Assignment 7: Time Series Analysis

Kendra Sultzer

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() #looks good
```

```
## [1] "C:/Users/kendr/Documents/Spring2021Classes/ENV872/Environmental_Data_Analytics_2021"
```

```
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
```

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

```
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right",
        legend.box.background = element_rect(color="black"), legend.background = element_blank(),
        plot.title=element_text(hjust = 0.5))
#makes font size bigger, puts legend on right, puts box around legend, centers title in middle
theme_set(mytheme) #setting for all subsequent plots

#2
Garinger_2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
Garinger_2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
Garinger_2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
Garinger_2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
Garinger_2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
Garinger_2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
Garinger_2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
Garinger_2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
Garinger_2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
Garinger_2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")
GaringerOzone <- rbind(Garinger_2010, Garinger_2011, Garinger_2012, Garinger_2013, Garinger_2014,
                       Garinger_2015, Garinger_2016, Garinger_2017, Garinger_2018, Garinger_2019)
#this last one combines these 10 datasets together
```

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(GaringerOzone$Date)
```

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

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

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

```r
# 4
Garinger_subset <-GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

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

# 6
GaringerOzone <- left_join(Days, Garinger_subset) #listing Days first b/c want to keep all these rows
```

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

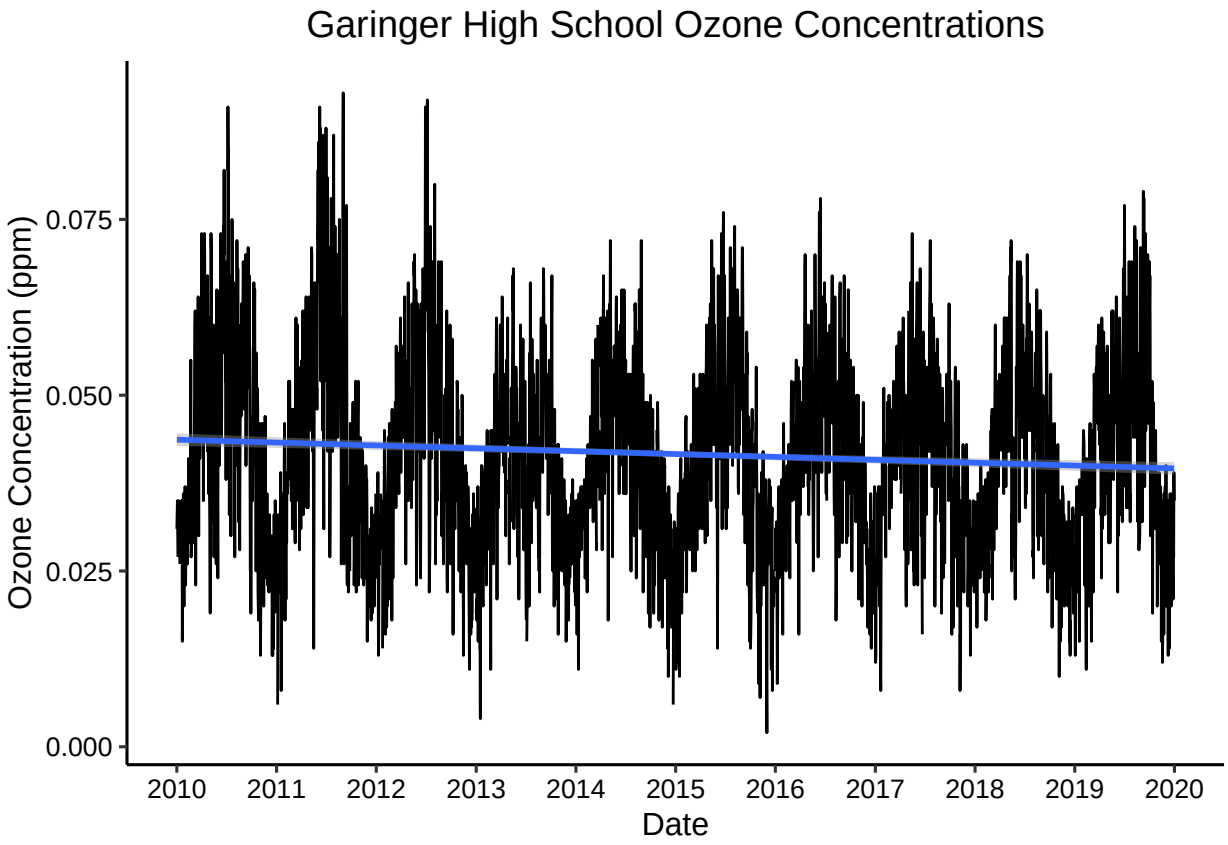
Visualize

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?

```r
#7
Ozone.time <- ggplot(GaringerOzone,aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method="lm")+
  labs(title = "Garinger High School Ozone Concentrations", y="Ozone Concentration (ppm)")+
  scale_x_date(date_labels="%Y", date_breaks = "1 year")
Ozone.time
```

