Assignment 7: Time Series Analysis

Shirley Fontanié

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] "/Users/shirleyfontanie/Documents/DUKE/EDA/Environmental_Data_Analytics_2022"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     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(dplyr)

mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"))
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

EPA_2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_RAW.csv", stringsAsFactors =

EPA_2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_RAW.csv", stringsAsFactors =

EPA_2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_RAW.csv", stringsAsFactors =

EPA_2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_RAW.csv", stringsAsFactors =

EPA_2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_RAW.csv", stringsAsFactors =

EPA_2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_RAW.csv", stringsAsFactors =

EPA_2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_RAW.csv", stringsAsFactors =

EPA_2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_RAW.csv", stringsAsFactors =

EPA_2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_RAW.csv", stringsAsFactors =

EPA_2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_RAW.csv", stringsAsFactors =


# COMBINE DATAFRAMES INTO ONE BIG DATAFRAME:
GaringerOzone <- rbind(EPA_2010,EPA_2011,EPA_2012,EPA_2013,EPA_2014,
                       EPA_2015,EPA_2016,EPA_2017,EPA_2018,EPA_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.subset <-GaringerOzone %>%
  select(Date,Daily.Max.8.hour.Ozone.Concentration,DAILY_AQI_VALUE)

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

colnames(Days) <- c("Date")

# 6
GaringerOzone2 <- left_join(Days,GaringerOzone.subset)
```

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

```
# I already had a dataset named GaringerOzone,
# Because the directions said so earlier
# (not my fault)
# so this will have to do.
# Please don't take points off okie byeeeee :) <3
```

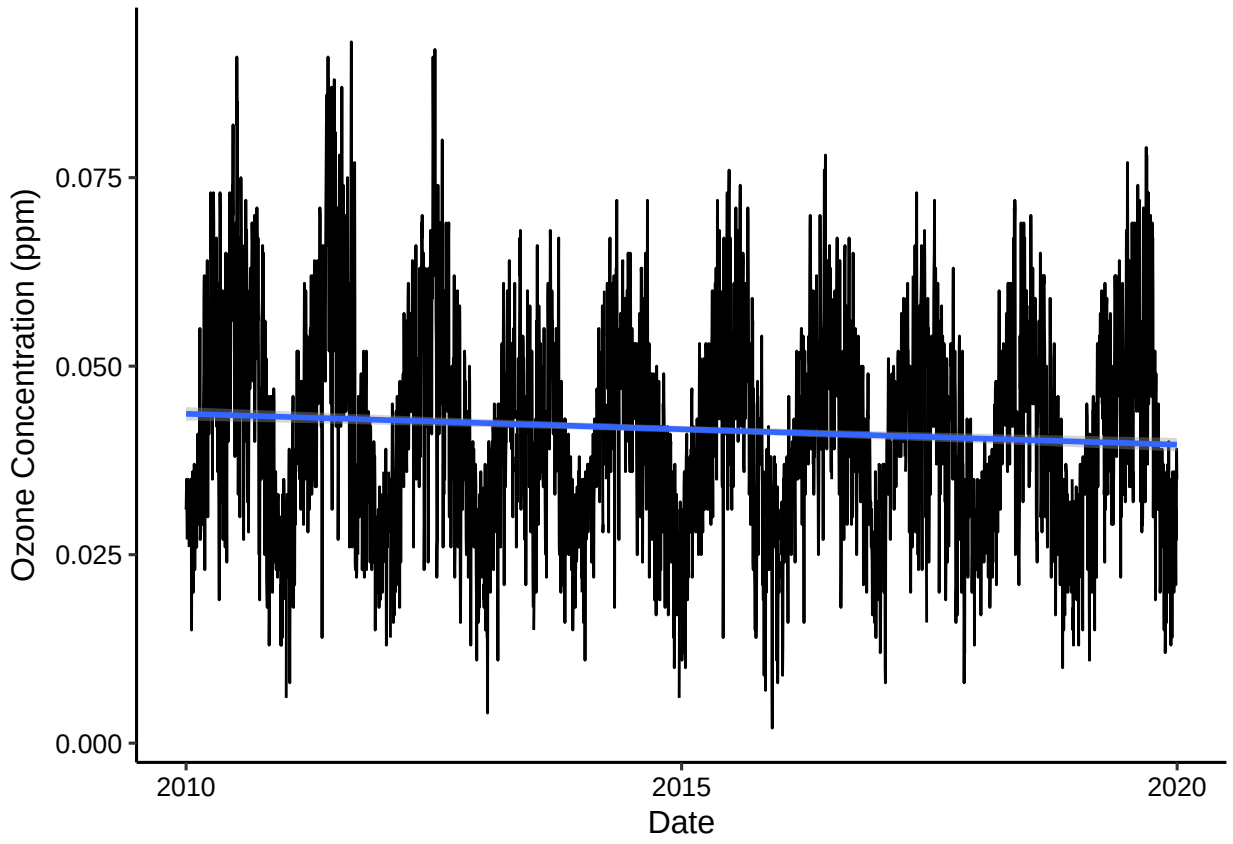
Visualize

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
GaringerPlot <- ggplot(GaringerOzone2, aes(x = Date, y =              Daily.Max.8.hour.Ozone.Concentrati
              geom_line()+
              geom_smooth(method = lm)+
  ylab(expression(paste("Ozone Concentration ", "(ppm)")))
print(GaringerPlot)
```

