Assignment 7: Time Series Analysis

Xuening Tang

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot 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
getwd()
```

```
## [1] "/Users/xueningtang/Desktop/R lab/EDA-Fall2022/Assignments"
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

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

```
## Warning: package 'zoo' was built under R version 4.1.2
```

```
library(trend)
library(dplyr)
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.1.2
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.1.2
```

```
my.theme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(my.theme)

#2
air.2010 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                         stringsAsFactors = TRUE)
air.2011 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                         stringsAsFactors = TRUE)
air.2012 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                         stringsAsFactors = TRUE)
air.2013 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                         stringsAsFactors = TRUE)
air.2014 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                         stringsAsFactors = TRUE)
air.2015 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                         stringsAsFactors = TRUE)
air.2016 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                         stringsAsFactors = TRUE)
air.2017 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                         stringsAsFactors = TRUE)
air.2018 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
```

```
                          stringsAsFactors = TRUE)
air.2019 <- read.csv(
  "../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                          stringsAsFactors = TRUE)
GaringerOzone <- rbind(
  air.2010, air.2011, air.2012, air.2013, air.2014,
  air.2015, air.2016, air.2017, air.2018, air.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
GaringerOzone$Date <-
  as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4
GaringerOzone.short <-
  select(GaringerOzone,
         Date,
         Daily.Max.8.hour.Ozone.Concentration,
         DAILY_AQI_VALUE)

# 5
Date <- seq(as.Date("2010-01-01"),by="day",length.out=3652)
Days <- as.data.frame(Date)

# 6
GaringerOzone <- left_join(Days, GaringerOzone.short, by = c("Date"))
```

Visualize

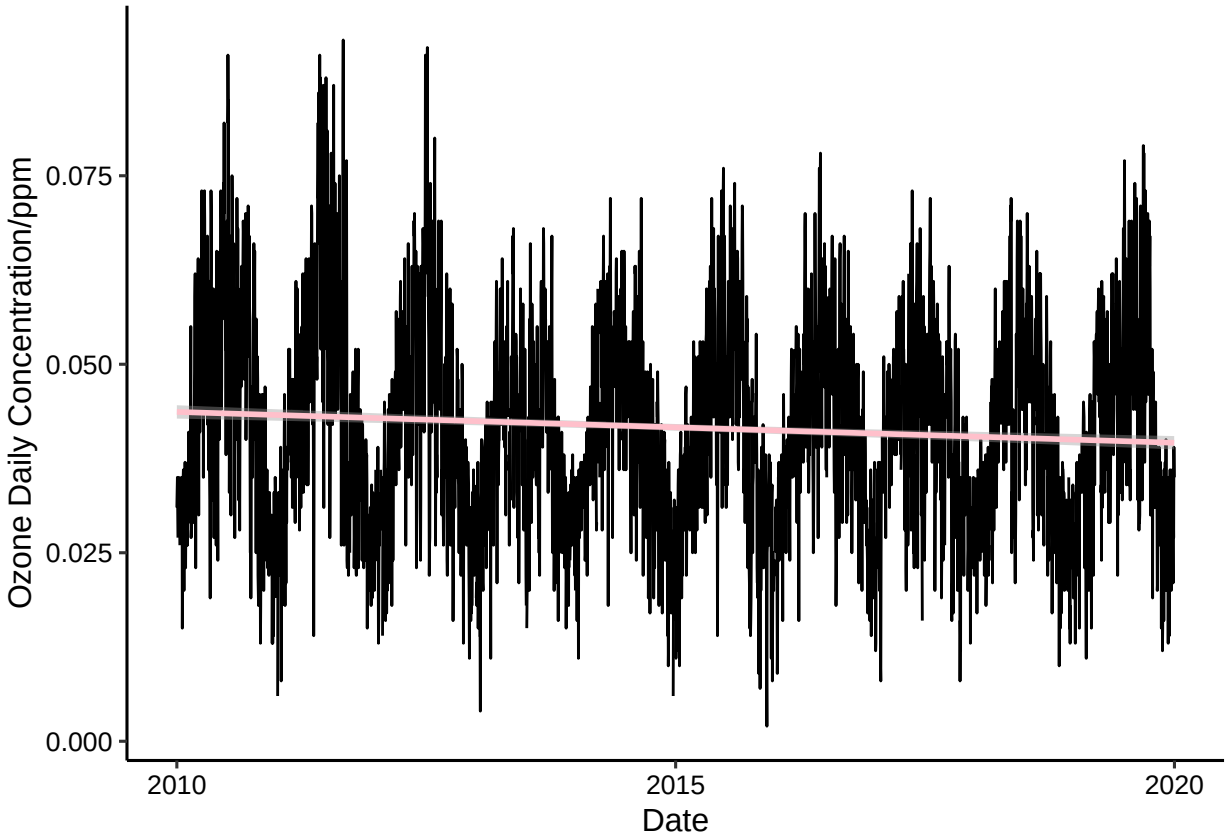
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
A7_PLOT1<-
  ggplot(GaringerOzone, aes(
    x = Date,
```