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

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



Answer: This plot does not show a strong trend in ozone concentration over time. The trend appears pretty flat, although there might be a slight decrease over time. However, it appears that there might be a sort of seasonal component within the data.

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(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration) #there are 63 NAs
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- zoo::na.approx(GaringerOzone$Daily.Max.8.hour.Ozo
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration) #no NAs!
```

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

Answer: We didn't use a piecewise constant or spline function because there were NAs directly next to each other. It was better to just use the "connect the dots" approach since ppms are continuous. The spline won't be accurate because it's quadratic so it will probably output values above the expected. For the piecewise, it's difficult to know which value is closest to it. Thus, the linear interpolation is the best bet.

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)

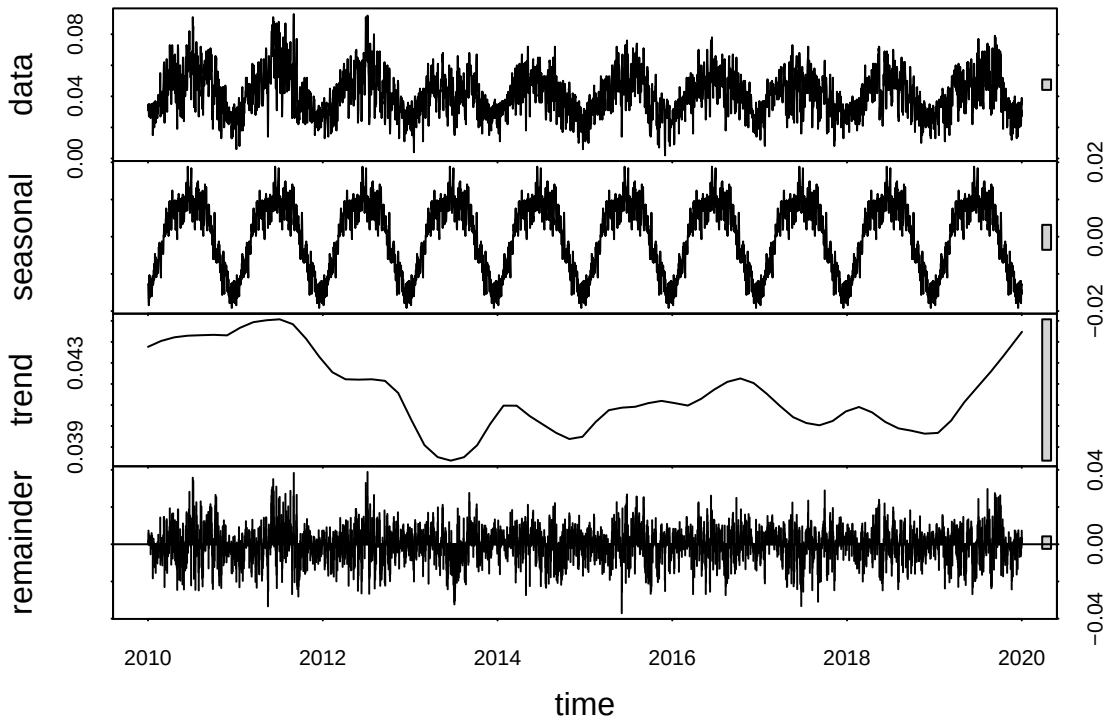
```
#9
GaringerOzone.monthly <- GaringerOzone  %>%
  mutate(Month = as.yearmon(Date, "%Y %m")) %>%
  group_by(Month)%>%
  summarise(Monthly.mean=mean(Daily.Max.8.hour.Ozone.Concentration))

