Assignment 7: Time Series Analysis

Britney Pepper

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

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=50),tidy=TRUE, echo = T)
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=60), tidy=TRUE, echo=TRUE)

#1
getwd()
```

```
## [1] "/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Data_Analytics_
```

```
#setwd("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Data_Analytics_

#install.packages("tidyverse")
#install.packages("lubridate")
#install.packages("zoo")
#install.packages("trend")
#install.packages("Kendall")
#install.packages("tseries")

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   1.0.8
## 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)
library(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#Theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
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

# reading the data sets
o3_2010 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2011 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2012 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2013 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2014 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2015 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2016 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2017 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2018 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)
o3_2019 <- read.csv("/Users/britneypepper/Desktop/ENVIRON 872 and L/GitHubRepositories/Environmental_Da
    stringsAsFactors = TRUE)


# combining the data sets
GaringerOzone <- rbind(o3_2010, o3_2011, o3_2012, o3_2013, o3_2014,
    o3_2015, o3_2016, o3_2017, o3_2018, o3_2019)
# contains 3589 observation and 20 variables
```

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
GaringerOzoneWrangle <- GaringerOzone %>%
    select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
```

```
# 5
Days <- as.data.frame(seq.Date(from = as.Date("2010/01/01"),
    to = as.Date("2019/12/31"), by = "day"))
names(Days)[1] <- "Date"

# 6
GaringerOzone <- left_join(Days, GaringerOzoneWrangle, by = "Date")
```

