Assignment 7: Time Series Analysis

Logan Loadholtz

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/loganloadholtz/Documents/DATA/Environmental_Data_Analytics_2021/Assignments"
```

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

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

#2
EPAair_O3_2010 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFa
```

```
EPAair_O3_2011 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFa
EPAair_O3_2012 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFa
EPAair_O3_2013 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFa
EPAair_O3_2014 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFa
EPAair_O3_2015 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFa
EPAair_O3_2016 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFa
EPAair_O3_2017 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFa
EPAair_O3_2018 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFa
EPAair_O3_2019 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFa

Garinger_Ozone <- rbind(EPAair_O3_2010, EPAair_O3_2011, EPAair_O3_2012, EPAair_O3_2013, EPAair_O3_2014,
```

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

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

```
## [1] "Date"
```

```
# 4
Garinger_Ozone_select <- select(Garinger_Ozone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_V



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



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

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

Visualize

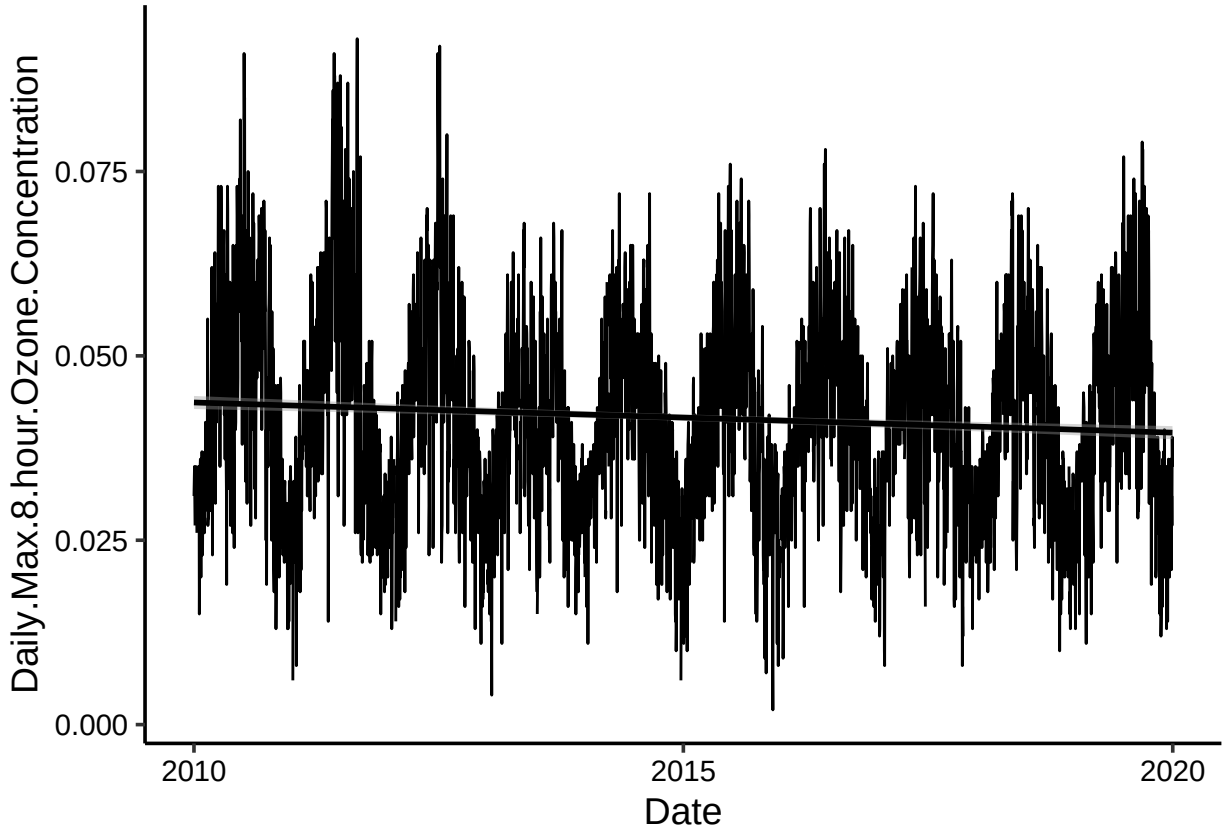
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
```

```
Ozone_lineplot <- ggplot(GaringerOzone, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method = lm, color="black")

print(Ozone_lineplot)
```

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

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



Answer: Over time, there is a general negative trend, showing that from 2010-2019, ozone concentration has been slighlty decreasing.

