Assignment 7: Time Series Analysis

Meghan Seyler

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] "Z:/ENV872/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)

theme7 <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(theme7)
```

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
Garinger2019_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2018_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2017_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2016_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2015_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2014_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2013_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2012_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2011_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2010_raw <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                         stringsAsFactors = TRUE)

Garinger2010.2019_raw<-rbind(Garinger2010_raw,Garinger2011_raw,
                             Garinger2012_raw,Garinger2013_raw,
                             Garinger2018_raw,Garinger2019_raw)
```

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
Garinger2010.2019_raw$Date <- mdy(Garinger2010.2019_raw$Date)

#confirm class as Date
class(Garinger2010.2019_raw$Date)
```

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

```
# 4
#wrangle the dataset
Garinger2010.2019_Processed<-Garinger2010.2019_raw%>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration,DAILY_AQI_VALUE)

write.csv(Garinger2010.2019_Processed, row.names = FALSE,
          file = "../Data/Processed/Garinger2010.2019_Processed.csv")

# 5
#generate new data frame with specific dates
Days<-data.frame(seq(as.Date("2010-01-01"),as.Date("2019-12-31"),"day"))

#rename column
names(Days)[1] <- "Date"

#take a look
head(Days)
```

```
##         Date
## 1 2010-01-01
## 2 2010-01-02
## 3 2010-01-03
## 4 2010-01-04
## 5 2010-01-05
## 6 2010-01-06
```

```
# 6
#join the two data frames
GaringerOzone<-merge(x=Days,y=Garinger2010.2019_Processed,by="Date", all.x=TRUE)

#confirm data frames joined correctly
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
```

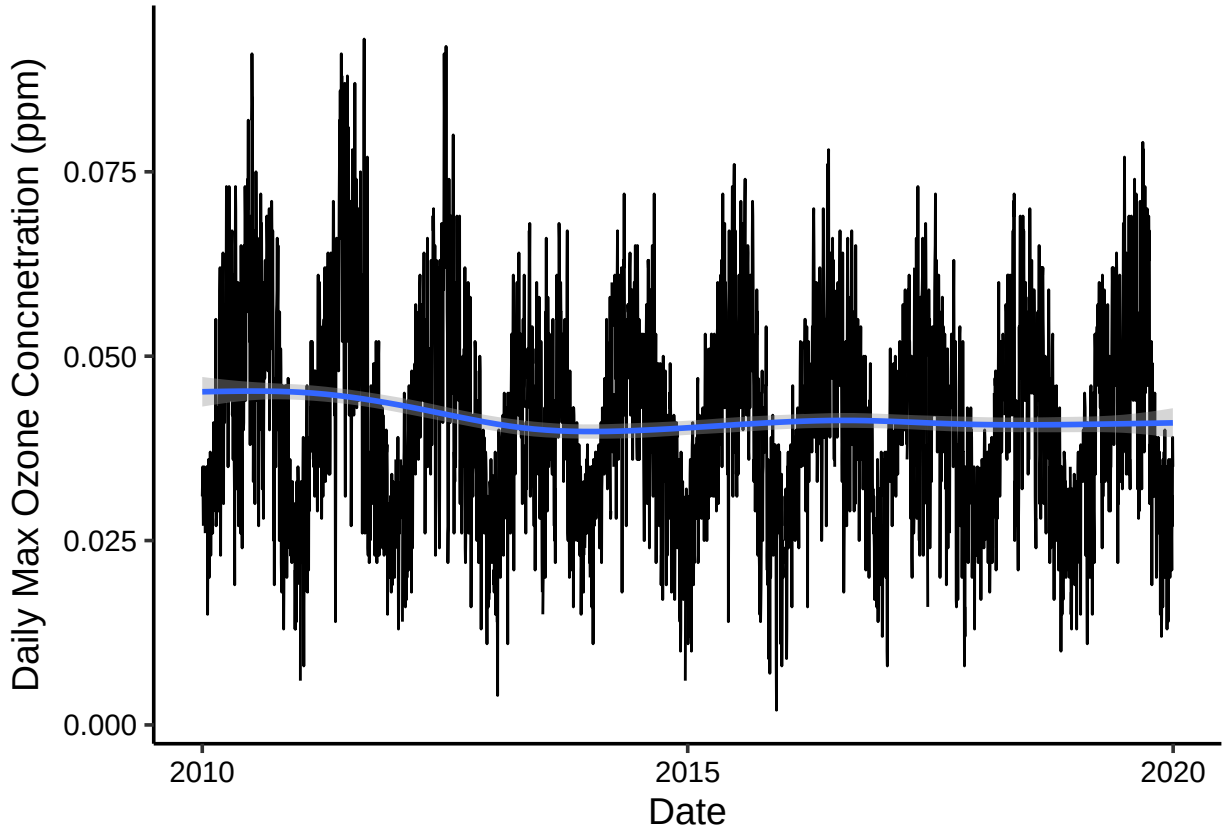
Visualize

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
#IS this right? or do they want a point plot with a smooth line
#how do I add every year to the y axis
GaringerOzone_ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,                    sta
                       frequency = 365)

ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth()+
  labs(x = "Date", y = expression("Daily Max Ozone Concnetration (ppm)"))
```

