Assignment 8: Time Series Analysis

Chunyi Xu

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

```
#Loading necessary packages (tidyverse, lubridate, here)
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   3.5.0     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)
```

```
## Warning: package 'zoo' was built under R version 4.3.3
```

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

```
library (trend)
```

```
## Warning: package 'trend' was built under R version 4.3.3
```

```
library (ggplot2)
library (here)
```

```
## Warning: package 'here' was built under R version 4.3.3
```

```
## here() starts at /Users/xuchunyi/Desktop/EDA_Spring2025/EDA_Spring2025
```

```
#Checking working directory is the project folder
here()
```

```
## [1] "/Users/xuchunyi/Desktop/EDA_Spring2025/EDA_Spring2025"
```

```
#Building my ggplot theme
ggplot.theme <- theme_classic(base_size = 15) +
  theme(plot.background = element_rect(color='white',fill='white'),
        plot.title = element_text(color='black', size = 15),
        axis.text = element_text(color = 'black',size = 10),
        panel.grid.minor = element_line(size = 0.5),
        panel.grid.major = element_line(size = 0.5),
        legend.background = element_rect(color='white', fill = 'white'),
        legend.position = "right",
        legend.title = element_text(color='black',size=10),
        legend.text = element_text(size = 10))
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
#Setting the theme as the default theme
theme_set(ggplot.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.

```r
#2
#Importing the ten datasets from the Ozone_TimeSeries folder with data from 2010-2019
#For 2010
Ozone_2010 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)

#For 2011
Ozone_2011 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)

#For 2012
Ozone_2012 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
  stringsAsFactors = TRUE)

#For 2013
Ozone_2013 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
  stringsAsFactors = TRUE)


#For 2014
Ozone_2014 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
  stringsAsFactors = TRUE)

#For 2015
Ozone_2015 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
  stringsAsFactors = TRUE)

#For 2016
Ozone_2016 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
  stringsAsFactors = TRUE)

#For 2017
Ozone_2017 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
  stringsAsFactors = TRUE)

#For 2018
Ozone_2018 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
  stringsAsFactors = TRUE)

#For 2019
Ozone_2019 <- read.csv(
  file = here("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
  stringsAsFactors = TRUE)
```

```
#Combining the ten datasets
GaringerOzone <-
  rbind(Ozone_2010, Ozone_2011, Ozone_2012, Ozone_2013, Ozone_2014,
        Ozone_2015, Ozone_2016, Ozone_2017, Ozone_2018, Ozone_2019)
```

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 columns to date objects
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

#Checking the class of the date column
class (GaringerOzone$Date)
```

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

```
# 4
#Wranglling the dataset so that it only contains specific variables
GaringerOzone_sub1 <-
  select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

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

#Renaming the column name in Days to "Date"
colnames (Days)  <-  "Date"

# 6
#Combining the data frames Days and GaringerOzone_sub1
GaringerOzone <- left_join(Days,GaringerOzone_sub1, by = "Date")
```