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
```

```
#There are 63 NAs here

GaringerOzone_clean <-
```

```
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )
#interpolation for missing datas

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

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

```
#there are no more NA values
```

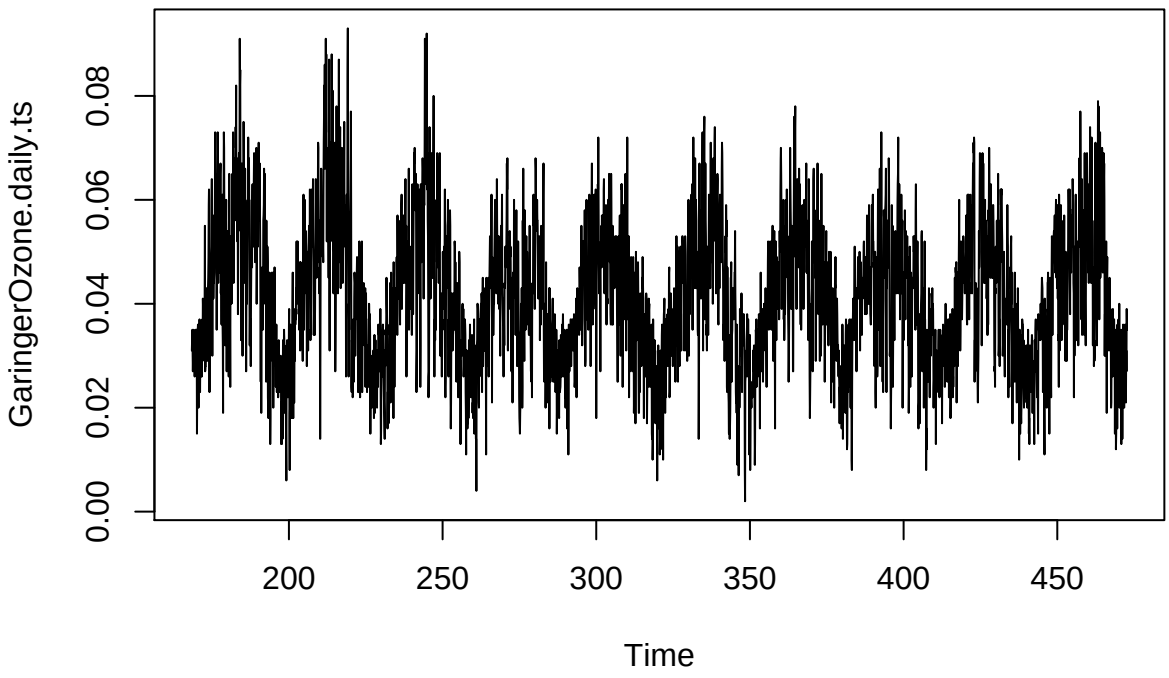
Answer: During the day, ozone concentration is constantly changing. Using linear interpolation is best here because it allows us to estimate an approximate point where the missing data point is. With a piecewise constant, the missing data would be assigned the same value as the nearest data value. Linear interpolation is the most simple method of interpolation.

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)

