Assignment 7: Time Series Analysis

Halina Malinowski

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/Dell Laptop/Documents/GitHub/EDA/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)
```

```
## Warning: package 'trend' was built under R version 4.1.3
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.1.3

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

```
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.1.3
```

```
mytheme <- theme_grey(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.

```r
#2
EPAair_2010 <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Ga

EPAair_2011<- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Gar

EPAair_2012<- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Gar

EPAair_2013 <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Ga

EPAair_2014 <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Ga

EPAair_2015 <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Ga

EPAair_2016 <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Ga

EPAair_2017 <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Ga

EPAair_2018 <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Ga

EPAair_2019 <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Raw/Ozone_TimeSeries/EPAair_O3_Ga

GaringerOzone <- rbind.data.frame(EPAair_2010, EPAair_2011, EPAair_2012, EPAair_2013, EPAair_2014, EPAa
```

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.

```r
# 3
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
class(GaringerOzone$Date)
```

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

```r
# 4
GaringerOzone_subset <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
Days <- as.data.frame(seq(as.Date('2010-01-01'), as.Date('2019-12-31'), by = "day"))
colnames(Days)
```

```
## [1] "seq(as.Date(\"2010-01-01\"), as.Date(\"2019-12-31\"), by = \"day\")"
```

```
names(Days)[names(Days)== "seq(as.Date(\"2010-01-01\"), as.Date(\"2019-12-31\"), by = \"day\")"] <- "Da

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

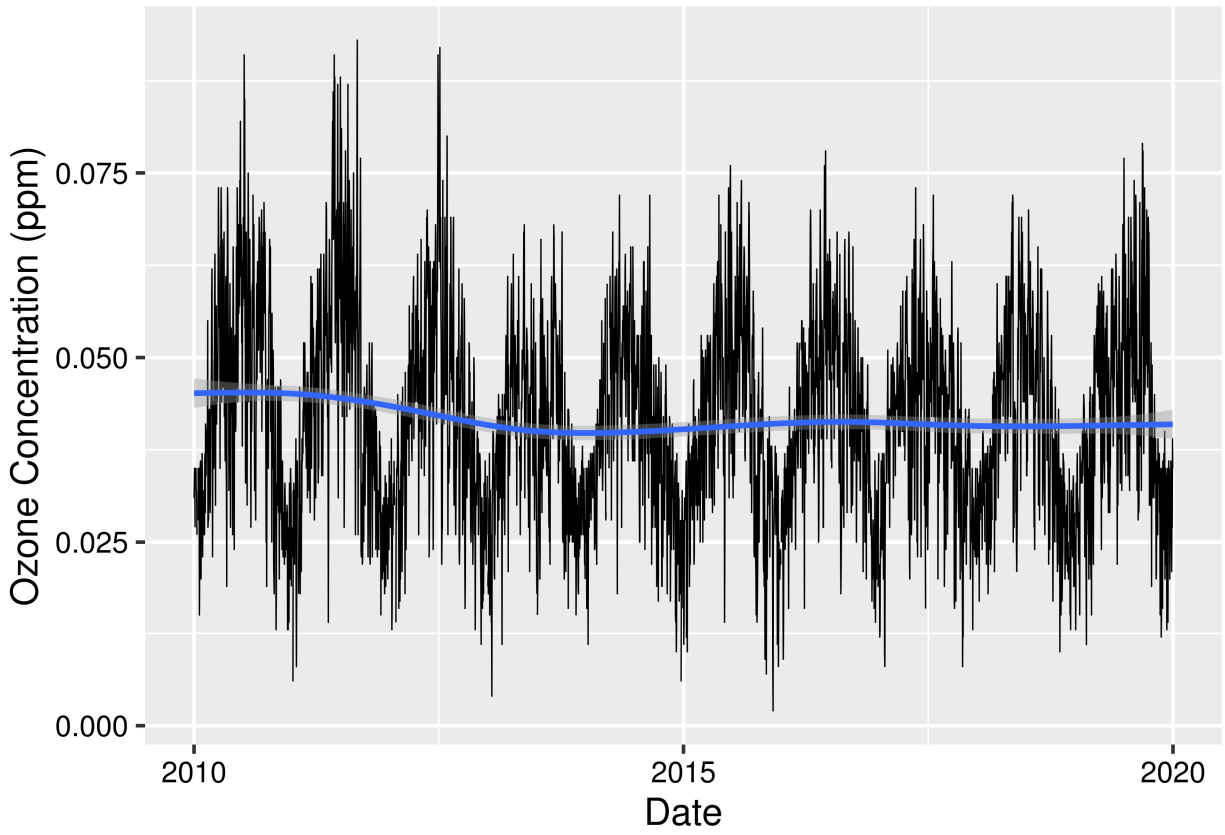
Visualize

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
ggplot(GaringerOzone)+
  geom_line(aes(y = Daily.Max.8.hour.Ozone.Concentration, x = Date), size = 0.25)+
  ylab("Ozone Concentration (ppm)")+
  geom_smooth(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

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



Answer: By examining the plot it looks like there is a seasonal trend for the data as is shown by the oscillations. There are 10 peaks and 10 troughs over 10 years which likely correlates with seasonal variations. The overall trend appreas fairly flat. However, the trend appears to decrease a little bit or be negatively correlated within the first few years of data collections.

