Assignment 7: Time Series Analysis

Isabel Zungailia

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.
#Check working directory
  getwd()
```

```
## [1] "/home/guest/EDA-Fall2022"
```

```
#Load packages
  library(tidyverse)
  library(lubridate)
  library(trend)
  library(zoo)
  library(Kendall)

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

```
#2. Import datasets (10) from the Ozone_TimeSeries folder in the Raw data folder
GaringerNC2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFac
GaringerNC2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFac
GaringerNC2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFac
GaringerNC2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFac
GaringerNC2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFac
GaringerNC2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFac
GaringerNC2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFac
GaringerNC2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFac
GaringerNC2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFac
GaringerNC2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFac

#Combine 10 datasets into a single dataframe named `GaringerOzone`
GaringerOzone <- rbind(GaringerNC2010, GaringerNC2011, GaringerNC2012, GaringerNC2013, GaringerNC2014, G
```

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. Set date column as date class
class(GaringerOzone$Date)
```

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

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

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

```
#4. Wrangle dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, an
GaringerOzone.wrangled <-
  GaringerOzone %>%
    select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

#5. Create a new data frame that contains a sequence of dates from 2010-01-01 to 2019-12-31
Days <- as.data.frame(seq.Date(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day"))

#Rename column name to Date
colnames(Days) <- c("Date")

#6. Combine the two data frames using a 'left_join' (named it "GarignerOzone2" since I already had a "G
GaringerOzone2 <- left_join(Days, GaringerOzone.wrangled)
```

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

Visualize

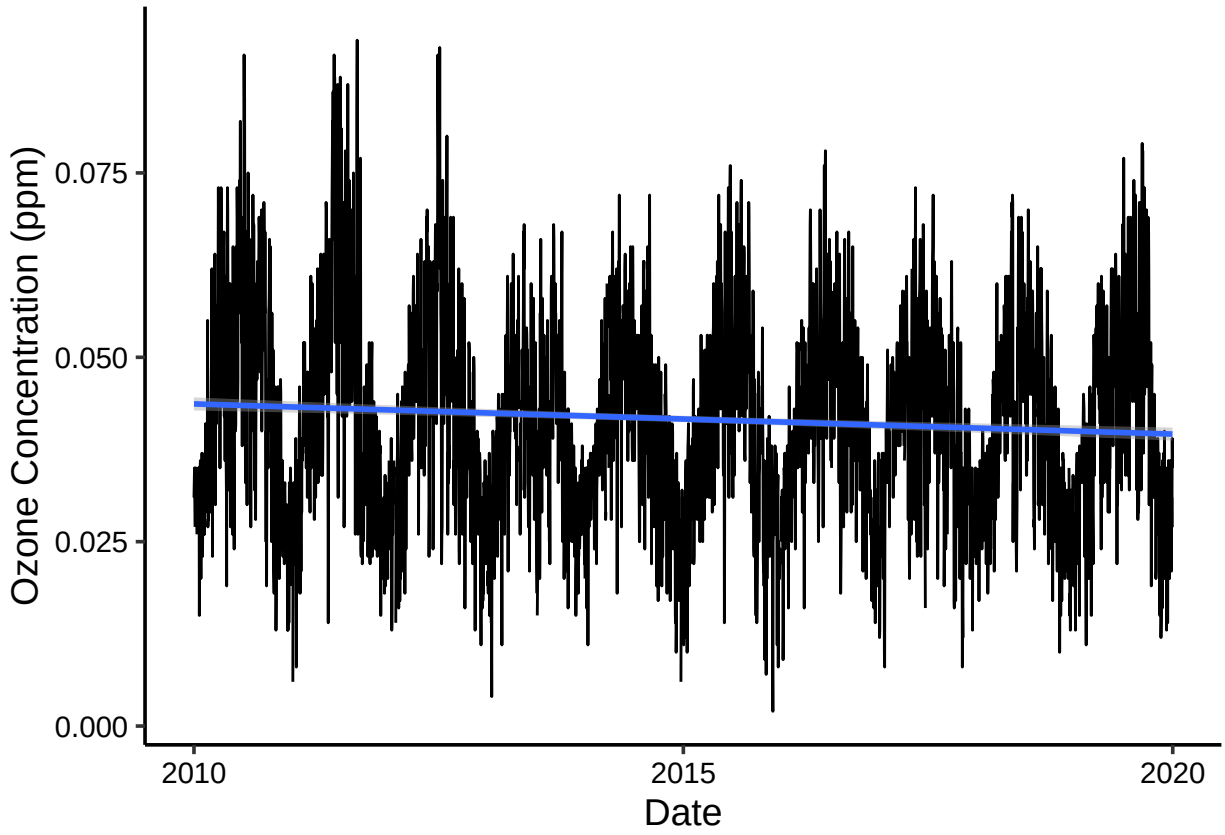
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7.Create a line plots that shows ozone concentrations (ppm) over time
OzoneConcentrations <- ggplot(GaringerOzone2, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  labs(x = "Date", y = "Ozone Concentration (ppm)") + #Rename axis
  geom_smooth(method = lm)

plot(OzoneConcentrations)
```

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

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



Answer: This plot suggests that that ozone concentrations have remained relatively stable over time, despite fluctuations throughout the seasons. The slope of the trend line is relatively flat, and the visible trend is likely due to seasonality.

