Assignment 7: Time Series Analysis

Meilin Chan

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 Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
#1
getwd()
```

```
## [1] "C:/Users/meili/OneDrive - Duke University/EDA/Environmental_Data_Analytics_2022/Assignments"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```
library(trend)

mytheme <- theme_bw(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")

theme_set(mytheme)
```

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.

```
#2
oz2010 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                   stringsAsFactors = TRUE)
oz2011 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                   stringsAsFactors = TRUE)
oz2012 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                   stringsAsFactors = TRUE)
oz2013 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                   stringsAsFactors = TRUE)
oz2014 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                   stringsAsFactors = TRUE)
oz2015 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                   stringsAsFactors = TRUE)
oz2016 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                   stringsAsFactors = TRUE)
oz2017 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                   stringsAsFactors = TRUE)
oz2018 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                   stringsAsFactors = TRUE)
oz2019 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                   stringsAsFactors = TRUE)

GaringerOzone <- list(oz2010, oz2011, oz2012,
                      oz2013, oz2014, oz2015,
                      oz2016, oz2017, oz2018, oz2019) %>%
  reduce(full_join) %>%
   rename(Oz.ppm = Daily.Max.8.hour.Ozone.Concentration)
```

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
class(GaringerOzone$Date)
```

```
## [1] "factor"
```

```
GaringerOzone$Date <- as.Date(GaringerOzone$Date,
                              format = "%m/%d/%Y")

# 4
clean_GaringerOzone <- GaringerOzone %>%
  select(Date, Oz.ppm, DAILY_AQI_VALUE)



# 5
library(dplyr)
Days <- as.data.frame(seq.Date(as.Date("2010-01-01"),
                               as.Date("2019-12-31"), "day"))
names(Days) <- c("Date")

# 6
GaringerOzone <- left_join(Days, clean_GaringerOzone)
```

```
## Joining, by = "Date"
```