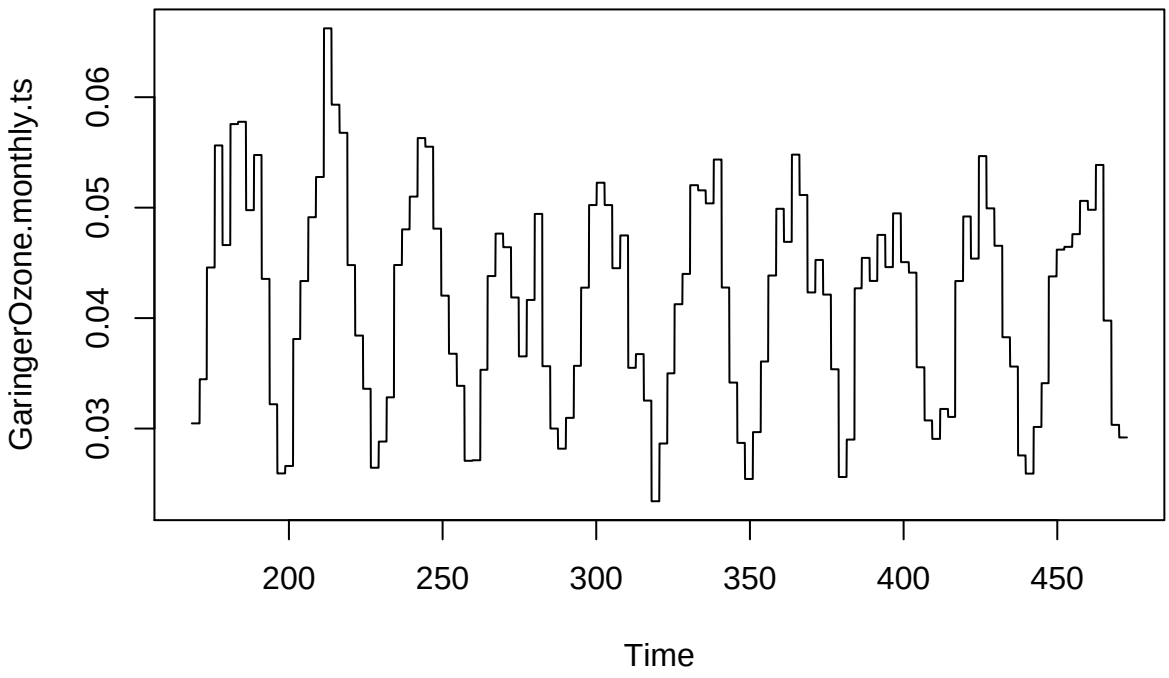
```
#9
GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(month = month(Date)) %>%
  mutate(year=year(Date)) %>%
 group_by(Month = format(as.Date(Date), '%m-%Y'))%>%
  mutate(MonthlyAvg= mean(Daily.Max.8.hour.Ozone.Concentration))%>%
  mutate(date=dmy(paste("01", month, year)))
```

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(GaringerOzone_clean$Date))
f_year <- year(first(GaringerOzone_clean$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration,
                   start=c(f_month, f_year),
                   frequency=12)

plot(GaringerOzone.daily.ts)
```

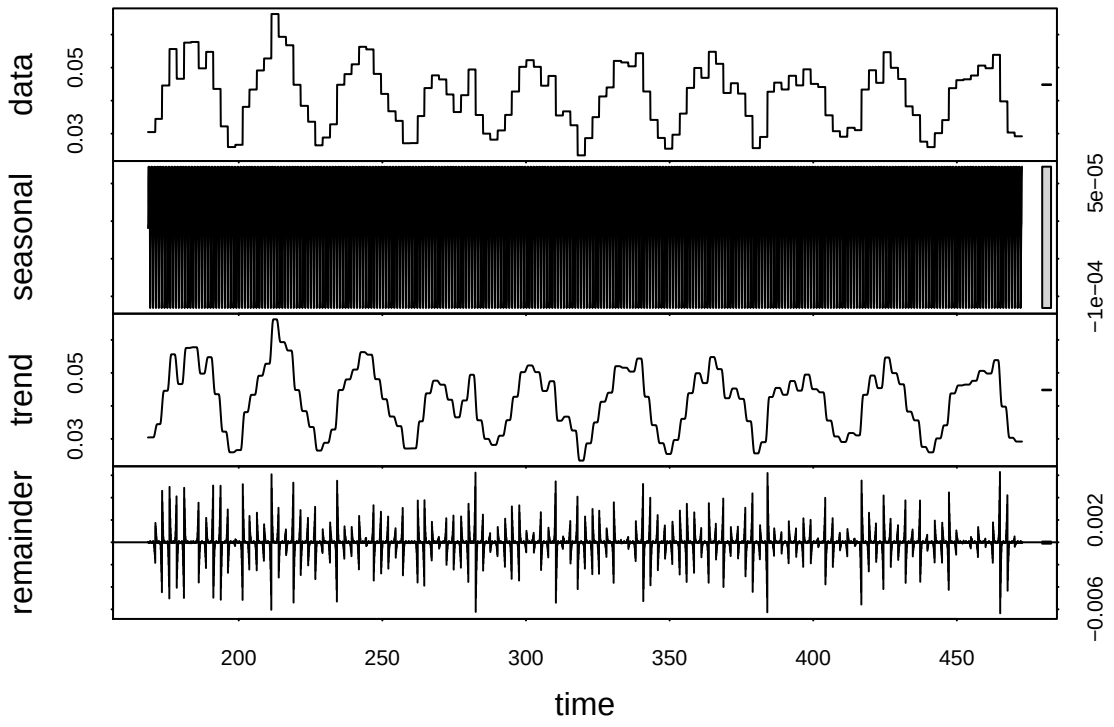



```
f_month.2 <- month(first(GaringerOzone.monthly$Date))
f_year.2 <- year(first(GaringerOzone.monthly$Date))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MonthlyAvg,
                 start=c(f_month.2, f_year.2),
                frequency=12)

