# Assignment 8: Time Series Analysis

Jenn McNeill

Fall 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
# Load packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
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 /Users/jennifermcneill/EDE_Fall2023/EDE_Fall2023
```

```
library(ggthemes)

# Check working directory
getwd()
```

```
## [1] "/Users/jennifermcneill/EDE_Fall2023/EDE_Fall2023"
```

```
here()
```

```
## [1] "/Users/jennifermcneill/EDE_Fall2023/EDE_Fall2023"
```

```
jenn_default_theme <- theme_classic() +
  theme(
    # Customize the plot background, title text, and axis title text
    panel.background = element_rect(fill = "grey97"),
    plot.title = element_text(size = 14,
                              face = "bold",
                              color = "violetred4"),
    axis.title.x = element_text(size = 10,
                                color = "tomato3"),
    axis.title.y = element_text(size = 10,
                                color = "tomato3"),

    # Customize the axis ticks and gridlines
    axis.text = element_text(size = 8, color = "grey44"),
    axis.line = element_line(color = "grey44"),
    panel.grid.major = element_line(color = "grey44",
                                    linetype = "dotted"),

    # Customize the legend
    legend.position = "right",
    legend.title = element_text(size = 12,
                                face = "bold",
                                color = "darkolivegreen"),
    legend.text = element_text(size = 10,
                               color = "darkolivegreen"),
    line = element_line(0.5))

# Set as default theme
theme_set(jenn_default_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.

```
#2

# Import csv
GaringerNC2010 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2011 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2012 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2013 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2014 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2015 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2016 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2017 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2018 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
  stringsAsFactors = TRUE)
GaringerNC2019 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
  stringsAsFactors = TRUE)

# Combine into a single dataframe
GaringerOzone <- rbind(GaringerNC2010,GaringerNC2011,GaringerNC2012,
                       GaringerNC2013,GaringerNC2014,GaringerNC2015,
                       GaringerNC2016,GaringerNC2017,GaringerNC2018,
                       GaringerNC2019)
```

