Assignment 7: Time Series Analysis

Austin Guimond

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] "/home/guest/R/EDA-Fall2022"
```

```
#load packages
library(tidyverse)
library(lubridate)
library(zoo)
library(ggplot2)
library(trend)
library(Kendall)
library(tseries)
library(dplyr)
#set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)

#2
Air2010 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                             stringsAsFactors = TRUE)
Air2011 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                             stringsAsFactors = TRUE)
Air2012 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                             stringsAsFactors = TRUE)
Air2013 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                             stringsAsFactors = TRUE)
Air2014 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                             stringsAsFactors = TRUE)
Air2015 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                             stringsAsFactors = TRUE)
Air2016 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                             stringsAsFactors = TRUE)
Air2017 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                             stringsAsFactors = TRUE)
Air2018 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                             stringsAsFactors = TRUE)
Air2019 <- read.csv("~/R/EDA-Fall2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                             stringsAsFactors = TRUE)
GaringerOzone <- rbind(Air2010,Air2011, Air2012, Air2013, Air2014, Air2015, Air2016, Air2017,
                       Air2018, Air2019)
```

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



# 4
```

```
GaringerOzone_Wrangle <- GaringerOzone%>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
colnames(GaringerOzone_Wrangle)
```

```
## [1] "Date"
## [2] "Daily.Max.8.hour.Ozone.Concentration"
## [3] "DAILY_AQI_VALUE"
```

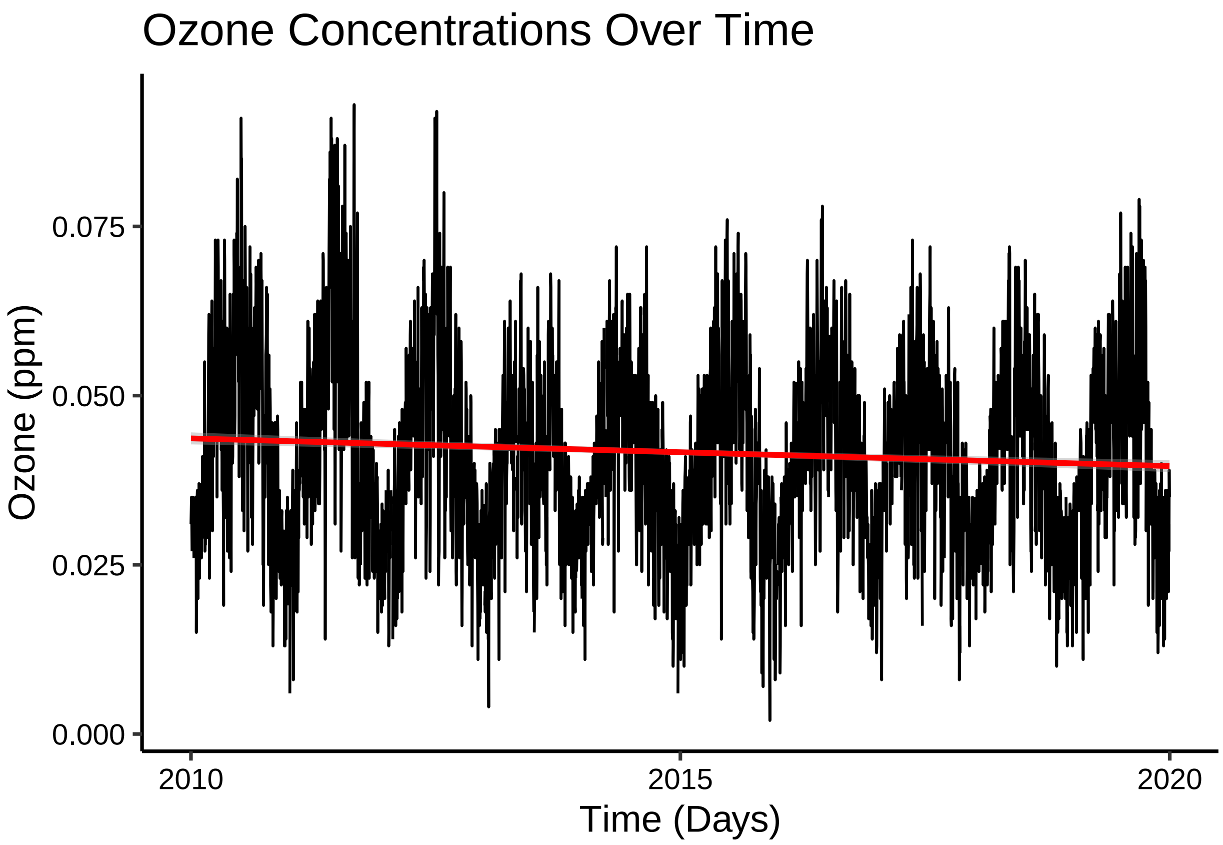
```
# 5
start_date <- as.Date("2010-01-01")
end_date <- as.Date("2019-12-31")
Date <- seq.Date(start_date, end_date, "days")
Days <- as.data.frame(Date)

# 6
GaringerOzone <- left_join(Days, GaringerOzone_Wrangle, 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
PPM_Plot <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line()+
  geom_smooth(method=lm, col= 'red')+
 ggtitle("Ozone Concentrations Over Time")+
  xlab("Time (Days)") + ylab("Ozone (ppm)")
print(PPM_Plot)
```