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. Use a linear interpolation to fill in missing daily data for ozone concentration
GaringerOzone2$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone2$Daily.Max.8.hour.Ozone.
summary(GaringerOzone2$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
```

```
#[Not asked for in the question...fill in missing daily data for AQI values]
#GaringerOzone2$DAILY_AQI_VALUE <- na.approx(GaringerOzone2$DAILY_AQI_VALUE)
#summary(GaringerOzone2$DAILY_AQI_VALUE)
```

Answer: Interpolation is a common method used to replace missing values in a dataset. It is important to note that we want to replace the missing values through interpolation to create a continuous observation - we do NOT want to remove them completely. Interoplation will not change the main properties of the distribution (min, max, mean, etc). Linear interpolations are a "connect the dots" approach - missing data values are assumed to fall between the previous and next measurement. A straight line is drawn between the two predetermined values to generate an average value for the missing values. For this dataset, a linear interpolation was used instead of a piecewise constant or spline interpolation methods. The piecewise constant, often referred to as the "nearest neightbor" approach, assigns the missing data to be equal to the measurement nearest to that date. Spline is similar to a linear interpretation except for that fact that it uses a quadratic function to interpolate (rather than a straight line). The linear interpretation is the best method to use for this dataset because it provides the most reasonable method for averaging out ozone concentration values.

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. Create a new data frame called `GaringerOzone.monthly`
GaringerOzone.monthly <-
  GaringerOzone2 %>%
  mutate(Month = month(Date)) %>% #Add month column
  mutate(Year = year(Date)) %>% #Add year column
  mutate(month_year = my(paste(Month,"-",Year))) %>% #Create a new date column
  group_by(month_year) %>%
  summarize(mean_monthly = mean(Daily.Max.8.hour.Ozone.Concentration)) #Mean ozone concentrations

summary(GaringerOzone.monthly)
```

```
##    month_year          mean_monthly
##  Min.   :2010-01-01   Min.   :0.02342
##  1st Qu.:2012-06-23   1st Qu.:0.03380
##  Median :2014-12-16   Median :0.04335
##  Mean   :2014-12-16   Mean   :0.04149
##  3rd Qu.:2017-06-08   3rd Qu.:0.04915
##  Max.   :2019-12-01   Max.   :0.06623
```

