Assignment 8: Time Series Analysis

Taylor Coleman

Spring 2023

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
#1 Loading packages, checking working directory, and setting ggplot theme
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- 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)
library(here)
```

```
## here() starts at /home/guest/ENVR872-RepositoryClone
```

```
here()
```

```
## [1] "/home/guest/ENVR872-RepositoryClone"
```

```
mytheme <- theme_classic(base_size = 10) +
  theme(axis.text = element_text(color = "black"),
        plot.title = element_text(hjust = 0.5),
        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 observations and 20 variables.

```
#2 Data import
Garinger2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                         stringsAsFactors = T)
Garinger2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                         stringsAsFactors = T)
Garinger2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                         stringsAsFactors = T)
Garinger2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                         stringsAsFactors = T)
Garinger2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                         stringsAsFactors = T)
Garinger2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                         stringsAsFactors = T)
Garinger2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                         stringsAsFactors = T)
Garinger2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                         stringsAsFactors = T)
```

```
Garinger2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                         stringsAsFactors = T)
Garinger2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                         stringsAsFactors = T)

GaringerOzone <- rbind(Garinger2010, Garinger2011, Garinger2012,
                         Garinger2013, Garinger2014, Garinger2015,
                         Garinger2016, Garinger2017, Garinger2018,
                         Garinger2019)
```

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

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

#5 Generating a daily dataset
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "1 day"))
colnames(Days) <- "Date"

#6 Merging new daily dataset with GaringerOzone
GaringerOzone <- left_join(Days, GaringerOzone)
```

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

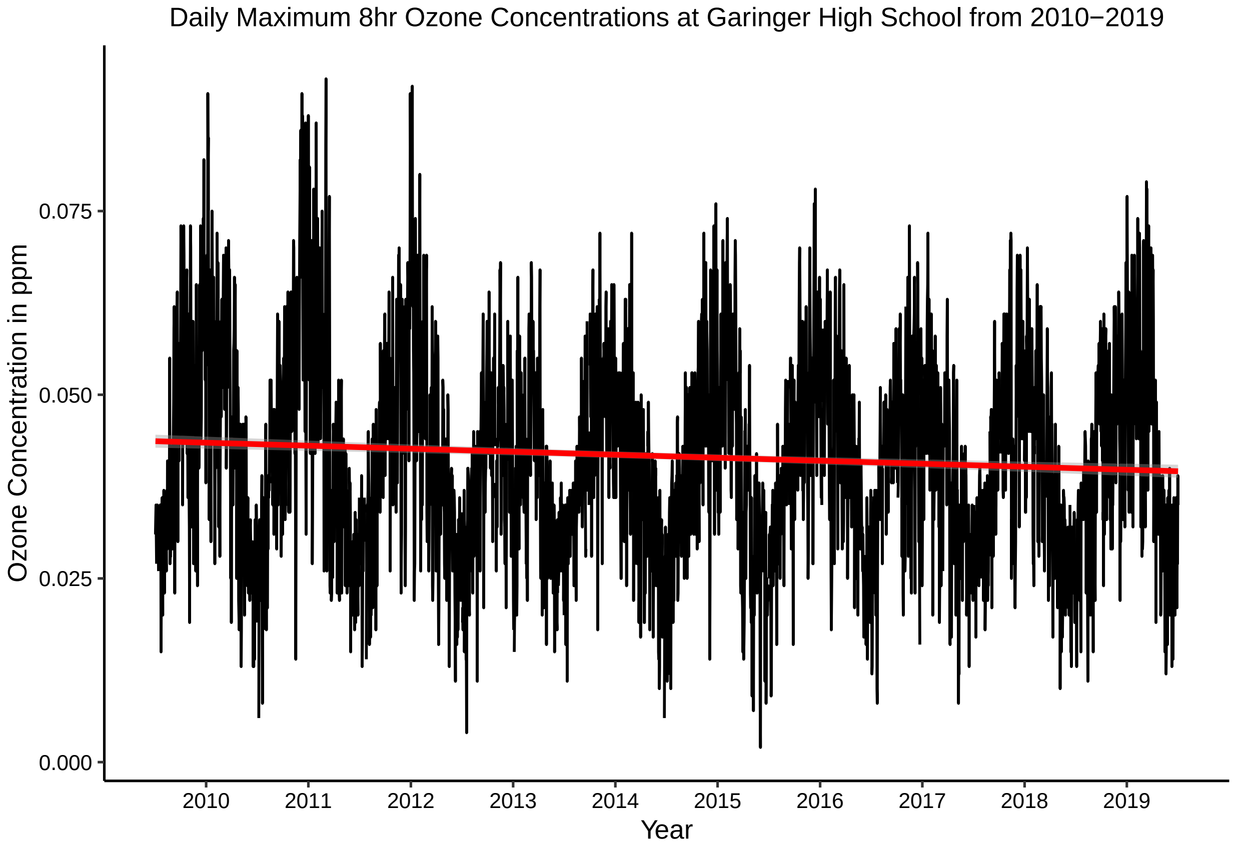
Visualize

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 Plotting O3 concentrations over time
ggplot(GaringerOzone,
       aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  theme(plot.title = element_text(size = 10)) +
  geom_line() +
  labs(title = "Daily Maximum 8hr Ozone Concentrations at Garinger High School from 2010-2019",
       y = "Ozone Concentration in ppm",
       x = "Year") +
  scale_x_date(breaks = "12 months", date_labels = "%Y") +
  geom_smooth(method = "lm", color = "red")
```