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

summary(GaringerOzone$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
```

Answer: Our data appear to follow linear trends oscillating between seasons therefore a linear interpolation was the best option because it interpolates values by looking at those before and after the gap and creating a straight line between the two then determining the missing value. A piecewise constant interpolation simply takes on the value of the point closest to the gap which would not accurately represent the behavior in our data which varies over time, moving linearly in the positive direction and then in the negative direction to create oscillations. A spline interpolation is also not as good of a fit because it uses a quadratic function to determine the value(s) in the gap and a linear relationship more closely matched our dataset.

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 %>%
  mutate(Month = month(Date))%>%
  mutate(Year = year(Date))%>%
  #mutate(Day = day(Date))%>%
  group_by(Month, Year)%>%
  summarise(Mean_Concentration = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

```
GaringerOzone.monthly$Day <- '1'
GaringerOzone.monthly$Date <- as.Date(with(GaringerOzone.monthly, paste(Year, Month, Day, sep = "-")),
```

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.

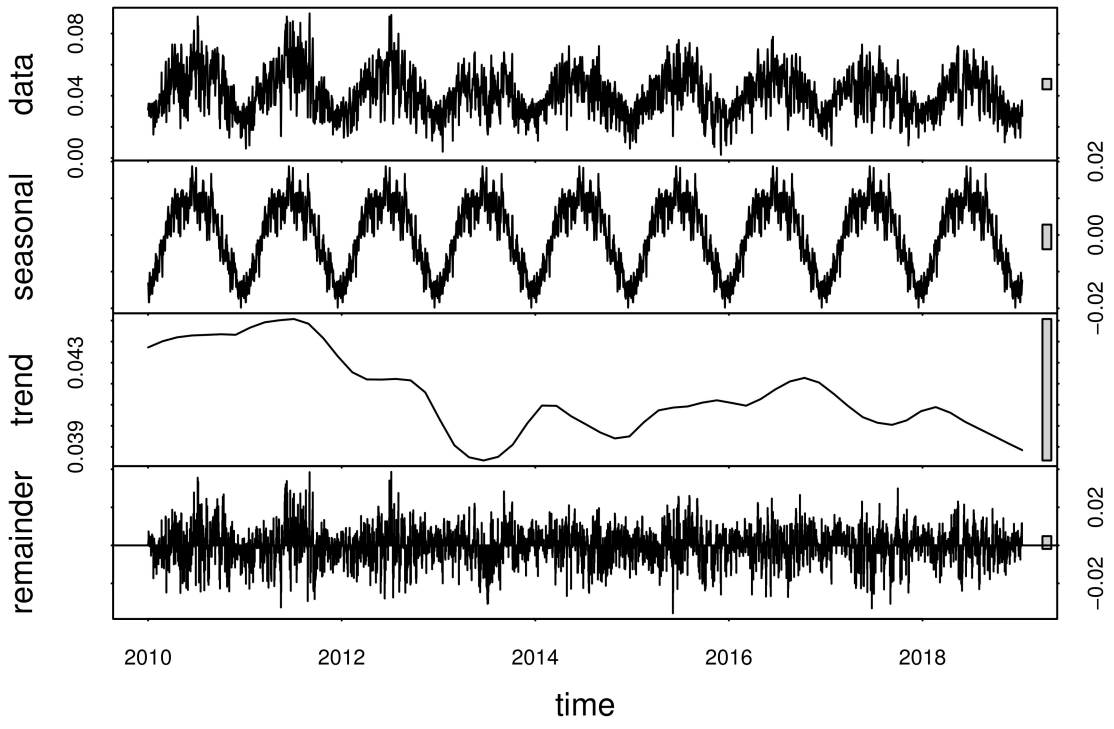
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration, start = c(2010,1,1),
                             end = c(2019,12,31), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Concentration, start = c(2010,1,1),
                               end = c(2019,12,1), frequency = 12)
```

