Assignment 8: Time Series Analysis

Jonathan Gilman

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

```
# import 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 error
```

```
library(lubridate)
library(here)
```

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

```
library(knitr)
library(agricolae)
library(dplyr)
library(zoo)
```

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

```
library(trend)

# check current working directory
here()
```

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

```
# 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.

```
#1

# import datasets from Ozone_TimeSeries
d1 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)
d2 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)
d3 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2012_raw.csv"),
  stringsAsFactors = TRUE)
d4 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2013_raw.csv"),
  stringsAsFactors = TRUE)
d5 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2014_raw.csv"),
  stringsAsFactors = TRUE)
d6 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2015_raw.csv"),
  stringsAsFactors = TRUE)
```

```
d7 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2016_raw.csv"),
  stringsAsFactors = TRUE)
d8 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2017_raw.csv"),
  stringsAsFactors = TRUE)
d9 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2018_raw.csv"),
  stringsAsFactors = TRUE)
d10 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2019_raw.csv"),
  stringsAsFactors = TRUE)


GaringerOzone <- rbind(d1, d2, d3, d4, d5, d6, d7, d8, d9, d10)
```

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 Date to character
GaringerOzone$Date <- as.character(GaringerOzone$Date)
# Check class
class(GaringerOzone$Date)
```

```
## [1] "character"
```

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

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

# 5
# generate daily dataset
Days <- data.frame(Date = seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day"))
```

```r
# 6
# combine dataframes
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")

# check the resulting dataframe
head(GaringerOzone)
```

```
##         Date Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
## 1 2010-01-01                                0.031              29
## 2 2010-01-02                                0.033              31
## 3 2010-01-03                                0.035              32
## 4 2010-01-04                                0.031              29
## 5 2010-01-05                                0.027              25
## 6 2010-01-06                                   NA              NA
```

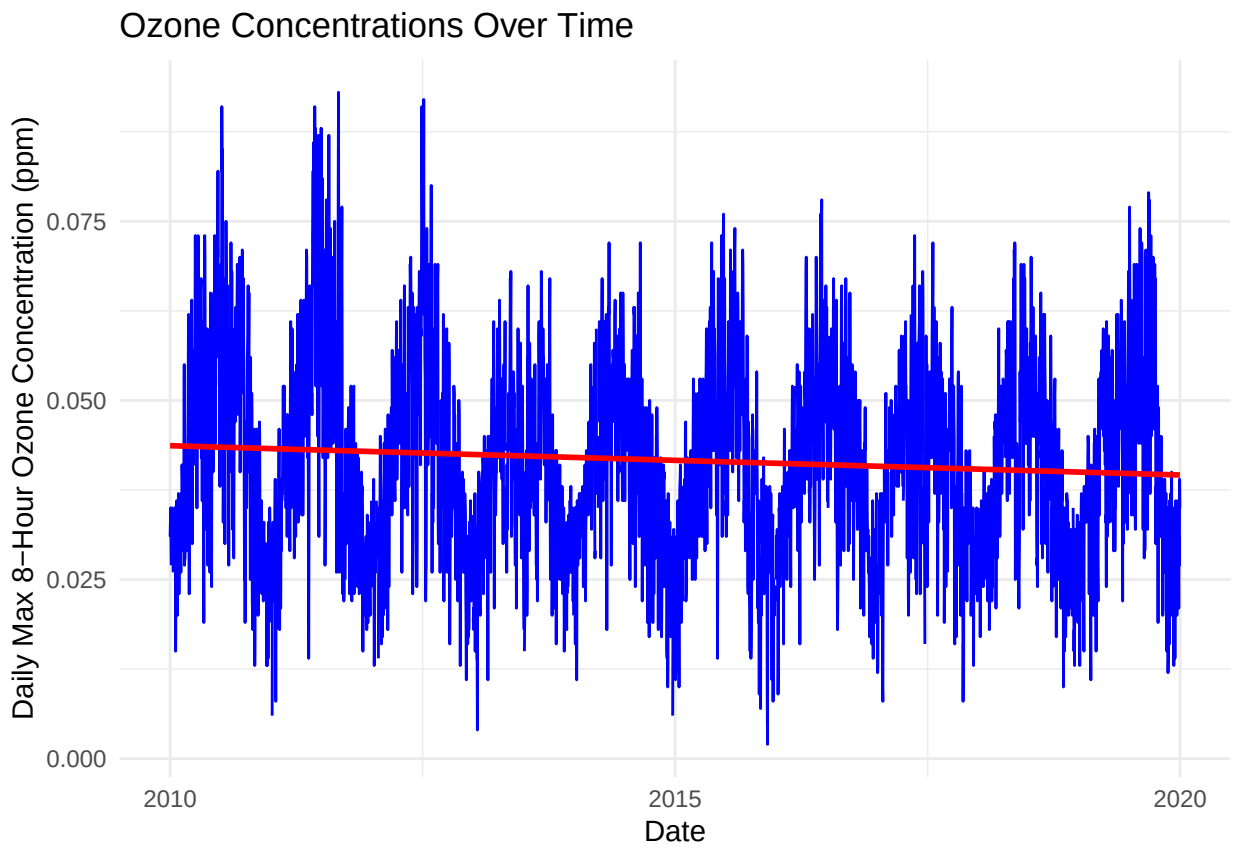
Visualize

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
# create line plot for ozone concentrations
ozone_plot <- ggplot(GaringerOzone, 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 = "Daily Max 8-Hour Ozone Concentration (ppm)") +
  theme_minimal()

print(ozone_plot)
```