```
    y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = lm, color="pink") +
  ylab("Ozone Daily Concentration/ppm")+
  xlab("Date")+
  my.theme
print(A7_PLOT1)
```

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

```
## Warning: Removed 63 rows containing non-finite values (stat_smooth).
```



Answer: The plot I printed shows a seasonal trend over time, the ozone concentration grows up when the weather becomes warmer in spring and summer and drops down as fall and winter coming. The avaerage trend based on years is not clearly going up or going down but quite stable.

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?

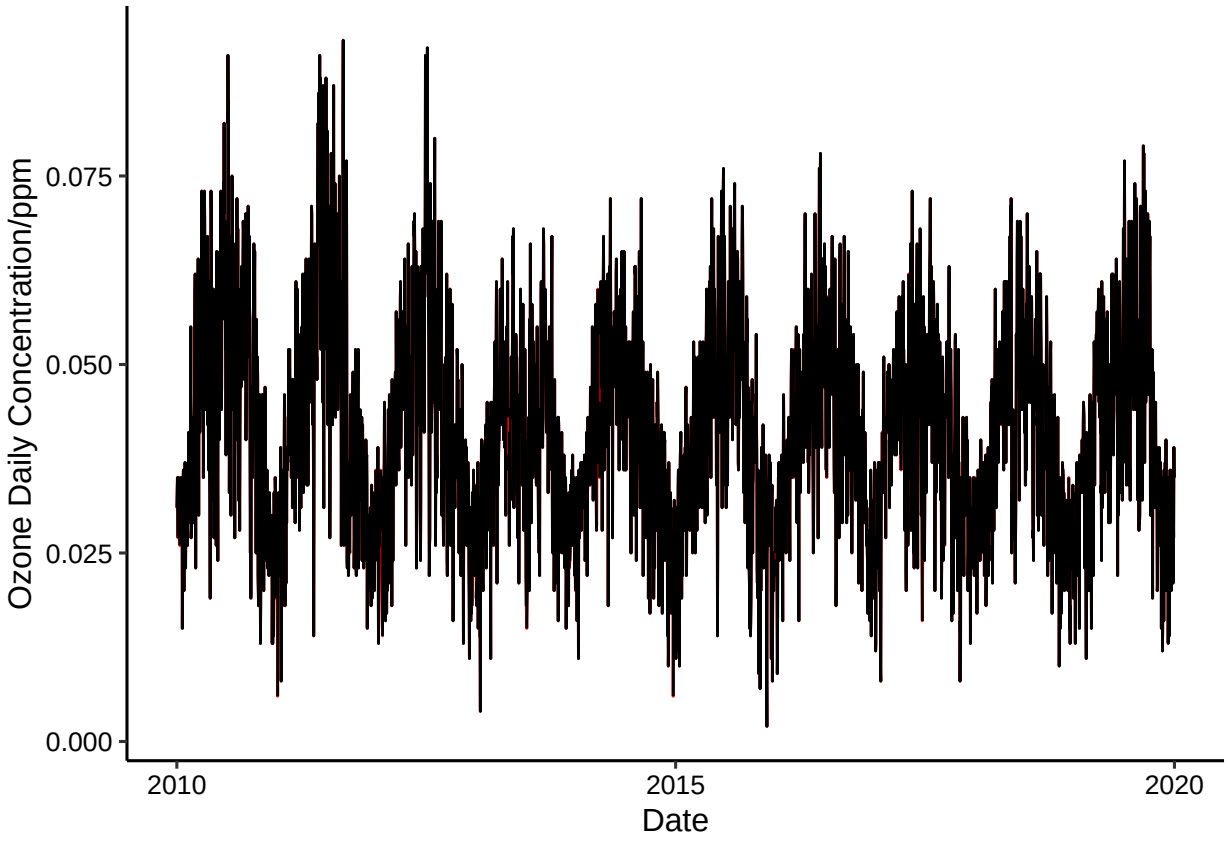
```
#8
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