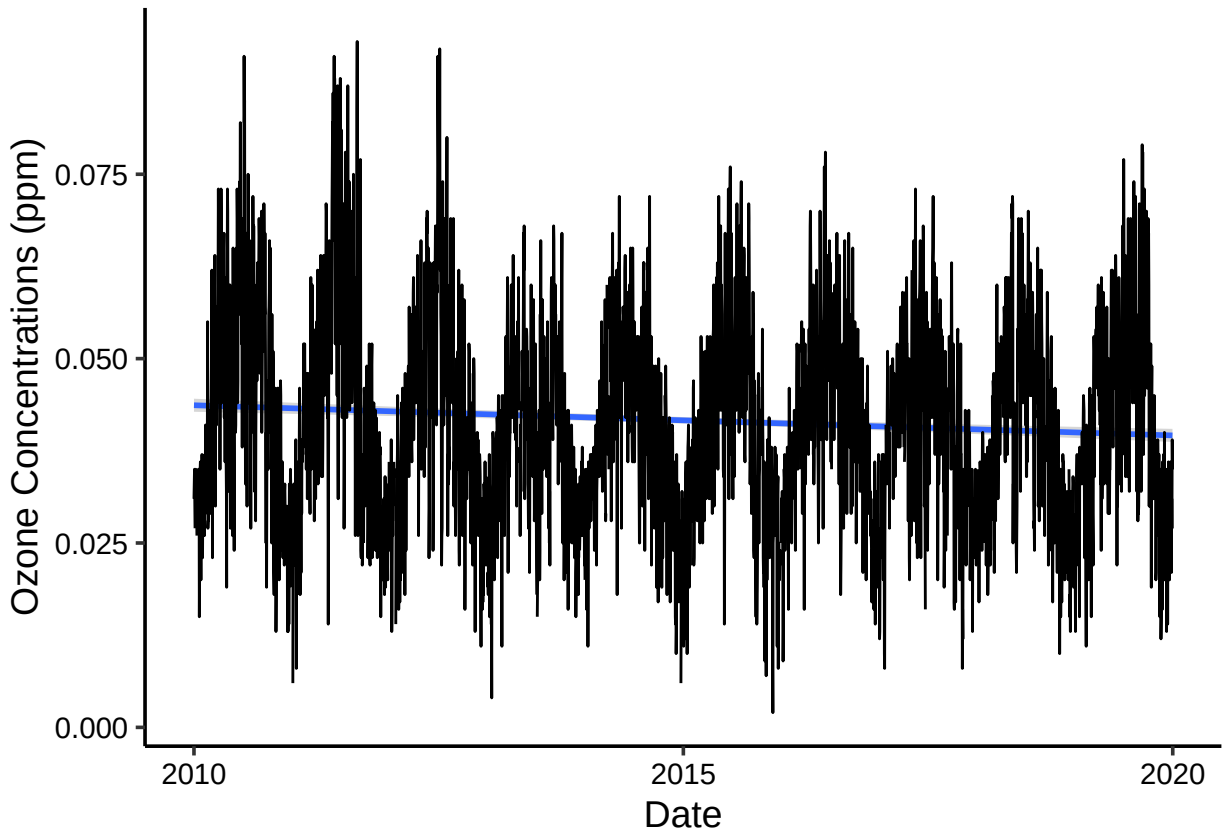
Visualize

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
GaringerOzone.line <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    ylab("Ozone Concentrations (ppm)") + xlab("Date") + geom_smooth(method = "lm") +
    geom_line()
print(GaringerOzone.line)
```

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

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



Answer: The plot does suggest a trend in ozone concentration over time. The linear trend line shows a slight negative relationship, but the trend line only represents the average ozone concentration instead of the pattern of the ozone concentration levels. The trend in the concentration is an oscillation where the ozone concentration level increases from the beginning of each the year until the middle of each year before it decreases until the end of the year.

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
head(GaringerOzone)
```

```