GaringerOzone.monthly$Date <- as.Date(GaringerOzone.monthly$Month, format="%b %Y")
```

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.

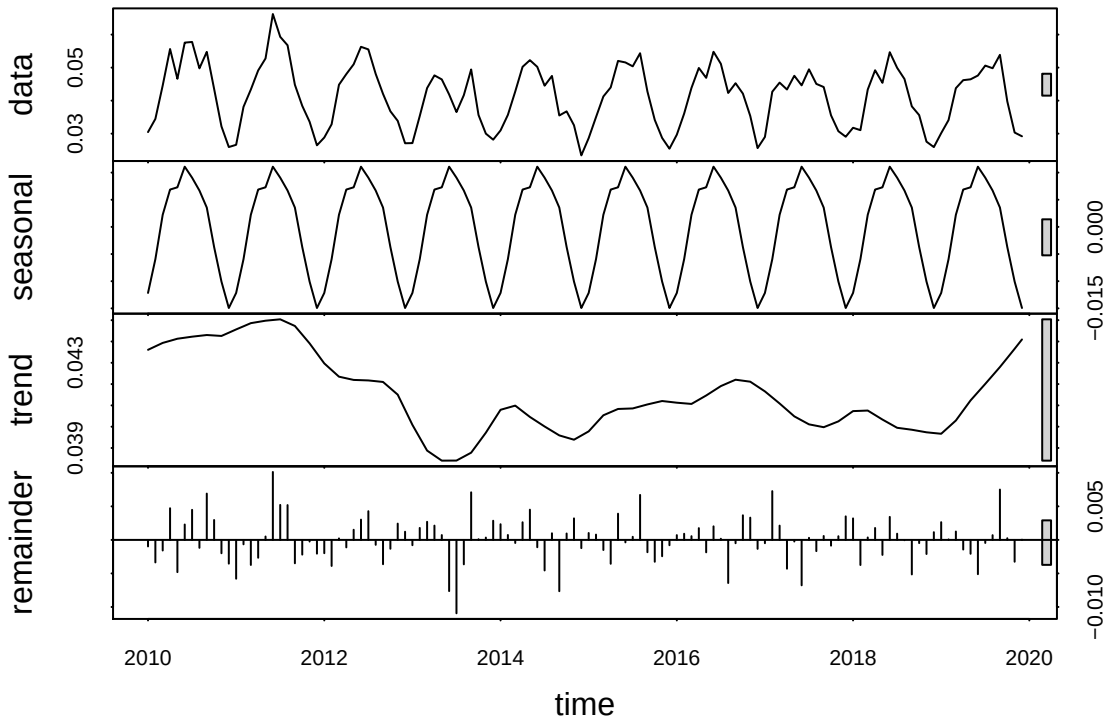
```
#10
f_month <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start=c(f_year,f_month),
                             frequency = 365) #measured daily
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Monthly.mean,
                               start=c(f_year,f_month),
                               frequency = 12) #measured monthly
```

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

```
#11
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(GaringerOzone.daily.decomp)
```




```
GaringerOzone.monthly.decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly.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
#run SMK tests
GaringerOzone.monthly.trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
GaringerOzone.monthly.trend2 <- smk.test(GaringerOzone.monthly.ts)

#inspect results
GaringerOzone.monthly.trend1
```

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

```
GaringerOzone.monthly.trend2 #p-value is small so there is a trend (data is not stationary)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

```
summary(GaringerOzone.monthly.trend1)
```

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

```
summary(GaringerOzone.monthly.trend2)
```