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

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



Answer: The trendline shows that 8 hour max ozone concentration have been declining between 2010 and 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
GaringerOzone_Clean <-
  GaringerOzone %>%
  mutate(Daily.Max.Clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %>%
  mutate(AQI = zoo::na.approx(DAILY_AQI_VALUE))


summary(GaringerOzone_Clean$Daily.Max.Clean)
```

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

```
summary(GaringerOzone_Clean$AQI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   30.00   38.00   41.41   47.00  169.00
```

Answer: The data appears to be very linear based on the graph from quuestion 7. Based on the linear trend, it makes the most sense to use a linear interpolation method which assumes missing data will continue the same trend. Use of a piecewise constant or nearest neighbor could make the data choppy and alter the trend that is present. Spline is similar to the linear interpolation but uses the qudratic formaula and therefore can reuqire more computing power and take more time for large datasets.

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)

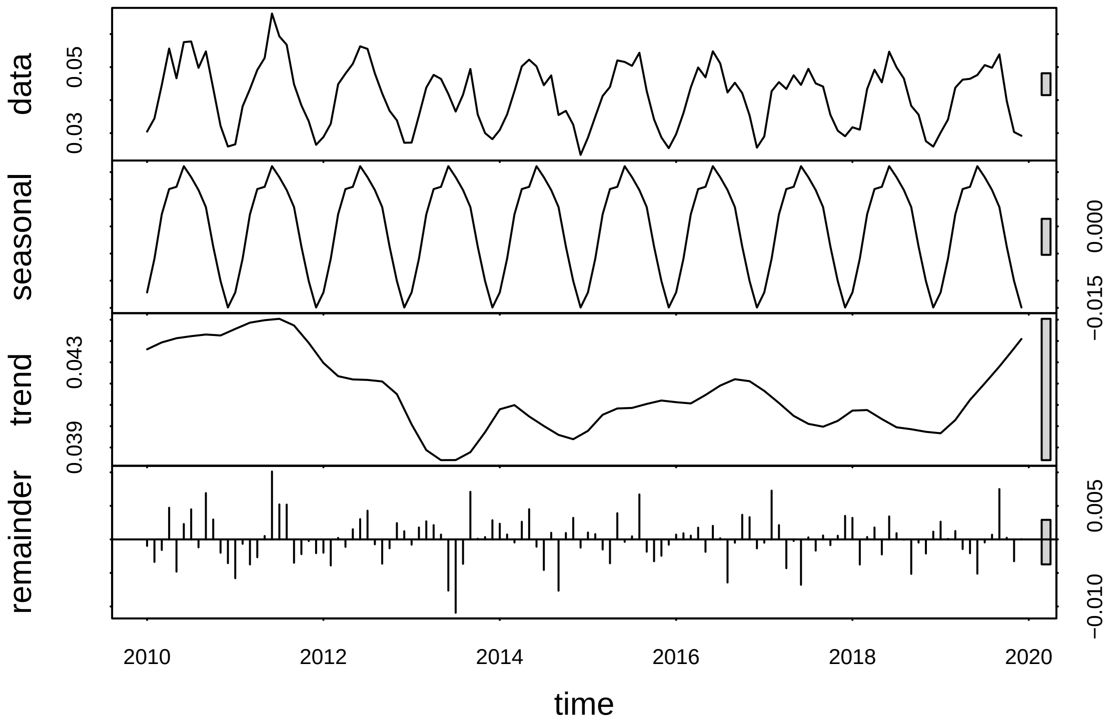
```
#9
GaringerOzone_Update <- GaringerOzone_Clean%>%
  select(Date, Daily.Max.Clean)