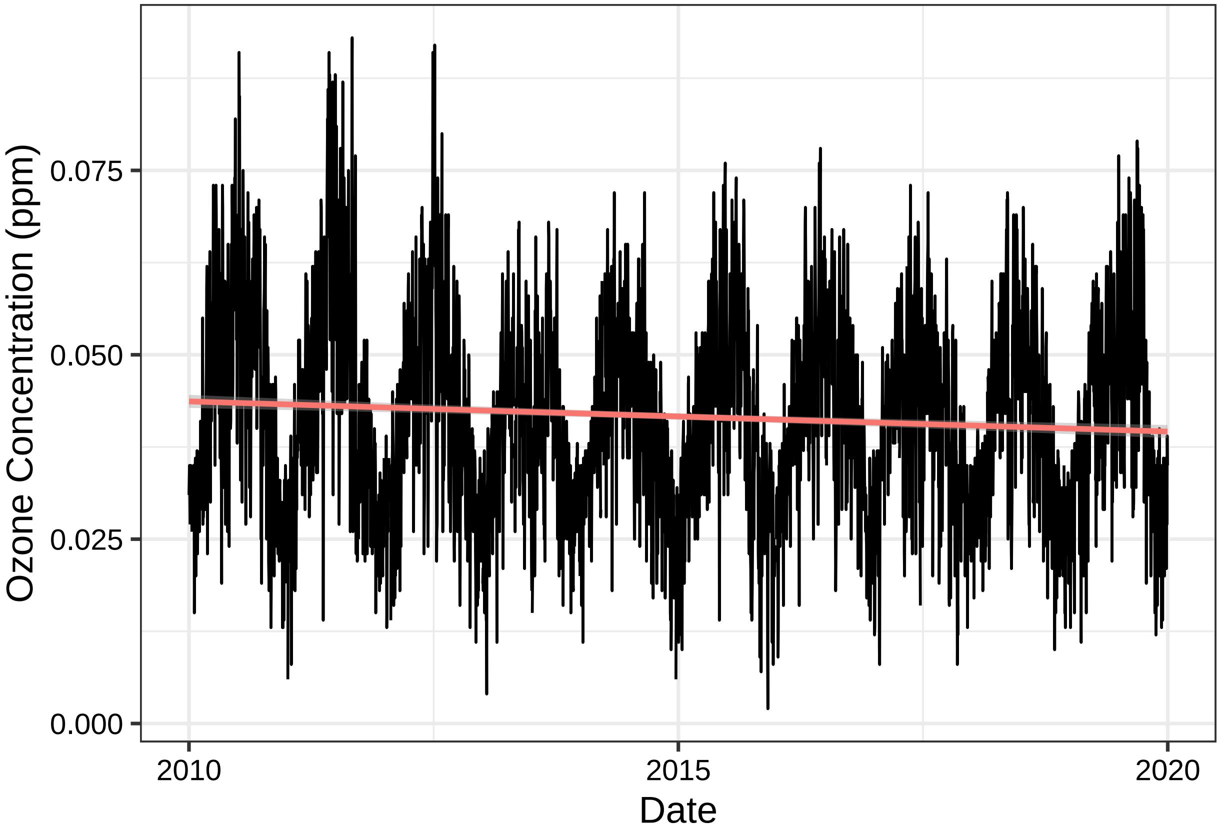
Visualize

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
line.Garinger <- ggplot(GaringerOzone,
                        aes(x= Date,
                            y = Oz.ppm)) +
  geom_line() +
  geom_smooth(method = lm, aes(color = "red")) +
  xlab("Date") +
  ylab("Ozone Concentration (ppm)") +
  theme(legend.position = "none")

print(line.Garinger)
```

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

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