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

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



Answer: My plot suggests that there is a slight decreasing trend in ozone concentrations at Garinger High School over time (and potential 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 Linear interpolation
GaringerOzone_clean <- GaringerOzone %>%
  mutate( Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )

summary(GaringerOzone_clean)
```

```
##       Date             Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04151                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                                                            NA’s   :63
```

> Answer: We used a linear interpolation to fill in the missing daily data for ozone concentrations because our missing data values fall between days that do have data and, considering the linear relationship or monotonic/deterministic trend that daily maximum 8hr ozone concentrations seem to have over time, these days can be used to "connect the dots" for those days that lack data. There is no need to apply the spline interpolation method because the relationship does not appear to be quadratic, and the piecewise constant interpolation method is inappropriate because there is no "nearest neighbor" when each date is the same "distance" away from each other (one day).

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 New data frame for aggregated mean O3 concentrations per month
GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(Month = month(ymd(Date)),
         Year = year(ymd(Date))) %>%
  aggregate(Daily.Max.8.hour.Ozone.Concentration ~ Month + Year, mean) %>%
  rename(Monthly.Mean.Ozone.Concentration = Daily.Max.8.hour.Ozone.Concentration)

# Creation of new Date column with 1st day of each month as month-year combinations
GaringerOzone.monthly$Date<-as.Date(with(GaringerOzone.monthly,paste(Year, Month, "01",
                                                                     sep = "-")),
                                    "%Y-%m-%d")
```

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.