plot(GaringerOzone.monthly.ts)
```

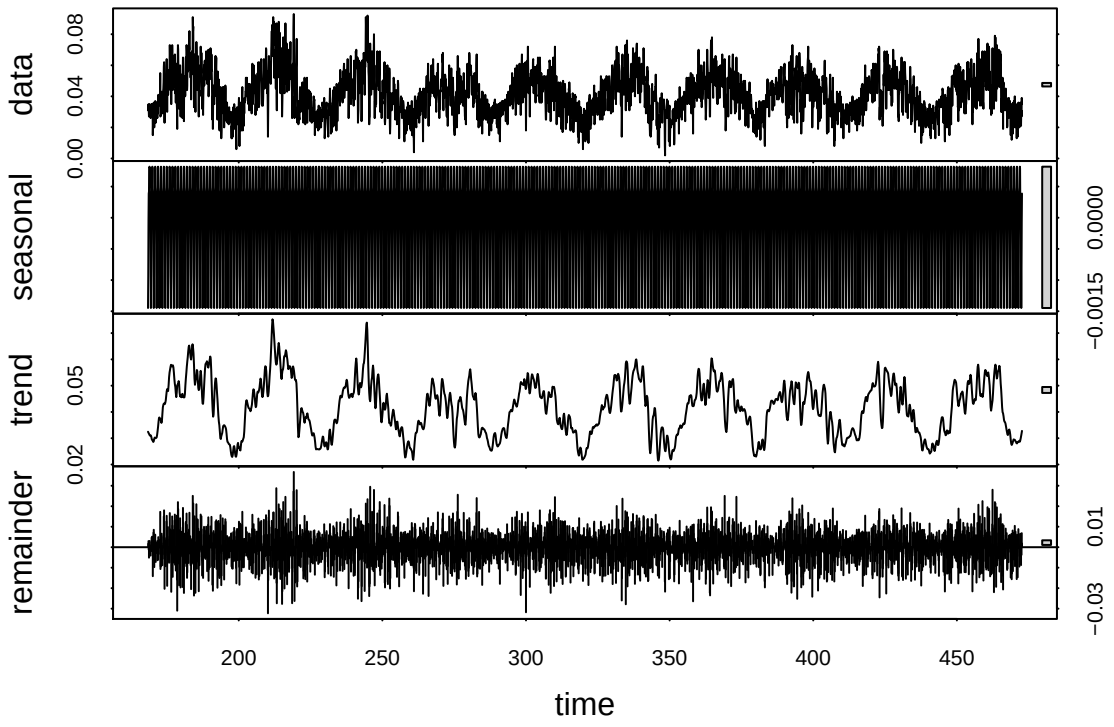


11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11

Garinger_Ozone.monthly_decomp <- stl(GaringerOzone.monthly.ts,
                                     s.window = "periodic")

plot(Garinger_Ozone.monthly_decomp)
```




```
GaringerOzone.daily_decomp <- stl(GaringerOzone.daily.ts,
                                  s.window = "periodic")
plot(GaringerOzone.daily_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

GaringerOzone.monthly_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

GaringerOzone.monthly_trend1
```

```
## tau = -0.0599, 2-sided pvalue =7.2237e-08
```

```
summary(GaringerOzone.monthly_trend1)
```

```
## Score =  -33095 , Var(Score) = 37763299
## denominator =  552314.2
## tau = -0.0599, 2-sided pvalue =7.2237e-08
```

