Assignment 8: Time Series Analysis

Tori Newton

Spring 2025

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

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

```
## [1] "/home/guest/EDA_Spring2025"
```

```
#Load 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(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/EDA_Spring2025
```

```
#Set ggplot theme
mytheme <- theme_classic(base_size = 14) +
  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.

```
#1: Import ten datasets
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 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: Set data column as a date class
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
class(GaringerOzone$Date) #Now Date
```

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

```
# 4: Wrangle dataset
tidy_dataset <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5: Create a new data frame
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"),
                          by = "day"))
names(Days) <- "Date"

# 6: Combine the data frames
GaringerOzone <- left_join(Days, tidy_dataset, by = "Date")
dim(GaringerOzone) #Has 3652 rows and 3 columns
```

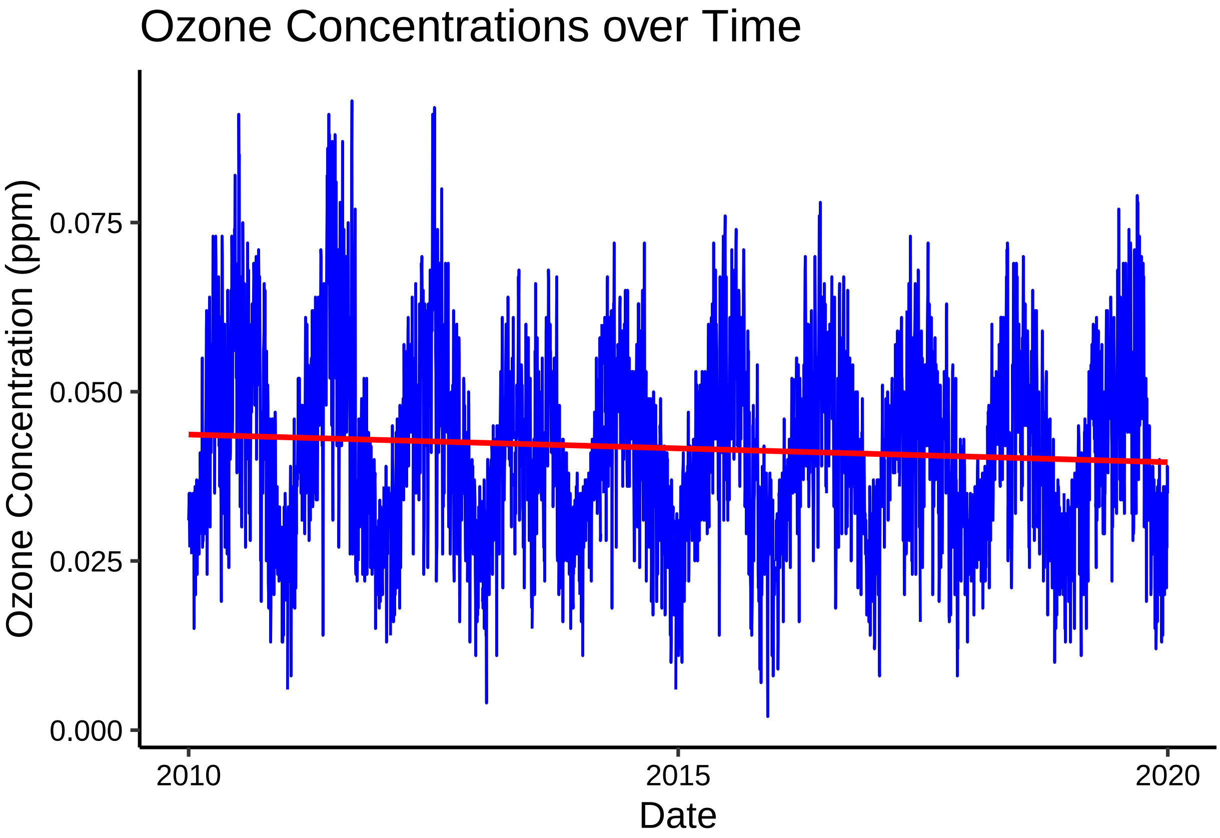
```
## [1] 3652    3
```

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: Line plot
GaringerOzone %>%
  ggplot(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Ozone Concentrations over Time",
       x = "Date",
       y = "Ozone Concentration (ppm)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 63 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



Answer:The red line is very slightly sloped downward, indicating a slight decrease in ozone concentration over time.

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?