## 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
GaringerOzone$Date <- mdy(GaringerOzone$Date)

# 4
GaringerOzone_Wrangled <- 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"), 1))
colnames(Days) <- "Date"

# 6
GaringerOzone <- Days %>%
  left_join(GaringerOzone_Wrangled, by = "Date")
```

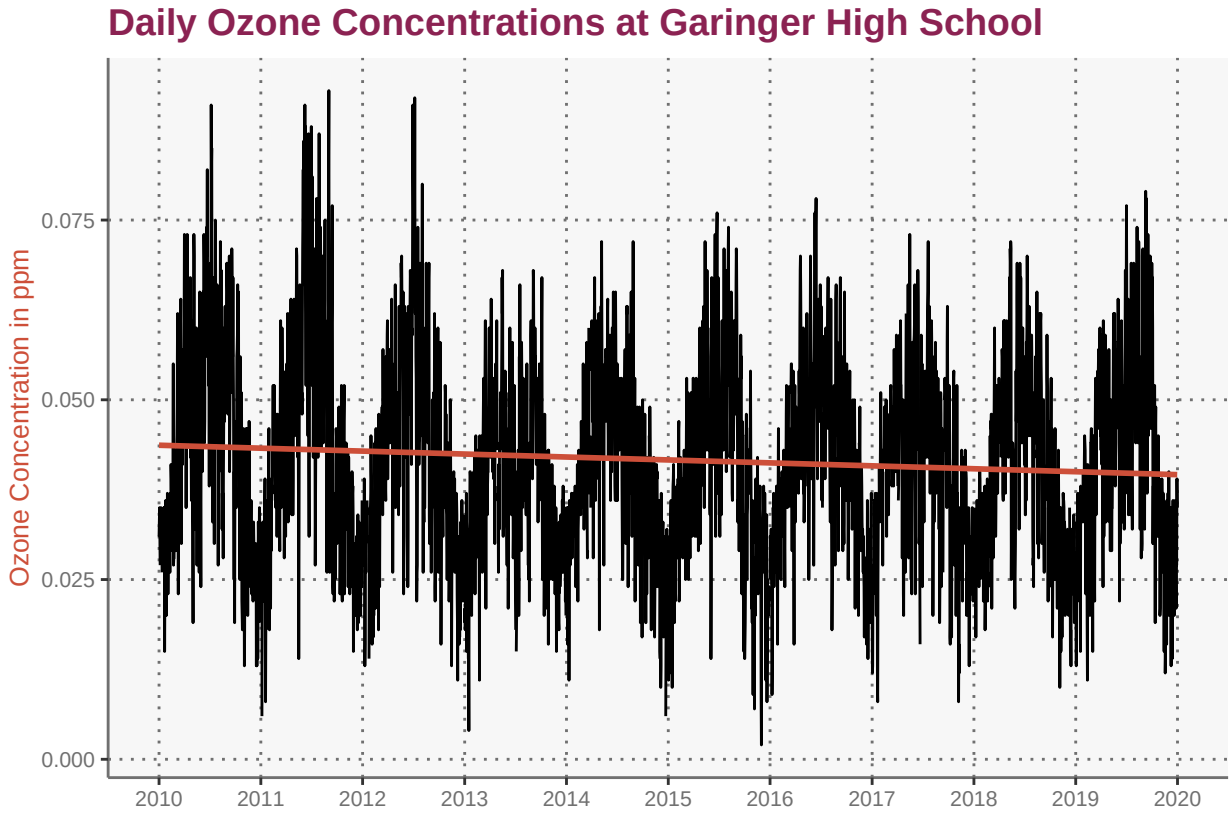
## Visualize

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() +
  geom_smooth(method = lm, se = FALSE, color = "tomato3") +
  xlab("") +
  ylab("Ozone Concentration in ppm") +
  ggtitle("Daily Ozone Concentrations at Garinger High School") +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year")
```

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

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




Answer: The trend line suggests that there is a slight downward trend in ozone concentration at Garinger High School from the beginning to the end of the 2010 decade. The daily data shows that there are yearly fluctuations that peak in the summer and dip in the winter.

## 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
GaringerOzone_Clean <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: We used a linear interpolation because we wanted to fill in missing data using a "connect the dots" approach that assumes missing values fall between the previous and next measurements. We did not use a piecewise constant because this would fill in all missing data with whichever measurement were made on the nearest date. We did not use spline interpolation because this would fill in missing data using a quadratic function between neighboring data points instead of a straight line like we used in our linear interpolation.

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)

```r
#9
GaringerOzone.monthly <-
  GaringerOzone_Clean %>%
  # Add a month column
  mutate(month = month(Date)) %>%
  # Add a year column
  mutate(year = year(Date)) %>%
  # Group data for each month of a given year
  group_by(year,month) %>%
  # Sum the data for each month of a given year
  summarize(aggregate_concentration = mean(Daily.Max.8.hour.Clean)) %>%
  # Add a column that gives each month-year combination as the first of the month
  mutate(first_of_month = as.Date(paste(year, month, "01", sep = "-")))
```

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

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.

