Assignment 8: Time Series Analysis

Jingze Dai

Fall 2024

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

```
# checking working directory
getwd()
```

```
## [1] "/home/guest/ENV872/EDE_Fall2024"
```

```
# loading libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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(trend)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(here)
```

```
## here() starts at /home/guest/ENV872/EDE_Fall2024
```

```
here()
```

```
## [1] "/home/guest/ENV872/EDE_Fall2024"
```

```
# setting theme
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        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 observation and 20 variables.

```
# 2 importing datasets
ozone2010 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)
ozone2011 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)
ozone2012 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
  stringsAsFactors = TRUE)
ozone2013 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
  stringsAsFactors = TRUE)
ozone2014 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
  stringsAsFactors = TRUE)
ozone2015 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
  stringsAsFactors = TRUE)
ozone2016 <- read.csv(here(
```

```
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
  stringsAsFactors = TRUE)
ozone2017 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
  stringsAsFactors = TRUE)
ozone2018 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
  stringsAsFactors = TRUE)
ozone2019 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
  stringsAsFactors = TRUE)

# combining dataset
GaringerOzone <- rbind(ozone2010,ozone2011,ozone2012,ozone2013,ozone2014,
                       ozone2015,ozone2016,ozone2017,ozone2018,ozone2019)

# checking dimensions to confirm it is 3589 x 20
dim(GaringerOzone)
```

```
## [1] 3589   20
```

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 to date class
GaringerOzone$Date <- mdy(GaringerOzone$Date)

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

# 5 creating a dataframe named "Days"
Days <- as.data.frame(seq(as.Date("2010-01-01"),
                          as.Date("2019-12-31"), by = "day"))
colnames(Days) <- "Date"

# 6 combining using left_join
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")
```