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

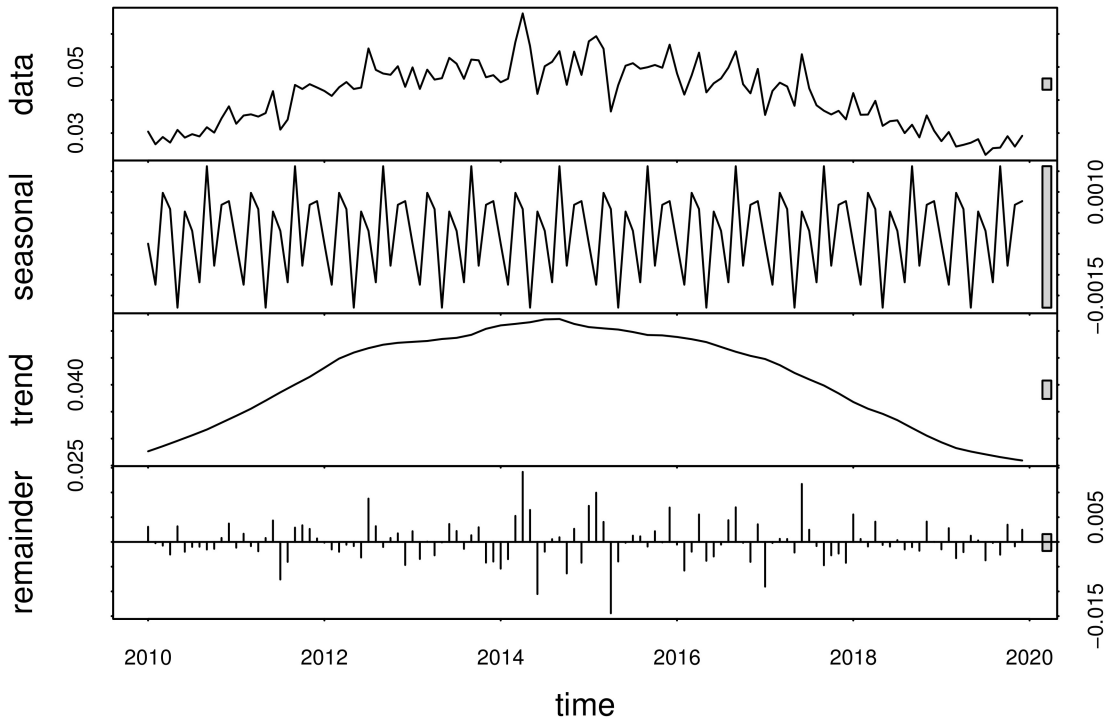
```
#11
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

plot(GaringerOzone.daily.decomposed)
```




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

plot(GaringerOzone.monthly.decompose)
```



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 = -0.1, 2-sided pvalue =0.16323
```

```
summary(GaringerOzone.monthly.trend1)
```

```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

```
GaringerOzone.monthly.trend2 <- trend::smk.test(GaringerOzone.monthly.ts)

GaringerOzone.monthly.trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
```

```
## data:  GaringerOzone.monthly.ts
## z = -1.3685, p-value = 0.1712
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -54 1500
```

