Assignment 7: Time Series Analysis

Gaby Antonova

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/gabri/OneDrive/Desktop/MPP Coursework/Spring 2022/ENVIRO 872/Environmental_Data_Analyti
```

```
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(tseries)
```

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

```
library(trend)

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

EPAair_O3_GaringerNC2010_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv


EPAair_O3_GaringerNC2011_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv
View(EPAair_O3_GaringerNC2010_raw)

EPAair_O3_GaringerNC2012_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv
View(EPAair_O3_GaringerNC2010_raw)

EPAair_O3_GaringerNC2013_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv
View(EPAair_O3_GaringerNC2010_raw)
```

```
EPAair_O3_GaringerNC2014_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv
View(EPAair_O3_GaringerNC2010_raw)

EPAair_O3_GaringerNC2015_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv
View(EPAair_O3_GaringerNC2010_raw)

EPAair_O3_GaringerNC2016_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv
View(EPAair_O3_GaringerNC2010_raw)

EPAair_O3_GaringerNC2017_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv
View(EPAair_O3_GaringerNC2010_raw)

EPAair_O3_GaringerNC2018_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv
View(EPAair_O3_GaringerNC2010_raw)

EPAair_O3_GaringerNC2019_raw <- read_csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv
View(EPAair_O3_GaringerNC2010_raw)


library(dplyr)

EPAAir_Data <- rbind(EPAair_O3_GaringerNC2010_raw,
EPAair_O3_GaringerNC2011_raw, EPAair_O3_GaringerNC2012_raw, EPAair_O3_GaringerNC2013_raw, EPAair_O3_Gari
```

Wrangle

3. Set your date column as a date class.

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

4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration

```
‘‘‘r
names(EPAAir_Data)<-str_replace_all(names(EPAAir_Data),
c(" " = "." , "-" = "." ))

EPAAir_Data_Subset <- select(EPAAir_Data, Date, Daily.Max.8.hour.Ozone.Concentration, 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.

```
# 5
summary(EPAAir_Data$Daily.Max.8.hour.Ozone.Concentration)
```

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

```
YearDays <- as.data.frame(seq(as.Date("2010/01/01"),
  as.Date("2019/12/31"), "days"))
colnames(YearDays) <- c("Date")

summary(YearDays)
```

```
##       Date
##  Min.   :2010-01-01
##  1st Qu.:2012-07-01
##  Median :2014-12-31
##  Mean   :2014-12-31
##  3rd Qu.:2017-07-01
##  Max.   :2019-12-31
```

```
# 6

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

GaringerOzone <- left_join(YearDays, EPAAir_Data_Subset,
by = c("Date"))
```