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

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



Answer: The plot suggests that there is a decreasing trend 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?

```
#8
# use linear interpolation to fill in missing data
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Co

head(GaringerOzone)  # View the first few rows to see if NAs have been filled
```

```
##         Date Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
## 1 2010-01-01                                0.031              29
## 2 2010-01-02                                0.033              31
## 3 2010-01-03                                0.035              32
## 4 2010-01-04                                0.031              29
## 5 2010-01-05                                0.027              25
## 6 2010-01-06                                0.030              NA
```

Answer: We used a linear interpolation because it is more straightforward than a piecewise constant or spline interpolation, so it makes less assumptions and provides a clearer analysis.

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
# create new dataframe
GaringerOzone.monthly <- GaringerOzone %>%
  # add year and month columns
  mutate(Year = year(Date),
         Month = month(Date)) %>%
  # group by year and month
  group_by(Year, Month) %>%
  summarize(Mean_Ozone_Concentration = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE)) %>%
  ungroup() %>%
  # create new data column
  mutate(Date = as.Date(paste(Year, Month, "01", sep = "-"), format = "%Y-%m-%d"))
```

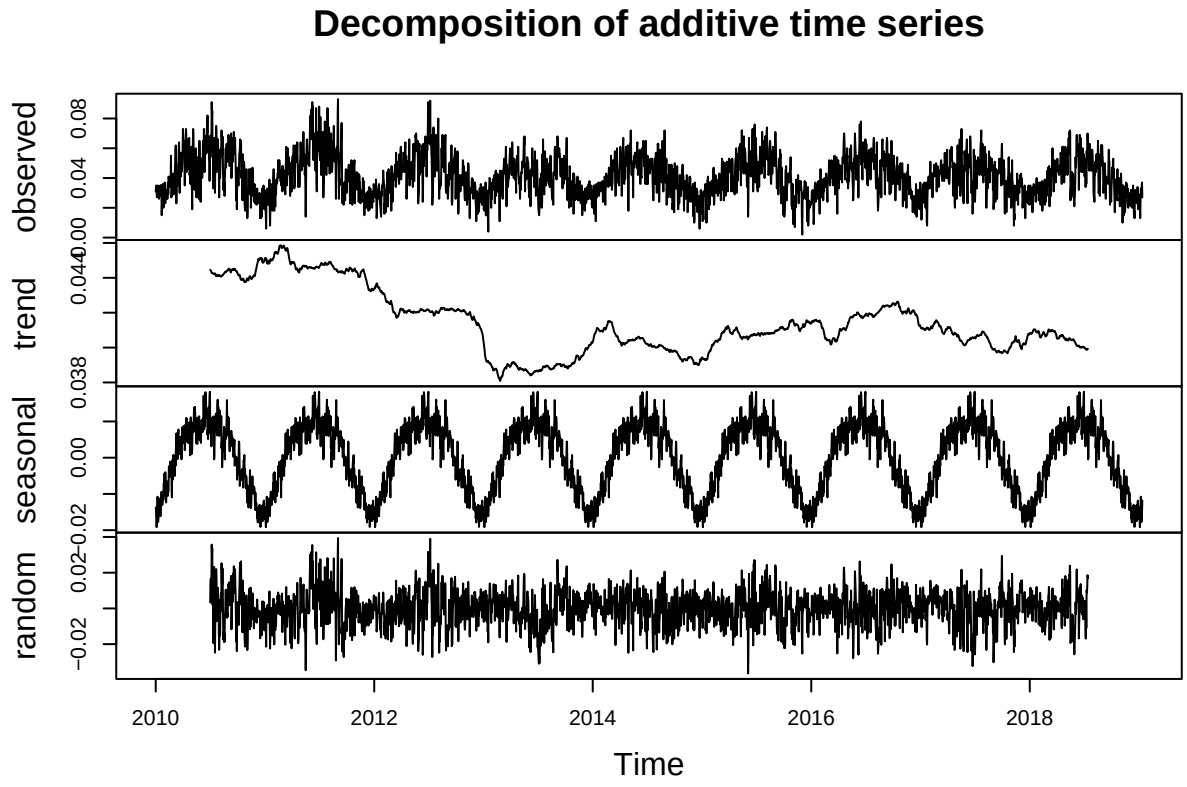
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

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.

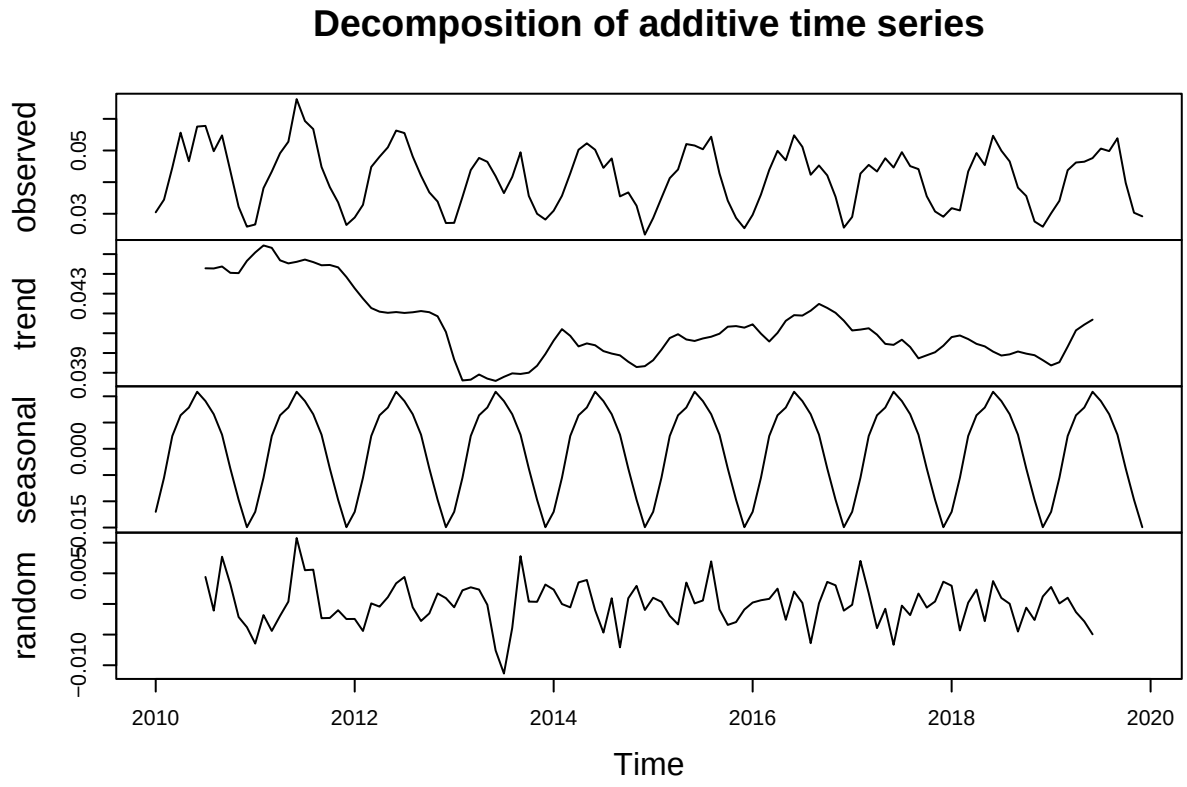
```
#10