GaringerOzone_Date <- GaringerOzone_Update %>%
mutate(month = month (Date)) %>%
mutate(year = year (Date))

GaringerOzone.monthly <- GaringerOzone_Date %>%
mutate( Date = my(paste0(month,"-", year)))

GaringerOzone.Monthly <- GaringerOzone.monthly %>%
  group_by(Date) %>%
  dplyr::summarize(value = mean (Daily.Max.Clean)) %>%
  as.data.frame()
```

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.

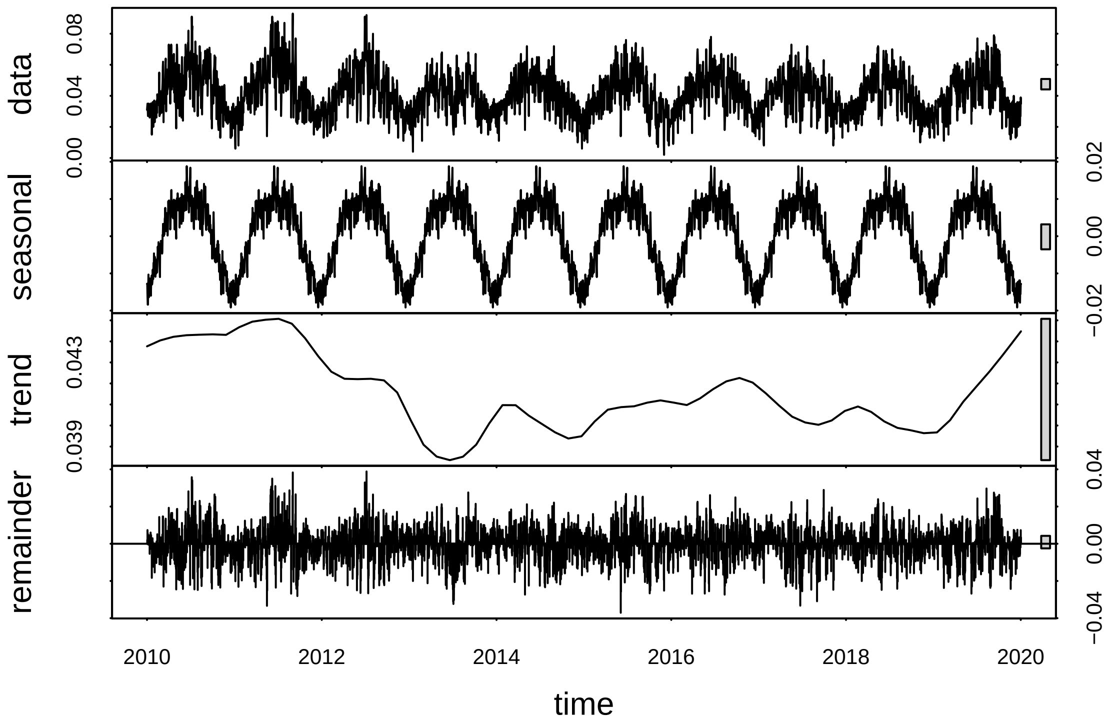
```
#10
GaringerOzone_ts <- ts(GaringerOzone_Update$Daily.Max.Clean, start = c(2010,01), frequency = 365)