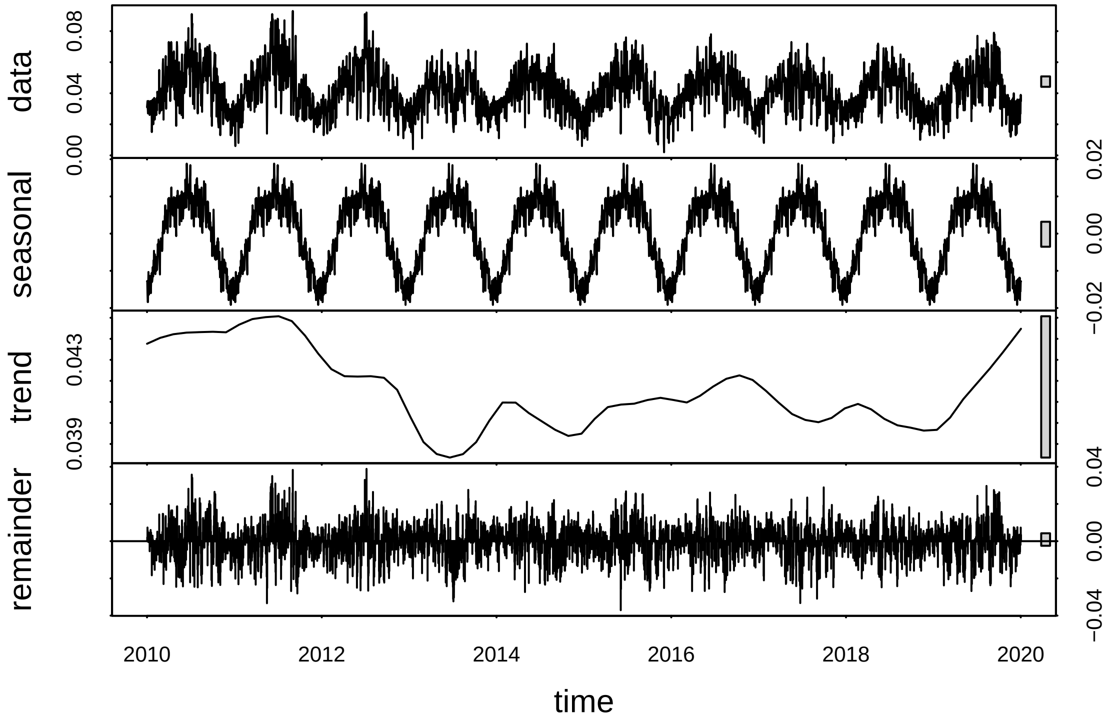
```
#10 Creating 2 time series objects from daily and monthly average ozone observations
f_month <- month(first(GaringerOzone_clean$Date))
f_year <- year(first(GaringerOzone_clean$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration,
                   start = c(f_year,f_month),
                   frequency = 365)
```

```
# Repetitive to redefine f_month and f_year, but wanted to repeat steps to check that values
# match with those from GaringerOzone_clean dataframe
f_month <- month(first(GaringerOzone.monthly$Date))
f_year <- year(first(GaringerOzone.monthly$Date))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Monthly.Mean.Ozone.Concentration,
                               start = c(f_year, f_month),
                               frequency = 12)
```

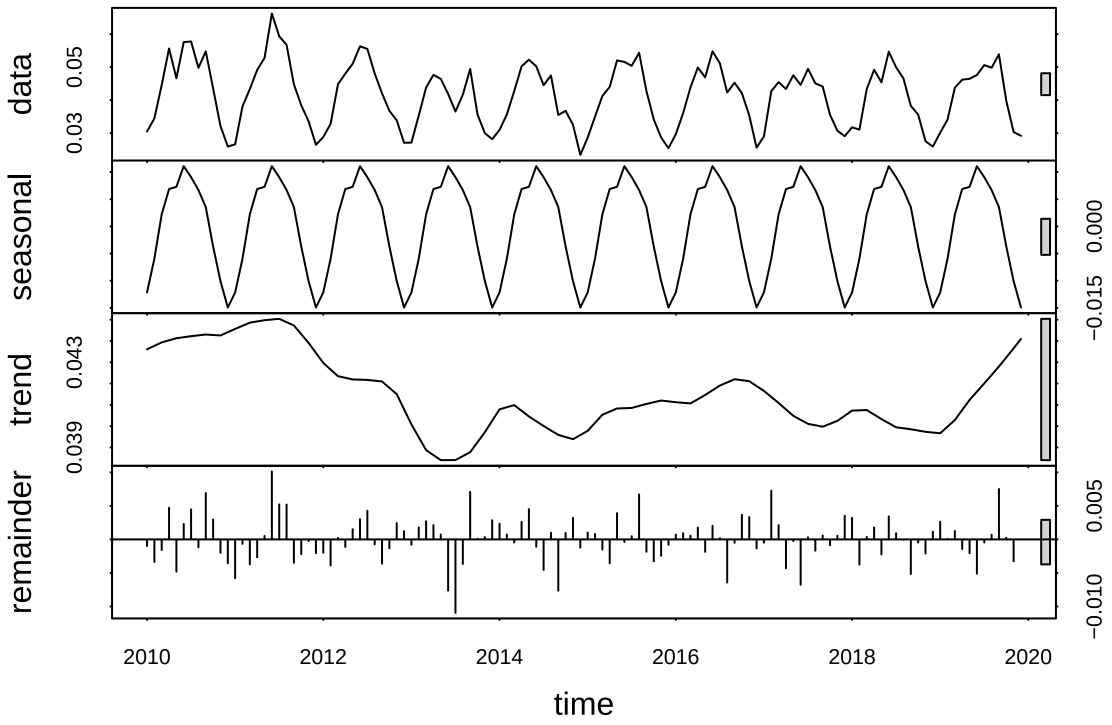
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11 Decomposing daily and monthly O3 time series objects
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