##         Date Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
## 1 2010-01-01                                0.031              29
## 2 2010-01-02                                0.033              31
## 3 2010-01-03                                0.035              32
## 4 2010-01-04                                0.031              29
## 5 2010-01-05                                0.027              25
## 6 2010-01-06                                   NA              NA
```

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

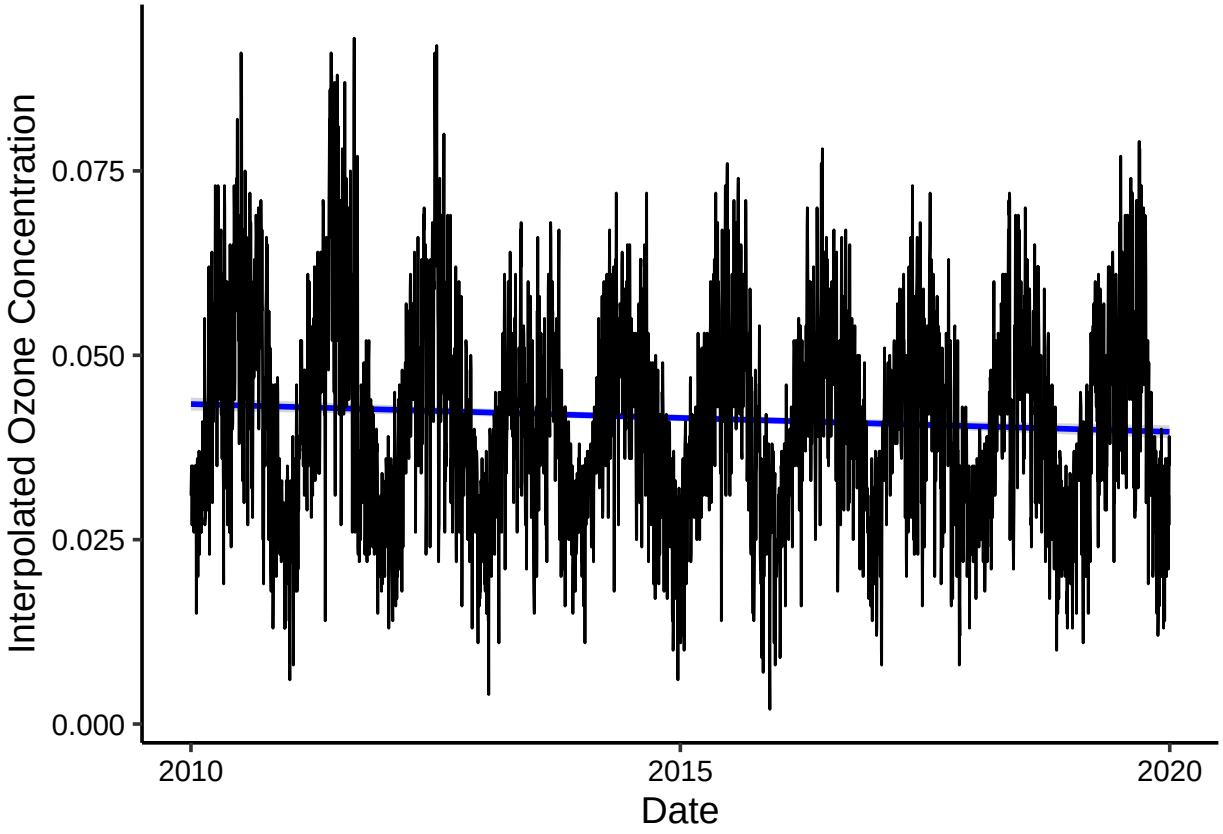
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
GaringerOzone_interp <- GaringerOzone %>%
    mutate(Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentra
# I really tried to get this all on the page but I couldn't
# :( so sorry! Here is what the code is says:
# mutate(Daily.Max.8.hour.Ozone.Concentration.clean =
# zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(GaringerOzone_interp$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
```

