# Assignment 7: Time Series Analysis

Emma Wellbaum

## 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.

```r
#1
# Check working directory
getwd()
```

```
## [1] "C:/Users/emmaw/Documents/ENV872/Environmental_Data_Analytics_2021"
```

```r
# Load necessary packages
library(tidyverse)
library(lubridate)
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.0.4
```

```r
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.0.4
```

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

```
theme_set(mytheme)

#2
# Import datasets
O3.10 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
O3.11 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
O3.12 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
O3.13 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
O3.14 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
O3.15 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
O3.16 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
O3.17 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
O3.18 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
O3.19 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

# Combine into a single data frame
GaringerOzone <-
  rbind(O3.10,O3.11,O3.12,O3.13,O3.14,O3.15,O3.16,O3.17,O3.18,O3.19)
```

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

# 4
# Select Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE
GaringerOzone <- select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration,
                        DAILY_AQI_VALUE)

# 5
# Create a new dataframe that contains a continuous daily sequence of dates
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), "days"))
colnames(Days) <- "Date"

# 6
# Join the datasets by "Date"
GaringerOzone <- left_join(Days, GaringerOzone)
```

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

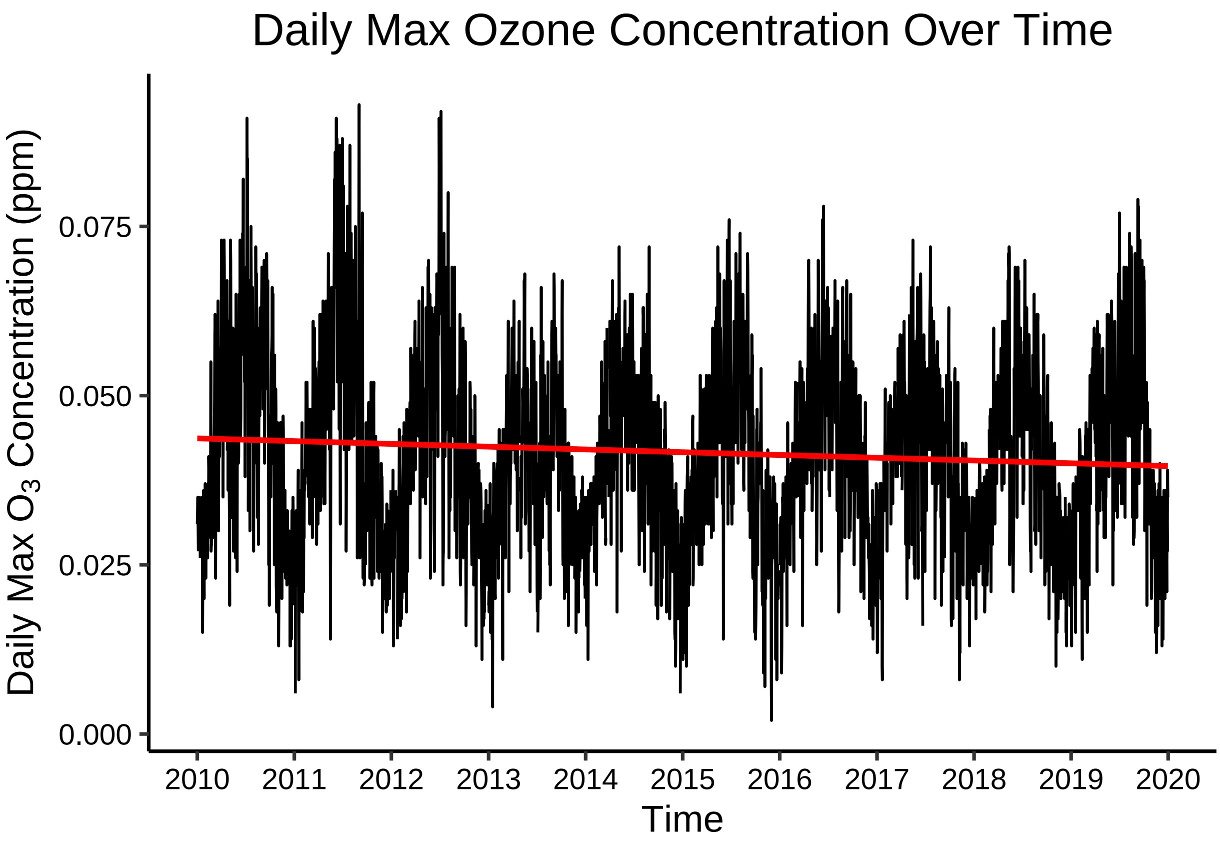
## Visualize

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
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(size = .5, color = "black") +
  geom_smooth(method=lm, se=FALSE, color = "red") +
  scale_x_date(breaks=("1 year"), date_labels = "%Y") +
  labs(x = "Time",
       y = expression("Daily Max O"[3]*" Concentration (ppm)"),
       title = "Daily Max Ozone Concentration Over Time") +
  theme(plot.title = element_text(hjust=0.5))
```