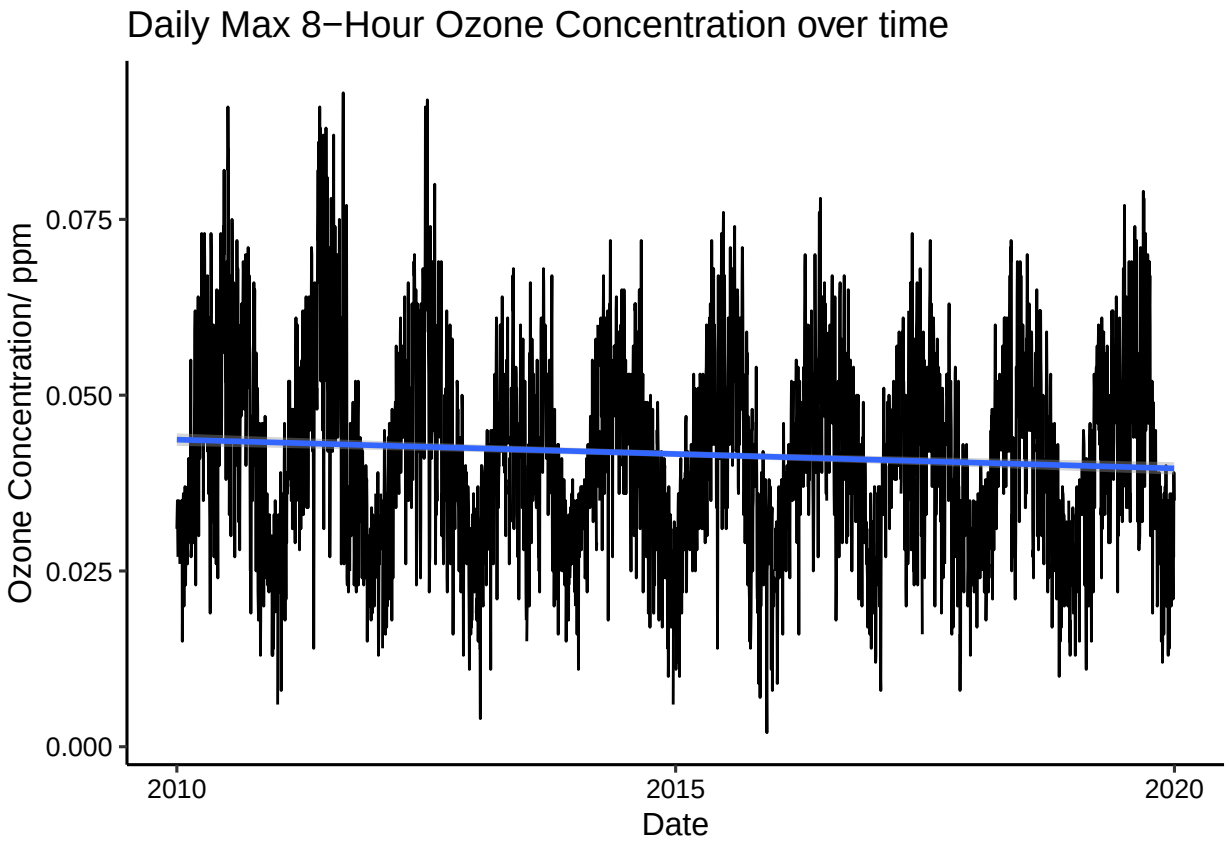
Visualize

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 visualization
ggplot(GaringerOzone, aes(x = Date,
                          y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  labs(title = "Daily Max 8-Hour Ozone Concentration over time",
           x = "Date", y = "Ozone Concentration/ ppm")+
  geom_smooth( method = lm )
```

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

```
## Warning: Removed 63 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



Answer: The trend line's slope is slightly going downwards, suggesting a slight decrease in ozone concentration over time. Moreover, we can observe that the peaks of cycles after the year 2013 decreased compared to peaks of cycles before 2013. However, the peaks after 2013 seems to be constant.

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

Answer: From the plot, we can observe that there are sinusoidal seasonal changes. In this case, we want to use linear interpolation because we want to preserve the seasonality by fitting the NA using a linear model based on its neighbor points. We do not want a piecewise constant because it simple carry forward the last observed number and will flatten the trend. We also do not want to use spline interpolation because it can overfit and add noise to the trend.

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 creating monthly mean concentration
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(Mean_Ozone_Concentration =
              mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = as.Date(paste(Year, Month, "01", sep = "-")))