```
GaringerOzone_interp_plot <- ggplot(GaringerOzone_interp, aes(x = Date,
    y = Daily.Max.8.hour.Ozone.Concentration.clean)) + ylab("Interpolated Ozone Concentration") +
    xlab("Date") + geom_smooth(method = "lm", colour = "blue") +
    geom_line()
print(GaringerOzone_interp_plot)
```

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



Answer: Because the dataset does not have many NA's, there are not many gaps in the data. Therefore, we would not need to use a piecewise constant because there is no need to assume that the missing measurements are equal to the nearest measurements in the data. Also, we are looking to see how the ozone concentrations have changed, so we would not want to assume that the gaps are where the data was constant. We do not need to use Spline because we have data from most days in the year, so there is not much need to go through the quadratic function to fill in the missing data. It is safe to assume that the missing data fall between previous and next measurements because of the daily measurements, and a straight line between the data is a good way to visualize the ozone concentrations changes.

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)

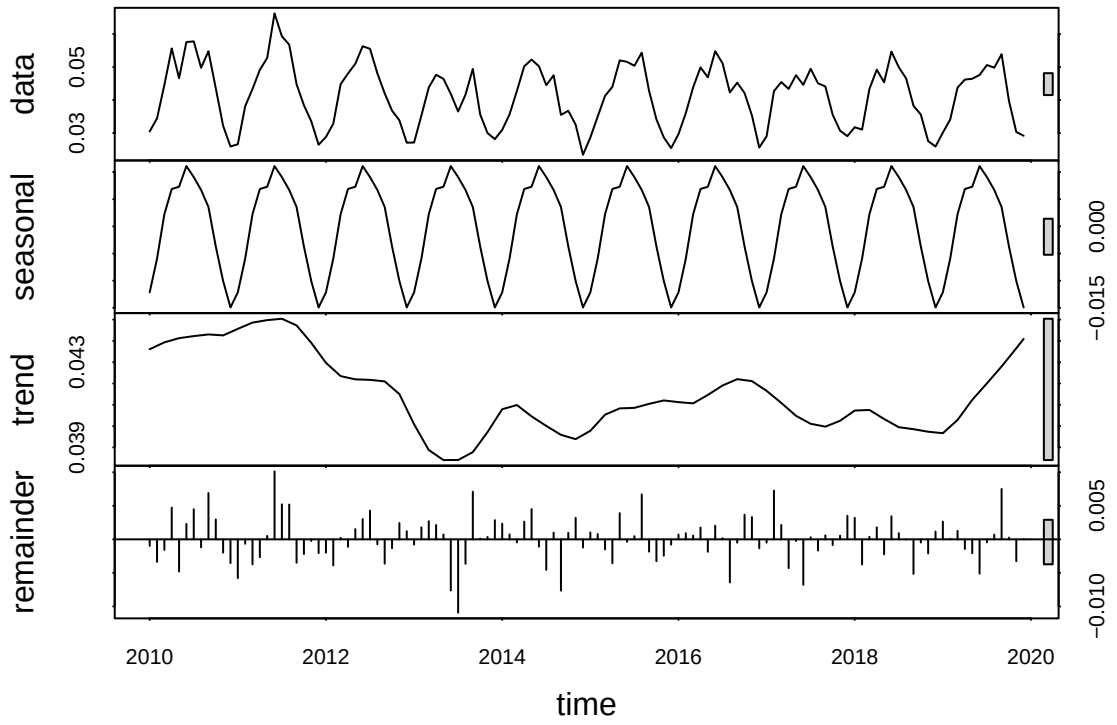
```
# 9
GaringerOzone.monthly <- GaringerOzone_interp %>%
    mutate(Month = month(Date)) %>%
    mutate(Year = year(Date)) %>%
    mutate(Date_M_Y = my(paste0(Month, "-", Year))) %>%
    dplyr::group_by(Date_M_Y) %>%
    dplyr::summarise(Month_Mean_O3 = mean(Daily.Max.8.hour.Ozone.Concentration.clean))
```

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.

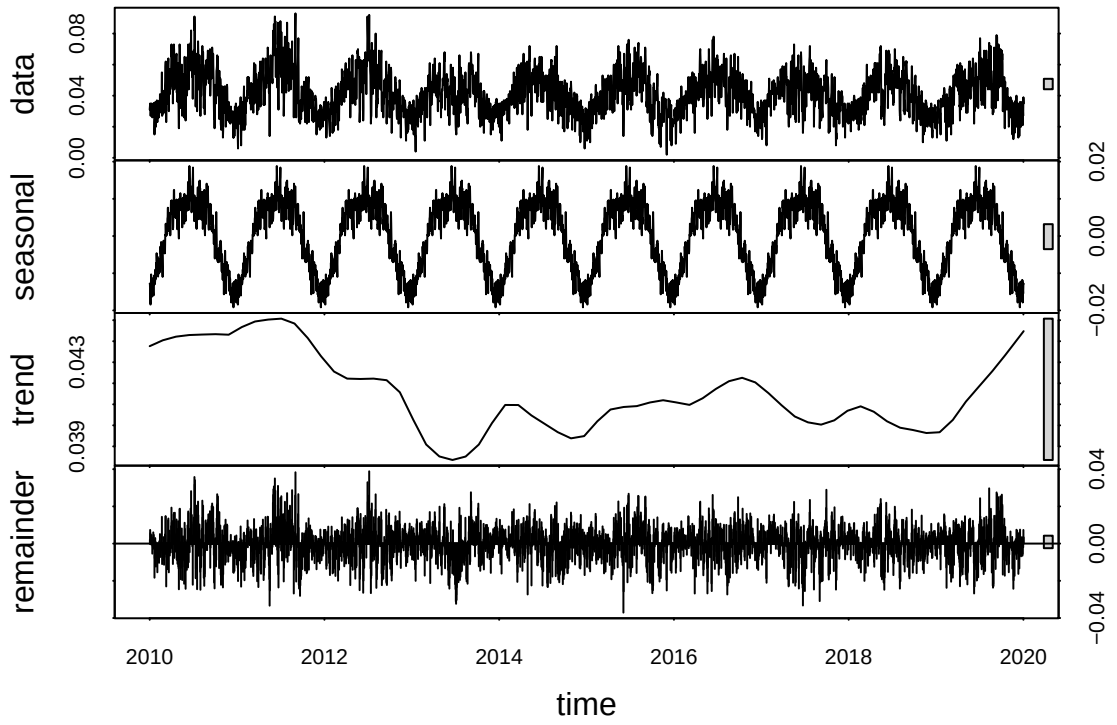
```r
# 10
f_month <- month(first(GaringerOzone_interp$Date))
f_year <- year(first(GaringerOzone_interp$Date))