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

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



Answer: My plot suggests there is a slight decrease in ozone concentration over time.

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

clean_GaringerOzone <- GaringerOzone2 %>%
  mutate(OzoneConc.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: We did not use a piecewise constant or spline interpolation because linear interpolation is "connecting the dots" that draws a straight line, when one plots the dataset. We did not use spline interpolation, because it is quadratic so it is better for dips or exponential trends. We did not use piecewise constant, it is called the "nearest neighbor" approach. Any missing data are assumed to be equal to the measurement made nearest to that date (could be earlier or later).In this scenario for this dataset, linear interpolation works best and is the cleanest result.

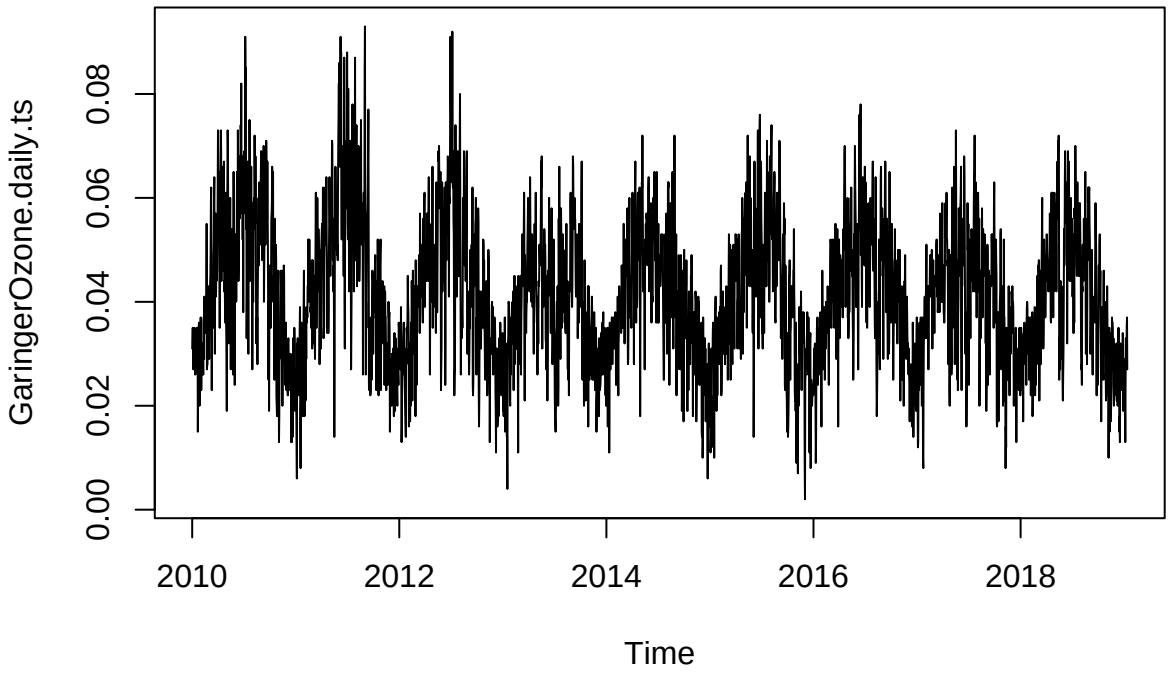
9. Create a new data frame called `GaringerOzone.monthly` that contains aggregated data: mean ozone concentrations for each month. In your pipe, you will need to first add columns for year and month to form the groupings. In a separate line of code, create a new Date column with each month-year combination being set as the first day of the month (this is for graphing purposes only)

```
#9
GaringerOzone.monthly <- clean_GaringerOzone %>%
    mutate(Month = month(Date)) %>%
    mutate(Year = year(Date)) %>%
```