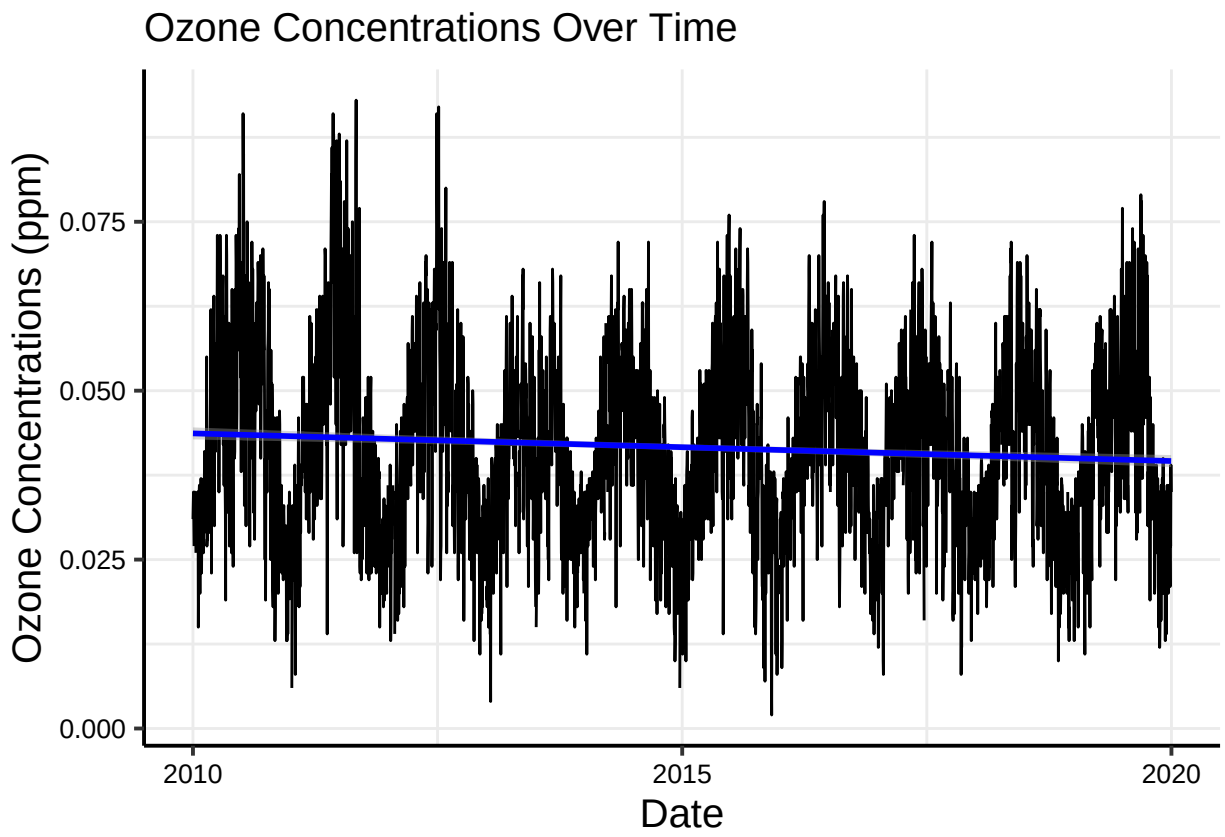
Visualize

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?

```r
#7
#Plotting ozone concentrations over time
Ozone_Concen_Trend <-
  ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth (method="lm", color = "blue")+
  labs(x = "Date", y = "Ozone Concentrations (ppm)") +
  labs(title = "Ozone Concentrations Over Time")
print(Ozone_Concen_Trend)
```

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

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



Answer: Yes, the downward sloping smooth line suggests a decreasing ozone concentration trend 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
#Using a linear interpolation to fill in missing daily data for ozone concentration
GaringerOzone_Fill <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration_Fill
         = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Linear interpolation fills in missing daily data by drawing a straight line between the known points to determine the values of the interpolated data on any given date. The piecewise constant method assumes the missing data point to be equal to the measurement made nearest to that date. Spline interpolation is similar to linear interpolation except that a quadratic function is used to interpolate rather than drawing a straight line. Among these three methods, linear interpolation becomes a better fit. The piecewise constant method may oversimplify the data distribution and reduce the daily variability in a dataset. Additionally, the linear interpolation method avoids the unnecessary complexity and fluctuations associated with spline interpolation, particularly when the trend in the dataset is predominantly linear.