Garinger.Ozone.monthly.ts <- ts(GaringerOzone.monthly$Month_Mean_O3,
    start = c(f_year, f_month), frequency = 12)

Garinger.Ozone.daily.ts <- ts(GaringerOzone_interp$Daily.Max.8.hour.Ozone.Concentration.clean,
    start = c(f_year, f_month), frequency = 365)

# monthly data is always frequency=12 and yearly data is
# always frequency=365
```

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

```r
# 11
Garinger.Ozone.monthly.ts.decomposed <- stl(Garinger.Ozone.monthly.ts,
    s.window = "periodic")
plot(Garinger.Ozone.monthly.ts.decomposed)
```




```r
Garinger.Ozone.daily.ts.decomposed <- stl(Garinger.Ozone.daily.ts,
    s.window = "periodic")
plot(Garinger.Ozone.daily.ts.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?

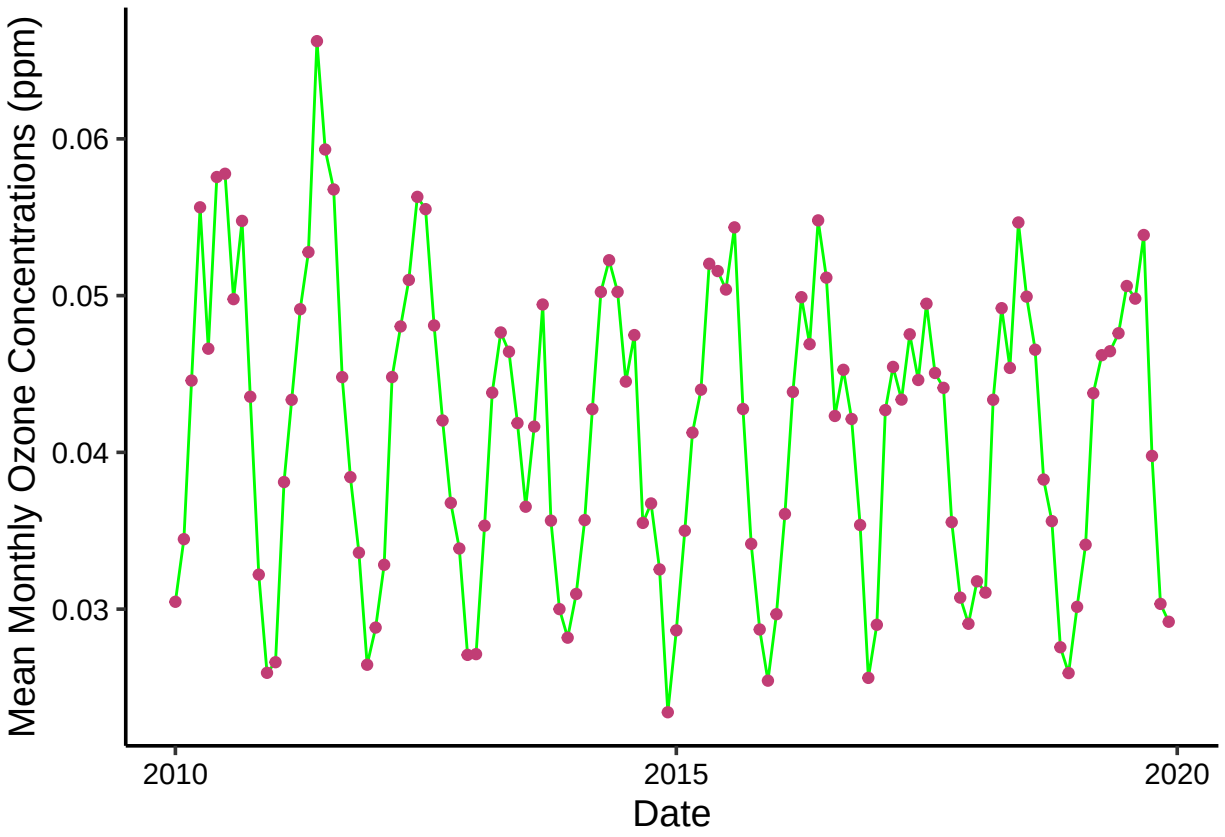
```
# 12 Run SMK test
Garinger.Ozone.monthly.ts.trend <- Kendall::SeasonalMannKendall(Garinger.Ozone.monthly.ts)
summary(Garinger.Ozone.monthly.ts.trend)
```

