Assignment 7: Time Series Analysis

Sara Diamond

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] "/Users/saradiamond/Documents/Environmental_Data_Analytics_2022"
```

```
setwd("/Users/saradiamond/Documents/Environmental_Data_Analytics_2022")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     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)

#setting theme
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
#loading the datasets
G1 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors = TRUE)
G2 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors = TRUE)
G3 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors = TRUE)
G4 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors = TRUE)
G5 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors = TRUE)
G6 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors = TRUE)
G7 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors = TRUE)
G8 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors = TRUE)
G9 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors = TRUE)
G10 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors = TRUE)

#combining them all into 1 dataset

GComplete <- rbind.data.frame(G1,G2,G3,G4,G5,G6,G7,G8,G9,G10, stringsAsFactors = TRUE)
```

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 changing to date

class(GComplete$Date) #checking the original class
```

```
## [1] "factor"
```

```
GComplete$Date <-as.Date(GComplete$Date,
                         format = "%m/%d/%Y") #changing factor to date
class(GComplete$Date) #rechecking class to make sur eit is date
```

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

```
# 4 wrangling the data

GCompleteWrangled <-
  GComplete %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration,
         DAILY_AQI_VALUE)
summary(GCompleteWrangled)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-03   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2015-01-04   Median :0.04100                      Median : 38.00
##  Mean   :2015-01-01   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-02   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
```

```
sum(is.na(GCompleteWrangled))
```

```
## [1] 0
```

```
# 5
#creating the dataframe
Daily.Data <-
as.data.frame(seq.Date(as.Date("2010/01/01"),
                       as.Date("2019/12/31"), "day"))

  colnames(Daily.Data) <- "Date" #changing the name of the column to say Date

# 6 combining the two datasets

GComplete <- left_join(Daily.Data, GCompleteWrangled)
```

```
## Joining, by = "Date"
```

Visualize

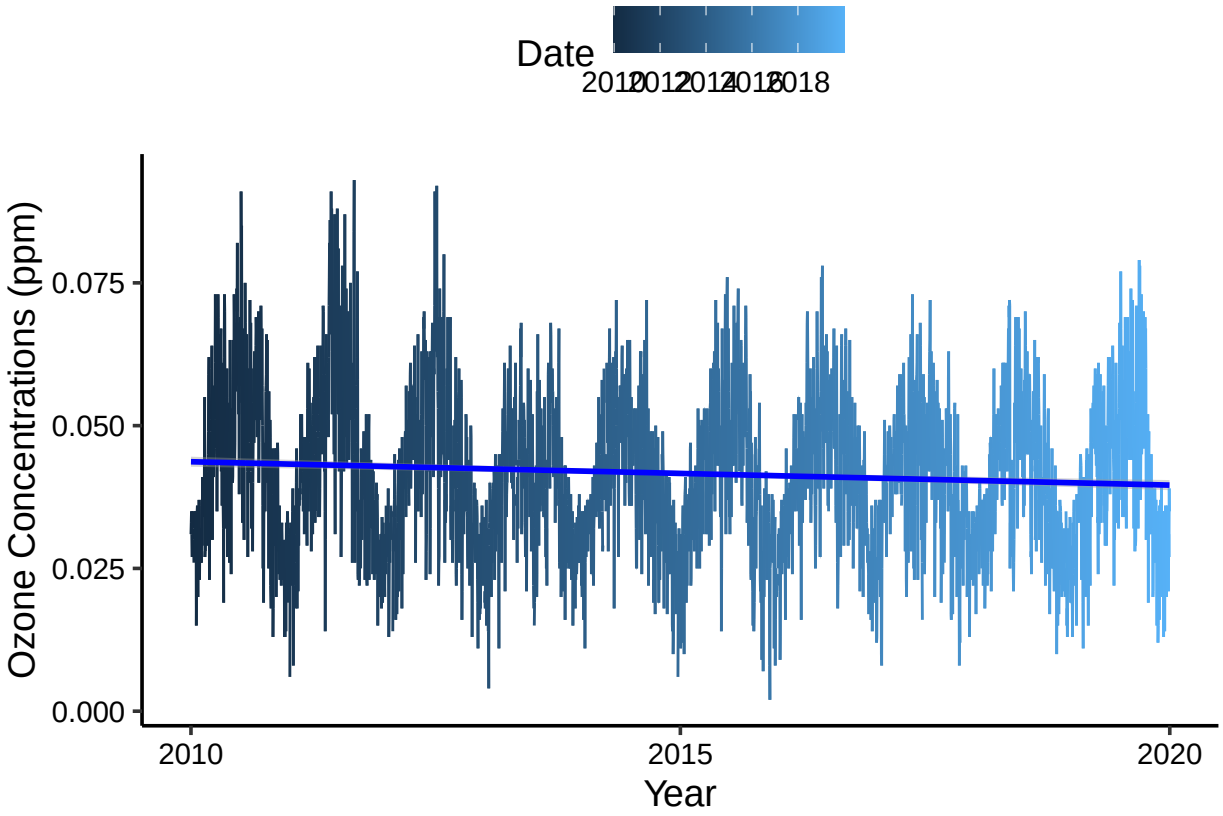
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
library(ggplot2)

