Assignment 8: Time Series Analysis

Rosie Wu

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

```
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)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
```

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

```
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
# Set 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.

```
# Load necessary package
library(dplyr)

# Set the path to the folder containing the datasets
data_folder <- "Data/Raw/Ozone_TimeSeries"

# List all files in the folder that have .csv extension
files <- list.files(data_folder, pattern = "*.csv", full.names = TRUE)

# Read all files into a list of data frames and then combine them
GaringerOzone <- files %>%
  lapply(read.csv) %>%  # Read each file into a list of data frames
  bind_rows()           # Combine all data frames in the list into one

# Check the structure and dimensions of the combined dataframe
str(GaringerOzone)
```

```
## 'data.frame':    3589 obs. of  20 variables:
##  $ Date                                : chr  "01/01/2010" "01/02/2010" "01/03/2010" "01/04/2010" ..
##  $ Source                              : chr  "AQS" "AQS" "AQS" "AQS" ...
##  $ Site.ID                             : int  371190041 371190041 371190041 371190041 371190041 3711
##  $ POC                                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.031 0.033 0.035 0.031 0.027 0.033 0.035 0.032 0.032
##  $ UNITS                               : chr  "ppm" "ppm" "ppm" "ppm" ...
##  $ DAILY_AQI_VALUE                     : int  29 31 32 29 25 31 32 30 30 28 ...
```

```
##  $ Site.Name                          : chr  "Garinger High School" "Garinger High School" "Garinge
##  $ DAILY_OBS_COUNT                    : int  17 17 17 17 17 17 17 17 17 17 ...
##  $ PERCENT_COMPLETE                   : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                 : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                 : chr  "Ozone" "Ozone" "Ozone" "Ozone" ...
##  $ CBSA_CODE                          : int  16740 16740 16740 16740 16740 16740 16740 16740 16740 1
##  $ CBSA_NAME                          : chr  "Charlotte-Concord-Gastonia, NC-SC" "Charlotte-Concord
##  $ STATE_CODE                         : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                              : chr  "North Carolina" "North Carolina" "North Carolina" "No
##  $ COUNTY_CODE                        : int  119 119 119 119 119 119 119 119 119 119 ...
##  $ COUNTY                             : chr  "Mecklenburg" "Mecklenburg" "Mecklenburg" "Mecklenburg
##  $ SITE_LATITUDE                      : num  35.2 35.2 35.2 35.2 35.2 ...
##  $ SITE_LONGITUDE                     : num  -80.8 -80.8 -80.8 -80.8 -80.8 ...
```

```
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. Convert the Date format in GaringerOzone (if needed)
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4. Select only the required columns
GaringerOzone_filtered <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5.
# Create a complete sequence of dates from 2010-01-01 to 2019-12-31
Days <- data.frame(Date = seq.Date(as.Date("2010-01-01"),
                                   as.Date("2019-12-31"),
                                   by = "day"))


#6. Use left_join to add missing dates and fill them with NA
GaringerOzone <- Days %>%
  left_join(GaringerOzone_filtered, by = "Date")