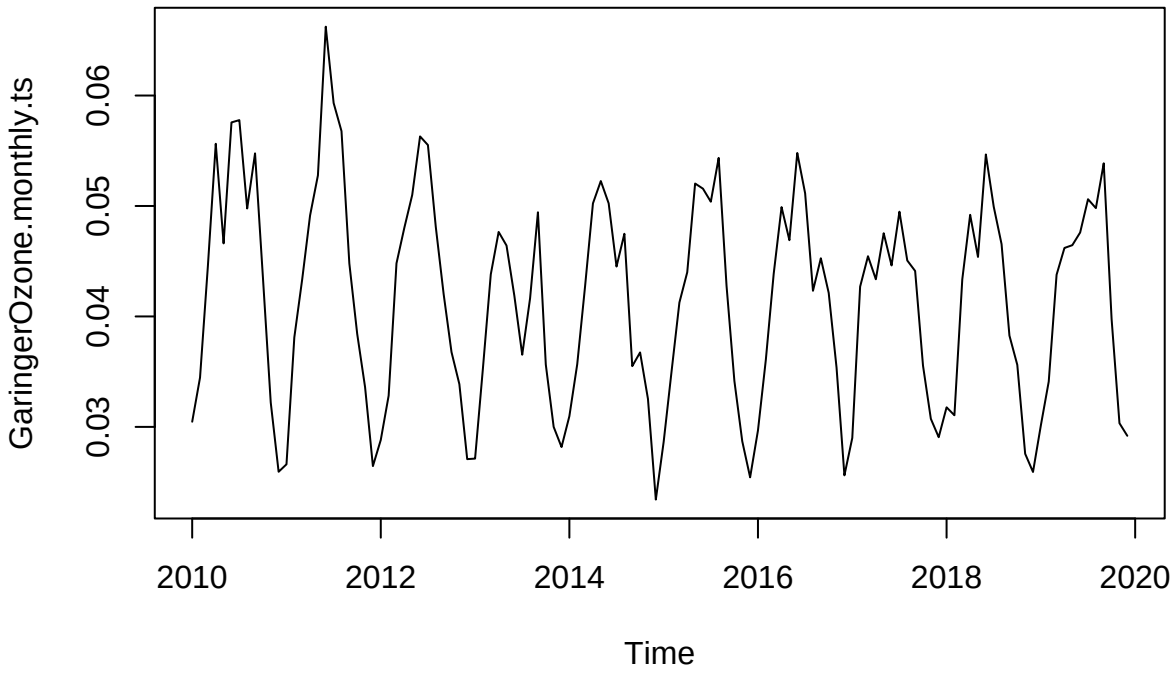
```
    mutate(Date = my(paste0(Month,"-",Year))) %>%
    group_by(Date) %>%
    summarize(Avg.Ozone.Concentrations = mean(OzoneConc.clean)) %>%
    select(Date,Avg.Ozone.Concentrations)

# smush month and date together so new column reads Jan 2010
# create new date column each month-year column set as first day of the month
```