#creating a line plot to show ozone over time
Ozone.Time <-
```

```
  ggplot(GComplete, aes(x=Date, y = Daily.Max.8.hour.Ozone.Concentration))+
  geom_line(aes(color=Date))+
  geom_smooth(method = 'lm', color = "blue")+
  labs(x = "Year", y = "Ozone Concentrations (ppm)")
print(Ozone.Time)
```

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

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



Answer: By looking at the plot it may seem like there could be an overall descreasing trend, but we are actually unsure of the overall trend because of the obvious increases and decreases in the seasonality data. In order to determine the trend we will have to run further tests, which would be a time series in this case.

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

#filling in the missing data with a linear interp.
GComplete.Days <-
  GComplete %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration_Clean =
           na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

```
summary(GComplete.Days$Daily.Max.8.hour.Ozone.Concentration_Clean)
```

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

```
#double checking the data
```

Answer: For this question, using a linear interpolation would be the best option rather than a piecewise or spline because a linear interpolation connects the data points from one point in time to another. Using a piecwise would apply equal values to each point closest to a certain point in time. We wouldn't use spline because it is a quadratic function and we are specifically wanting to use linear here.

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 creating new data drame and wrangling data with pipes
library(dplyr)
GaringerOzone.monthly <-
  GComplete.Days %>%
  mutate(Month = month(Date),
         Year = year (Date)) %>%
  mutate(Date = my(paste0(Month, "-", Year))) %>%
  group_by(Date, Month, Year) %>%
  summarise(mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration_Clean)) %>%
  select(mean_Ozone, Date)
```

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

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

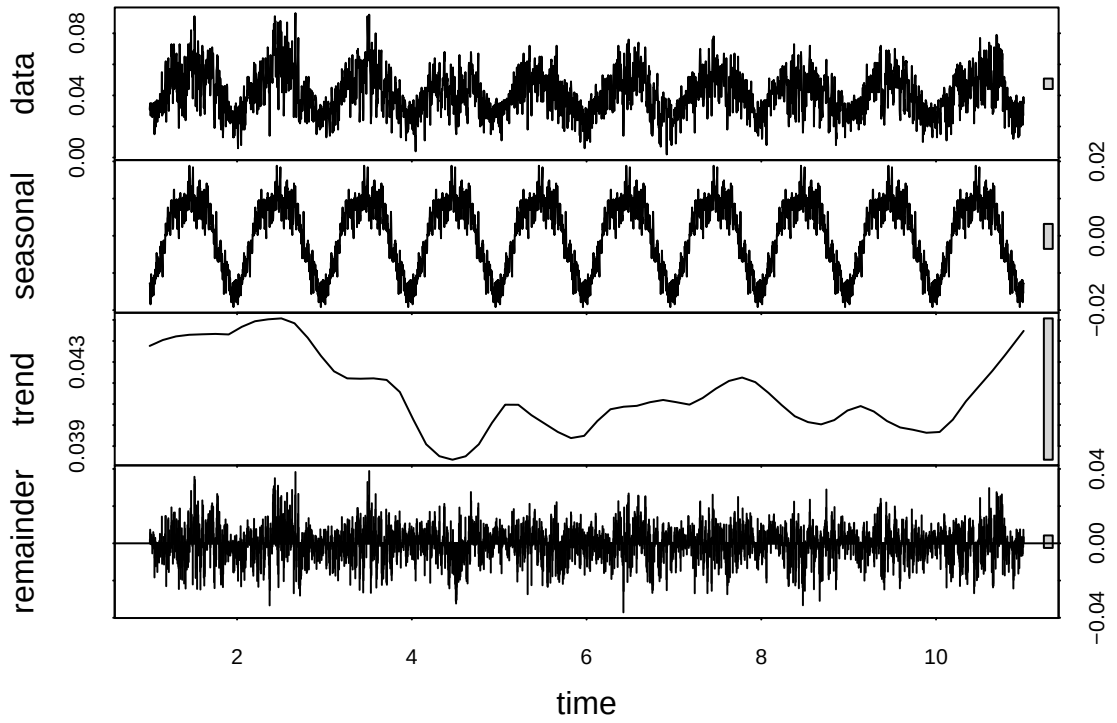
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```
#10