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

summary(
  GaringerOzone.clean$Daily.Max.8.hour.Ozone.Concentration.clean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

```
ggplot(GaringerOzone.clean) +
  geom_line(aes(x = Date,
                y = Daily.Max.8.hour.Ozone.Concentration.clean),
            color = "red") +
  geom_line(aes(x = Date,
                y = Daily.Max.8.hour.Ozone.Concentration),
            color = "black") +
  my.theme+
  ylab("Ozone Daily Concentration/ppm")
```



Answer: Because using linear interpolation to fill in missing daily data for ozone concentration could assume the missing data to fall between the previous and next measurement, with a straight line drawn between the known points determining the values of the interpolated data on any given date. However, Piecewise constant assumes missing data to be equal to the measurement made nearest to that date (could be earlier or later), which is not suitable for our ideal model. Also, Spline is not a good one to make assumptions for missing data because it uses quadratic function to interpolate rather than drawing a straight line.

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(Month = month(Date))%>%
  mutate(Year = year(Date))%>%
  group_by(Year, Month)%>%
  summarise(MeanOzone =
              mean(Daily.Max.8.hour.Ozone.Concentration.clean))
```

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

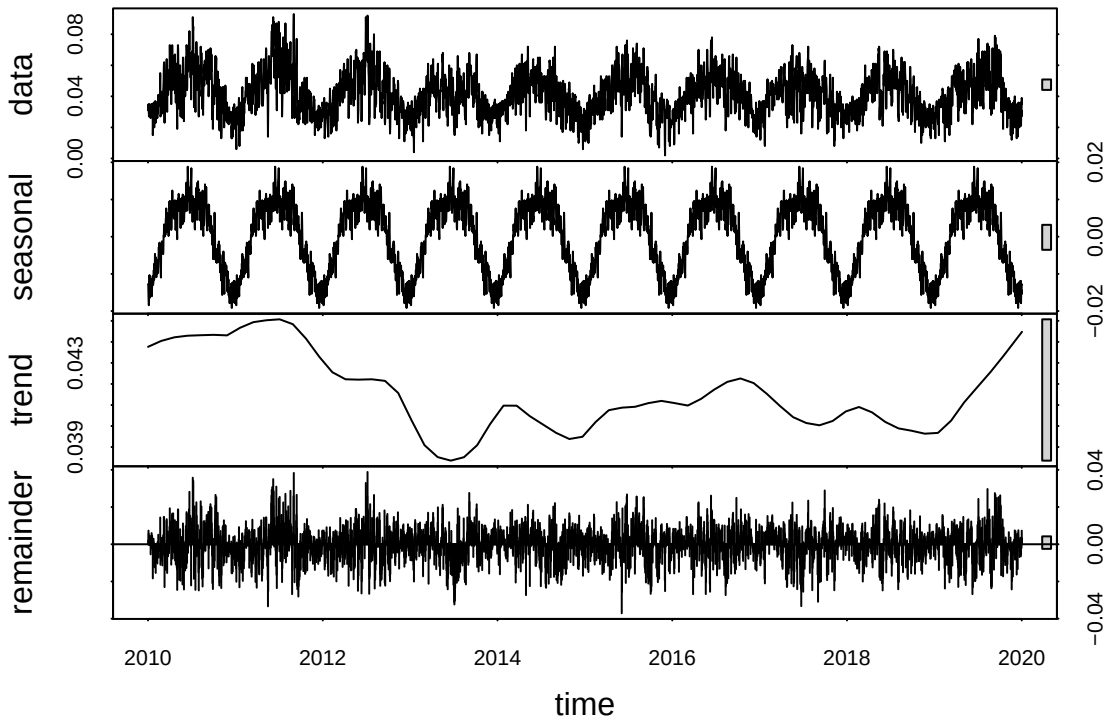
```
GaringerOzone.monthly$DATE <-
  as.yearmon(paste
             (GaringerOzone.monthly$Year,
               GaringerOzone.monthly$Month), "%Y %m")
```

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.

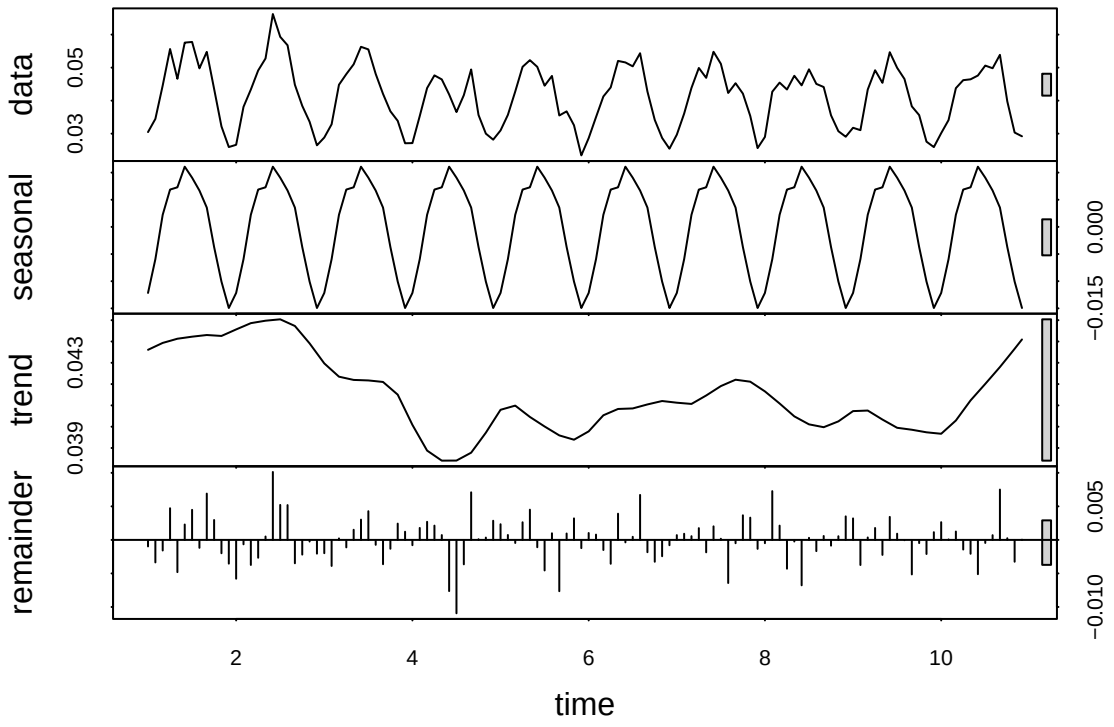
```
#10
f_month1 <- month(first(GaringerOzone.clean$Date))
f_year1 <- year(first(GaringerOzone.clean$Date))
GaringerOzone.daily.ts <-
  ts(GaringerOzone.clean$Daily.Max.8.hour.Ozone.Concentration.clean,
     start = c(f_year1,f_month1), frequency = 365)

