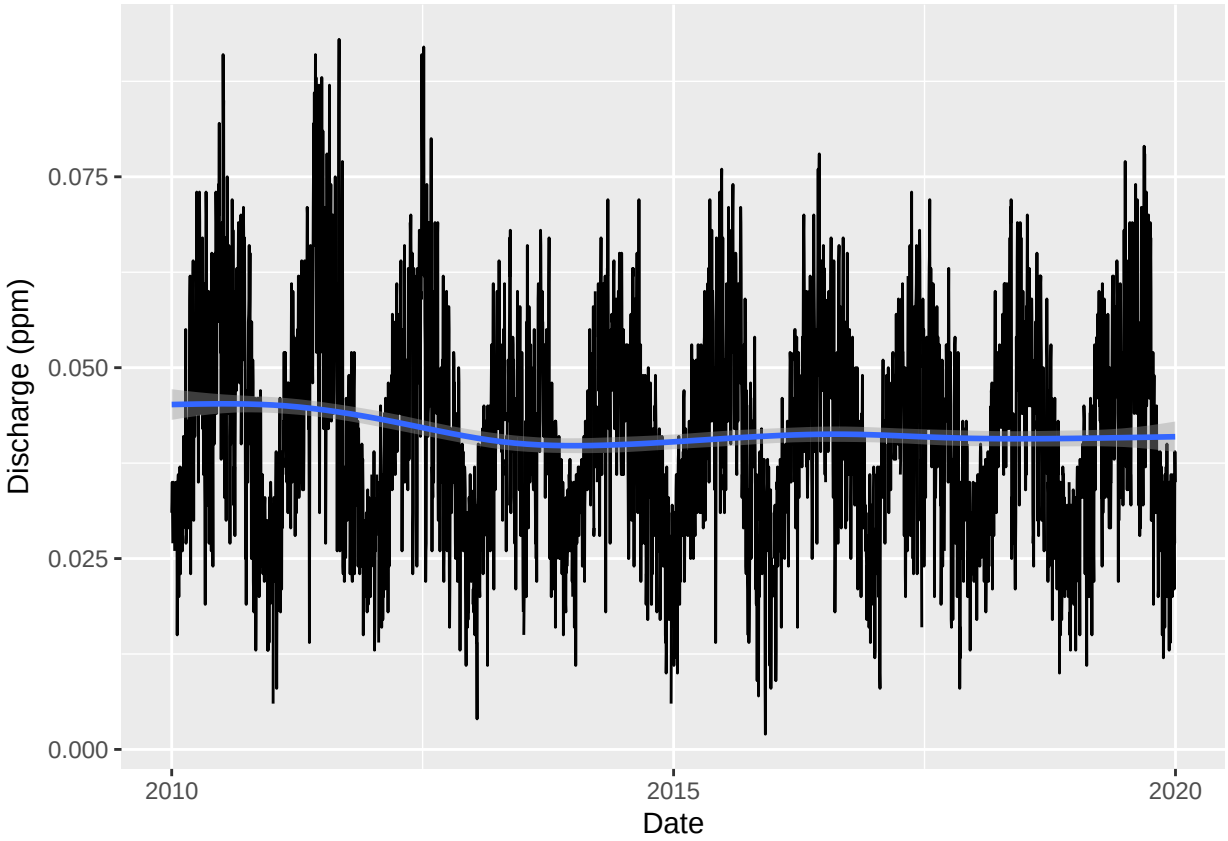
6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined data frame GaringerOzone.

```
# 3

#Date format month - day- year
GaringerOzone$Date <- mdy(GaringerOzone$Date)

# 4

GaringerOzoneWrangled <- select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration,
DAILY_AQI_VALUE)

# 5

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

#Changing column name produced in the 'Days' data frame to 'Date'

colnames(Days) <- "Date"

# 6

#Joining wrangled and days data

GaringerOzone <- left_join( Days, GaringerOzoneWrangled, 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() + labs(x = "Date", y =
expression("Discharge (ppm)"))
```



Answer: Overtime, it seems that discharge levels stay consistently close to 0.050 ppm. With a slight decline around 2013- 2014.