fdaily <- day(first(GComplete.Days$Date)) #taking the first day

fmonthly <- month(first(GaringerOzone.monthly$Date)) #taking first month

fyear <- year(first(GaringerOzone.monthly$Date)) #taking first year

#daily time series
GaringerOzone.daily.ts <- ts(GComplete.Days$Daily.Max.8.hour.Ozone.Concentration_Clean,
                             start = c(fdaily,fmonthly, fyear), frequency=365)
#monthly time series
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_Ozone,
                               start = c(fmonthly,fyear), frequency=12) #monthly time series
```

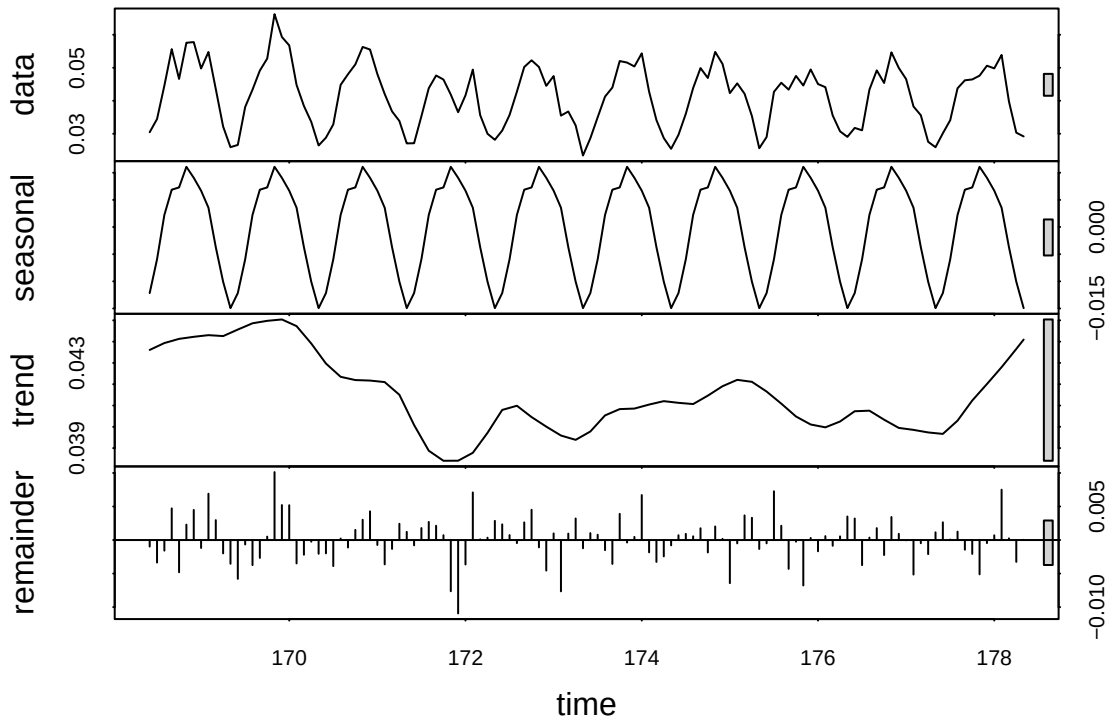
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11
```

```
#daily decomposition
GComplete.Days.decompostion <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(GComplete.Days.decompostion)
```