f_month2 <- first(GaringerOzone.monthly$Month)
f_year2 <- first(GaringerOzone.clean$Year)
GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$MeanOzone,
     start = c(f_year2,f_month2), 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")
plot(GaringerOzone.daily.decomposed)
```




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

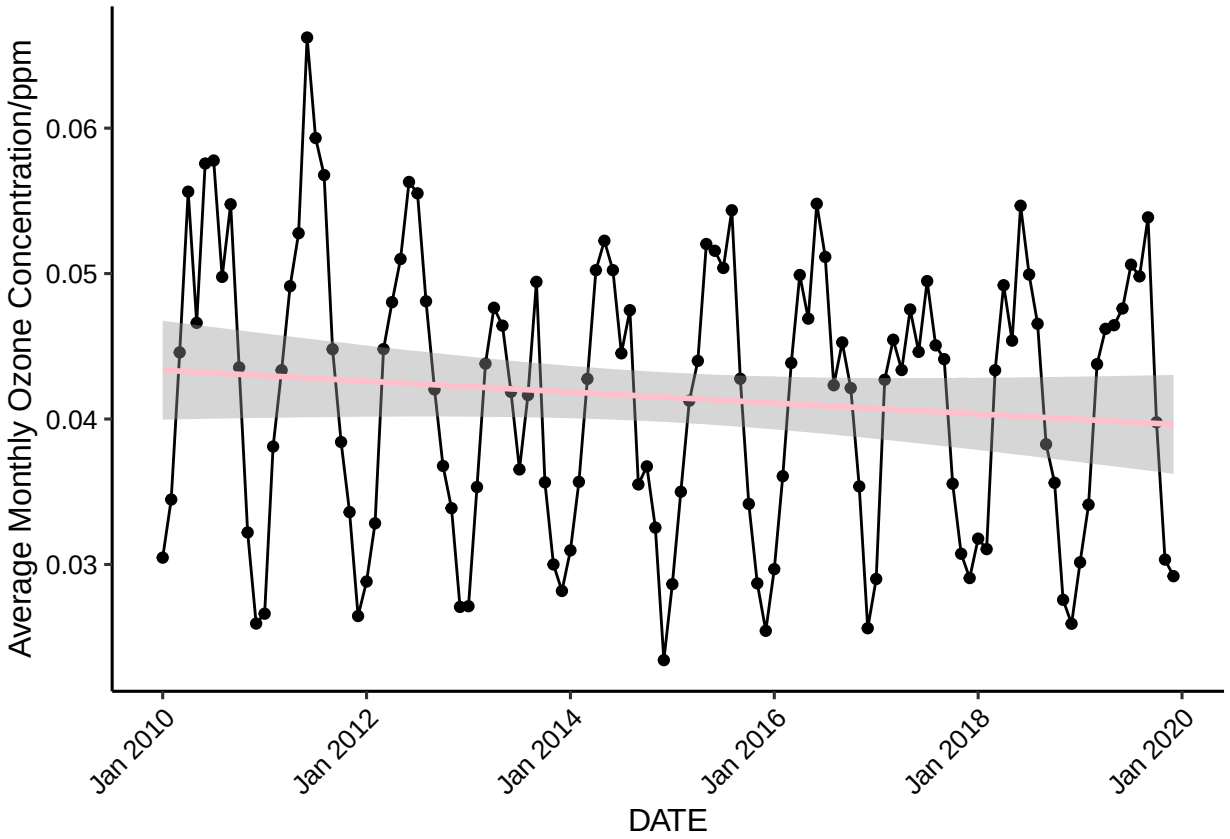
```
#12
GaringerOzone.monthly.trend <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
```

Answer: Because the seasonal Mann-Kendall could help to dipict the trend in one specfic season or month, which could fit best for the seasonal data. So in this case it is more appropriate. For other monotonic trend analysis methods, linear regression, Mann-Kendall, Spearman Rho and Augmented Dickey Fuller all lack of the seasonality analysis ability.

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
A7_PLOT2 <-
ggplot(GaringerOzone.monthly, aes(x = DATE, y = MeanOzone)) +
  geom_point() +
  geom_line() +
  ylab("Average Monthly Ozone Concentration/ppm") +
  geom_smooth(method = lm, color="pink")+
  my.theme+
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))
print(A7_PLOT2)
```

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



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: Acoording to the 10-year Ozone concentration daily data we got, there is a clear seasonal trend and slightly going down overall trend observed from the linear plot. In order to reduce the noise of the data, we calculated the mean value of each month and the monthly data refelcts very perfect seasonal trend in the time series model components. The Seasonal MannKendall model also show the statitical significant to reject there isno seasonal trend in our data. So the ozone concentration fluctuates with the season, higher concentration in warm seasons and lower in cold seasons.

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
GaringerOzone.monthly.component <-
  as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

GaringerOzone.monthly.component <-
```

```
  mutate(GaringerOzone.monthly.component,
         Observed = GaringerOzone.monthly$MeanOzone,
         Date = GaringerOzone.monthly$DATE)


#16
GaringerOzone.monthly.trend2 <-
 MannKendall(GaringerOzone.monthly.ts)
```

Answer: From the model of MannKendall, the two-side p-value is large than the 0.1 so we can not reject the null hypothesis which means there is no trend in the data. However, the two side p-value from Seasonal MannKendall is samller then 0.05, it shows the statistic significance that there is seasonal trend in the data. Also, the Seasonal Kendall model has higher statitics and magnitude of Kendall score and lower denomination and variance of S, which show the more statistical significance of the data's seasonal trend.