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

Answer: The season Mann-Kendall is most appropriate because we are looking at monthly averages within the dataset. This monthly change can be considered seasonal trends, and the only monotonic trend analysis that is for seasonality is the seasonal Mann-Kendall analysis. The data is also non-parametric, so it fits the sesason Mann-Kendall analysis.

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
GaringerOzoneMonthlyPlot <- ggplot(GaringerOzone.monthly) + geom_line(aes(y = Month_Mean_O3,
    x = Date_M_Y), color = "green") + geom_point(aes(y = Month_Mean_O3,
    x = Date_M_Y), color = "#c13d75ff") + ylab("Mean Monthly Ozone Concentrations (ppm)") +
    xlab("Date")
print(GaringerOzoneMonthlyPlot)
```



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: Based on the graphs, there appear to be seasonal trends. Each year, the seasonal graph has the same pattern present and there does not seem to be much variation in the trend with a small remainder.The statistical test from the seasonl Mann-Kendall returned a p-value of 0.0467, which is less than an alpha value of 0.05. Therefore we would say that there is seasonal trends occuring over the years (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.

```
# 15
Garinger.No.Seasonality <- as.data.frame(Garinger.Ozone.monthly.ts.decomposed$time.series[,
    2:3])

f_month <- month(first(GaringerOzone_interp$Date))
f_year <- year(first(GaringerOzone_interp$Date))

Garinger.Ozone.monthly.No.Seasonality.ts <- ts(Garinger.No.Seasonality,
    start = c(f_year, f_month), frequency = 12)
```

```r
# 16 Run MK test
Garinger.No.Seasonality.Monthly.trend <- MannKendall(Garinger.Ozone.monthly.No.Seasonality.ts)
summary(Garinger.No.Seasonality.Monthly.trend)
```

```
## Score =  -16300 , Var(Score) = 1545533
## denominator =  28680
## tau = -0.568, 2-sided pvalue =< 2.22e-16
```

```r
# Run SMK test
Garinger.Ozone.monthly.ts.trend <- Kendall::SeasonalMannKendall(Garinger.Ozone.monthly.ts)
summary(Garinger.Ozone.monthly.ts.trend)
```

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

Answer: The results for the non-seasonal Ozone monthly series have a very small p-value that is less than an alpha of 0.05. Therefore, we would say that there are trends occuring in the non-seasonal Ozone monthly data (Score = -16300 , Var(Score) = 1545533, denominator = 28680, tau = -0.568, 2-sided pvalue =< 2.22e-16). Both the Mann Kendal test and the seasonal Mann Kendall test produce p-values that are smaller than 0.05. Therefore we would say that there are both season trends and non-seasonal trends occuring in the Ozone monthly series data.