Assignment 8: Time Series Analysis

Emily Guyu Yang

Fall 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

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] "/home/guest/EDE_Fall2024"
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

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

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
library(tseries)
```

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

```r
# Set theme
mytheme <- theme_classic(base_size = 10) +
  theme(axis.text = element_text(color = "dark grey"),
        legend.position = "top",
        plot.title = element_text(size = 13, hjust = 0.5))
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
ozone_GHS_2010 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
ozone_GHS_2011 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
ozone_GHS_2012 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
ozone_GHS_2013 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
ozone_GHS_2014 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
ozone_GHS_2015 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
ozone_GHS_2016 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
ozone_GHS_2017 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
ozone_GHS_2018 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
ozone_GHS_2019 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GraingerOzone <- rbind(
  ozone_GHS_2010, ozone_GHS_2011, ozone_GHS_2012, ozone_GHS_2013, ozone_GHS_2014,
  ozone_GHS_2015, ozone_GHS_2016, ozone_GHS_2017, ozone_GHS_2018, ozone_GHS_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
class(GraingerOzone$Date)
```

```
## [1] "character"
```

```
# set date column
GraingerOzone$Date <- mdy(GraingerOzone$Date)
class(GraingerOzone$Date)
```

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

```
# 4
GraingerOzone <- subset(GraingerOzone,
                        select = c(Date, Daily.Max.8.hour.Ozone.Concentration,
                                   DAILY_AQI_VALUE))

# 5
Day = data.frame(DATE = seq.Date(from = as.Date("2010-01-01"),
                                 to = as.Date("2019-12-31"),
                                 by = "day"))
colnames(Day)[1] <- "Date"

# 6
GraingerOzone <- left_join(Day, GraingerOzone, by = "Date")
dim(GraingerOzone)
```

```
## [1] 3652    3
```

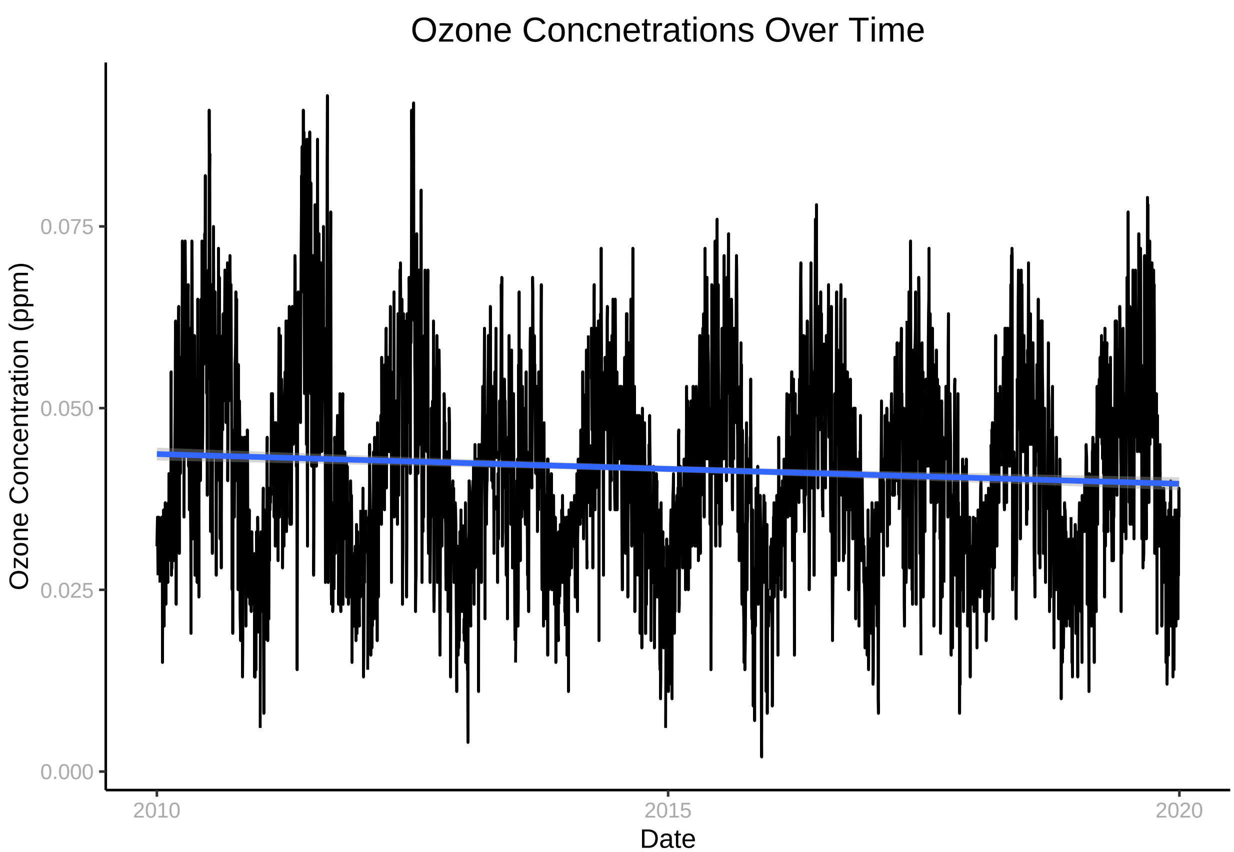
Visualize

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?