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

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



Answer: The plot suggests a negative linear trend in Ozone concentration, but perhaps not a very strong one. The slope of the trendline is not very steep at all.

## 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
# Examine the data for NA values
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
```

```
# Replace missing data for ozone concentration
GaringerOzone <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

# Check to see that the NA values are gone
summary(GaringerOzone$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: A piecewise constant interpolation would have been less appropriate for our data than a linear interpolation because rather than “connecting the dots” to fill the missing values, each missing value would have been assigned the same value of its “nearest neighbor.” This would likely skew our results, as the missing ozone concentrations could be assigned values observed on earlier or later dates. We did not use a spline interpolation because our trend does not look polynomial whatsoever and a spline interpolation uses a quadratic function rather than a linear function.

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)

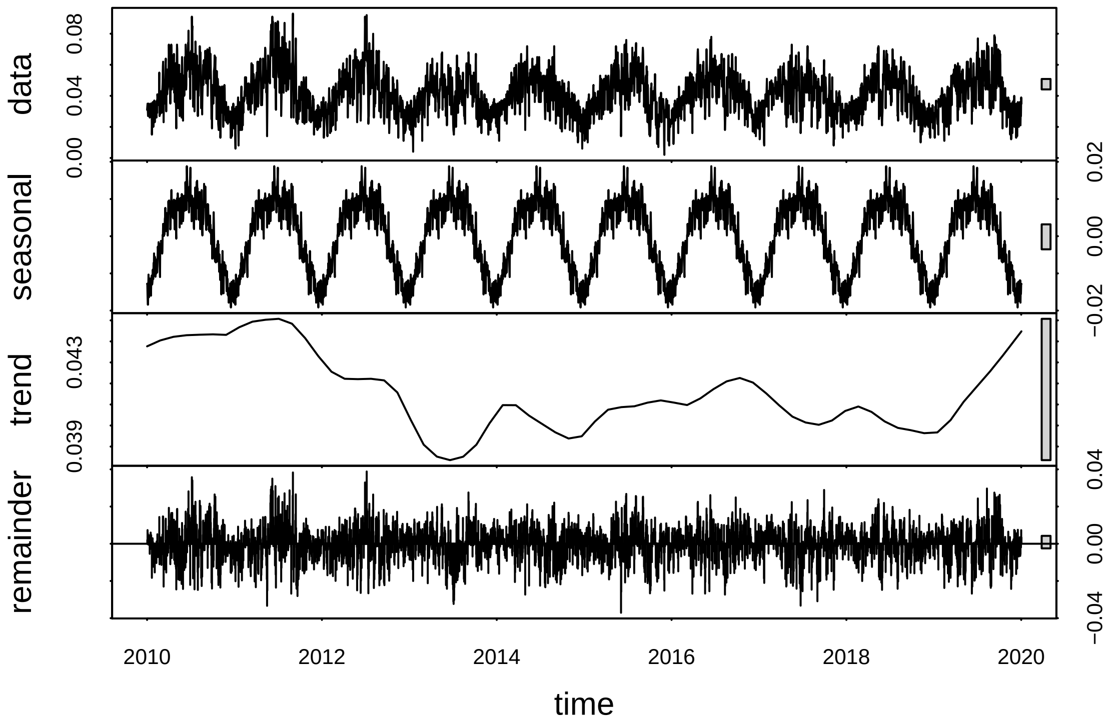
```
#9
GaringerOzone.monthly <-
  GaringerOzone %>%
  mutate(Month = month(Date), Year = year(Date)) %>%
  mutate(Date = dmy(paste0("01-",Month,"-",Year))) %>%
  group_by(Date) %>%
  summarize(Mean.Ozone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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.

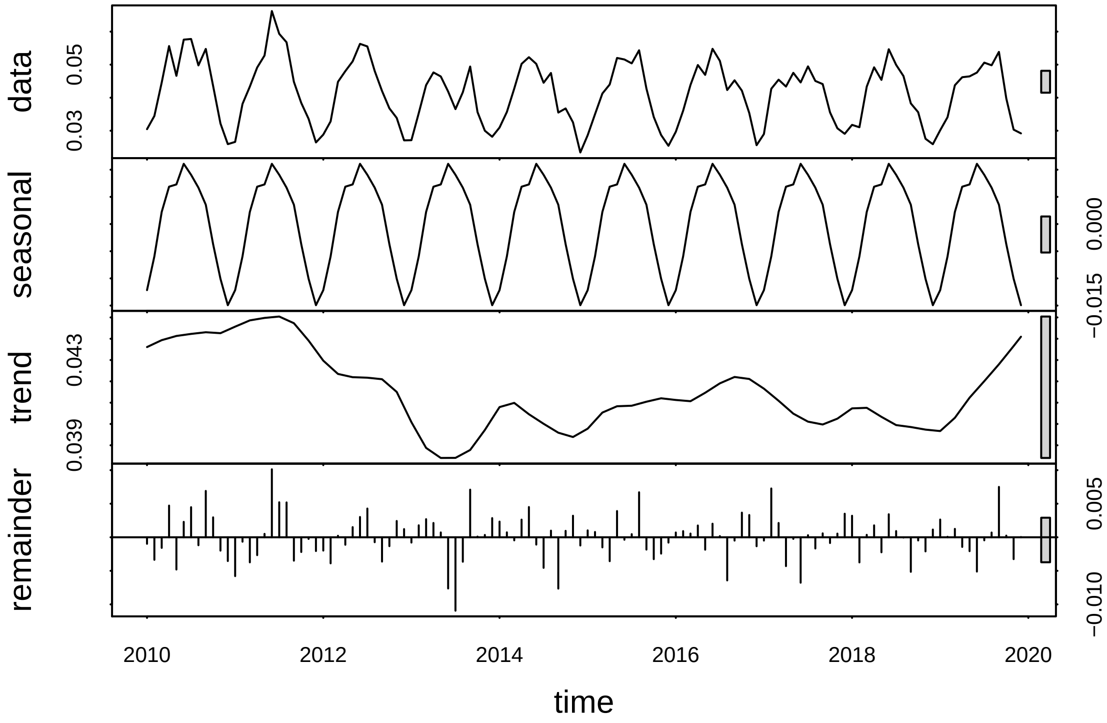
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                   start = c(2010,1),
                   frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Ozone,
                   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
# Decompose the daily time series and plot its components
GaringerOzone.daily_Decomposed <-
  stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily_Decomposed)
```



```
# Decompose the monthly time series and plot its 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
# Run SMK tests
monthly_smk1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
monthly_smk2 <- trend::smk.test(GaringerOzone.monthly.ts)

# Inspect Results
summary(monthly_smk1)
```