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(clean_GaringerOzone$OzoneConc.clean, start = c(2010,01,01), end = c(2019,1
plot(GaringerOzone.daily.ts)
```

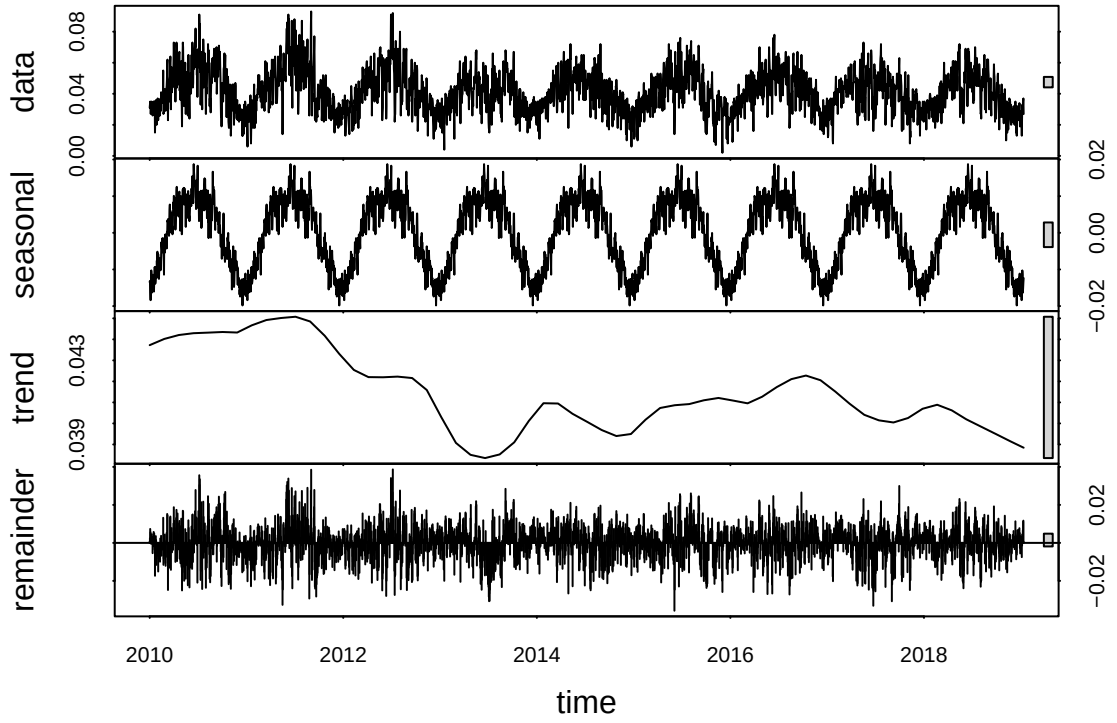



```
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Avg.Ozone.Concentrations, start = c(2010,01,01), e
plot(GaringerOzone.monthly.ts)
```

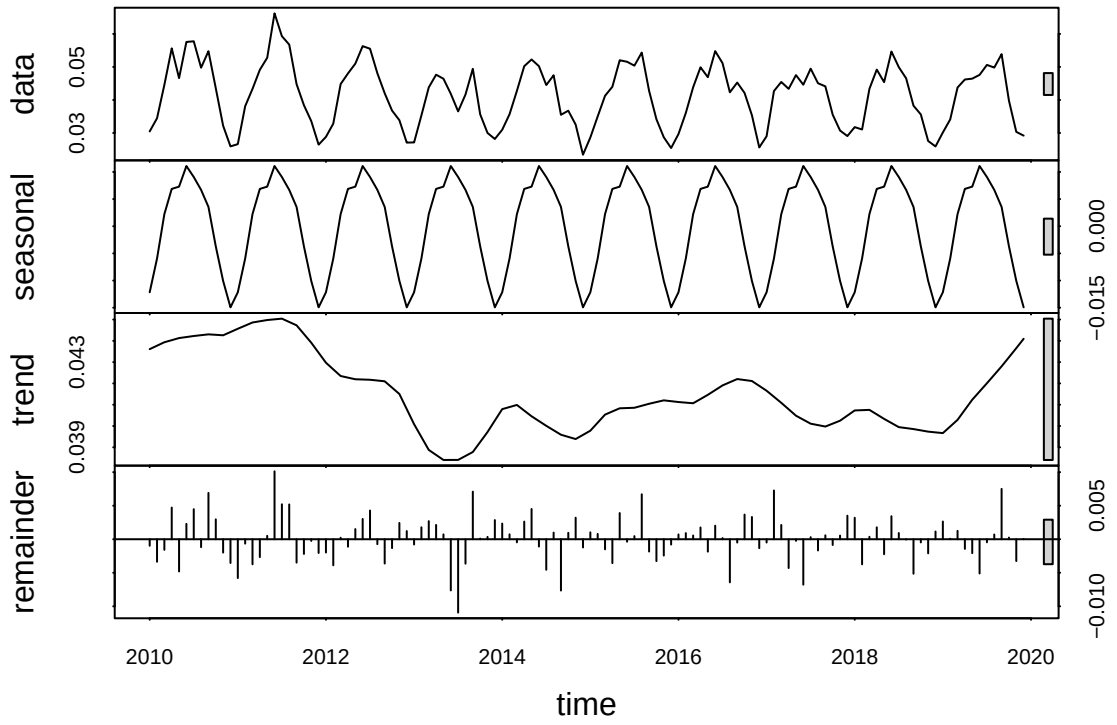


11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

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




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