```r
#7
GH_ozone_plot <-
ggplot(GraingerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  labs(
    title = "Ozone Concnetrations Over Time",
       x = "Date",
       y = "Ozone Concentration (ppm)") +
  geom_smooth( method = lm ) +
  mytheme
print(GH_ozone_plot)
```

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

```
## Warning: Removed 63 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



Answer: According to the linear trend, there is a slight decrease in the ozone concentration overtime.

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(GraingerOzone)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
```

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

summary(GraingerOzone)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04151                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                                                            NA's   :63
```

Answer: Piecewise constant interpolation fills missing values with the nearest known value, which is unsuitable for fluctuating data like ozone concentrations. Spline interpolation smooths the data with curves, but it can introduce unrealistic peaks or dips. Linear interpolation, a "connect-the-dots" method, provides a straightforward and realistic estimate of gradual day-to-day changes. Therefore, it is the most appropriate choice for daily ozone data.

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
GraingerOzone.monthly <- GraingerOzone %>%
  mutate(Year = year(Date),
         Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(
    monthly.mean = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE)
  ) %>%
  ungroup()%>%
  mutate(Date = as.Date(paste(Year, Month, "01", sep = "-")))
```

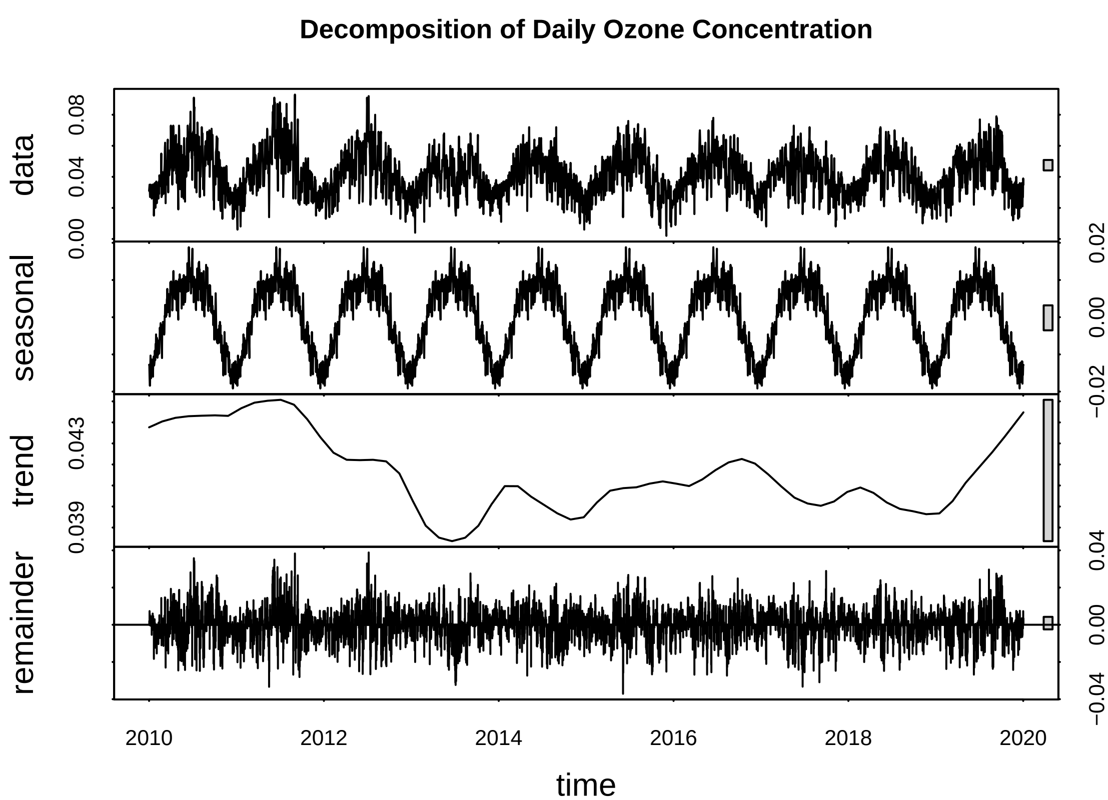
```
## 'summarise()' has grouped output by 'Year'. You can override using the
## '.groups' argument.
```

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.

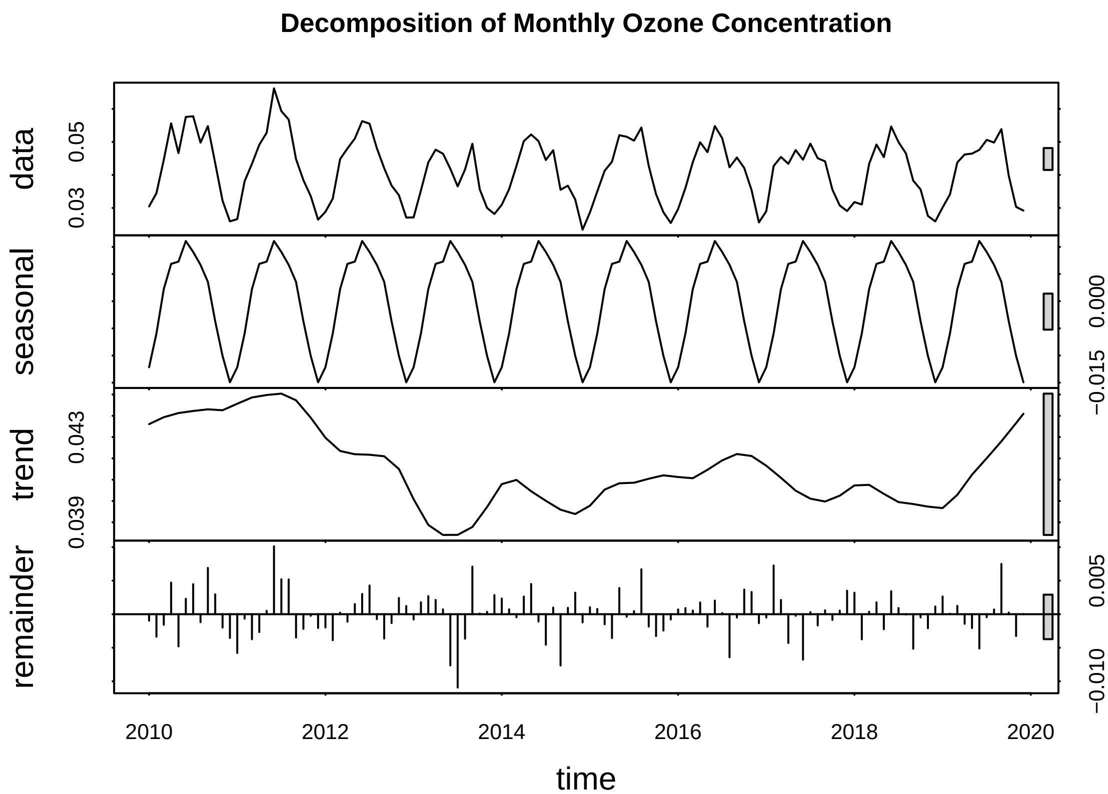
```
#10
GraingerOzone.daily.ts <- ts(GraingerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010, 1),
                             frequency = 365)

GraingerOzone.monthly.ts <- ts(GraingerOzone.monthly$monthly.mean,
                               start = c(2010, 1),
                               frequency = 12)
```

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

```
#11
GraingerOzone.daily.decomp <- stl(
  GraingerOzone.daily.ts, s.window = "periodic"
  )
plot(GraingerOzone.daily.decomp,
     main = "Decomposition of Daily Ozone Concentration")
```



```r
GraingerOzone.monthly.decomp <- stl(
  GraingerOzone.monthly.ts, s.window = "periodic"
  )
plot(GraingerOzone.monthly.decomp,
     main = "Decomposition of Monthly Ozone Concentration")
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

```r
#12
GraingerOzone.monthly.trend <- Kendall::SeasonalMannKendall(GraingerOzone.monthly.ts)

summary(GraingerOzone.monthly.trend)
```