Answer: There does seem to be a trend in ozone concentrations over time. Ozone concentrations tend to cycle through an increase and decrease pattern over time - the concentrations could follow a seasonal trend based on the plot. From 2010 to 2020, there seems to be indication of a slight decrease in ozone concentrations.

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?

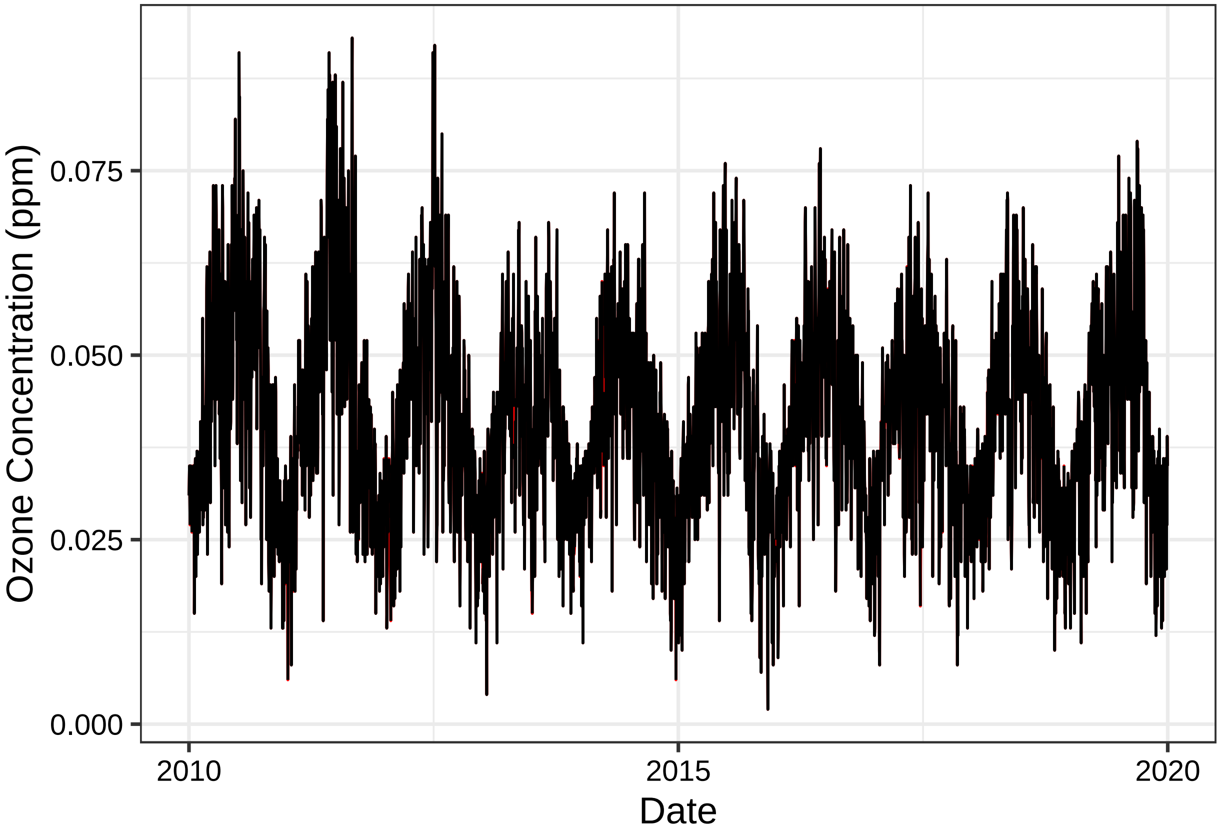
```
#8
GaringerOzone.clean <- GaringerOzone %>%
  mutate(Oz.ppm.clean = zoo::na.approx(Oz.ppm),
         DAILY_AQI_VALUE.clean = zoo::na.approx(DAILY_AQI_VALUE))
summary(GaringerOzone)
```