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 a new data frame that contains aggregated monthly ozone data
GaringerOzone.monthly <-
  GaringerOzone_Fill %>%
  mutate(Month = month(Date))%>%
  mutate(Year = year(Date)) %>%
  mutate(Date = my(paste0(Month,"-",Year))) %>%
  group_by(Date)%>%
  summarise(Monthly_Mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration_Fill))
```

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.

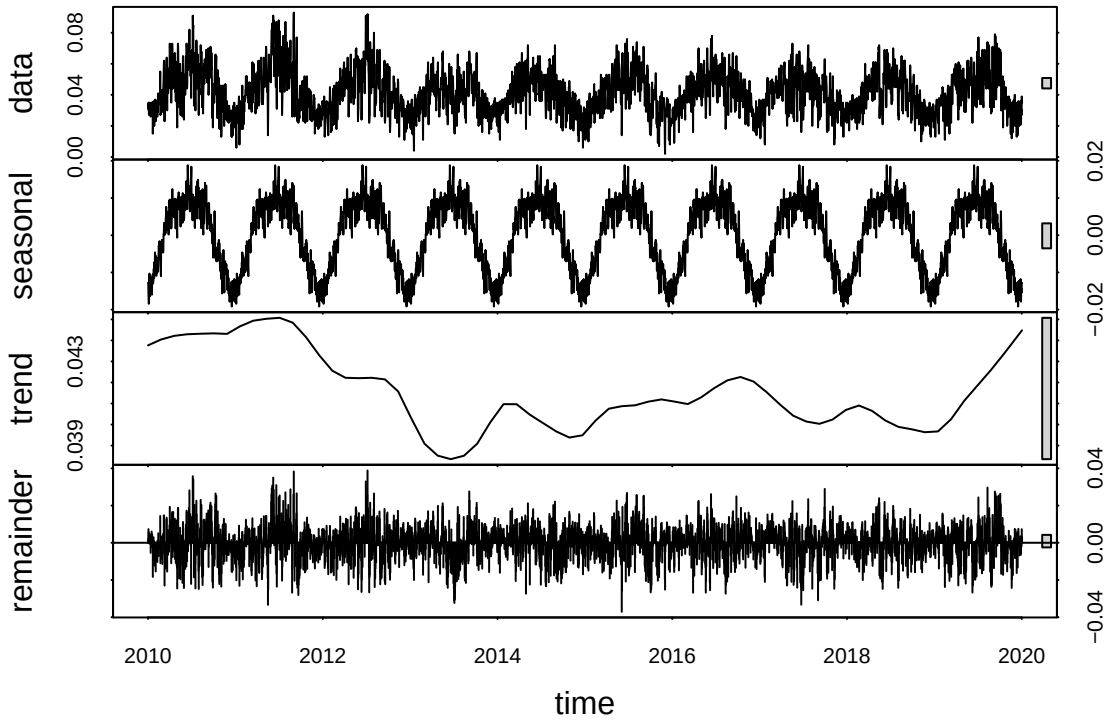
```
#10
#Generating the time series object based on the dataframe of daily observations
GaringerOzone.daily.ts <- ts(GaringerOzone_Fill$Daily.Max.8.hour.Ozone.Concentration_Fill,
                             start = c(2010,1,1), frequency = 365)

#Generating the time series object based on the dataframe of monthly observations
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Monthly_Mean_Ozone,
                               start = c(2010,1), frequency = 12)
```

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

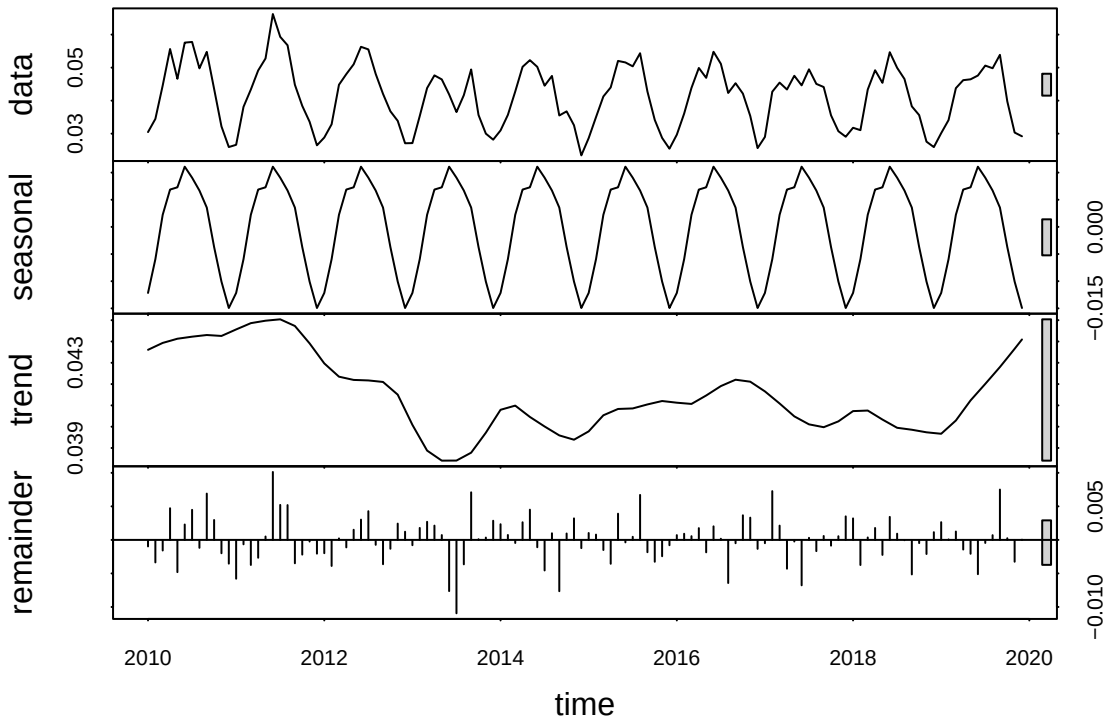
```r
#11
#Daily
# Generating the decomposition
GaringerOzone.daily_Decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