# Check the dimensions of the final data frame
dim(GaringerOzone)  # Should be 3652 rows and 3 columns
```

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

Visualize

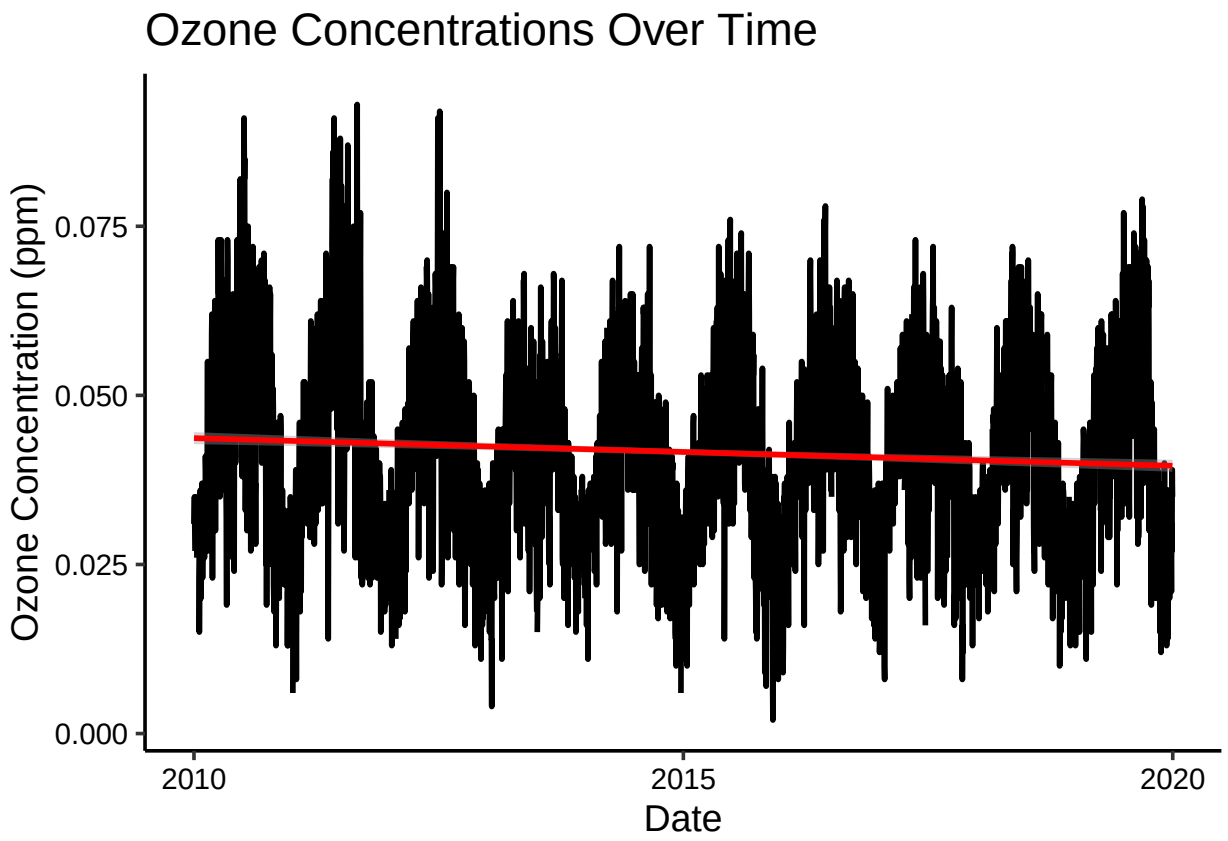
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
# Load necessary packages
library(ggplot2)

# plot concentrations using ggplot
GaringerOzone_7plot <-
  ggplot(GaringerOzone, aes(x = Date,
                            y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(color = "black", size = 1) +
  geom_smooth(method = lm, color = "red") +  # Smoothed line
  labs(title = "Ozone Concentrations Over Time",
       x = "Date",
       y = "Ozone Concentration (ppm)") +
  theme(mytheme)
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
print(GaringerOzone_7plot)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’

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



Answer: The plot shows a slight decreasing trend of 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?

```
#8
# check if there were NAs first in the ozone concentration column
summary(GaringerOzone)
```

```
##       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   :0.04163                      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                           NA's   :63
```

```
GaringerOzone_clean8 <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )
# check again, now the ozone concentration doesn't have NAs
summary(GaringerOzone_clean8)
```

```
##       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: - Piecewise constant is also known as a "nearest neighbor" approach. Any missing data are assumed to be equal to the measurement made nearest to that date, so that's not the most accurate approach. It may not capture trends or variations between data points effectively, especially if there are natural changes in the underlying phenomenon. - Spline constant draws a quadratic function that is used to interpolate rather than drawing a straight line, which is complex and, If the spline degree is too high or the number of knots is too large, it can fit noise rather than the underlying trend. - Linear, in comparison, is simpler to use. It also has less Data Sensitivity: It does not require complex parameter tuning. The resulting data series is continuous, which is often desirable in time series analysis. It's also more suitable for small gaps where adjacent data points are relatively close in value, which fits our case.