GaringerOzone.monthly
```

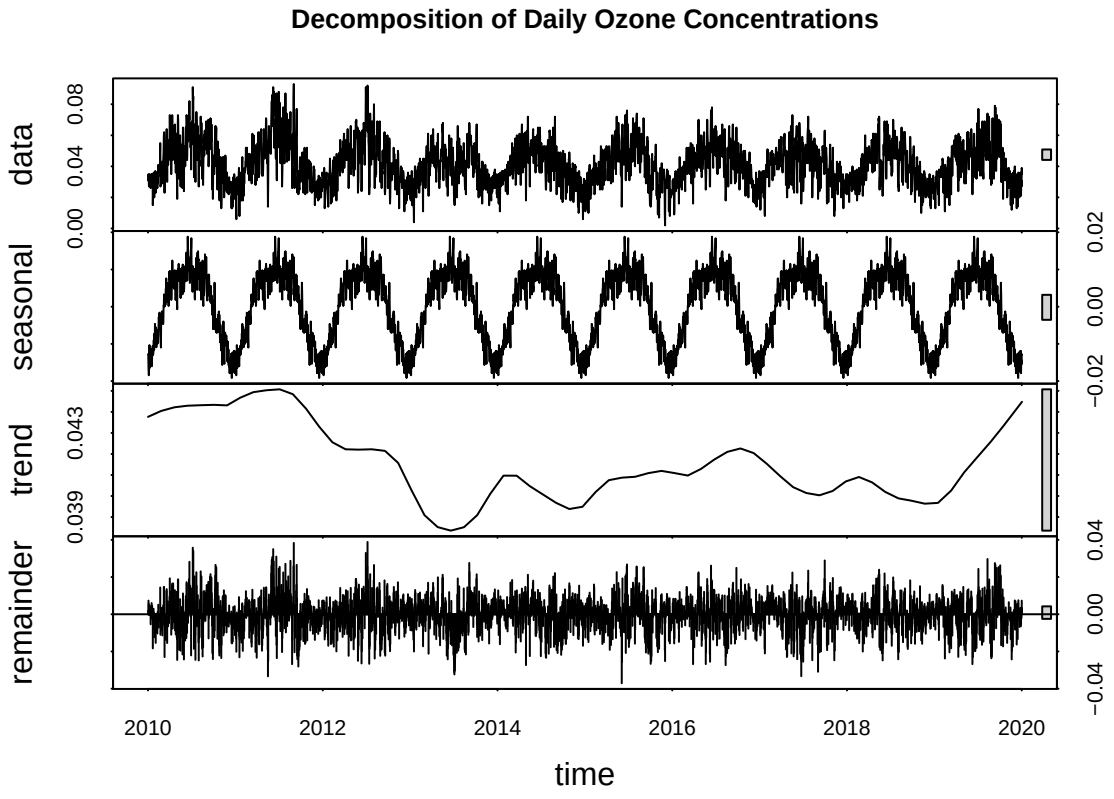
```
## # A tibble: 120 x 4
## # Groups:   Year [10]
##     Year Month Mean_Ozone_Concentration Date
##    <dbl> <dbl>                    <dbl> <date>
##  1  2010     1                   0.0305 2010-01-01
##  2  2010     2                   0.0345 2010-02-01
##  3  2010     3                   0.0446 2010-03-01
##  4  2010     4                   0.0556 2010-04-01
##  5  2010     5                   0.0466 2010-05-01
##  6  2010     6                   0.0576 2010-06-01
##  7  2010     7                   0.0578 2010-07-01
##  8  2010     8                   0.0498 2010-08-01
##  9  2010     9                   0.0548 2010-09-01
## 10  2010    10                   0.0435 2010-10-01
## # i 110 more rows
```

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.

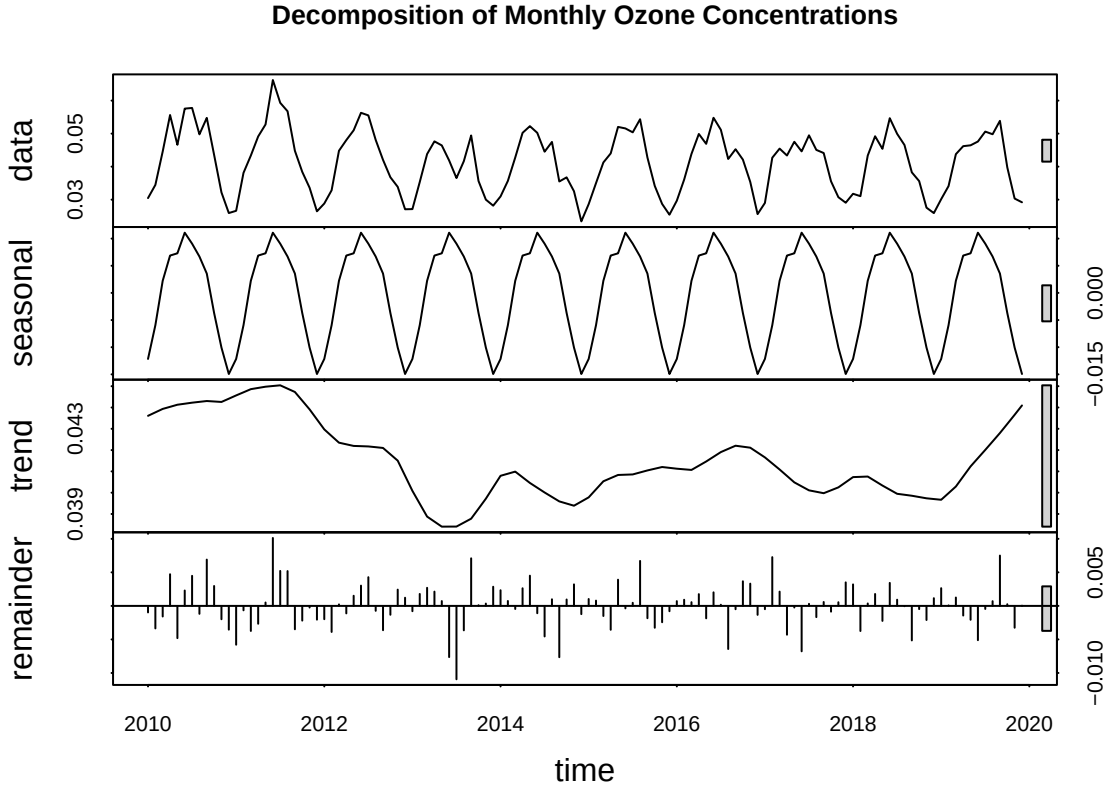
```
#10 time series objects
# daily
GaringerOzone.daily_ts <-
  ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
     start = c(2010,1),
     frequency = 365)