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

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



Answer: This plot suggests a cyclical pattern because the cycles appear to be occuring in periods longer than 1 year in length. Also, the trend appears to be downward indicating that ozone may be decreasing from 2010 to 2020.

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

#interpolate missing observations in Daily.Max.8.hour.Ozone.Concentration column
GaringerOzone_Clean<-
  GaringerOzone%>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

#check to make sure summary statistics stay the same and NAs are replaced
#in the ozone concentration column (Daily.Max.8.hour.Ozone.Concentration)
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
```

```
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
```

Answer: We used a linear interpolation because we're interested in connecting the dots before and after the missing data point. We did not use a piecewise constant because in this approach missing data are assumed to be equal to the measurement made nearest to that date. We did not use a spline interpolation becuase this approach is used on quadratic data and our data is not quadratic.

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),
         Year = year(Date))%>%
  mutate(Date = my(paste0(Month, "-", Year)))%>%
  dplyr::group_by(Date,Month,Year)%>%
  dplyr::summarise(MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration))%>%
  select(MeanOzone,Date)
```

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

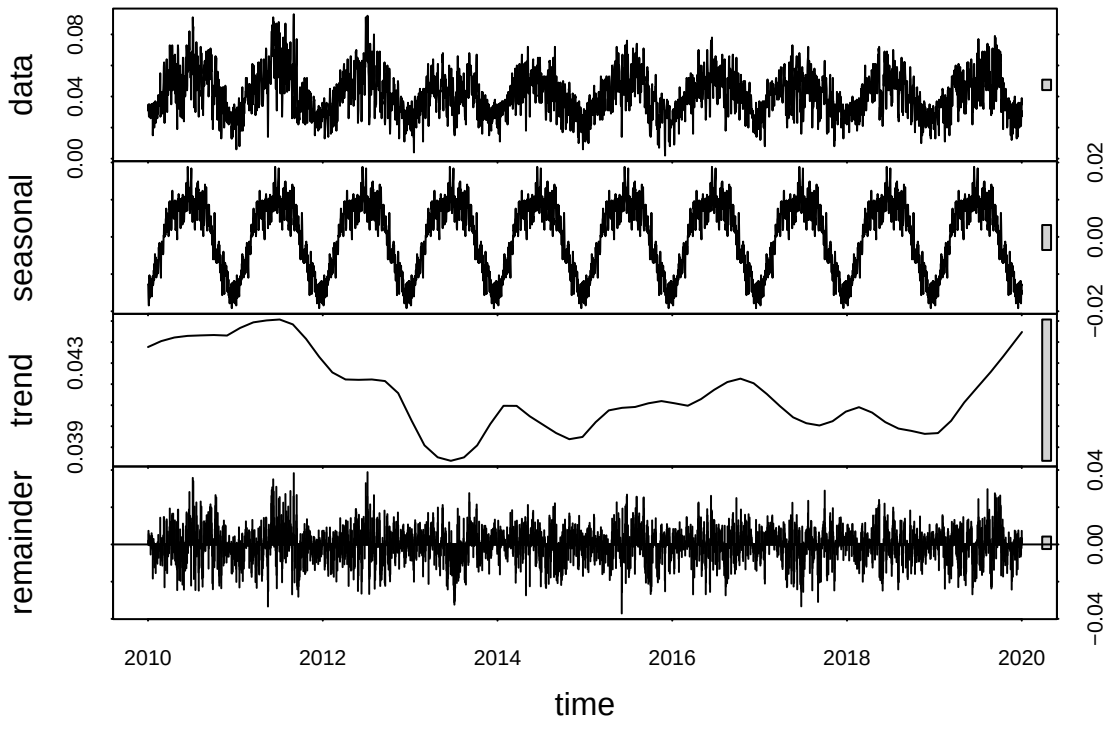
```
## Adding missing grouping variables: `Month`
```