```r
#8: Use a linear interpolation
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <-
na.approx (GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
           x = GaringerOzone$Date)
```

Answer: Piecewise constant internpolation would fill in missing values with the most recent non-missing value. It assumes that the value remains constant between the known data points which would not be appropriate for ozone concentrations. Spline interpolation fits a smooth curve through the data points and would be more appropriate with data that exhibits nonlinear trends. Linear interpolation is simpler and typically more appropriate for daily time series like this one.

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
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Year = year(Date), Month=month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(Mean_Ozone =mean(Daily.Max.8.hour.Ozone.Concentration,
                             na.rm = TRUE)) %>%
  mutate(Date=as.Date(paste(Year, Month, "01", sep = "-"))) %>%
  ungroup()
```

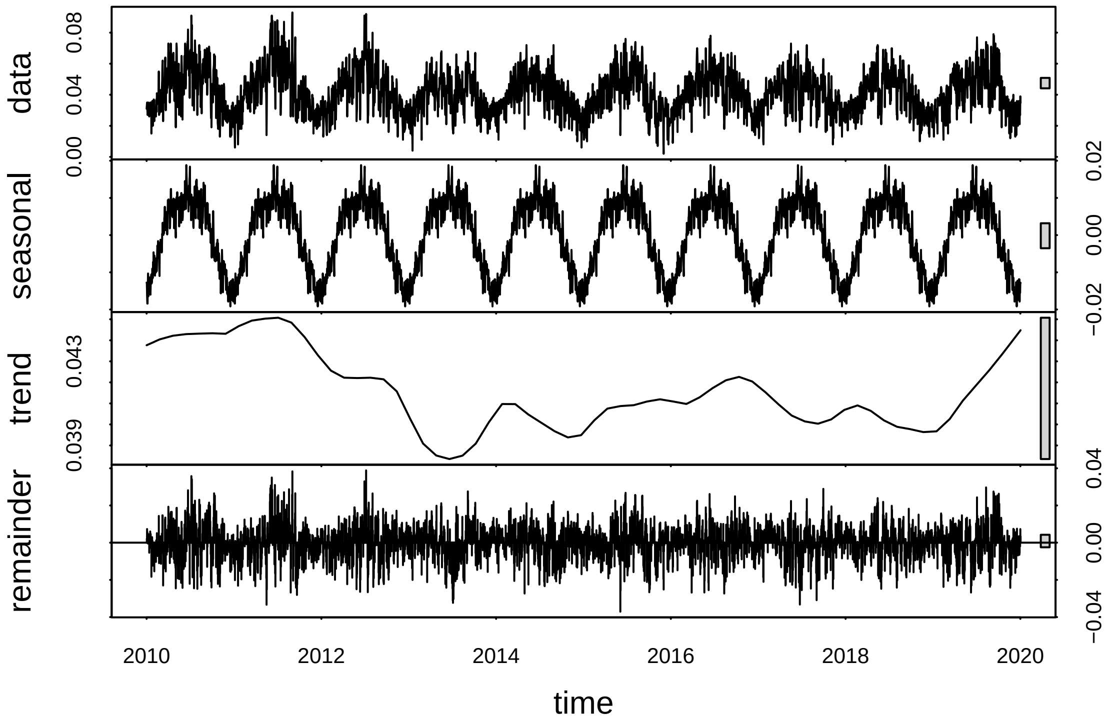
```
## 'summarise()' has grouped output by 'Year'. You can override using the
## '.groups' argument.
```

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.