# Visualizing the decomposed series.
plot(GaringerOzone.daily_Decomposed)
```




```r
#Monthly
# Generating the decomposition
GaringerOzone.monthly_Decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

# Visualizing the decomposed series.
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 monotonic trend analysis for the monthly Ozone series.
Monthly_Trend_analysis <- trend::smk.test(GaringerOzone.monthly.ts)
Monthly_Trend_analysis
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

```
summary (Monthly_Trend_analysis)
```

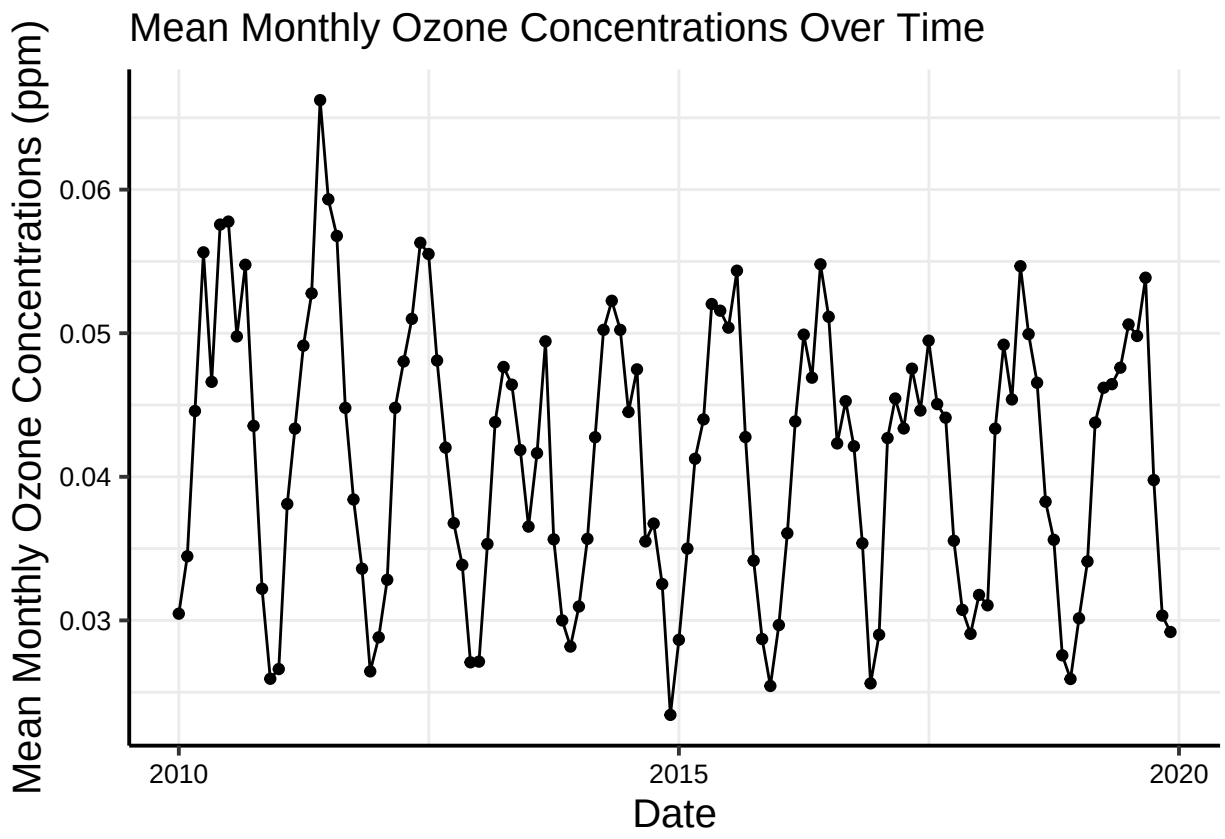
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
```