```
##       Date                Oz.ppm        DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200   Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200   1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100   Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163   Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100   3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300   Max.   :169.00
##                       NA's   :63        NA's   :63
```

```
summary(GaringerOzone.clean)
```

```
##       Date                Oz.ppm        DAILY_AQI_VALUE   Oz.ppm.clean    
##  Min.   :2010-01-01   Min.   :0.00200   Min.   :  2.00   Min.   :0.00200  
##  1st Qu.:2012-07-01   1st Qu.:0.03200   1st Qu.: 30.00   1st Qu.:0.03200  
##  Median :2014-12-31   Median :0.04100   Median : 38.00   Median :0.04100  
##  Mean   :2014-12-31   Mean   :0.04163   Mean   : 41.57   Mean   :0.04151  
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100   3rd Qu.: 47.00   3rd Qu.:0.05100  
##  Max.   :2019-12-31   Max.   :0.09300   Max.   :169.00   Max.   :0.09300  
##                       NA's   :63        NA's   :63                        
##  DAILY_AQI_VALUE.clean
##  Min.   :  2.00       
##  1st Qu.: 30.00       
##  Median : 38.00       
##  Mean   : 41.41       
##  3rd Qu.: 47.00       
##  Max.   :169.00       
## 
```

```
lin.int.Garinger <- ggplot(GaringerOzone.clean) +
  geom_line(aes(x = Date, y = Oz.ppm.clean), color = "red") +
  geom_line(aes(x = Date, y = Oz.ppm), color = "black") +
  ylab("Ozone Concentration (ppm)")