GaringerOzone.Monthly_ts <- ts(GaringerOzone.Monthly$value, start = c(2010,01), frequency = 12)
```

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

```
#11
Monthly_decomp <- stl(GaringerOzone.Monthly_ts,s.window = "periodic")
plot(Monthly_decomp)
```



```
Daily_decomp <- stl(GaringerOzone_ts,s.window = "periodic")
plot(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
Monthly_Ozone_Trend <- Kendall::SeasonalMannKendall(GaringerOzone.Monthly_ts)
summary(Monthly_Ozone_Trend)
```

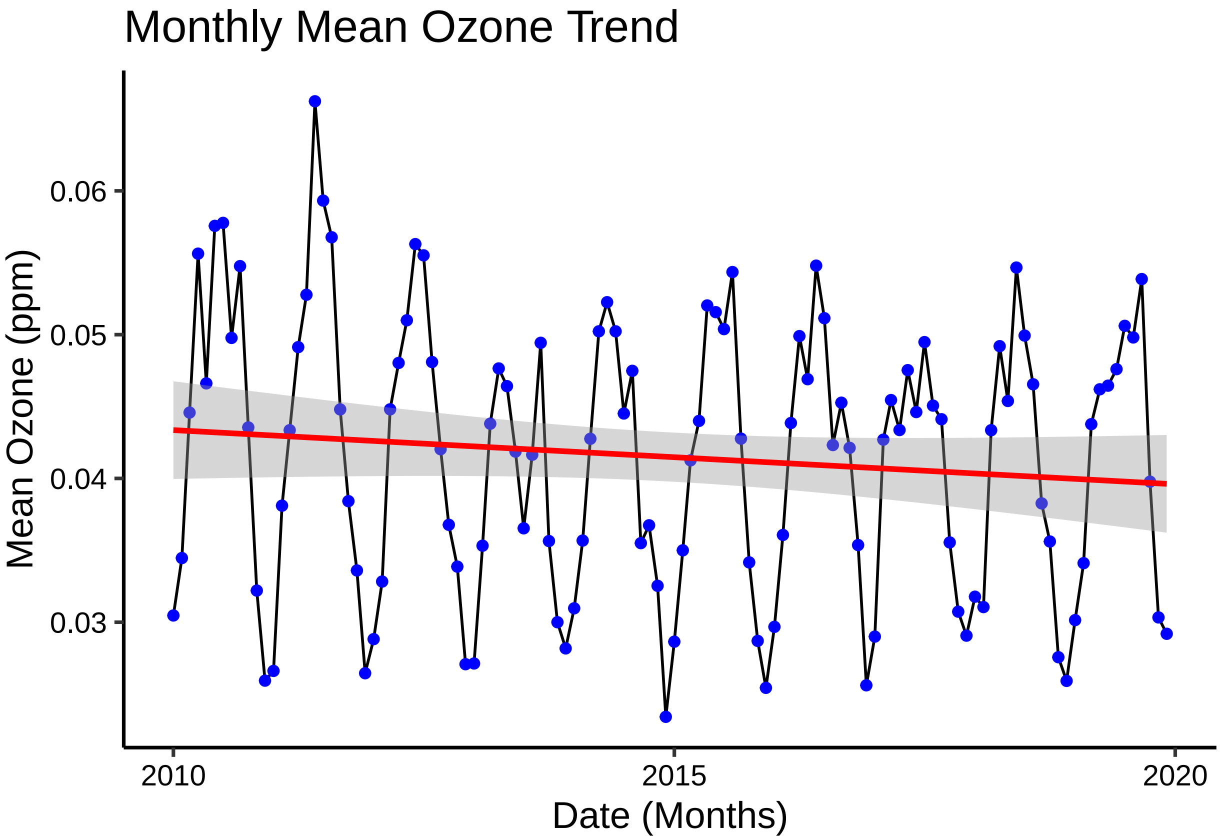
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The dataset we are using is based on means from month to month. Therefore, seasonality is a variable in the data that influences the trend. For this analyis, we need to use the seasonal Mann Kendall due to the presence of seasonal variation. If we subtracted seasonality from the data, we could use just the Mann Kendall Test.

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
Monthly_Ozone_Plot <- ggplot(GaringerOzone.Monthly, aes(x = Date, y = value)) +
  geom_line() +
  geom_point(alpha = 1, size = 1.5, color="blue")+
 ggtitle("Monthly Mean Ozone Trend")+
  geom_smooth(method=lm, col= 'red')+
  xlab("Date (Months)") + ylab("Mean Ozone (ppm)")
print(Monthly_Ozone_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.

Study question: Have ozone concentrations changed over the 2010s at this station? > Answer: The graph of monhtly mean ozone values shows a trend of lower ozone values over time. The P value from the seasonal Mann Kendall test is also below .05, which indicates the trend is significant and we can reject our null hupothesis (2-sided pvalue =0.046724).

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
Monthly_decomp_Season <- stl(GaringerOzone.Monthly_ts,s.window = "periodic")
summary(Monthly_decomp_Season)
```

```
##  Call:
##  stl(x = GaringerOzone.Monthly_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal                trend              remainder
##  Min.   :-0.014935333   Min.   :0.03840568   Min.   :-0.010941198
##  1st Qu.:-0.006975626   1st Qu.:0.04013416   1st Qu.:-0.001986892
##  Median : 0.002895178   Median :0.04107812   Median : 0.000087698
##  Mean   : 0.000000000   Mean   :0.04150482   Mean   :-0.000011519
##  3rd Qu.: 0.006981830   3rd Qu.:0.04246828   3rd Qu.: 0.002048822
##  Max.   : 0.011093186   Max.   :0.04504168   Max.   : 0.010131384
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##      0.013957     0.002334  0.004036      0.015350
##    %  90.9         15.2      26.3          100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 1201 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 121 2 2
##  $ inner: int 2
##  $ outer: int 0
```