```
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 12:   S = 0  11  125  0.244  0.894  0.37109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: This is because the seasonal Mann-Kendall is designed to handle the seasonality in a dataset. From the graph we created in Q7, we can see equally spaced upward and downward movements, which suggests seasonality may exist in this dataset.

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
#Plotting mean monthly ozone concentrations over time
Monthly_Ozone_Concen_Trend <-
  ggplot(GaringerOzone.monthly, aes(x = Date, y = Monthly_Mean_Ozone)) +
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Mean Monthly Ozone Concentrations (ppm)") +
  labs(title = "Mean Monthly Ozone Concentrations Over Time")
print(Monthly_Ozone_Concen_Trend)
```



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 p-value of the seasonal Mann-Kendall test is less than 0.05 (statistical test output: z = -1.963, p-value = 0.04965). This result suggests that we need to reject the null hypothesis that the mean monthly ozone concentration trend is stationary. Instead, we have a statistically significant trend in this dataset. However, it is important to note that this p-value is only marginally below the 0.05 threshold. To put the result in the context of the research question, this test result indicates that the mean monthly ozone concentrations has changed over the 2010 to 2019 at this station. However, from the SMK test, we can also see the trend breakdown for each season of the year. For each season, the decreasing trend is not so pronounced, as their p-values are all greater than 0.05. This may explain why the overall p-value from the seasonal Mann-Kendall test is only slightly under 0.05. Nonetheless, the overall trend remains statistically significant due to its p-value being less than 0.05.

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
#Removing seasonality for monthly ozone concentrations
Monthly_Ozone_Nonseas <- GaringerOzone.monthly.ts - GaringerOzone.monthly_Decomposed$time.series[,1]
```

```r
#16
#Running the Mann Kendall test on the non-seasonal Ozone monthly series
Nonsea_Monthly_Trend_analysis <- trend::mk.test(Monthly_Ozone_Nonseas)
Nonsea_Monthly_Trend_analysis
```

```
##
##  Mann-Kendall trend test
##
## data:  Monthly_Ozone_Nonseas
## z = -2.672, n = 120, p-value = 0.00754
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.179000e+03  1.943657e+05 -1.651376e-01
```

Answer: From the Mann Kendall test on the non-seasonal Ozone monthly series, we can get the z-value is 2.672 and the p-value is 0.00754. Since the p-value is lower than 0.05, this result suggests that we need to reject the null hypothesis that the mean monthly ozone concentration trend is stationary in this dataset without the seasonal series. Instead, we have a statistically significant trend in this dataset. The p-value of the Mann Kendall test is much lower than the p-value of the Seasonal Mann Kendall test, while the latter is only marginally below the 0.05 threshold. This difference may suggest that the trend in the complete series (revealed by the seasonal Mann Kendall test) is less significant and less pronounced than the trend in the dataset without the seasonal series (revealed by the Mann Kendall test). Also, the seasonal variation in the mean monthly Ozone concentrations over time may dilute and confuse the trend in the dataset. Therefore, the difference of the results given by the Mann Kendall test and the seasonal Mann Kendall test indicates the necessity to account for seasonal fluctuations while evaluating trends.