print(lin.int.Garinger)
```



Answer: Spline is similar to a linear interpolation, except instead of drawing a straight line, spline would perform a quadratic function to connect gaps. Piecewise constant would fill in gaps by assuming that data would be the same as measurements from nearby dates. Based on our plot, increasing and decreasing trends within a seasonal cycle look to be linear and using spline would not fill in data gaps in a way that would align with the overall trend. Piecewise constant would also not fill in data gaps that would follow the linear increasing and decreasing trends. From tne dataframe itself, it looks that data day to day does not tend to be equal to each other - there is either a decrease or increase compared to the measurement the day before. Thus using linear interpolation was best for this data set as it would align with the overall linear increasing and decreasing trend within each seasonal cycle.

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)

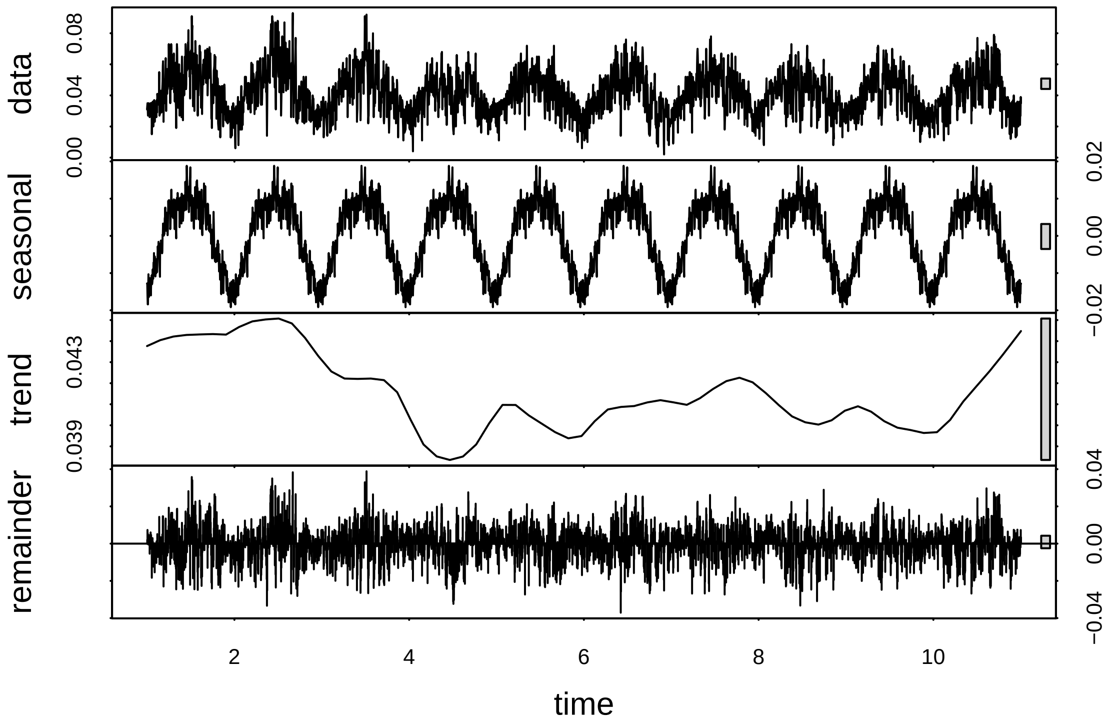
```
#9
GaringerOzone.monthly <- GaringerOzone.clean %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  mutate(F_month = my(paste0(Month, "-", Year))) %>%
  group_by(F_month) %>%
  dplyr::summarise(Mean_Ozone = mean(Oz.ppm.clean))
```

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.

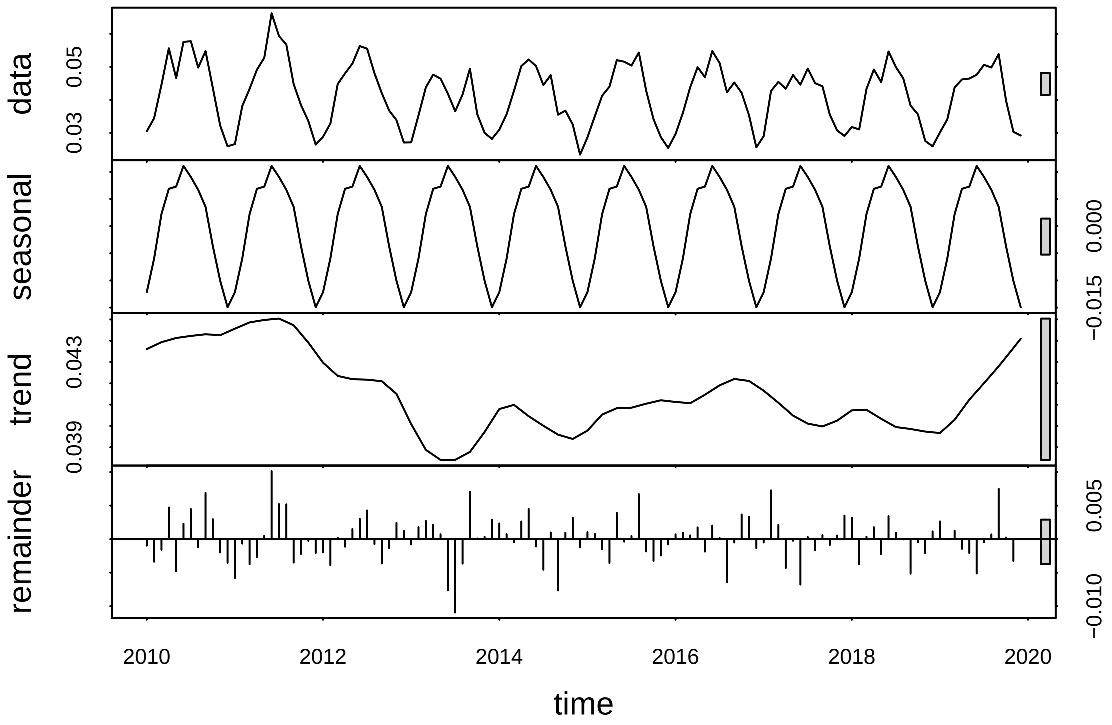
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone.clean$Oz.ppm.clean,
                             frequency = 365)


first.month <- month(first(GaringerOzone.monthly$F_month))
first.year <- year(first(GaringerOzone.monthly$F_month))

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

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