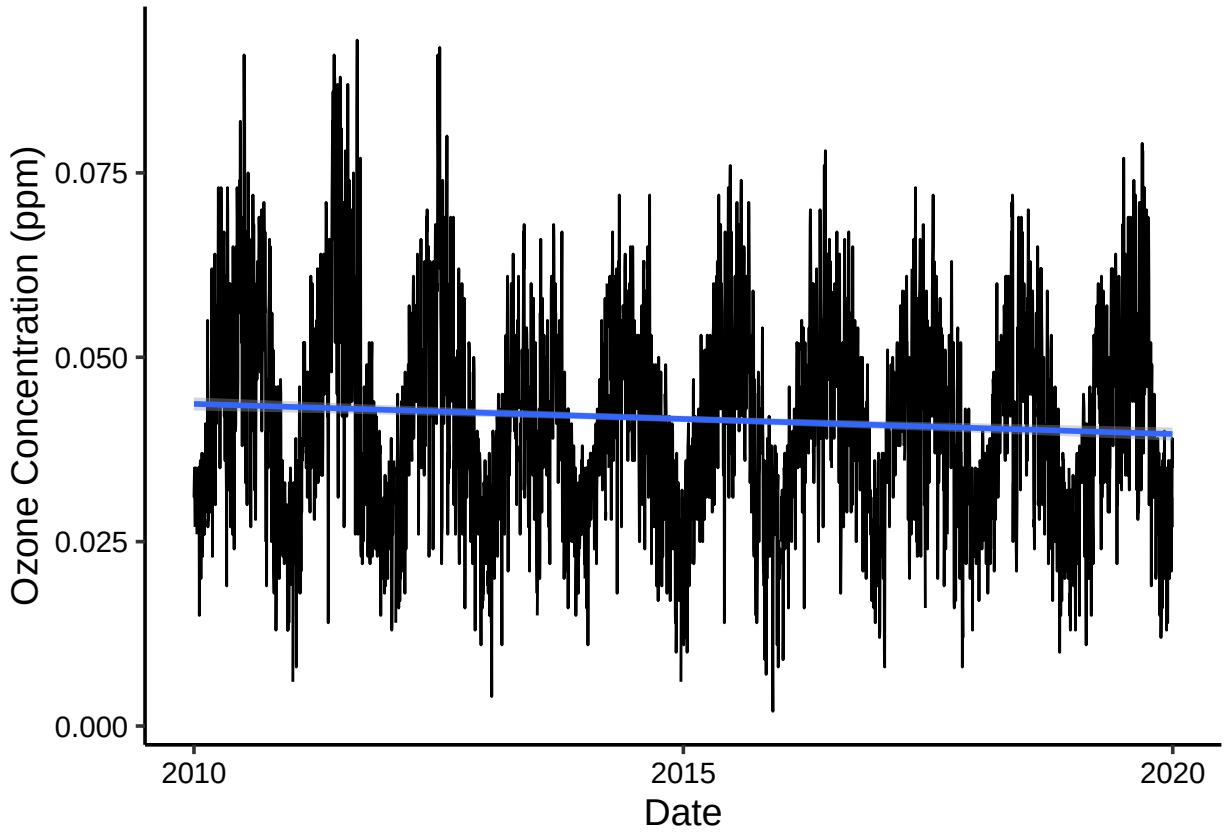
Visualize

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

GaringerOzonePlot <- ggplot(GaringerOzone, aes(Date, Daily.Max.8.hour.Ozone.Concentration)) +
geom_line()+
ylab("Ozone Concentration (ppm)")+
geom_smooth( method = lm )

print(GaringerOzonePlot)
```

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

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



Answer: No, the plot does not suggest there is a trend or change in ozone overtime, however, there is a lot of variation in the data and it is unclear if it is seasonal, random, etc. so it is hard to tell, based on this plot.

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
```

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

summary(GaringerOzoneClean$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 the linear interpolation because the intervals of missing data are relatively short. We didn't use piecewise constant or spline interpolation because we can see that there is variation because it makes more sense to take the average of adjacent observations rather than take the value of the nearest points since there is significant seasonality and the data is not constant. The data is also slowly varying - there aren't significant jumps so the spline interpolation would not be appropriate.