# monthly
GaringerOzone.monthly_ts <-
  ts(GaringerOzone.monthly$Mean_Ozone_Concentration,
     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 decomposation
# daily
GaringerOzone.daily_ts_Decomposed <- stl(GaringerOzone.daily_ts,
                                         s.window = "periodic")
plot(GaringerOzone.daily_ts_Decomposed,
     main = "Decomposition of Daily Ozone Concentrations")
```




```
# monthly
GaringerOzone.monthly_ts_Decomposed <- stl(GaringerOzone.monthly_ts,
                                           s.window = "periodic")
plot(GaringerOzone.monthly_ts_Decomposed,
     main = "Decomposition of Monthly Ozone Concentrations")
```



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 using seasonal
GaringerOzone.monthly_trend <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly_ts)

# Inspect results
GaringerOzone.monthly_trend
```

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

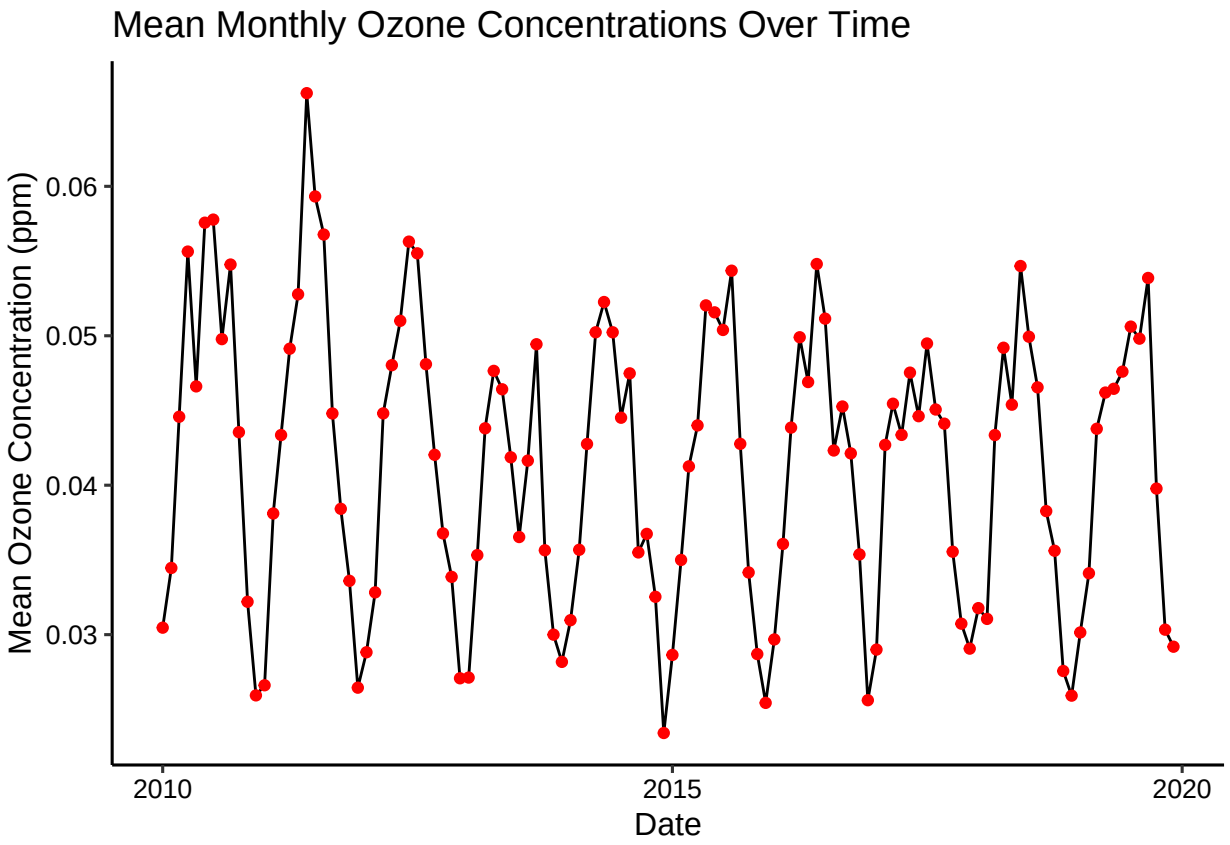
```
summary(GaringerOzone.monthly_trend)
```