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_clean8 %>%
  # Add year and month columns
  mutate(Year = year(Date),
         Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(Mean_Ozone_Concentration = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE),
            .groups = "drop") %>%
  # Create a new Date column as the first day of each month
  mutate(Date = as.Date(paste(Year, Month, "01", sep = "-")))
```

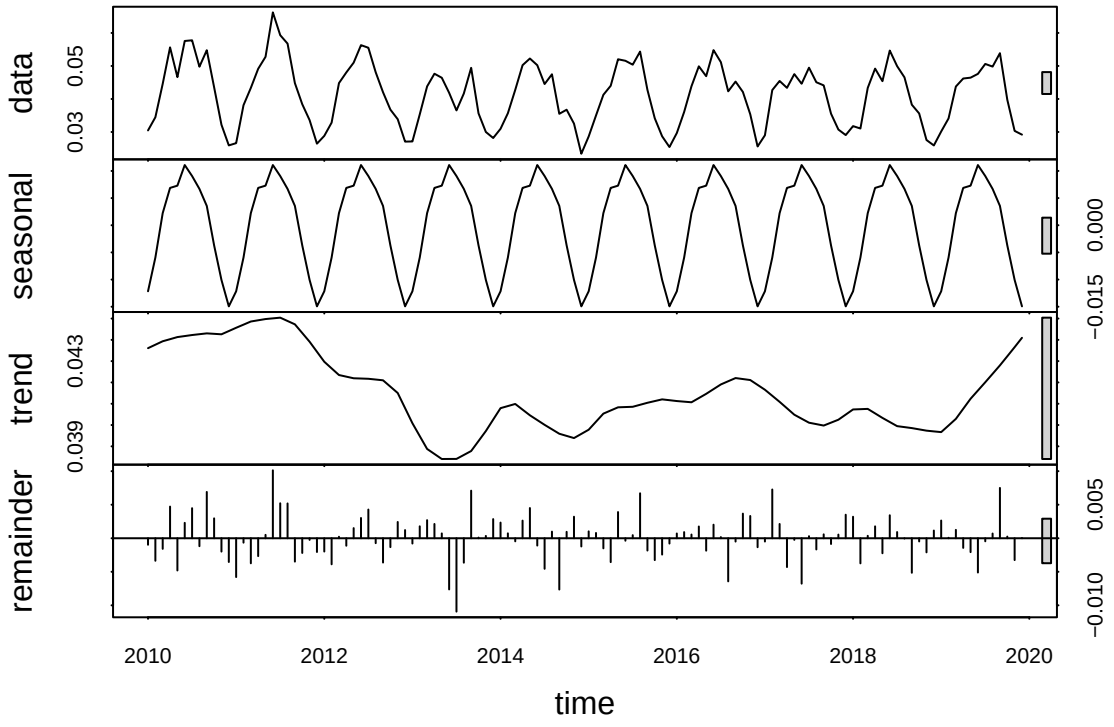
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```
#10
# first review the start and end date of the dataset,
# Generate time series (trend test needs ts, not data.frame)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone_Concentration,
                   start=c(2010, 1),
```

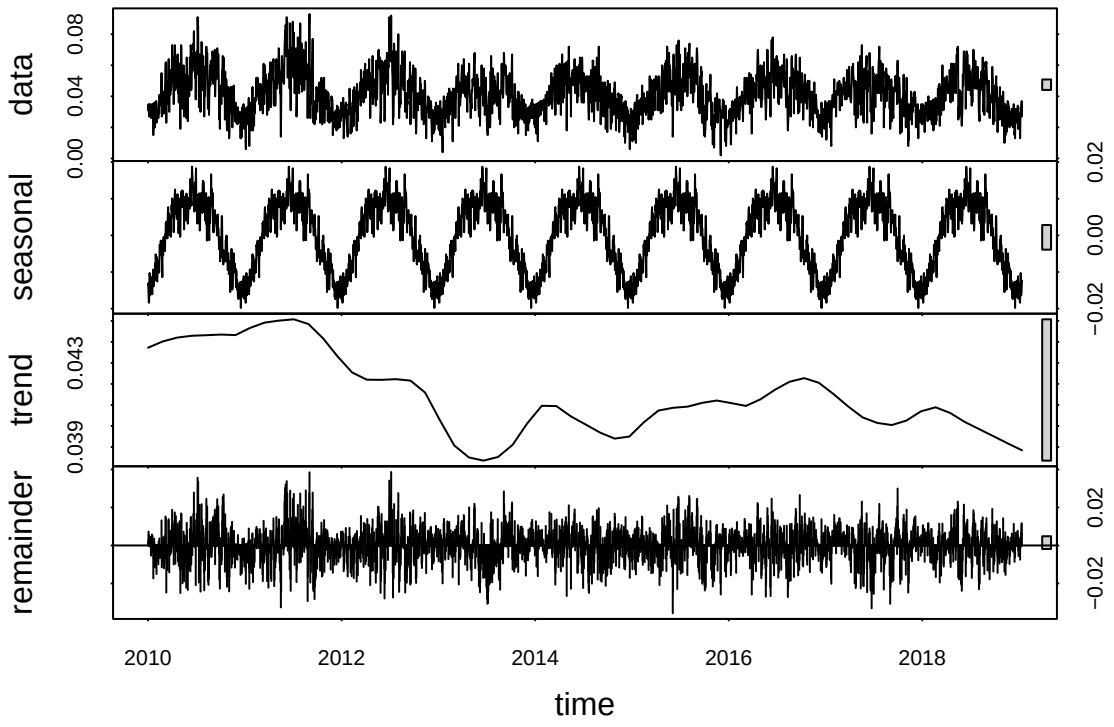
```
                    end = c(2019, 12) ,
                     frequency=12)
GaringerOzone.daily.ts <- ts(GaringerOzone_clean8$Daily.Max.8.hour.Ozone.Concentration,
                  start = c(2010, 1, 1),  # Start date: 2010-01-01
                  end = c(2019, 12, 31),   # End date: 2019-12-31
                  frequency = 365)      # Frequency: daily observations
```