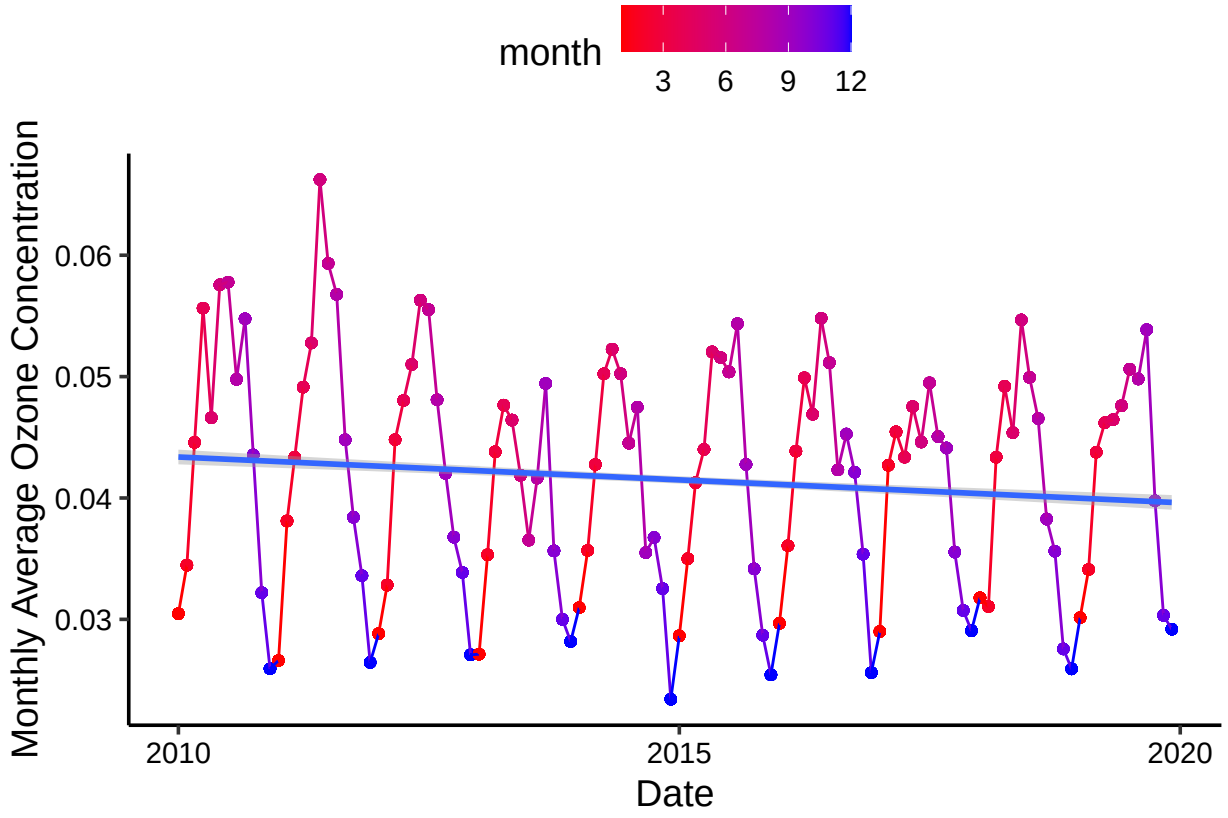
Answer: The seasonal Mann Kendall tests allows for a test analyze trends in seasonal data. In this case, our seasonal trend is "month". In our case here, a trend in the overall series won't be analyzed, but instead if there is a trend from month to month.

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

Plot_Monthly <- ggplot(GaringerOzone.monthly,
                       aes(x=date, y=MonthlyAvg, color=month))+
  scale_color_gradient(low= "red", high="blue")+
```

```
  geom_point()+
  xlab('Date')+
  ylab('Monthly Average Ozone Concentration')+
  geom_line()+
  geom_smooth(method = lm)


print(Plot_Monthly)
```

```
## `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.

    Research question: Have ozone concentrations changed over the 2010s at this station? Answer: When looking at the mean ozone concentration over time, it helps to use colors to differentiate the months. We can see in this plot that earlier months of the year (such as January and February) and later months of the year (October, November, December) have lower means. The highest means are found in the summer months, where we can see there is a peak. Overall, when using geom_smooth, it looks like there has been a slight decrease in ozone from 2010 to 2019.

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.monthly.ts_Components <- Garinger_Ozone.monthly_decomp$time.series[,2:3]

#16
nonseasonalmonthlyozone <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts_Components)


nonseasonalmonthlyozone
```

```
## tau = -0.519, 2-sided pvalue =< 2.22e-16
```

```r
summary(nonseasonalmonthlyozone)
```

```
## Score =  -1144465 , Var(Score) = 301061162
## denominator =  2207044
## tau = -0.519, 2-sided pvalue =< 2.22e-16
```

Answer: The Test statistic in the complete series is -0.0599, and the test statistic on the non-seasonal monthly series is -0.519. The pvalue of the complete series is 7.22e-08 and the pvalue of the non-seasonal monthly is 2.22e-16. Because the p value is less than 0.05 in both series, we can reject the null and conclude that there is a trend present in both series.