```
# data
# seasonal
# trend
# remainder
```

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)

monthly_ozone_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
monthly_ozone_trend1
```

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

```
summary(monthly_ozone_trend1)
```

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

```
monthly_ozone_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
monthly_ozone_trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

```
summary(monthly_ozone_trend2)
```

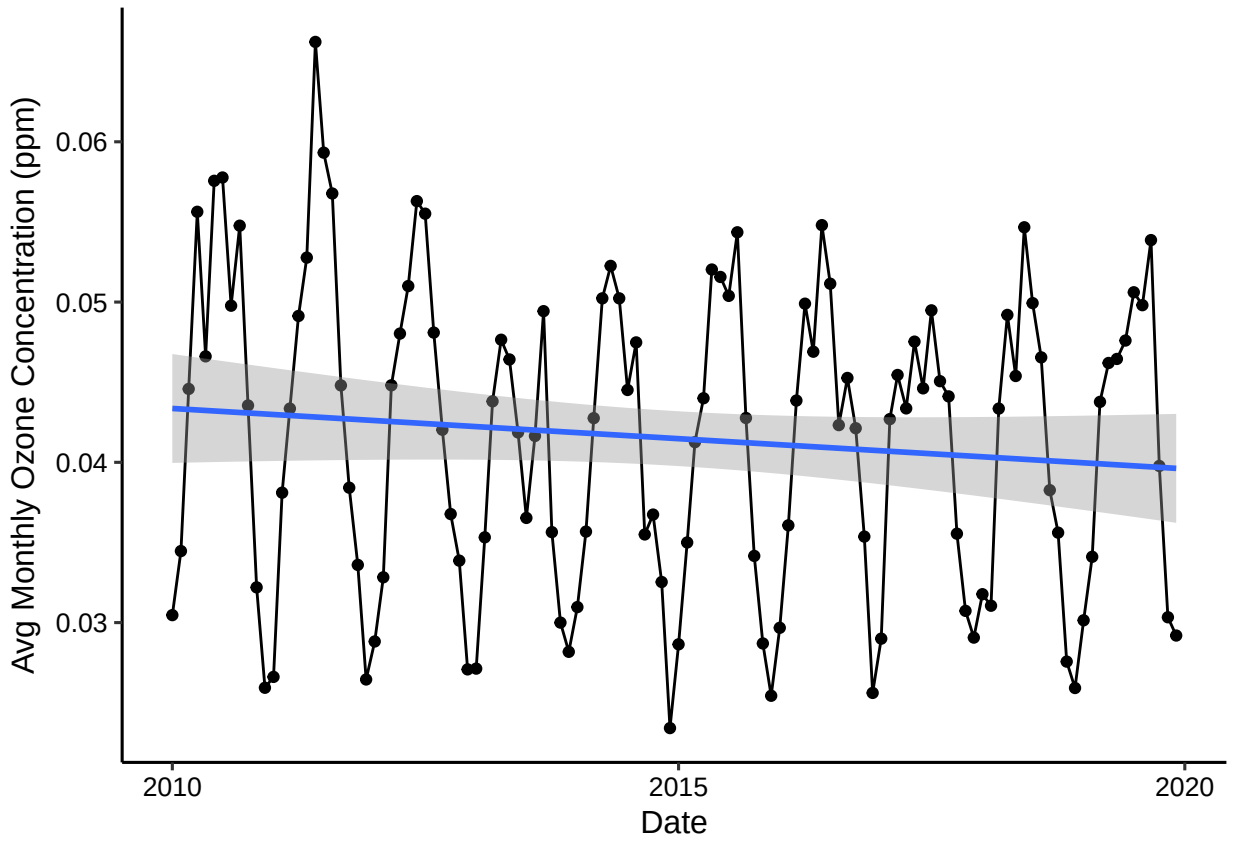
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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: The seasonal Mann-Kendall is most appropriate, because it considers seasonality in a dataset. Since we are analyzing monthly data, seasonality in ozone concentrations can vary based on the month i.e. the season.

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
MonthlyOzonePLOT <- ggplot(GaringerOzone.monthly,
  aes(x = Date, y = Avg.Ozone.Concentrations))+
  geom_point()+
  geom_line()+
  geom_smooth(method=lm)+
  ylab(expression(paste("Avg Monthly Ozone Concentration ", "(ppm)")))
print(MonthlyOzonePLOT)
```