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

```
#11
#decompose month
GaringerOZ_month_ts_decomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
plot(GaringerOZ_month_ts_decomp)
```




```
# decompose daily
GaringerOZ_daily_ts_decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(GaringerOZ_daily_ts_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?

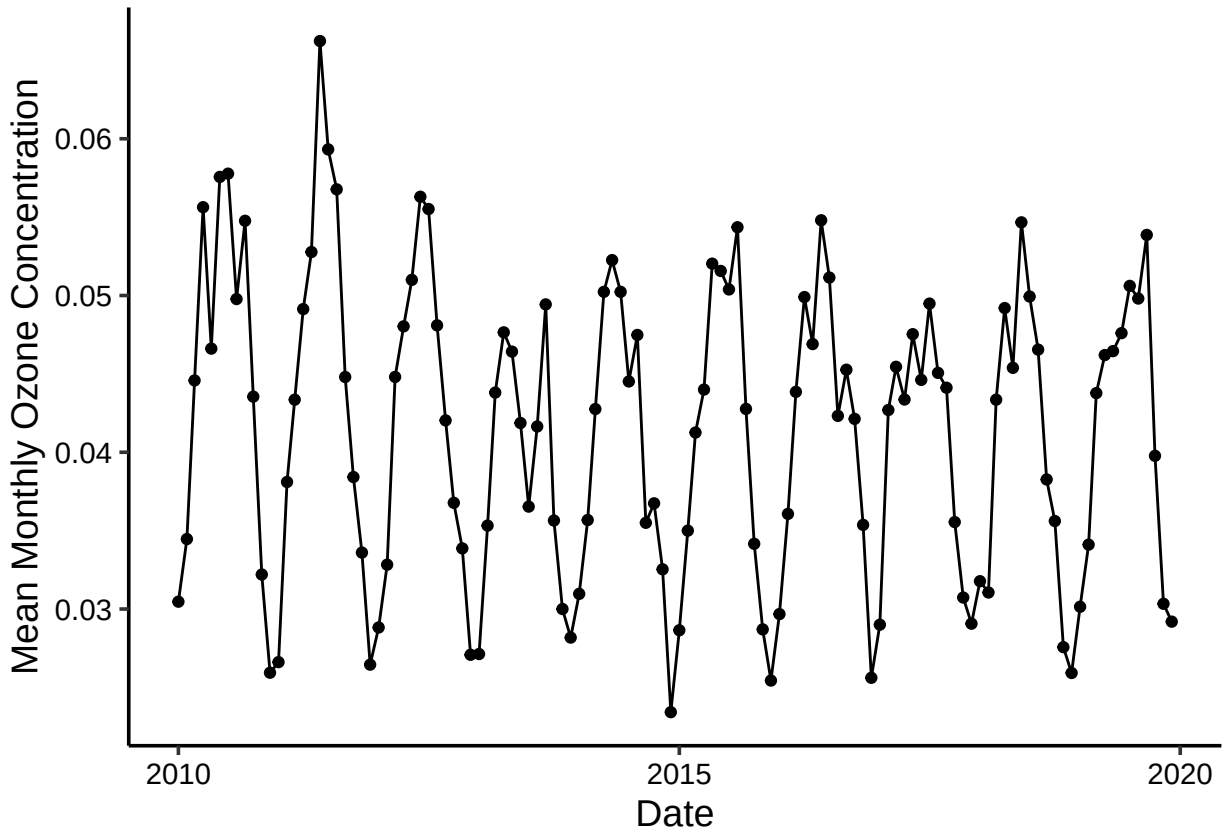
```
#12
# Run SMK test
GaringerOZ_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOZ_trend1)
```

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

Answer: It should be. - Monthly ozone data exhibits seasonal patterns, making the seasonal Mann-Kendall test appropriate for detecting trends within this structure. - The test does not assume normality, making it suitable for environmental data that may have outliers or non-normal distributions that we don't know of. - It effectively identifies monotonic trends while accounting for seasonal variations, ensuring more accurate results. - Independent Seasonal Trends: The test can analyze trends separately for each season, allowing for a clearer understanding of changes over time in a seasonal pattern.

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.