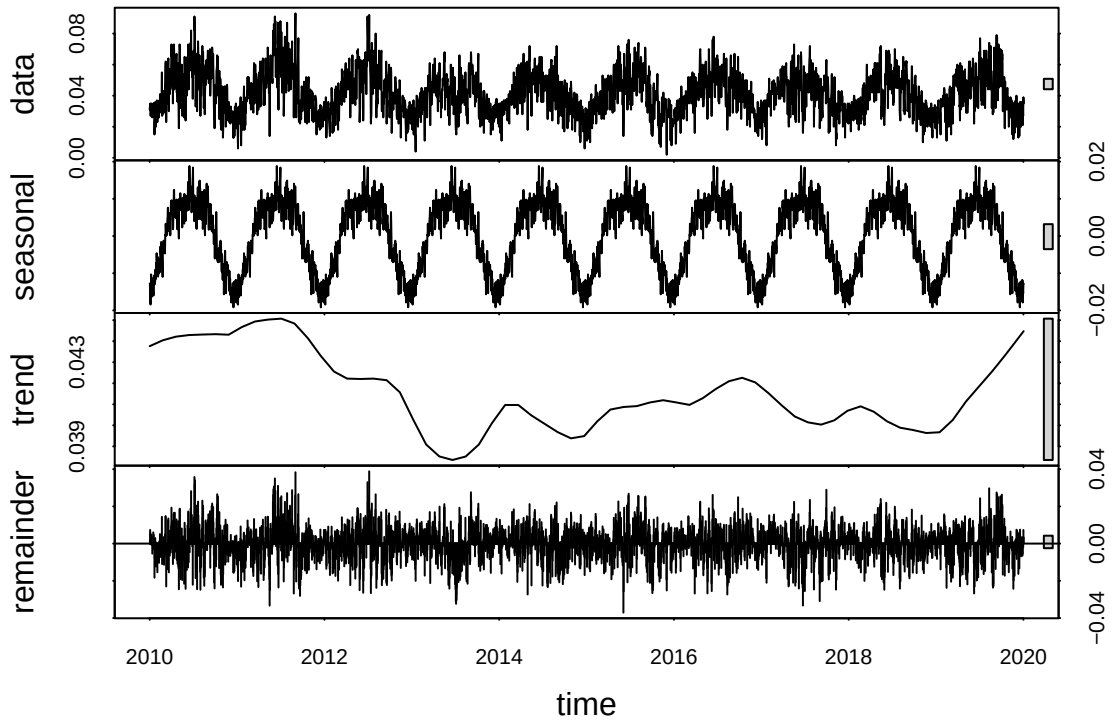
```r
#10
GaringerOzone.daily.ts <- ts(GaringerOzone_Clean$Daily.Max.8.hour.Clean,
                             frequency = 365,
                             start = c(2010,01))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$aggregate_concentration,
                               frequency = 12,
                               start = 2010)
```

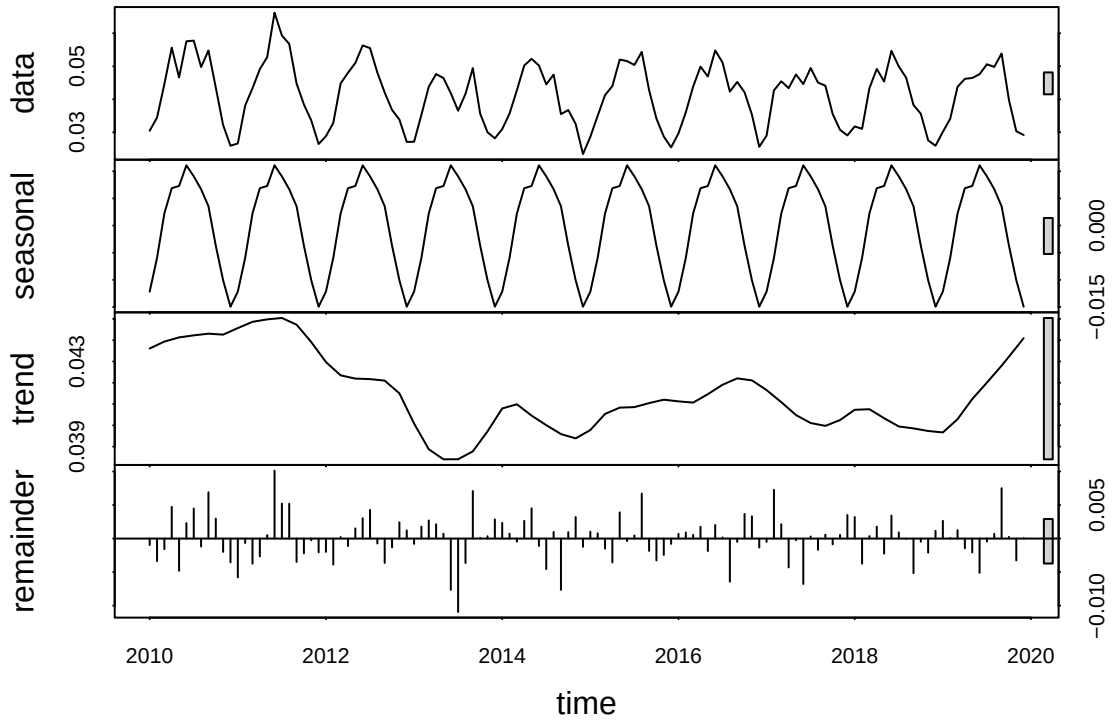
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