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( Daily.Max.8.hour.Ozone.Concentration
  = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer:Linear interpolation is most common for missing data with short intervals, which is relevant for daily ozone data as it is by day. Piecewise constant assumes data to be constant or equal, which is not always true day by day. Spline is more complex, but isn't linear which is the best suited relationship for the data we are working with.

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)

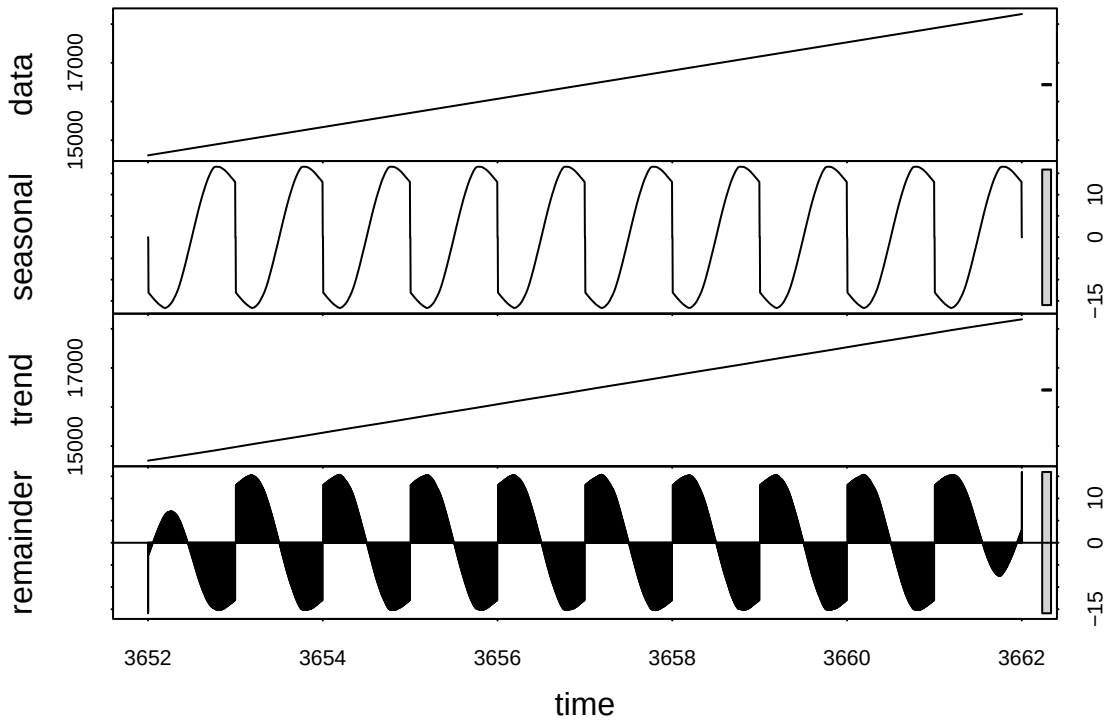
```
#9

GaringerOzone.monthly <-  GaringerOzone_clean %>%
  mutate(year = year(Date), month = month(Date)) %>%
  group_by(year, month) %>%
  summarize(mean_ozone =
  mean(Daily.Max.8.hour.Ozone.Concentration))


GaringerOzone.monthly$Date <- as.Date(paste(GaringerOzone.monthly$year, GaringerOzone.monthly$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.

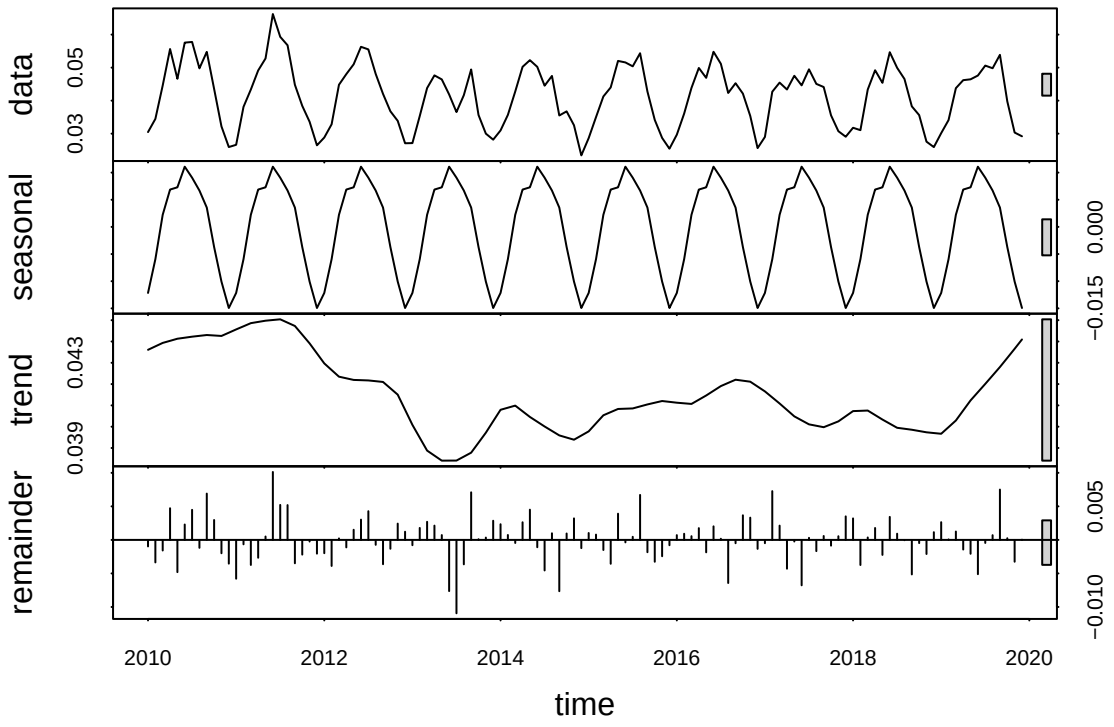
```
#10