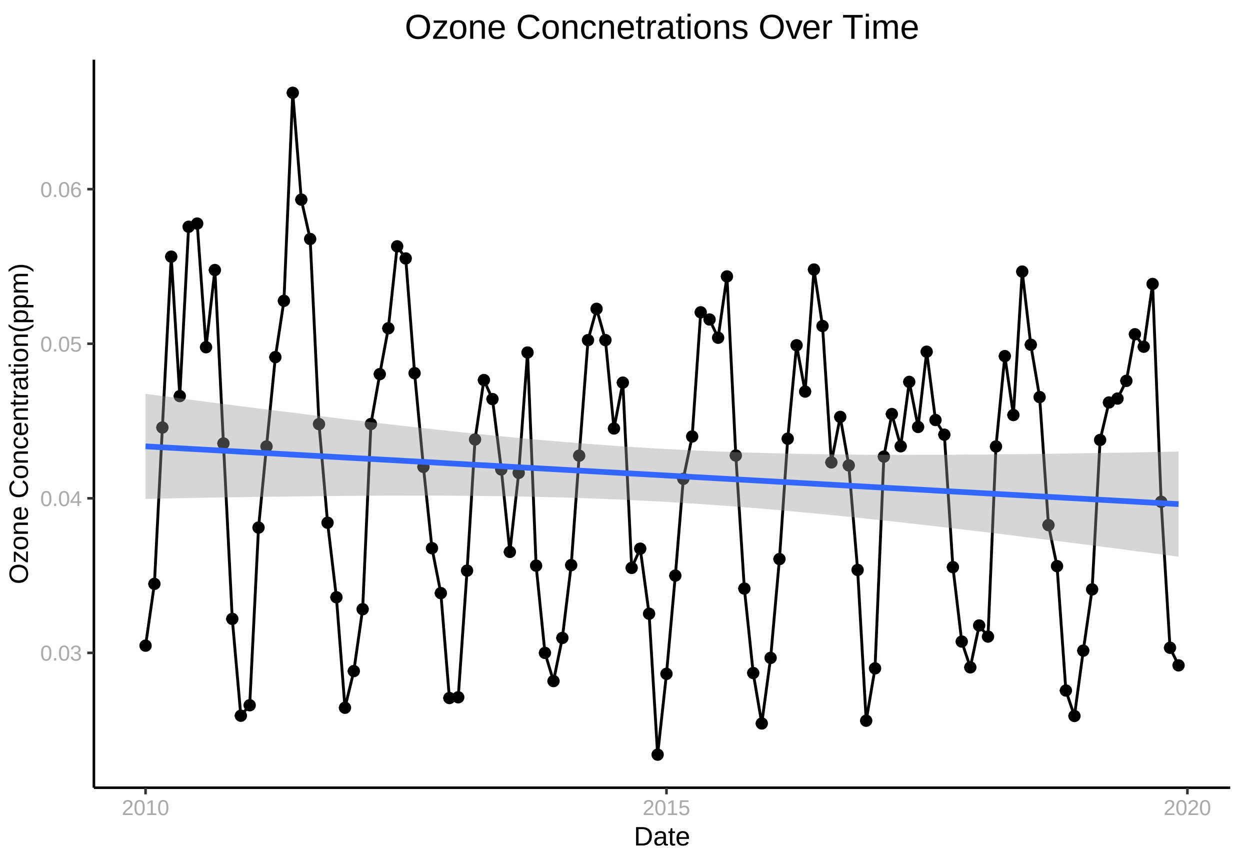
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann-Kendall test is appropriate for analyzing the monthly ozone series because it accounts for seasonal patterns in the data. Ozone concentrations vary by month due to environmental factors. The seasonal Mann-Kendall test isolates the monthly effects and result in a more accurate assessment of the underlying trend.

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.

```r
# 13
GraingerOzone.monthly.plot <- ggplot(GraingerOzone.monthly,
                                     aes(x = Date,
                                         y = monthly.mean)) +
  geom_line() +
  geom_point() +
  labs(
    title = "Ozone Concnetrations Over Time",
       x = "Date",
       y = "Ozone Concentration(ppm)") +
  geom_smooth( method = lm ) +
  mytheme
print(GraingerOzone.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 results from the graph shows a slight downward trend, and the seasonal Mann-Kendall test show a negative tau of -0.143 and p-value of 0.047, meaning it is statistically significant. These suggest that ozone levels have decreased slightly (Score = -77 , Var(Score) = 1499).

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(GraingerOzone.monthly.decomp$time.series)

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

#16
GraingerOzone.monthly.trend <-
  Kendall::MannKendall(GraingerOzone.monthly.ts)
summary(GraingerOzone.monthly.trend)
```

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

> Answer: The Seasonal Mann-Kendall test found a significant downward trend on the complete time series (tau = -0.143, p = 0.0467), while the Mann-Kendall test did not show a significant trend (tau = -0.0594, p = 0.0337) on deseasonalized sereis. It shows that the complete time series is largly influenced by seasonal variations.