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



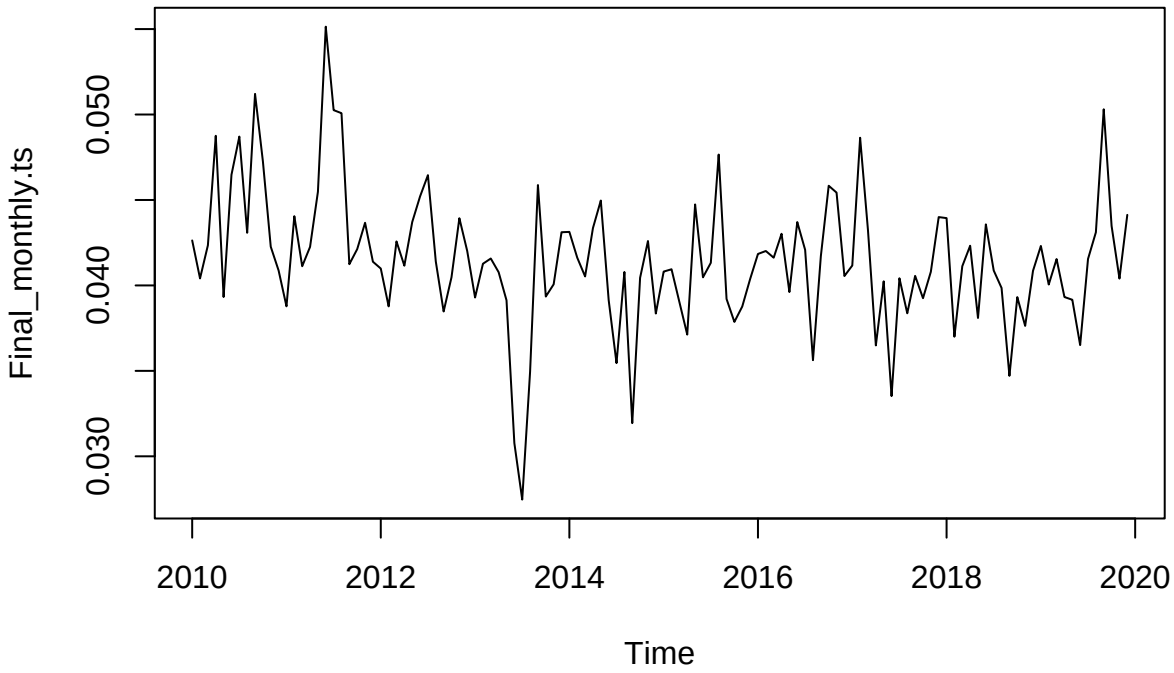
14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: Over the 2010's, ozone concentrations have changed at this station. Ozone concentrations have gradually decreased over time. Generally,the results show that ozone concentrations is higher in the middle of the year, and decrease in beginning and end of each year. Statistically, the p-values are below 0.05, meaning it is signficant and we reject the null hypothesis.

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

Final_monthly <- Extracted_monthly %>%
  mutate(Nonseasonal = trend+remainder) %>%
  select(Nonseasonal)

Final_monthly$Date <- GaringerOzone.monthly$Date

Final_monthly.ts <- ts(Final_monthly$Nonseasonal,start = c(2010,01,01), end = c(2019,12,01), frequency =
plot(Final_monthly.ts)
```




```
# SEASONAL TREND AND REMAINDER COMBINE TO MAKE DATA GRAPH
# skip over first column then 1 2 and 3

#16

#use normal mann kendall
FINAL.MONTHLY.TREND <- Kendall::MannKendall(Final_monthly.ts)
FINAL.MONTHLY.TREND
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
summary(FINAL.MONTHLY.TREND)
```

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

```
# tau indicates your trend (negative means trending down)
# BELOW 0.05 P VALUE: significant
# to compare: the larger your tau the greater your trend
# look at tau and pvalue. lower tau is less significant, higher tau is stronger
```

Answer: The Mann Kendall test on the nonseasonal Ozone monthly series has a greater negative trend due to a larger tau value of -0.165 vs the tau value of -0.143 for the Seasonal Mann Kendall test on the complete monthly series. The Mann Kendall test on the nonseasonal Ozone monthly series is more significant due to its p-value that is below 0.05 (p-value = 0.007) vs the p-value of 0.046 of the Seasonal Mann Kendall test on the complete monthly series. Both tests are significant, but the Mann Kendall test is just more significant.