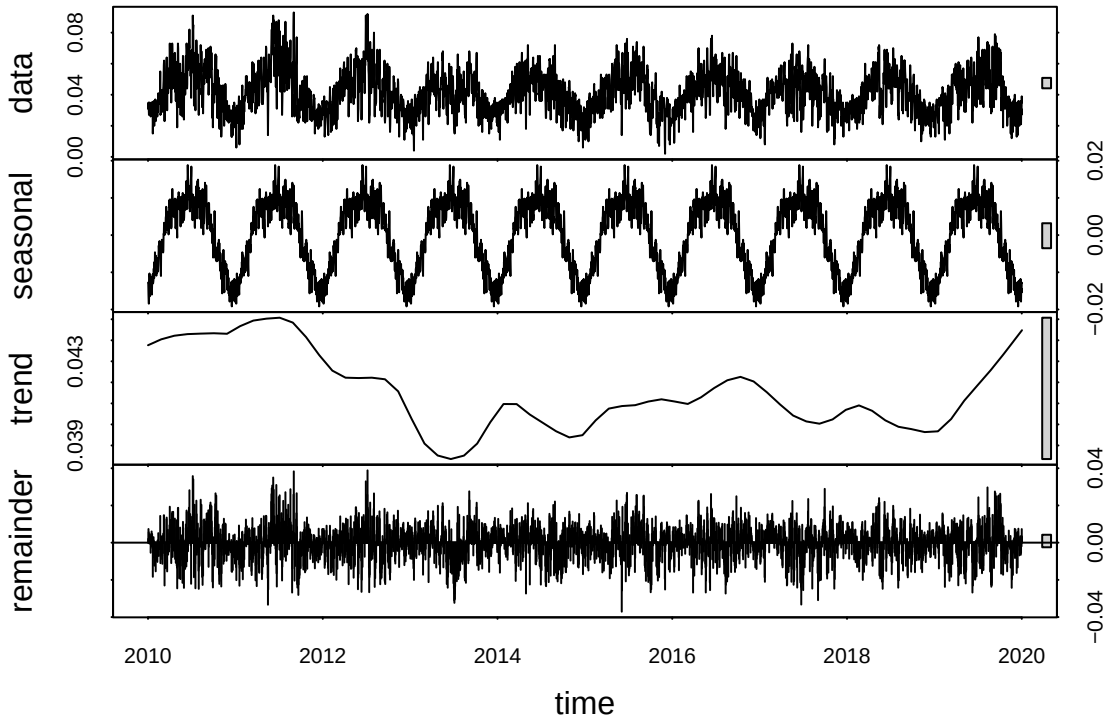
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```
#10. Create a time series object named `GaringerOzone.daily.ts`
GaringerOzone.daily.ts <- ts(GaringerOzone2$Daily.Max.8.hour.Ozone.Concentration, start = c(2010, 01),
frequency = 365)
```

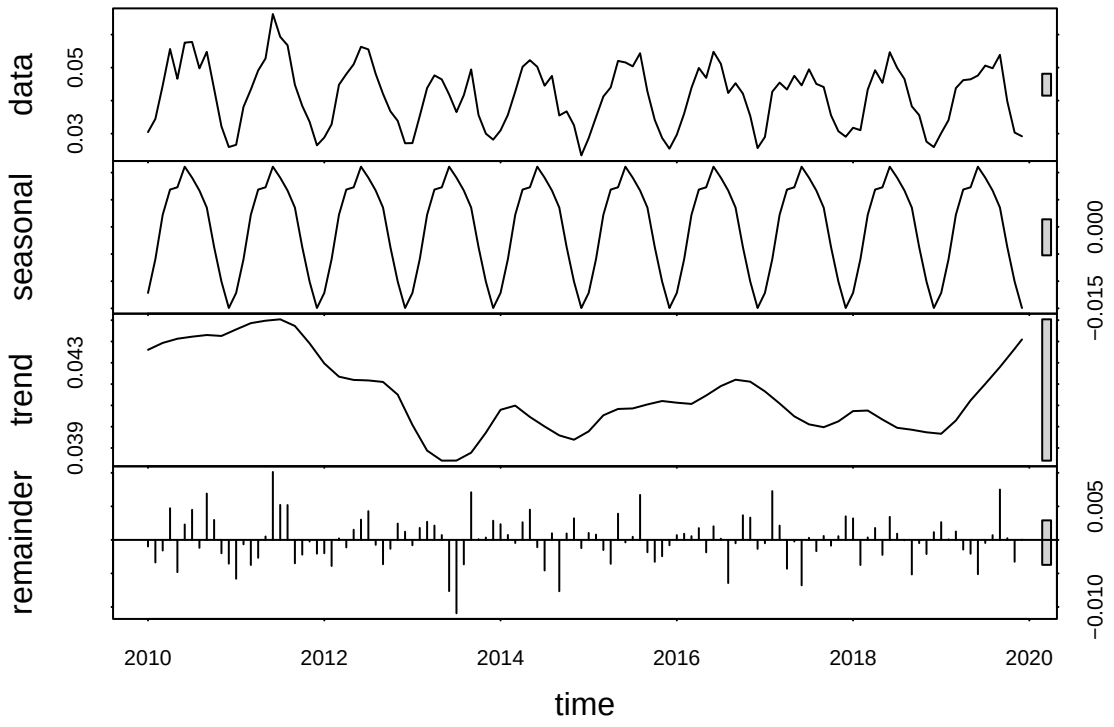
```
#Create a time series object named `GaringerOzone.monthly.ts`
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_monthly, start = c(2010, 01),
frequency = 12)
```

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

```
#11. Decompose and plot the daily time series objects/components
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.decomposed)
```




```
#Decompose and plot the monthly time series objects/components
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
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?