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.

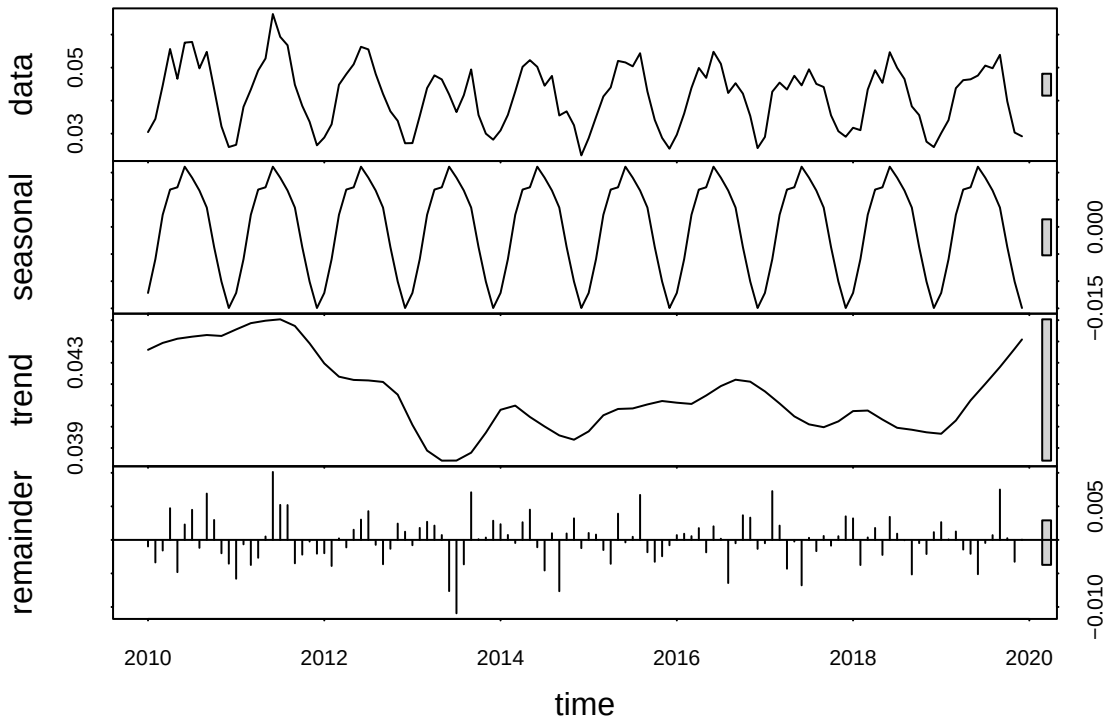
```r
#10

#######do we need to set an end? or will it just continue on for the rest of the data set?
GaringerOzone.daily.ts<-ts(GaringerOzone_Clean$Daily.Max.8.hour.Ozone.Concentration, start = c(2010,1),

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

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

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

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

plot(GaringerOzone_daily_Decomposed)
```




```r
plot(GaringerOzone_monthly_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

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

#Inspect Results
GaringerOzone.monthly_trend1
```