```
#11
Garinger.daily.decomp <- stl(GaringerOzone.daily.ts,
                             s.window = "periodic")
plot(Garinger.daily.decomp)
```



```
Garinger.monthly.decomp <- stl(GaringerOzone.monthly.ts,
                               s.window = "periodic")
plot(Garinger.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
library(Kendall)

Garinger_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
Garinger_trend
```

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

```
summary(Garinger_trend) #p-value < 0.05 - reject null hypothesis that there is no trend
```

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

```
Garinger_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
summary(Garinger_trend2) #p-value is greater than 0.05 in all the seasons - stationary trend
```

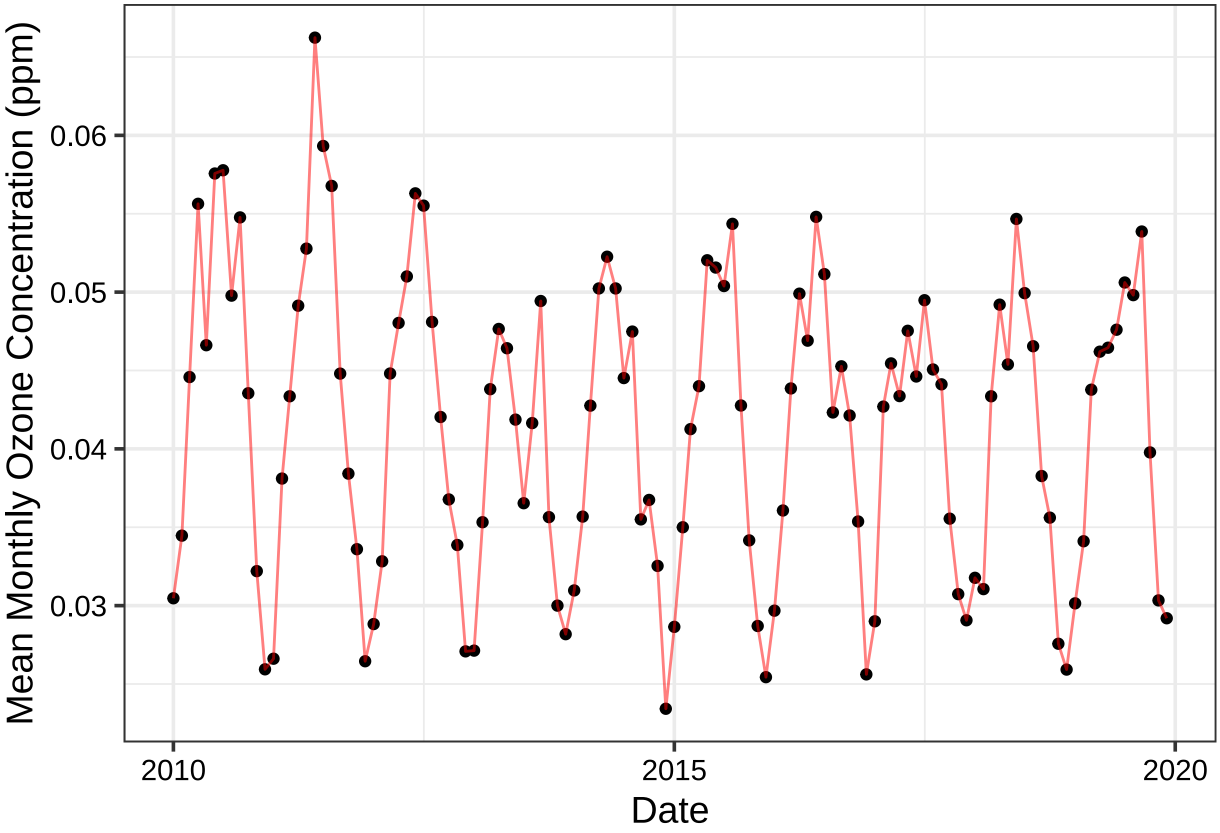
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
```

```
## 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: Looking at the monthly time series plot, we can see that ozone concentrations follow a seasonal trend with cycles of increasing and decreasing concetnration observations. So right away we can determine that a linear regression would not be a good monotonic trend analysis to apply. We can also see that the observations are non-parametric (especially a ozone concentration observations. Although Mann-Kendall and Spearman Ro are for non-parametric trends, they are used for non-seasonality. Dickey FUller cannot be used as it is for stochastic observations. Thus, the seasonal Mann-Kendall is the most appropriate analysis to use as it is for seasonal AND non-parametric trends.

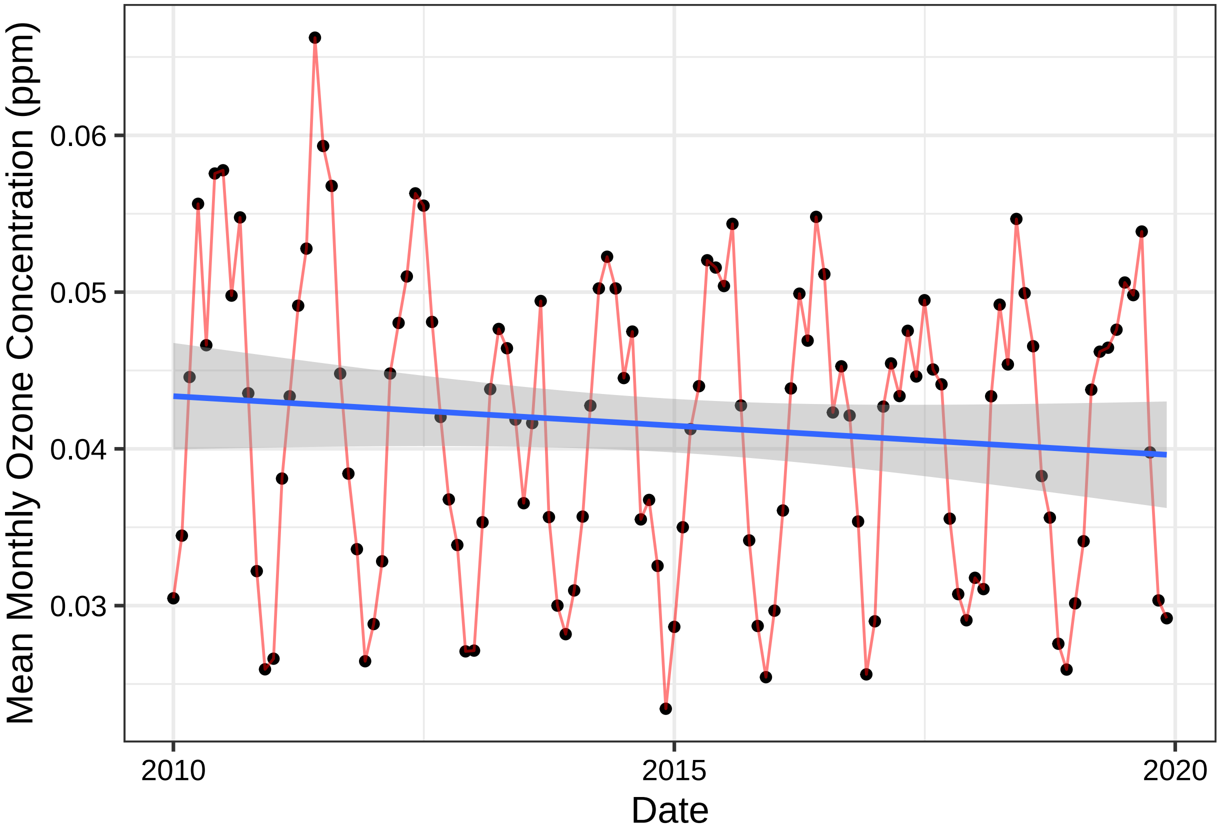
13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13
Garinger.monthly.time <- ggplot(GaringerOzone.monthly,
                                aes(x = F_month, y = Mean_Ozone)) +
  geom_point() +
  geom_line(color = "red", alpha = 0.5) +
  xlab("Date") +
  ylab("Mean Monthly Ozone Concentration (ppm)")

plot(Garinger.monthly.time)
```



```r
trend.Garinger.monthly.time <- ggplot(GaringerOzone.monthly,
                                      aes(x = F_month, y = Mean_Ozone)) +
  geom_point() +
  geom_line(color = "red", alpha = 0.5) +
  geom_smooth(method = lm) +
  xlab("Date") +