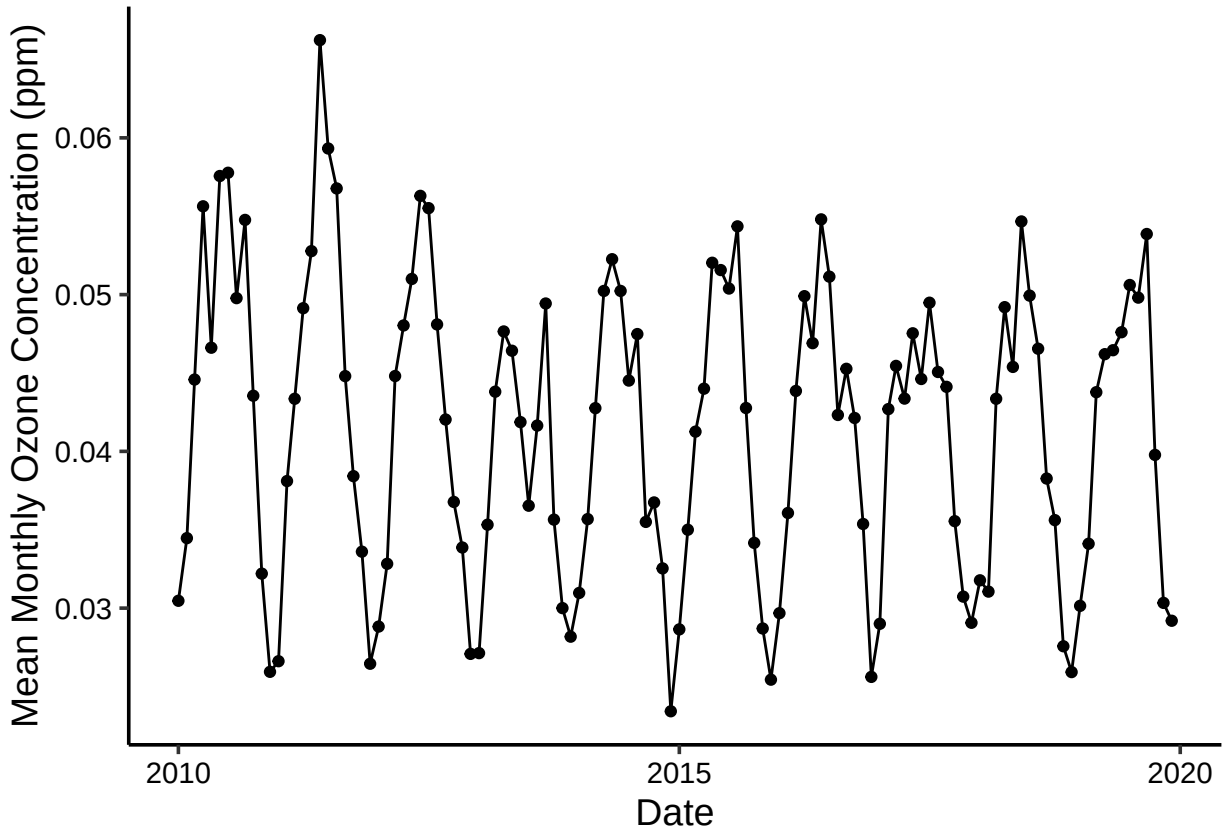
```
#12. Monotonic trend analysis - seasonal Mann-Kendall test
GaringerOzone_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
GaringerOzone_trend
```

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

Answer: The seasonal Mann-Kendall test is a specific test, used for monotonic trend analysis, that takes into account seasonality. It is the only test that can be used for seasonality - it runs a separate test on each of the seasons individually.

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. Plot monthly ozone concentrations over time
MeanMonthlyOzone.plot <- ggplot(GaringerOzone.monthly, aes(x = month_year, y = mean_monthly)) +
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Mean Monthly Ozone Concentration (ppm)")
print(MeanMonthlyOzone.plot)
```



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 research question asked: Have ozone concentrations changed over the 2010s at this station? This graphs displays the fluctuation of ozone concentrations throughout the 2010s. During the winter months (particularly December and January), the mean ozone concentrations are the lowest, and the highest values are seen during the summer months (particularly May-August). The minimum value of ozone concentration during this time frame was 0.02342 ppm, the maximum ozone concentration value was 0.06623 ppm (on 2011-06-01), and the mean value was 0.04149 ppm. The seasonal Mann-Kendall test generated a tau value of -0.143 and a p-value of ~0.046724. The p-value is very close to 0.05, which suggests that the relationship between ozone concentrations and time has only a small level of significance. The tau value is used to interpret the strength of the relationship between the variables (ranges between -1 and 1), and a negative tau value indicates that the variables are inversely related.

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. Subtract the seasonal component from the `GaringerOzone.monthly.ts`
GaringerOzone.monthly.nonseasonal <- GaringerOzone.monthly.ts - GaringerOzone.monthly.decomposed$time.s

#16. Run the Mann Kendall test on the non-seasonal Ozone monthly series
GaringerOzone.monthly.MannKendall <- Kendall::MannKendall(GaringerOzone.monthly.nonseasonal)
GaringerOzone.monthly.MannKendall
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
summary(GaringerOzone.monthly.MannKendall)
```

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

Answer: After subtracting the seasonal component from the running the `GaringerOzone.monthly.ts` and re-running the Mann Kendall test on the non-seasonal Ozone monthly series, the tau value comes out to -0.165 and the p-value is 0.0075402. This p-value is significantly smaller than the value from the seasonal Mann Kendall test in #12 (0.046724), which suggests that there is a significant trend in the time series that is not due to seasonality. Additionally, the tau value is used to interpret the strength of the relationship between the variables (ranges between -1 and 1), and a negative tau value (-0.165) indicates that the variables are inversely related.