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

Monthly_Components <- mutate(Monthly_Components,
        Observed = GaringerOzone.Monthly$value,
        Date = GaringerOzone.Monthly$Date)

OzoneSeasonAdj <- Monthly_Components %>%
  mutate(Subtract.Season = Monthly_Components$Observed - Monthly_Components$seasonal)
summary(OzoneSeasonAdj)
```

```
##     seasonal             trend          remainder             Observed
##  Min.   :-0.014935   Min.   :0.03841   Min.   :-1.094e-02   Min.   :0.02342
##  1st Qu.:-0.006976   1st Qu.:0.04013   1st Qu.:-1.987e-03   1st Qu.:0.03380
##  Median : 0.002895   Median :0.04108   Median : 8.770e-05   Median :0.04335
##  Mean   : 0.000000   Mean   :0.04150   Mean   :-1.152e-05   Mean   :0.04149
##  3rd Qu.: 0.006982   3rd Qu.:0.04247   3rd Qu.: 2.049e-03   3rd Qu.:0.04915
##  Max.   : 0.011093   Max.   :0.04504   Max.   : 1.013e-02   Max.   :0.06623
##       Date            Subtract.Season
##  Min.   :2010-01-01   Min.   :0.02747
##  1st Qu.:2012-06-23   1st Qu.:0.03932
##  Median :2014-12-16   Median :0.04120
##  Mean   :2014-12-16   Mean   :0.04149
##  3rd Qu.:2017-06-08   3rd Qu.:0.04325
##  Max.   :2019-12-01   Max.   :0.05514
```

```
#16
Monthly_Ozone_NonSeasonal_Trend <- Kendall::MannKendall(OzoneSeasonAdj$Subtract.Season)
summary(Monthly_Ozone_NonSeasonal_Trend)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer:The results of the non-seasonal Mann Kendall test also indicate that the trend of declining ozone over time is signficant even when seasonality is removed. The p value for the seasonally corrected test was 0.0075402, indicating an even more significant trend of decline compared to the P value of the seasonal test.