# Plotting components for each decomposed time series object
plot(GaringerOzone.daily.decomposed)
```



```
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 Running a seasonal Mann-Kendall test
GaringerOzone.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Analyzing results
GaringerOzone.monthly.trend
```

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

```
summary(GaringerOzone.monthly.trend)
```

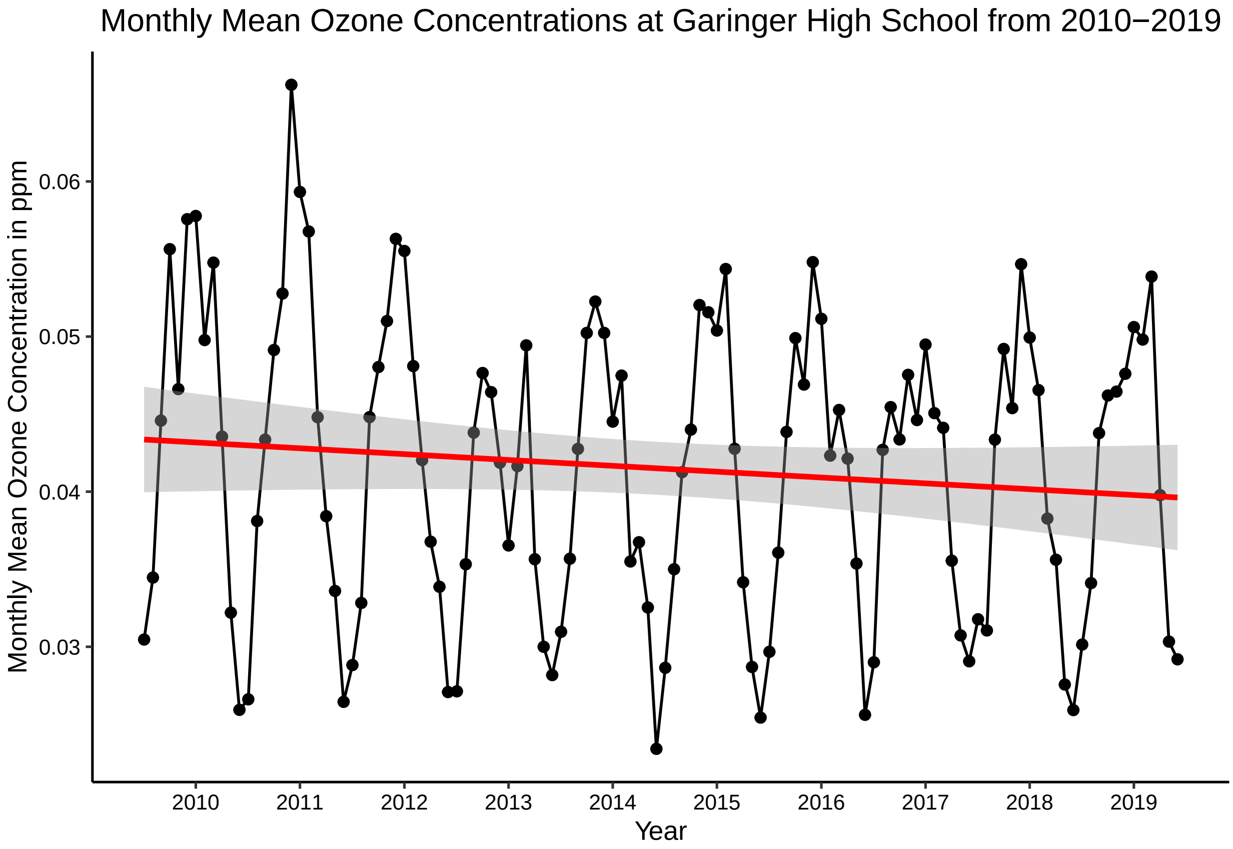
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann-Kendall test is most appropriate because there is seasonality in the ozone daily and monthly average concentrations data and the seasonal Mann-Kendall test is the only monotronic trend analysis test that can be used for seasonal data.

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 Plotting mean monthly O3 concentrations over time
ggplot(GaringerOzone.monthly,
       aes(x = Date, y = Monthly.Mean.Ozone.Concentration)) +
  geom_point() +
  geom_line() +
  labs(title = "Monthly Mean Ozone Concentrations at Garinger High School from 2010-2019",
       x = "Year",
       y = "Monthly Mean Ozone Concentration in ppm") +
  scale_x_date(breaks = "12 months", date_labels = "%Y") +
  geom_smooth(method = "lm", color = "red")
```

```
## `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: Results from the seasonal Mann-Kendall test reported a 2-sided p-value less than the significance level of 0.05 as well as a very small, negative score, suggesting a statistically significant decreasing trend in monthly average ozone concentrations over time (p = 0.046724, S = -77). Therefore, we can reject the null hypothesis that ozone concentrations are independent and identically distributed, in favor of the alternative hypothesis that states that ozone concentrations follow a trend. In the context of our research question, ozone concentrations have changed (decreased) over the 2010s at the Garinger High School station!

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 from the Seasonal Mann-Kendall on the complete series.