f_month <- month(first(Days$Date))
f_year <- year(first(GaringerOzone.monthly$Date))

GaringerOzone.daily.ts <- ts(Days$Date, start = c(3652,1),
frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone, start = c(f_year,f_month),
frequency = 12)
```

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

```
#11

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

plot(Daily_decomp)
```




```
plot(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

install.packages("Kendall")
library(Kendall)

Monthly_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

Monthly_trend1
```

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

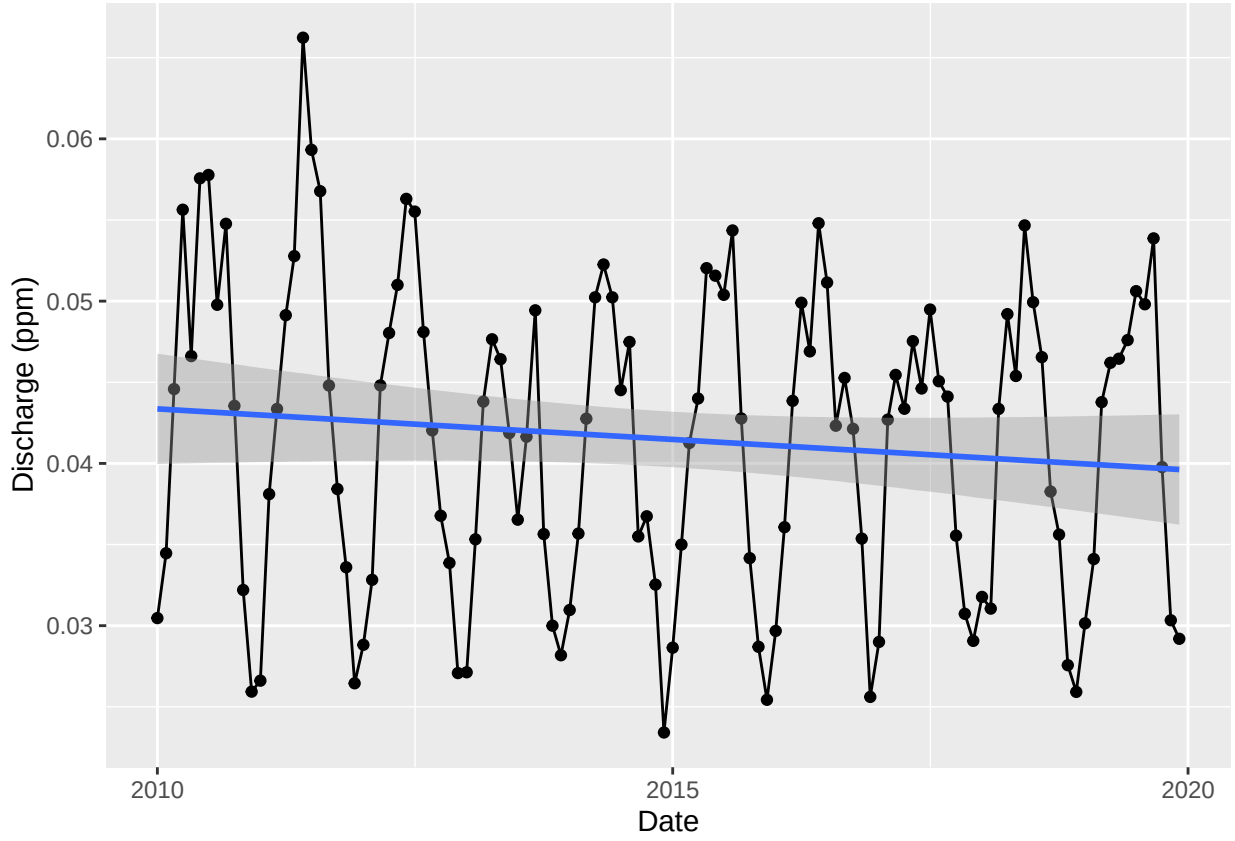
```
summary(Monthly_trend1)
```

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

Answer: The Mann-Kendall is most appropriate as it is non-parametic and missing data is allowed, it does not require data to be normally distributed. Seasonal is best as it shows a fluctuation of time with different months of the year, or seasonality.

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

monthly_data_plot <- ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
  geom_point() +
  geom_line() +
  ylab("Discharge (ppm)") +
  geom_smooth(method = lm)
print(monthly_data_plot)
```



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 plot seems to show a decreasing monthly trend over the years. While this analyis is only limited to the Garinger area, these trends can help with overall decisions centering around air quality and publich health. This trend is also statistically signifigant where $P < 0.05$ (Score = -77 , Var(Score) = 1499 denominator = 539.4972, tau = -0.143, 2-sided pvalue =0.046724).

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

GaringerOzone.Components <- as.data.frame(Monthly_decomp$time.series[,1:3])

GaringerOzone.Components <- mutate(GaringerOzone.Components,
Observed = GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration,
Date = GaringerOzone.monthly$Date)


#16

Nonseasonal_trend1 <- Kendall::MannKendall(GaringerOzone.monthly.ts)
Nonseasonal_trend1
```

```
## tau = -0.0594, 2-sided pvalue =0.33732
```

```r
summary(Nonseasonal_trend1)
```

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

Answer: The p-value for the Mann-Kendall test on the non-seasonal Ozone monthly series is 0.33732, while the p-value for the Seasonal Mann-Kendall test on the complete series is 0.046724. This means that the seasonal component has a significant impact on the trend, as the p-value for the Seasonal Mann-Kendall test is lower.