```
summary(GaringerOzone.monthly.trend2)
```

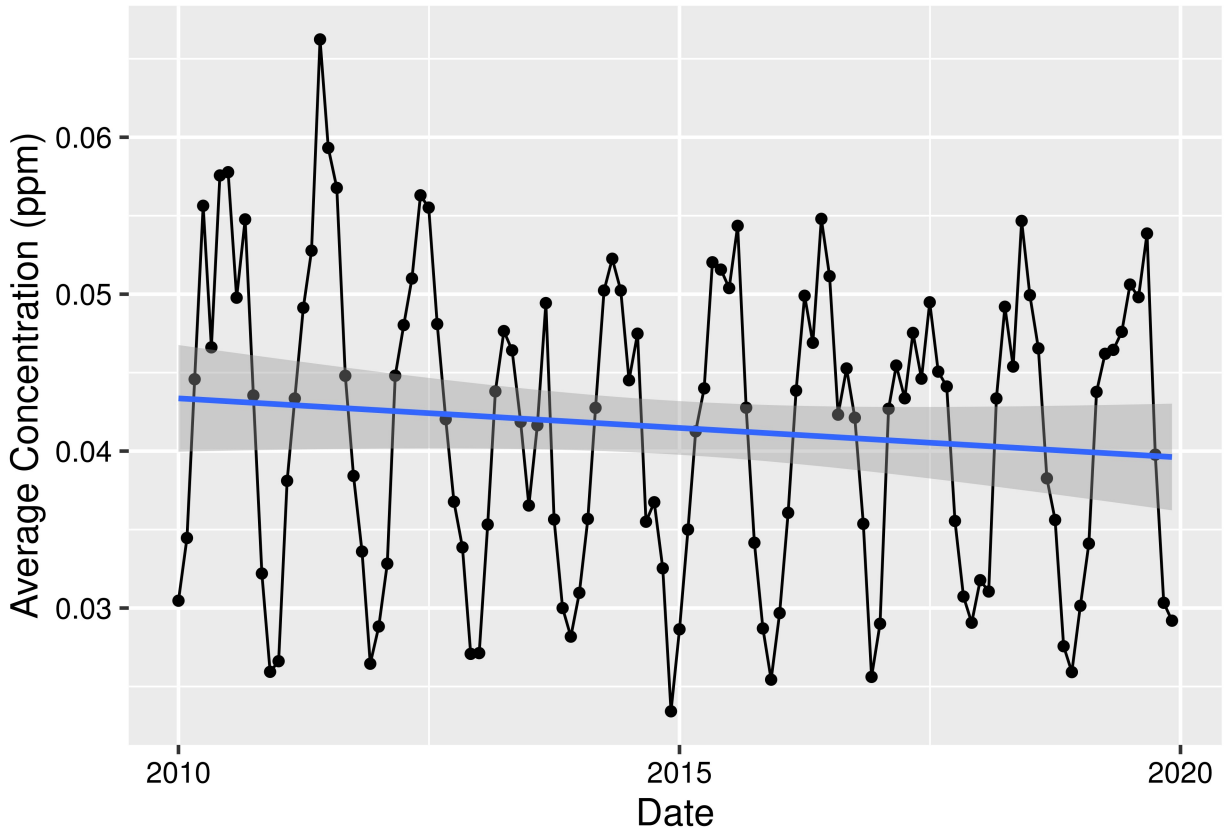
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                      S varS    tau      z Pr(>|z|)
## Season 1:   S = 0    1  125  0.022  0.000  1.00000
## Season 2:   S = 0    5  125  0.111  0.358  0.72051
## Season 3:   S = 0   -3  125 -0.067 -0.179  0.85803
## Season 4:   S = 0    1  125  0.022  0.000  1.00000
## Season 5:   S = 0   -9  125 -0.200 -0.716  0.47427
## Season 6:   S = 0    1  125  0.022  0.000  1.00000
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -3  125 -0.067 -0.179  0.85803
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -11  125 -0.244 -0.894  0.37109
## Season 11:   S = 0 -15  125 -0.333 -1.252  0.21050
## Season 12:   S = 0  -5  125 -0.111 -0.358  0.72051
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The seasonal Mann-Kendall analysis is most appropriate because the data exhibits a seasonal trend and this analysis is the only one which we learned that considers seasonal trends. This way we can examine any changes in mean concentration over time while also considering a seasonal trend. All other analysis including linear regression, Mann-Kendall, Spearman Rho, and Augmented Dickey Fuller do not consider seasonal 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
GaringerOzone.monthly.plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Concentration)) +
  geom_point() +
  geom_line() +
  ylab("Average Concentration (ppm)") +
  geom_smooth( method = lm )
print(GaringerOzone.monthly.plot)
```

```
## `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: From our analysis it appears that there is no monthly seasonal trend in mean ozone concentration at Garinger High School in North Carolina as the p-value > 0.05. ($p = 0.1712$). From the graph above it appears that overall there is a slightly negative trend in the data from 2010 - 2020. After examining the second Seasonal Mann-Kendall test it is seen that there is no significant variation for any season and p-value > 0.05 for each season with very low S values.

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

GaringerOzone.monthly.components <- mutate(GaringerOzone.monthly.components,
        Observed = GaringerOzone.monthly$Mean_Concentration,
        Date = GaringerOzone.monthly$Date)

GaringerOzone.monthly.components.ts <-
  ts(GaringerOzone.monthly.components$Observed,
```

```
  start = c(2010,1), frequency = 12)

#16
GaringerOzone.monthly.trend3 <- Kendall::MannKendall(GaringerOzone.monthly.components.ts)

GaringerOzone.monthly.trend3
```

```
## tau = -0.105, 2-sided pvalue =0.088483
```

```
summary(GaringerOzone.monthly.trend3)
```

```
## Score =  -752 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.105, 2-sided pvalue =0.088483
```

Answer: The results from the Mann-Kendall analysis fit the data better than the results from the Seasonal Mann-Kendall. Although neither analysis was significant at a p-value < 0.05. The first analysis Seasonal Mann-Kendall gave a p-value of 0.1712 and the second analysis Mann-Kendall gave a p-value of 0.088. Although this analysis fits better it is still not significant and the null hypothesis is accepted that there is no change over time. This can easily be observed in the previous graphs where the overall trend line did not depcit a steep slope and was nearly flat demonstrating little change in mean ozone concentration in ppm over the 10 year period of the study.