```r
# 13
#Visualization
GaringerOZ_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone_Concentration)) +
  geom_point() +
  geom_line() +
  ylab("Mean Monthly Ozone Concentration")
print(GaringerOZ_plot)
```



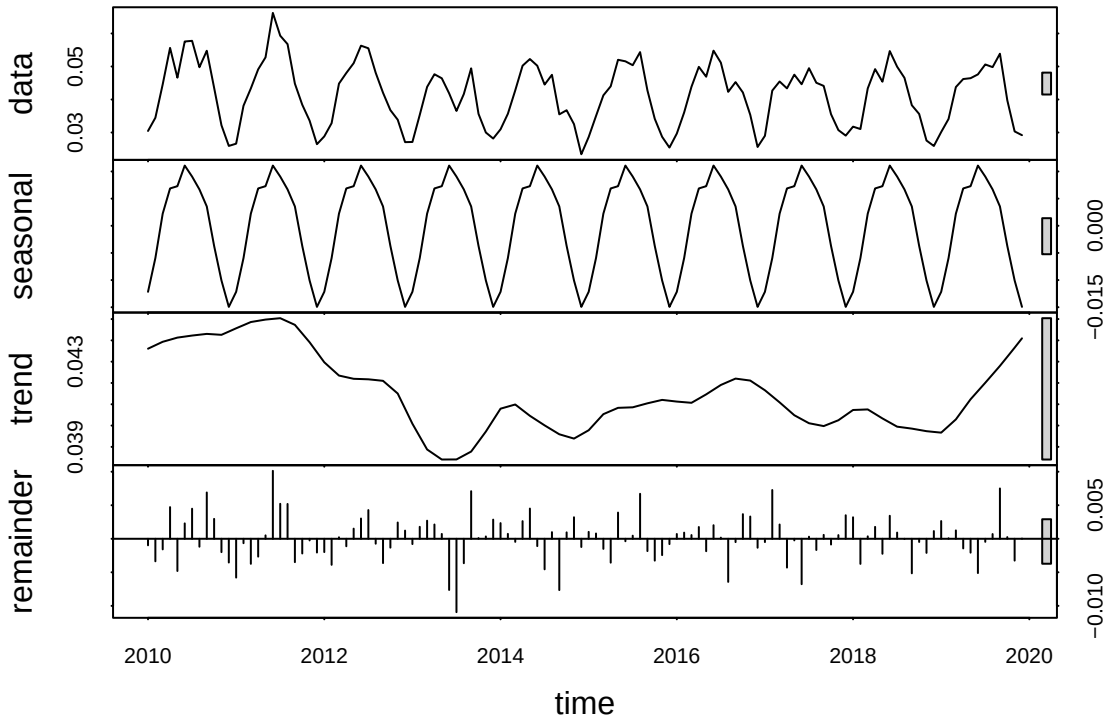
14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: According to the p-value from the skm test < 0.05, this could imply it's strong and significant linear trend. Also the tau and score are negative, so the direction of the trend is going down overtime. The trend seems to follow a seasonal pattern – by observing the line graphs by monthly average Ozone concentrations from the previous question, the Ozone concentration peaks during middle (or summer) of of the year, and the beginning and end of the year, which are the colder months, should have lower monthly concentration. Meanwhile, the average concentration trend is slowly decreasing over years in the measured decade (2010-2019).

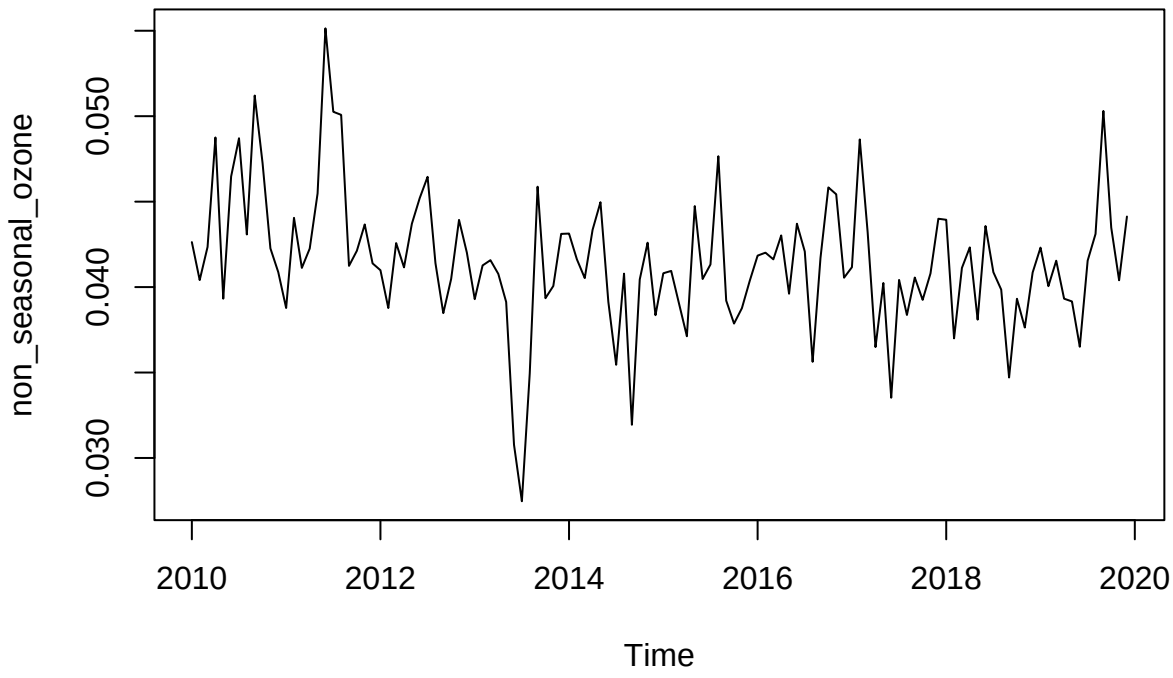
15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

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.

```r
#15
# Decompose the time series
decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(decomposed)
```




```r
# Extract the seasonal component
seasonal_component <- decomposed$time.series[, "seasonal"]

# Subtract the seasonal component from the original time series
non_seasonal_ozone <- GaringerOzone.monthly.ts - seasonal_component

plot(non_seasonal_ozone)
```




```
#16
result_non_seasonal <- MannKendall(non_seasonal_ozone)
summary(result_non_seasonal)
```

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

Answer: The p-value for the seasonal is 0.046724, and the non-seasonal is 0.0075. So they are both significant (<0.05), so there should be a significant trend for the ozone concentration change over time. The non-seasonal test shows a lower tau (-0.165) than seasonal one(-0.143), but they are both negative, which means the trend is generally going down. The Var(score) for seasonal (1499) is lower than non-seasonal (194365.7), so the non-seasonal time series have less variance than seasonal test results. The score for non-seasonal result shows a larger magnitute, which means the negative trend could be more drastic than the seasonal result. From visual obeservation of the plot for non-seasonal test, it seems that the time series analysis trend of the ozone concentration data is going up more visibly over the years.