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

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

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

```
#pvalue is 0.046 which rounds to 0.05 so we accept the null and do NOT have a trend


GaringerOzone.monthly_trend2<-trend::smk.test(GaringerOzone.monthly.ts)
#Inspect results
GaringerOzone.monthly_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(GaringerOzone.monthly_trend2)
```

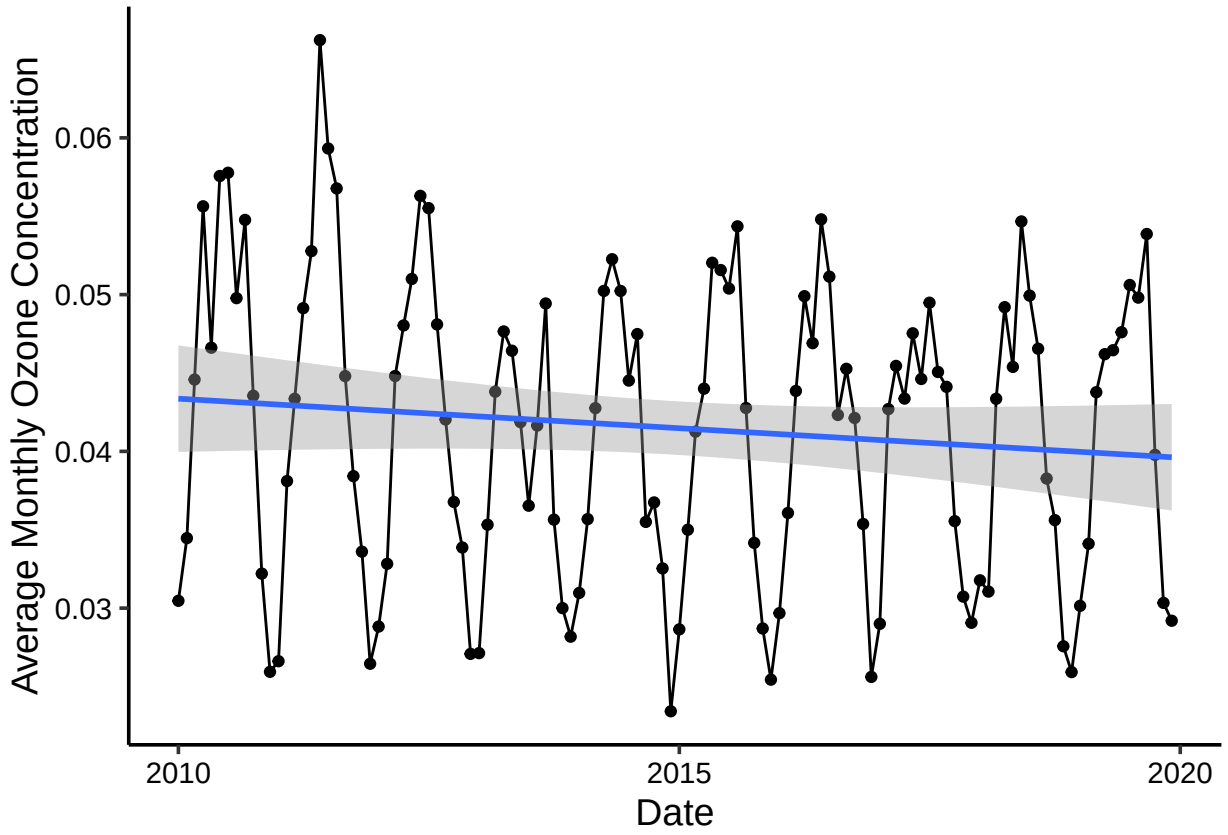
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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 our data set is seasonal and the other tests should not be used on seasonal data. Seasonality was confirmed by rning the stl test. An alternative method would be to remove the seasonality of the dataset and use one of the other tests (linear regression, Mann-Kendall, SpearmanRho).