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

library(lubridate)
GaringerOzone.monthly <- GaringerOzoneClean %>%
mutate(Month =month(Date)) %>%
mutate(Year =year(Date)) %>%
mutate (Date = my(paste0(Month, "-", Year))) %>%
dplyr::group_by(Date, Month, Year) %>%
dplyr::summarise(mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
select(mean_Ozone, Date)
```

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

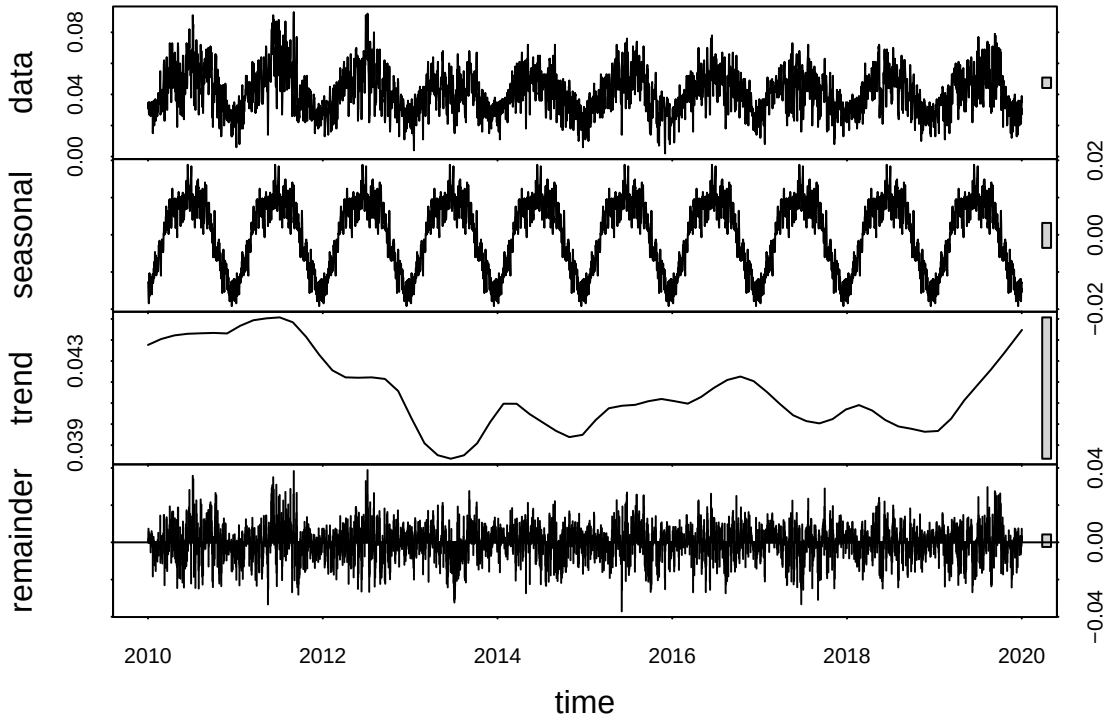
```
## Adding missing grouping variables: `Month`
```

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.

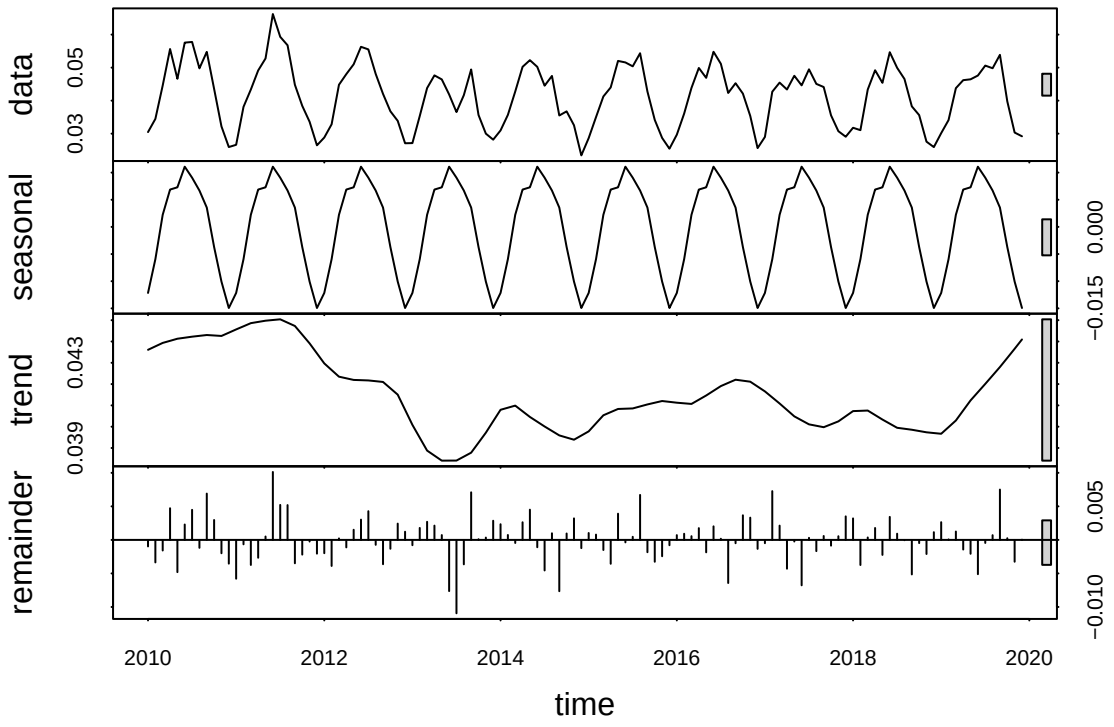
```
#10

f_month <- month(first(GaringerOzoneClean$Date))
f_year <- year(first(GaringerOzoneClean$Date))
f_day <- day(first(GaringerOzoneClean$Date))

GaringerOzone.daily.ts <- ts(GaringerOzoneClean$Daily.Max.8.hour.Ozone.Concentration,
start=c(f_year, f_month, f_day),
frequency = 365)


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

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

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

```
#11


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




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



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

Garinger_data_SMK <- Kendall::SeasonalMannKendall(Garinger.monthly.ts)
summary(Garinger_data_SMK)
```