```
#15 Creating new dataframe: Subtracting seasonal component from mean monthly O3 concentrations
GaringerOzone.monthly.nonseasonal <- as.data.frame(GaringerOzone.monthly.decomposed$time.series[, 2:3])

# Merging "trend" and "remainder" columns
GaringerOzone.monthly.nonseasonal <- unite(GaringerOzone.monthly.nonseasonal, components,
                                           c(trend, remainder))

# Adding date and mean ozone columns to the dataframe
GaringerOzone.monthly.nonseasonal <- mutate(GaringerOzone.monthly.nonseasonal,
         Observed = GaringerOzone.monthly$Monthly.Mean.Ozone.Concentration,
         Date = GaringerOzone.monthly$Date)

# Converting GaringerOzone.monthly.nonseasonal to new time series object
GaringerOzone.monthly.nonseasonal.ts <- ts(GaringerOzone.monthly.nonseasonal$Observed,
                              start = c(2010, 1),
                              frequency = 12)

#16 Running Mann-Kendall test (non-seasonal)!
GaringerOzone.monthly.trend2 <- Kendall::MannKendall(GaringerOzone.monthly.nonseasonal.ts)
GaringerOzone.monthly.trend3 <- Kendall::MannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.trend3)
```

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

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

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

```
# Note to self: These report the exact same score and 2-sided p-value!
# This makes sense because seasonality is not taken into account by the MK test -
# only the seasonal MK test can do this.
# Therefore, inappropriately running the Mann-Kendall test on GaringerOzone.monthly.ts
# produces the same results as running the Mann-Kendall test on
# GaringerOzone.monthly.nonseasonal.ts. This is my hypothesis at least!
```

> Answer: Running the Mann-Kendall test on the nonseasonal version of the monthly mean ozone concentration time series results in a slightly larger negative number for the reported score than the seasonal Mann-Kendall due to its smaller number of observations given the data's monthly aggregation. Simply considering this score suggests a decreasing trend over time for average ozone concentrations at Garinger High School. However, because the Mann-Kendall test also reports a 2-sided p-value that is greater than the significance level of 0.05, we ultimately cannot conclude that there is a trend in ozone concentrations as we fail to reject the null hypothesis that ozone is independent and identically distributed (p = 0.33732). It would appear that removing the seasonal component from the GaringerOzone data frame results in removal of the trend.