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

```
summary(monthly_smk2)
```

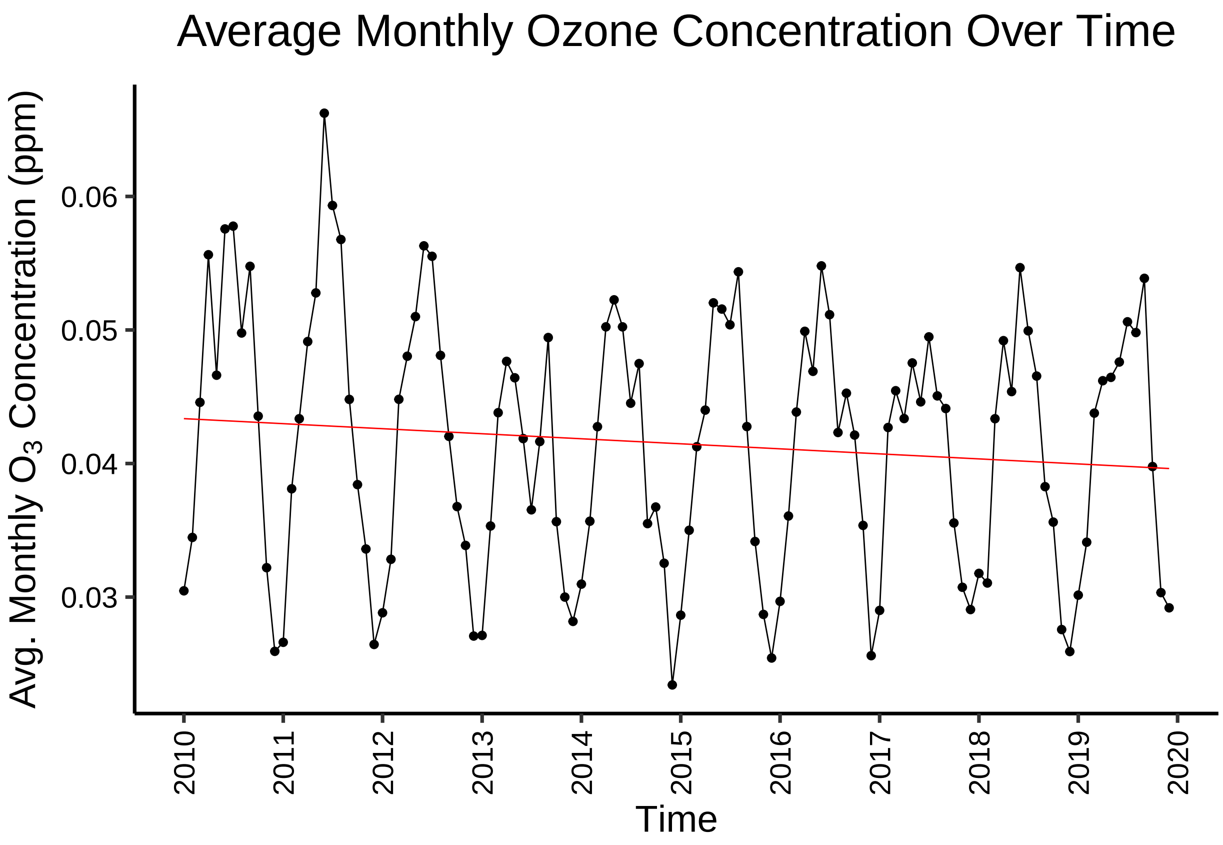
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                        S  varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15   125  0.333   1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 12:   S = 0  11  125  0.244  0.894  0.37109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The monthly data is seasonal. Unless we remove the seasonal component, the only option is the Seasonal 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 = Mean.Ozone)) +
  geom_point(size=1) +
  geom_line(size=0.25) +
  #scale_y_continuous(n.breaks = 9) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(x = "Time", y = expression("Avg. Monthly O"[3]*" Concentration (ppm)"),
       title = expression("Average Monthly Ozone Concentration Over Time")) +
  theme(axis.text.x = element_text(angle=90, hjust=1, vjust=.5),
        plot.title = element_text(hjust=0.5)) +
  geom_smooth(method = lm, se=FALSE, color = "red", size = 0.25)
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.

    Answer: Ozone concentrations have decreased over the 2010s at this station, as indicated by the statistically significant negative trend between monthly ozone concentrations over time (SMK tau=-0.143; p-value=0.046724). This negative trend is relatively subtle based on the SMK test. The tau value is fairly close to zero and the p-value is just under 0.5. While there is a statistically significant trend at this station overall, the month-specific ozone concentrations at the station have not changed significantly over the 2010s (monthly_smk2).

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_Components <-
  as.data.frame(GaringerOzone.monthly_Decomposed$time.series[,1:3]) %>%
  mutate(Observed = GaringerOzone.monthly$Mean.Ozone,
        Date = GaringerOzone.monthly$Date)

GaringerOzone.monthly.noseason.ts <-
  ts(data = Monthly_Components$Observed - Monthly_Components$seasonal,
                       start = c(2010,1),
                       frequency = 12)
#16
```

```
# Run MK test
Noseason_MKtest <- trend::mk.test(GaringerOzone.monthly.noseason.ts)
Noseason_MKtest
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.monthly.noseason.ts
## z = -2.672, n = 120, p-value = 0.00754
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.179000e+03  1.943657e+05 -1.651376e-01
```

> Answer: The results of the Mann-Kendall test performed on our seasonally adjusted data are more statistically significant than the results of our Seasonal Mann-Kendall test on the complete series. First, the p-value from the Mann-Kendall test was more statistically significant than the p-value from the Seasonal Mann-Kendall test (0.0075402 vs. ) The magnitude of tau increased when the seasonal component of our data was removed (-0.165 non-seasonal vs. -0.143 seasonal) and the p-value seasonality is removed (seasonal: -0.143 seasonal vs. 0.046724). Second, the magnitude of tau was greater when the seasonal component was removed (-0.165 vs. -0.143). This indicates that adjusting for the variation in ozone concentration that occurs seasonally may help us to evaluate changes in ozone concentration over time more accurately.