# create a daily time series object
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010, 1),
                             end = c(2019, 12),
                             frequency = 365)

# create a monthly time series object
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone_Concentration,
                               start = c(2010, 1),
                               end = c(2019, 12),
                               frequency = 12)
```

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

```
#11
# decompose the daily time series object
GaringerOzone.daily.decomposed <- decompose(GaringerOzone.daily.ts)

# plot
plot(GaringerOzone.daily.decomposed)
```




```r
# decompose the monthly time series object
GaringerOzone.monthly.decomposed <- decompose(GaringerOzone.monthly.ts)

# plot
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

# run seasonal Mann-Kendall trend analysis
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone_Concentration,
                               start = c(2010, 1),
                               frequency = 12)

# run seasonal Mann-Kendall test
mann_kendall_result <- smk.test(GaringerOzone.monthly.ts)

print(mann_kendall_result)
```

```
##
##  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
```

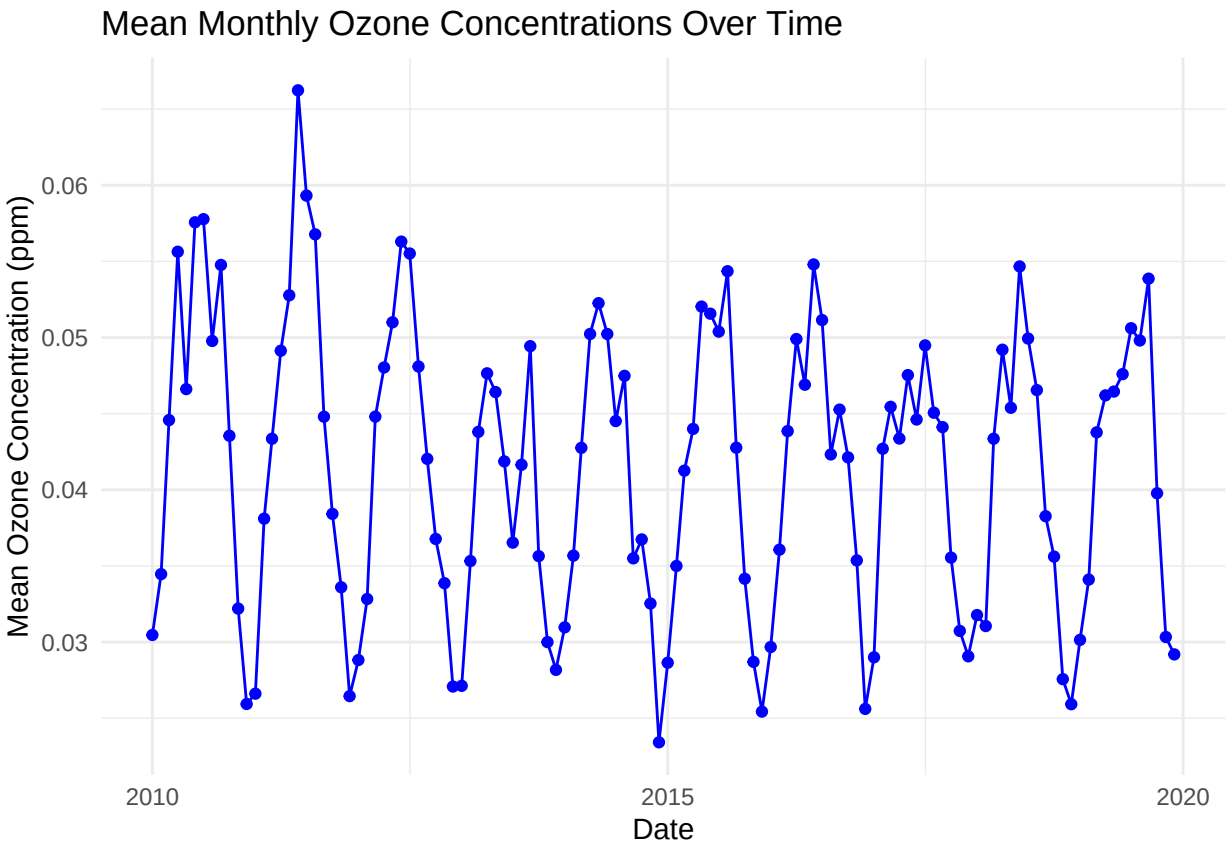
```r
summary(mann_kendall_result)
```