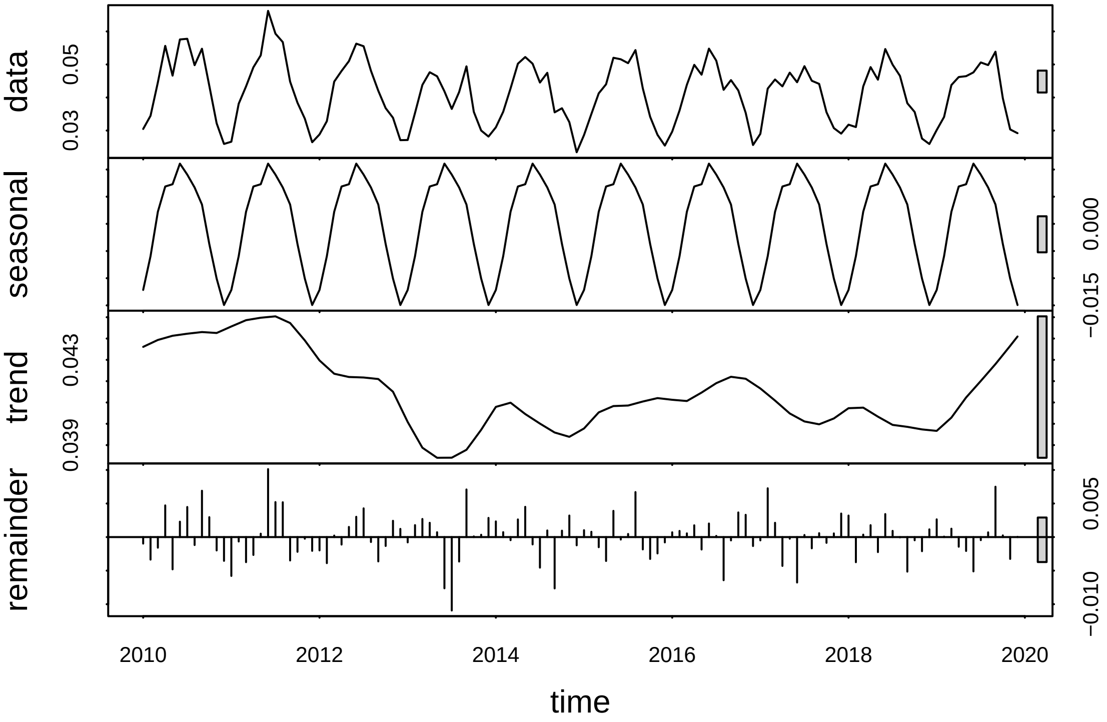
```
#10: Generating time series objects
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             frequency=365,
                             start=c(2010,1))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone,
                               frequency=12,
                               start=c(2010,1))
```

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

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



```
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?

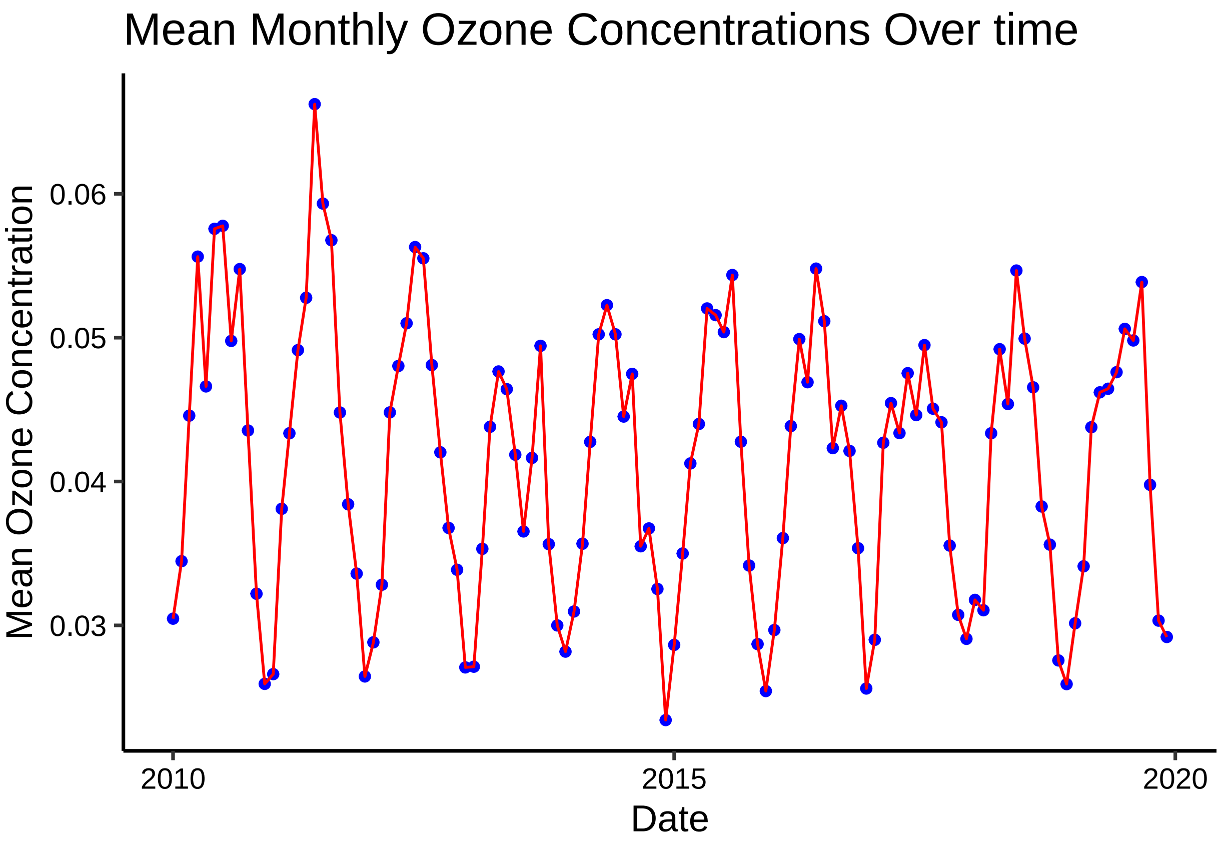
```
#12: Seasonal Mann-Kendall
Ozone_monthy_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(Ozone_monthy_trend)
```

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

Answer: Seasonal Mann-Kendall is most appropriate because it is a non-parametric test that is often used to detect trends in time series data when the data has a seasonal component. This test is particularly suitable for time series with periodic or seasonal variations (such as monthly ozone levels) because it accounts for the repeating patterns in the data while analyzing the overall trend.

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: Plot depicting mean monthly ozone concentrations over time

GaringerOzone.monthly %>%
  ggplot(aes(x= Date, y=Mean_Ozone)) +
  geom_point(color="blue") +
  geom_line(color="red") +
  labs(title = "Mean Monthly Ozone Concentrations Over time", x = "Date",
       y = "Mean Ozone Concentration")
```