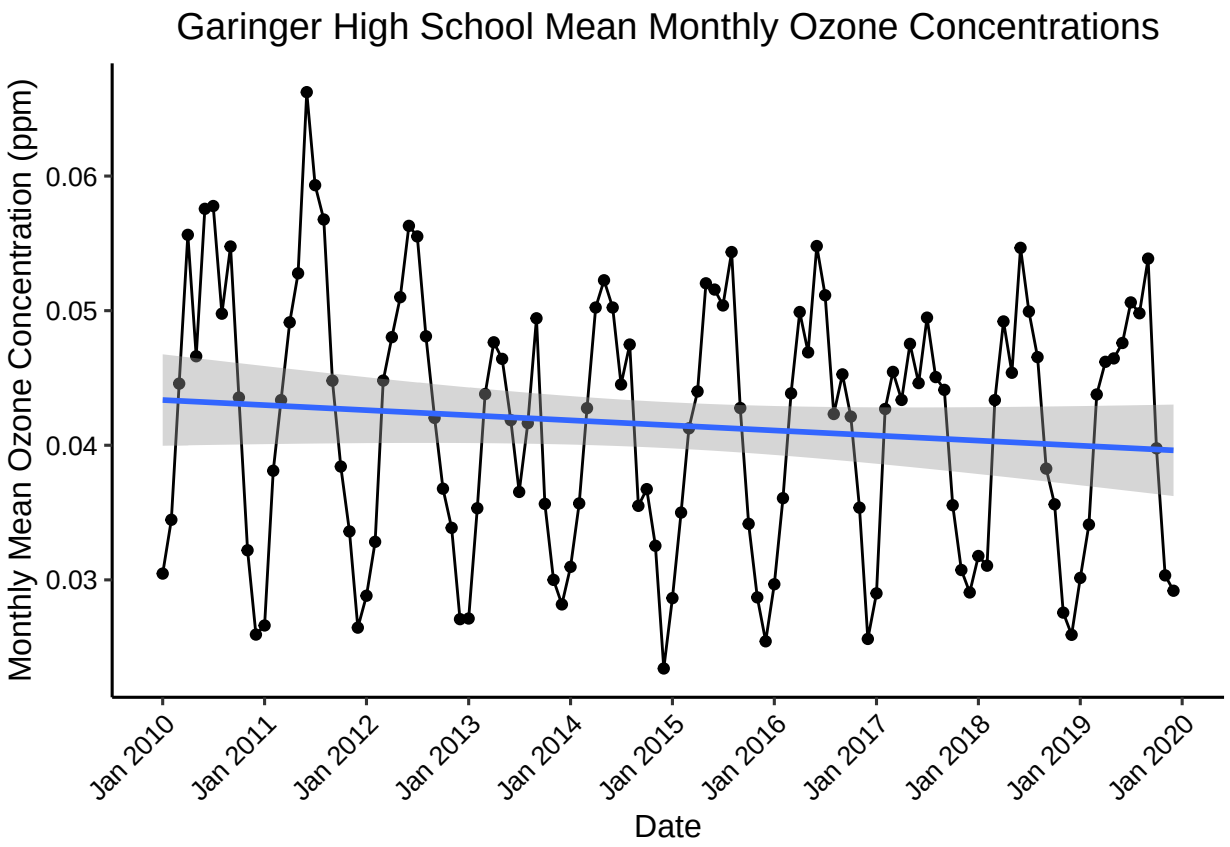
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                      S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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 seasonal Mann-Kendall is the most appropriate trend analysis here since there's an obvious seasonal component to the data (as seen in the plotting of the components). If there wasn't a seasonal component, we would consider one of the other tests like linear regression, regular Mann-Kendall, or Spearman Rho.

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
Mean.month.ozone.plot <-
  ggplot(GaringerOzone.monthly, aes(x=Date, y = Monthly.mean))+
  geom_line()+
  geom_point()+
  labs(title="Garinger High School Mean Monthly Ozone Concentrations",
       y="Monthly Mean Ozone Concentration (ppm)")+
  scale_x_date(date_labels = "%b %Y", date_breaks = "1 year")+
  theme(axis.text.x = element_text(angle = 45, hjust=1))+
  geom_smooth(method="lm")
Mean.month.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: From our analysis, we can say that ozone concentrations have not changed significantly over the 2010s at Garinger High School in North Carolina (p=0.05). As the date increases, the mean ozone concentrations have not increased or decreased. Even though there appears to be a seasonal trend, the results of the seasonal Mann-Kendall test do not support this.

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.ozone.components <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])
#extracting seasonal, trend, and remainder from time series

Monthly.ozone.components <- mutate(Monthly.ozone.components,
                                   Observed=GaringerOzone.monthly$Monthly.mean,
                                   Date=GaringerOzone.monthly$Month)
#adding in observed data and month
```

```
#remove seasonality (subtract seasonal values from observed values)
ozone.ts.noseason <- ts(data=Monthly.ozone.components$Observed-Monthly.ozone.components$seasonal,
                        start=c(f_year,f_month),
                        frequency=12)


#16
#apply Mann Kendall
MK_ozone <- trend::mk.test(ozone.ts.noseason)
MK_ozone
```

```
##
##  Mann-Kendall trend test
##
## data:  ozone.ts.noseason
## 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 data without the seasonal component is now showing a significant trend ($p = 0.007$) according to the Mann Kendall test. Thus, without the seasonal component, ozone concentrations have changed over the 2010s at Garinger High School in North Carolina. As the date has increased, the monthly mean ozone concentrations have decreased.