```
##
##  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: Ozone fluctuations appear to fluctuate seasonally, so it makes sense that the Mann-Kendall is most appropriate.

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

ozone_plot <- ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone_Concentration)) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  labs(title = "Mean Monthly Ozone Concentrations Over Time",
       x = "Date",
       y = "Mean Ozone Concentration (ppm)") +
  theme_minimal()
ozone_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: The Seasonal Mann-Kendall test shows a slight downward trend in monthly ozone concentrations from 2010 to 2019, with a z-value of -1.963 and a p-value of 0.04965, indicating this trend is statistically significant. However, most individual seasons did not show significant trends, as their p-values were above 0.05. Only December showed a positive trend, but it was not significant. Overall, while there are some changes, the trends in ozone levels are not consistent across the seasons.

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
# decompose the monthly time series
GaringerOzone.monthly.decomposed <- decompose(GaringerOzone.monthly.ts)

# extract the seasonal component
seasonal_component <- GaringerOzone.monthly.decomposed$seasonal

# subtract the seasonal component from the original series
```

```r
GaringerOzone.non_seasonal.ts <- GaringerOzone.monthly.ts - seasonal_component

# check the first few values of the non-seasonal series
head(GaringerOzone.non_seasonal.ts)
```

```
## [1] 0.04246981 0.03997928 0.04214067 0.04924980 0.03875421 0.04670050
```

```r
#16
library(trend)

# Run the Mann-Kendall test on the non-seasonal ozone series
non_seasonal_mann_kendall_result <- mk.test(GaringerOzone.non_seasonal.ts)

# Inspect the results
print(non_seasonal_mann_kendall_result)
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.non_seasonal.ts
## z = -2.6039, n = 120, p-value = 0.009216
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.149000e+03  1.943657e+05 -1.609356e-01
```

Answer: The non-seasonal Mann-Kendall test shows a significant decrease in ozone levels, with a z-value of -2.6039 and a p-value of 0.009216, meaning ozone concentrations are going down over time. In comparison, the seasonal Mann-Kendall test indicated a smaller trend (z = -1.963, p-value = 0.04965). This means that even though there are seasonal changes, the overall trend in ozone levels is clearer when we ignore those seasonal effects.