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



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

#running mann-kendall with seasonality

GaringerOzone.monthly.MK <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.MK)
```

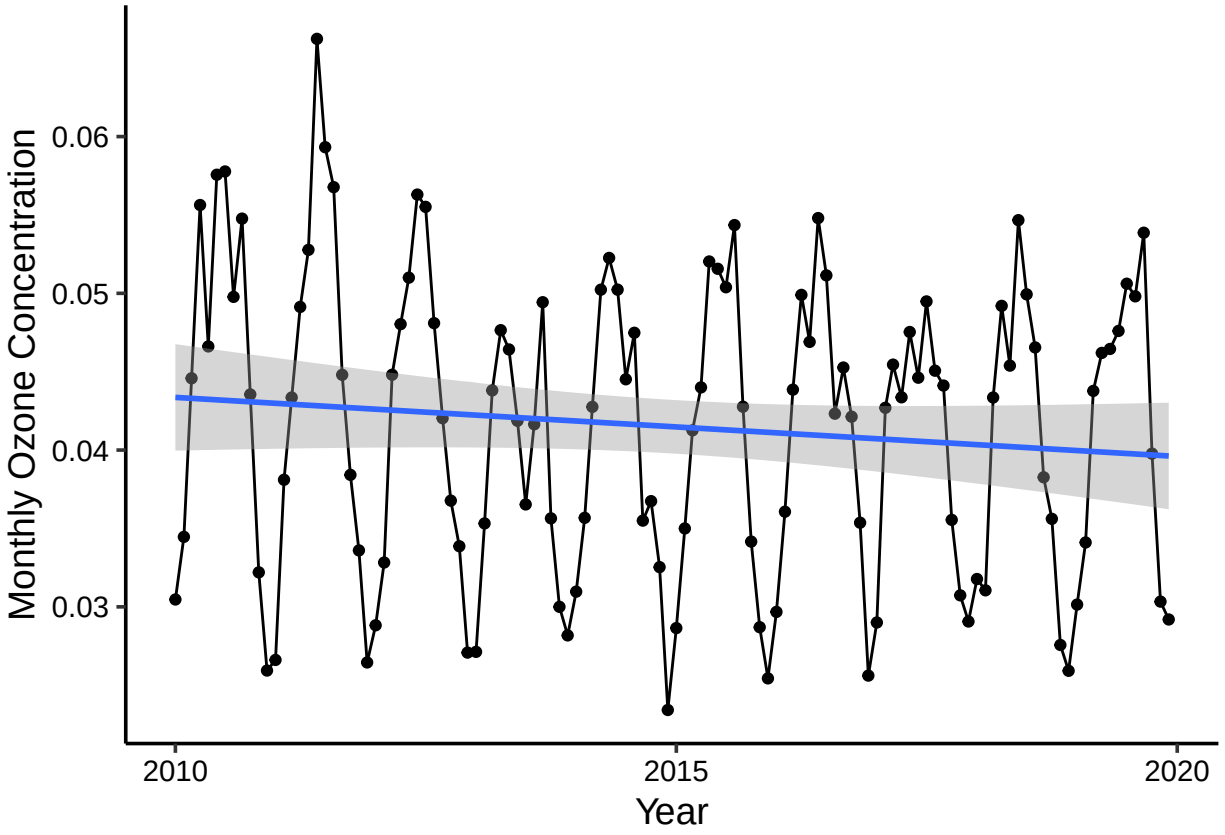
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann-Kendall is appropriate because we are dealing specifically with seasonal data and this test is the only one that we can use to measure the seasonal data 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
Garinger.Ozone.monthly.plots <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_Ozone)) +
  geom_point() +
  geom_line() +
  labs(x = "Year", y = "Monthly Ozone Concentration") +
  geom_smooth(method = lm)
print(Garinger.Ozone.monthly.plots)
```

```
## `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 reject the null hypothesis that there has been no change in ozone levels over the last decade and it actually seems to be decreasing over time according to the plot (Score = -77, Var(Score) = 1499, denominator = 539.4972, tau = -0.143, 2-sided pvalue = 0.0467)

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
#removing the seasonal column
GaringerOzone.monthly.ts.components <-
  as.data.frame(GaringerOzone.monthly.decomposition$time.series[,2:3])

#adding the remaining to together into the observed
GaringerOzone.monthly.ts.components<-
  mutate(GaringerOzone.monthly.ts.components,
         Observed = GaringerOzone.monthly.ts.components$trend +
           GaringerOzone.monthly.ts.components$remainder,
         Date = GaringerOzone.monthly$Date)
#16

fmonth.2 <- month(first(GaringerOzone.monthly.ts.components$Date))
#taking the first month and year
fyear.2 <- year(first(GaringerOzone.monthly.ts.components$Date))

#making another time series for the observed to run MK test
GaringerOzone.monthly.second.ts <- ts(GaringerOzone.monthly.ts.components$Observed,
                                      start = c(fmonth.2,fyear.2), frequency=12)
#now running the MK test (non-seasonal)
GaringerOzone.monthly.2 <-Kendall::MannKendall(GaringerOzone.monthly.second.ts)
summary(GaringerOzone.monthly.2)
```

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

Answer: In this test, we would also reject the null hypothesis because the p-value provided in the results is less than 0.05 meaning that there is a change in overall ozone levels over the last decade. Compared to the seasonal Mann-Kendall test, the p-value is lower which could mean there is greater support to accept the alternative hypothesis of there being a change in the ozone levels over time. This happened when we took out the seasonal data column. Like the previous test, the tau is still negative meaning we can conclude that the ozone levels are decreasing over time. In this case they are appearing to decrease more than in the seasonal test (Score = -1179, Var(Score) = 194365.7, denominator = 7139.5, tau = -0.165, 2-sided pvalue = 0.0075)