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

# Plot the decomposed components
plot(GaringerOzone.daily.ts.decomposed)
```




```
plot(GaringerOzone.monthly.ts.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 test
Ozone_Monthly_SMK <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
Ozone_Monthly_SMK
```

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

```
summary(Ozone_Monthly_SMK)
```

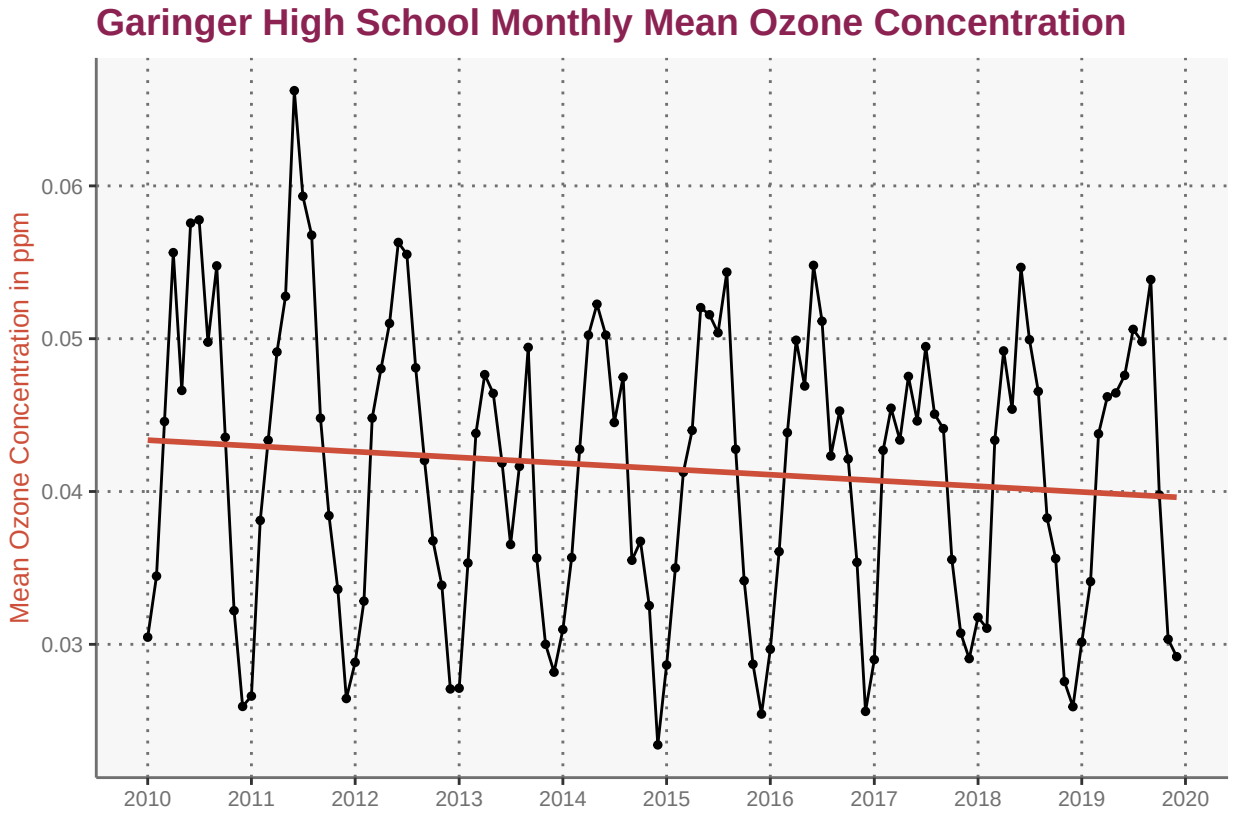
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The plots that I produced in #11 showed a strong seasonal component in the monthly time series, so it is best to start the trend analysis using a test that factors in seasonality. None of the other three tests (Linear Regression, Mann-Kendall, Spearnan Rho) have this feature.

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
ggplot(GaringerOzone.monthly,
       aes(x = first_of_month,
           y = aggregate_concentration)) +
  geom_point(size = 1) +
  geom_line() +
  ylab("Mean Ozone Concentration in ppm") +
  xlab("") +
  ggtitle("Garinger High School Monthly Mean Ozone Concentration") +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
  geom_smooth(method = lm, se = FALSE, color = "tomato3")
```

```
## `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: Considering the research question, "Have ozone concentrations changed over the 2010s at this station?," we can discern from the graph that ozone concentrations have experienced a steady downward trend over the 2010 decade. Fitting the data with a linear regression model gives us this trendline. Our statistical Seasonal Mann-Kendall test corroborates this information because it provides a tau value of -0.143. This tau value being below zero represents a monotonic downward trend over time. The pvalue of 0.046724, less than 0.05, shows that the results from this test are statistically significant and we should explore running the test with the seasonal component removed.

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
# Separate the seasonal, trend, and remainder from the decomposed ts
GaringerOzone.monthly.components <- as.data.frame(