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: The plot of mean monthly ozone concentrations over time shows seasonal fluctuations in ozone levels throughout the 2010s. The result from the Seasonal Mann-Kendall test indicates a negative monotonic trend in ozone concentrations over the 2010s (tau=-0.143, p-value = 0.046724). The p-value is below 0.05, suggesting that the trend is statistically significant. The slope is not very strong (-0.143) but it still indicates a decline over the study period. In conclusion, based on the Seasonal Mann-Kendall test, there is evidence to suggest a slight decline in ozone concentrations at this station during the 2010s.

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: Subtract the seasonal component
#Decompose the time series
GaringerOzone.monthly.ts_decomposed <- stl(GaringerOzone.monthly.ts,
                                           s.window = "periodic")

#Extract the components into a data frame
GaringerOzone.monthly.ts_components <- as.data.frame(GaringerOzone.monthly.ts_decomposed$time.series[,1:3])

#Mutate to add the Observed and Date columns
GaringerOzone.monthly.ts_components <-
  mutate(GaringerOzone.monthly.ts_components,
         Observed = GaringerOzone.monthly$Mean_Ozone,
         Date = GaringerOzone.monthly$Date)

#Subtract the seasonal component from the observed series
GaringerOzone.deseasonalized <-
```

```
  GaringerOzone.monthly.ts_components$Observed -
  GaringerOzone.monthly.ts_components$seasonal

#Mutate to add the deseasonalized column
GaringerOzone.monthly.ts_components <-
  mutate(GaringerOzone.monthly.ts_components,
         Deaseasonalized = GaringerOzone.deseasonalized)

#View the updated data frame
head(GaringerOzone.monthly.ts_components)
```

```
##       seasonal      trend     remainder   Observed       Date Deaseasonalized
## 1 -0.012164159 0.04360892 -0.0009770197 0.03046774 2010-01-01      0.04263190
## 2 -0.005945745 0.04377124 -0.0033612105 0.03446429 2010-02-01      0.04041003
## 3  0.002231834 0.04393356 -0.0015847518 0.04458065 2010-03-01      0.04234881
## 4  0.006878411 0.04403138  0.0047235448 0.05563333 2010-04-01      0.04875492
## 5  0.007292088 0.04412919 -0.0048083781 0.04661290 2010-05-01      0.03932081
## 6  0.011093186 0.04417744  0.0022960356 0.05756667 2010-06-01      0.04647348
```

```
#16: Mann Kendall test

Nonseasonal_monthly_Ozone <- Kendall::MannKendall(GaringerOzone.deseasonalized)
summary(Nonseasonal_monthly_Ozone)
```

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

> Answer: Based on the Mann Kendall test, the Score has a negative value which indicates a decreasing trend over time. This is consistent with the results from the Seasonal Mann Kendall test. For the Mann Kendall test, the tau value of -0.165 indicates a weak negative monotonic trend, meaning that there is a slight decreasing trend in the data which is also consistend with the results from the Seasonal Mann Kendall. The p-value of the Mann Kendall (0.0075402) indicates that the results are statistically significant. Overall, the results for both the Mann Kendall and the Seasonal Mann Kendall are similar and both indicate a weak but statistically significant decreasing trend in the data over time.