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 = MeanOzone)) +
  geom_point() +
  geom_line() +
  ylab("Average Monthly Ozone Concentration") +
  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: The graph depicts that there might be an average decrease of monthly ozone over-time. This ambiguity is not surprising considering the p-value was just below 0.05 at 0.046. ############include output from statsitical test

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
# We can extract the components and turn them into data frames

GaringerOzone_monthly.ts_Components <- as.data.frame(GaringerOzone_monthly_Decomposed$time.series)

GaringerOzone_monthly.ts_Components <-
  GaringerOzone_monthly.ts_Components%>%
  mutate(Observed = GaringerOzone_monthly$MeanOzone, Date = GaringerOzone_monthly$Date)%>%
  mutate(NonSeasonal=Observed - seasonal)

#16

GaringerOzone.monthly_Components_trend1<-Kendall::MannKendall
(GaringerOzone_monthly.ts_Components$NonSeasonal)
```

```
##   [1] 0.04263190 0.04041003 0.04234881 0.04875492 0.03932081 0.04647348
##   [7] 0.04871023 0.04307797 0.05120815 0.04725211 0.04226527 0.04087082
##  [13] 0.03877706 0.04405289 0.04112300 0.04225492 0.04548211 0.05514015
##  [19] 0.05025862 0.05007797 0.04124148 0.04212308 0.04366527 0.04138695
##  [25] 0.04098674 0.03877333 0.04257462 0.04115492 0.04370791 0.04520681
##  [31] 0.04645216 0.04140056 0.03847481 0.04047792 0.04393193 0.04201598
##  [37] 0.03929319 0.04126717 0.04157462 0.04077159 0.03912727 0.03077348
##  [43] 0.02746829 0.03494894 0.04587481 0.03934888 0.04006527 0.04311275
##  [49] 0.04313190 0.04162432 0.04052623 0.04335492 0.04496598 0.03914015
##  [55] 0.03545216 0.04078765 0.03194148 0.04044566 0.04259860 0.03835469
##  [61] 0.04080932 0.04094575 0.03902623 0.03712159 0.04474017 0.04047348
##  [67] 0.04132313 0.04765862 0.03920815 0.03786501 0.03876527 0.04037082
##  [73] 0.04184158 0.04201471 0.04162300 0.04302159 0.03961114 0.04370681
##  [79] 0.04208120 0.03562636 0.04170815 0.04583275 0.04543193 0.04054824
##  [85] 0.04116416 0.04864217 0.04321978 0.03648826 0.04024017 0.03352348
##  [91] 0.04041991 0.03836830 0.04055815 0.03925211 0.04079860 0.04399985
##  [97] 0.04393835 0.03699932 0.04112300 0.04232159 0.03809501 0.04357348
## [103] 0.04087152 0.03985217 0.03470815 0.03931662 0.03763193 0.04085469
## [109] 0.04230932 0.04005289 0.04154236 0.03932159 0.03915952 0.03650681
## [115] 0.04154894 0.04311023 0.05030815 0.04347792 0.04039860 0.04412888
```

```
GaringerOzone.monthly_Components_trend1
```

```
## function (x)
## {
##     Kendall(1:length(x), x)
## }
## <bytecode: 0x000000002b969800>
## <environment: namespace:Kendall>
```

Answer: Once the trend is removed the p value decreases from 0.046 to 0.0075. Thus, when the trend is removed there is an average decrease of monthly ozone overtime.