  GaringerOzone.monthly.ts.decomposed$time.series[,1:3])

# Create a new df that includes seasonal, trend, remainder,
# aggregate, date, and the new aggregate after subtracting seasonal
GaringerOzone.monthly.without.seasonality <-
```

```
  mutate(GaringerOzone.monthly.components,
         Observed_Aggregate = GaringerOzone.monthly$aggregate_concentration,
         Date = GaringerOzone.monthly$first_of_month,
         Observed_Aggregate_Without_Seasonality = Observed_Aggregate - seasonal)

# Create a new time series object of the aggregate values without seasonality
GaringerOzone.monthly.ts.without.seasonality <- ts(
  GaringerOzone.monthly.without.seasonality$Observed_Aggregate_Without_Seasonal,
                                          frequency = 12,
                                          start = 2010)


#16
# Run MK test
Ozone_Monthly_MK <- Kendall::MannKendall(
  GaringerOzone.monthly.ts.without.seasonality)

# Inspect results
Ozone_Monthly_MK
```

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

```
summary(Ozone_Monthly_MK)
```

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

```
# Compare with the results from the SMK test from before
summary(Ozone_Monthly_SMK)
```

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

Answer: The non-seasonal and seasonal tau values are -0.165 and -0.143, respectively. Both of the tau values being negative proves what we discerned from our trend line: there is a monotonic downward trend of ozone concentration at Garinger High School over the 2010 decade. When we remove seasonality from the data, there is a stronger downward trend of ozone concentration because we are blocking out the noise of seasonal changes that contribute to ozone level changes. The p value for the non-seasonal Mann-Kendall test is 0.0075402, which is much closer to 0 than the p value for the seasonal Mann-Kendall test. This shows that there is very little evidence against the null hypothesis that ozone concentration does not change with time, and the results of the test are statistically significant.