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

Answer: There is an obvious sinusoidal seasonal pattern, with ozone levels being lower in the winter and higher in the summer. In this case, the seasonal Mann-Kendall analysis prevents the seasonal cycle from interfering with the trend. Moreover, the distribution of ozone concentration is non-parametric, thus seasonal Mann-Kendall analysis fits the best. Lastly, seasonal Mann-Kendall analysis allows missing 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 mean monthly ozone concentration over time
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone_Concentration)) +
  geom_line() +
  geom_point(color = "red", size = 1.5) +
  labs(title = "Mean Monthly Ozone Concentrations Over Time",
    x = "Date", y = "Mean Ozone Concentration (ppm)")
```



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 the graph, we can observe that the peak mean monthly ozone concentration slightly decreases since 2012. Based on the Mann-Kendall trend analysis, there is a statistically significant negative trend in ozone concentrations as year progresses, indicating that the monthly mean concentration decreases overtime. Kendall's tau of -0.143 suggests a weak decline in ozone levels . The two-sided p-value of 0.0467 suggests that this trend is statistically significant at 95% confidence interval. Thus, the answer to the research question is "Yes, ozone concentrations changed (decreased) at this station over the years". (Score = -77 , Var(Score) = 1499, denominator = 539.4972, tau = -0.143, 2-sided pvalue =0.046724)

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 substract seasonal component
monthly_components <-
  as.data.frame(GaringerOzone.monthly_ts_Decomposed$time.series[,1:3])

non.seasonal_ts <- GaringerOzone.monthly_ts - monthly_components$seasonal

#16 performing Mann Kendall test
non.seasonal_trend <-
  Kendall::MannKendall(non.seasonal_ts)

# displaying results
non.seasonal_trend
```

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

```
summary(non.seasonal_trend)
```

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

Answer: When running the Mann-Kendall analysis on the non-seasonal ozone monthly series, we can observe a stronger and more significant trend as compared to the Seasonal Mann-Kendall analysis on the monthly mean series. The new tau value is -0.165, indicating a stronger negative relationship as compared to the previous -0.143 value. The two-sided p value of 0.00754 is one order lower than the previous p value of 0.0467, indicating that the trend is statistically more significant (at the 1% level as compared to the 5% level). This suggests that by removing the seasonality and running a Mann-Kendall test have improved the trend analysis. (Score = -1179 , Var(Score) = 194365.7, denominator = 7139.5, tau = -0.165, 2-sided pvalue =0.0075402)