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

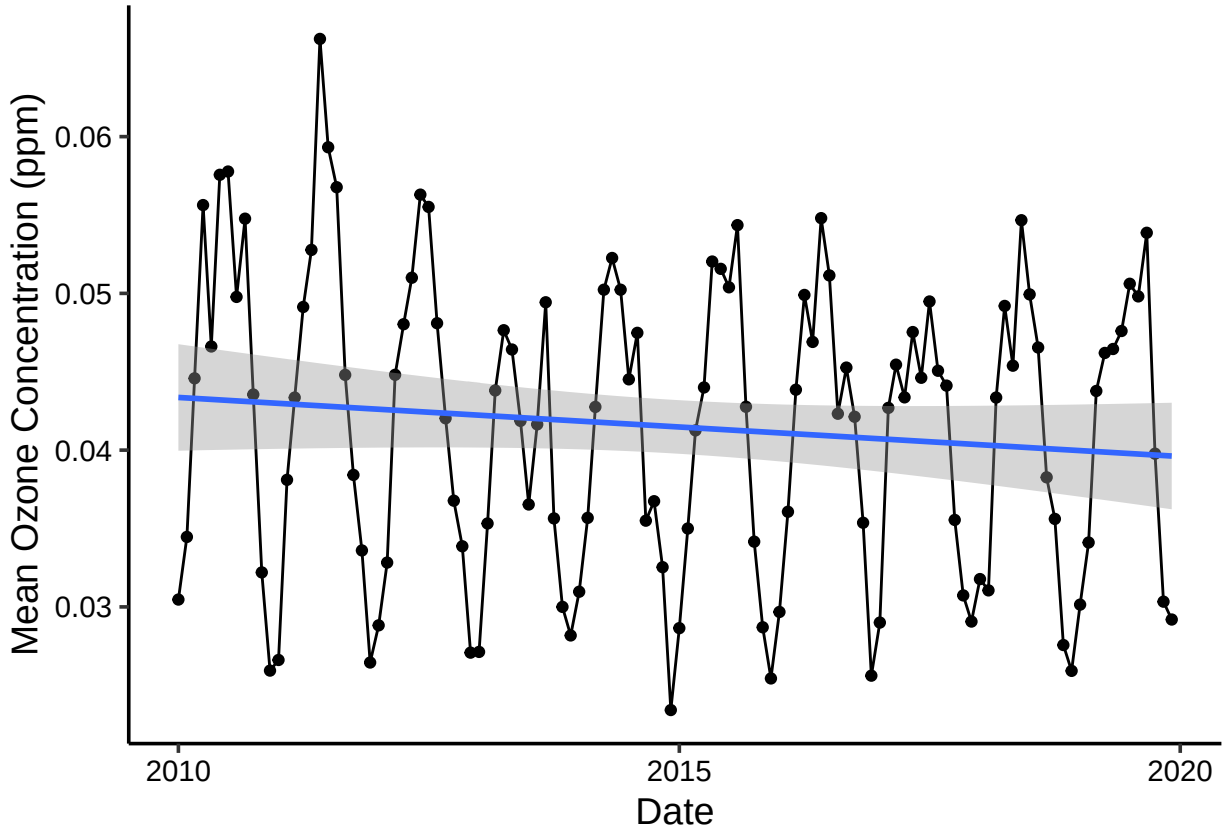
Answer: The seasonal Mann-Kendall test is the most appropriate because this ozone data has seasonality as seen in the decomposed time series plot and is non-parametric.

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
#Visualization
Garinger_data_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_Ozone)) +
  geom_point() +
  geom_line() +
```

```
  ylab("Mean Ozone Concentration (ppm)") +
  geom_smooth( method = lm )
print(Garinger_data_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: We can reject the null hypothesis that the mean ozone concentration at Garinger High School are not changing over time at the 0.05 level. The mean ozone concentration is changing over time at Garinger High School (p-value= 0.0467). Based on the plot, the mean ozone concentration appears to be decreasing.

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.monthly_Components <- as.data.frame(GaringerMonthlyDecomp$time.series[,1:3])
```

```
Garinger.monthly_Components <- mutate(Garinger.monthly_Components,
        Observed = GaringerOzone.monthly$mean_Ozone,
        Date = GaringerOzone.monthly$Date)


Garinger.monthly_Components <- select(Garinger.monthly_Components,
trend, remainder, Observed, Date)

Garinger.monthly.trend.ts <- ts(Garinger.monthly_Components,
start=c(f_year, f_month),
frequency=12)



#16

Garinger_data_trend_SMK <- Kendall::MannKendall(Garinger.monthly.trend.ts)
summary(Garinger_data_trend_SMK)
```

```
## Score =  49276 , Var(Score) = 12326265
## denominator =  114959
## tau = 0.429, 2-sided pvalue =< 2.22e-16
```

Answer: When the seasonality is subtracted, the p-value decreases to 2.22e-16 meaning that the seasonality was masking some of the variation in mean ozone over time. We still reject the null hypothesis that mean ozone concentration are not changing over time at Garinger High School, but at the 0.01 level.