  ylab("Mean Monthly Ozone Concentration (ppm)")

plot(trend.Garinger.monthly.time)
```

```
## `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: Our study question was whether or not ozone concentrations at this particular station changed over time since 2010. From our seasonal Mann Kendall test, we find that we should reject our null-hypothesis that there is no change/no change in trend of ozone concentrations at this station since 2010 onwards (Score = -77 ,Var(Score) = 1499, denominator = 539.4972, tau = -0.143, 2-sided pvalue =0.046724).Thus since 2010, there have been changes in monthly ozone concentrations. From our smk.test(), we see that each season has a p-value > 0.05 meaning within each season, there is no large change in trends. Across S-values, there are increasing and decreasing trends (some S-value is greater than 0 and other values are less than 0). However, since values are close to 0, the trend seems to be more stationary than fluctuating.

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
noseason.GaringerOzone.monthly <- as.data.frame(Garinger.monthly.decomp$time.series[,2:3])

noseason.GaringerOzone.monthly <- noseason.GaringerOzone.monthly %>%
  mutate(Date = GaringerOzone.monthly$F_month,
         Mean_Ozone = GaringerOzone.monthly$Mean_Ozone) %>%
  select(Date, Mean_Ozone, trend, remainder)
```

```
fmonth <- month(first(noseason.GaringerOzone.monthly$Date))
fyear <- year(first(noseason.GaringerOzone.monthly$Date))

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

#16

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

```
## tau = -0.595, 2-sided pvalue =< 2.22e-16
```

```
summary(noseason.Garinger_trend) #p-value < 0.05 --> reject null hypothesis
```

```
## Score =  -68444 , Var(Score) = 12326265
## denominator =  114959
## tau = -0.595, 2-sided pvalue =< 2.22e-16
```

> Answer: The p-value is less than 0.05 indicating that we can reject the null hypothsis. This means that after the removal of the seasonal component in our data set and through the utilization of the Mann Kendall monotonic trend analysis, we can accept the hypothesis that there is a trend within our data set. Our analysis for question 16 is the same as the analysis found for the Seasonal Mann Kendall as it found that p-value < 0.05.