Assignment 8: Time Series Analysis

Siyu Dong

Spring 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

```
# getwd()

library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
library(dplyr)
library(ggplot2)

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.

```
GaringerNC2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                         stringsAsFactors = TRUE)
GaringerNC2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                         stringsAsFactors = TRUE)

GaringerOzone <- bind_rows(GaringerNC2010,
                           GaringerNC2011, GaringerNC2012, GaringerNC2013,
                           GaringerNC2014, GaringerNC2015, GaringerNC2016,
                           GaringerNC2017, GaringerNC2018, GaringerNC2019)
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
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

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

# 5
```

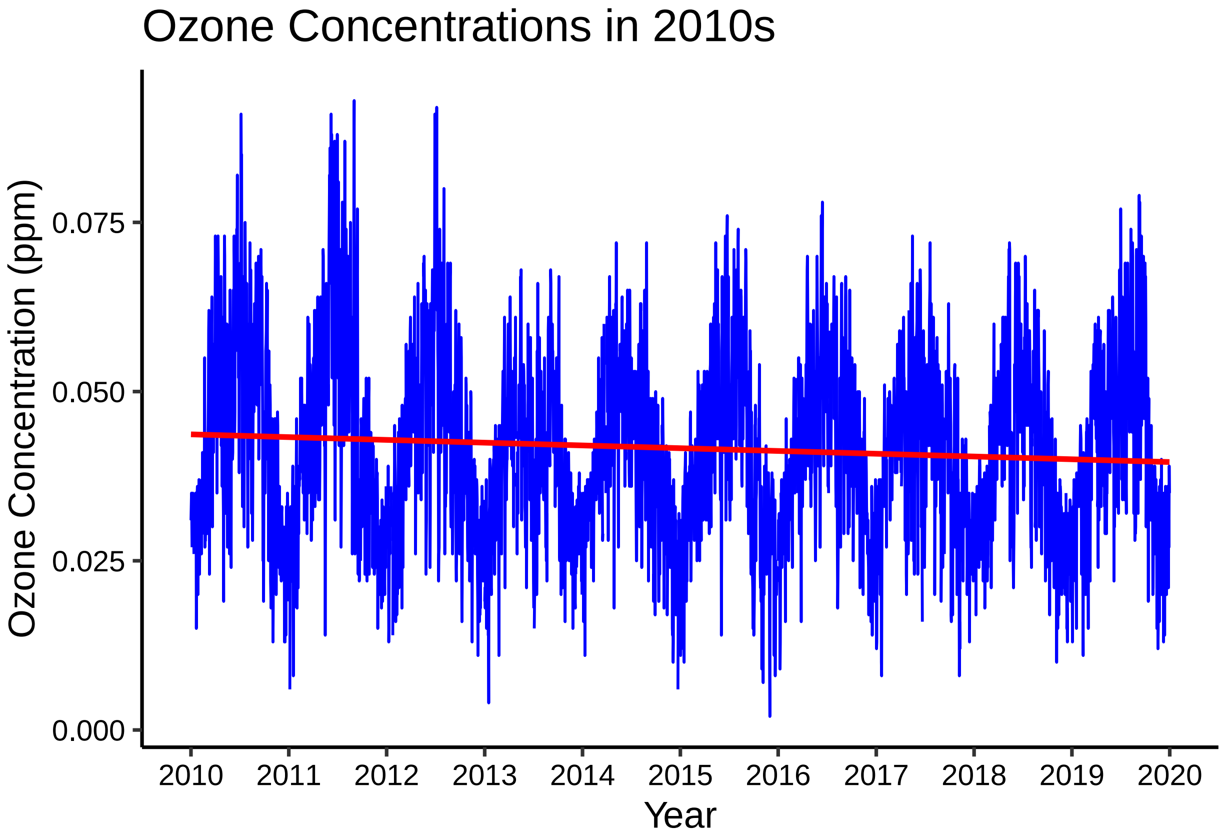
```r
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day"))
colnames(Days) <- "Date"

# 6
GaringerOzone <- left_join(Days, GaringerOzone_Selected, by = "Date")
dim(GaringerOzone)
```

```
## [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?

```r
#7
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(x = "Year", y = "Ozone Concentration (ppm)",
       title = "Ozone Concentrations in 2010s") +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
  mytheme
```



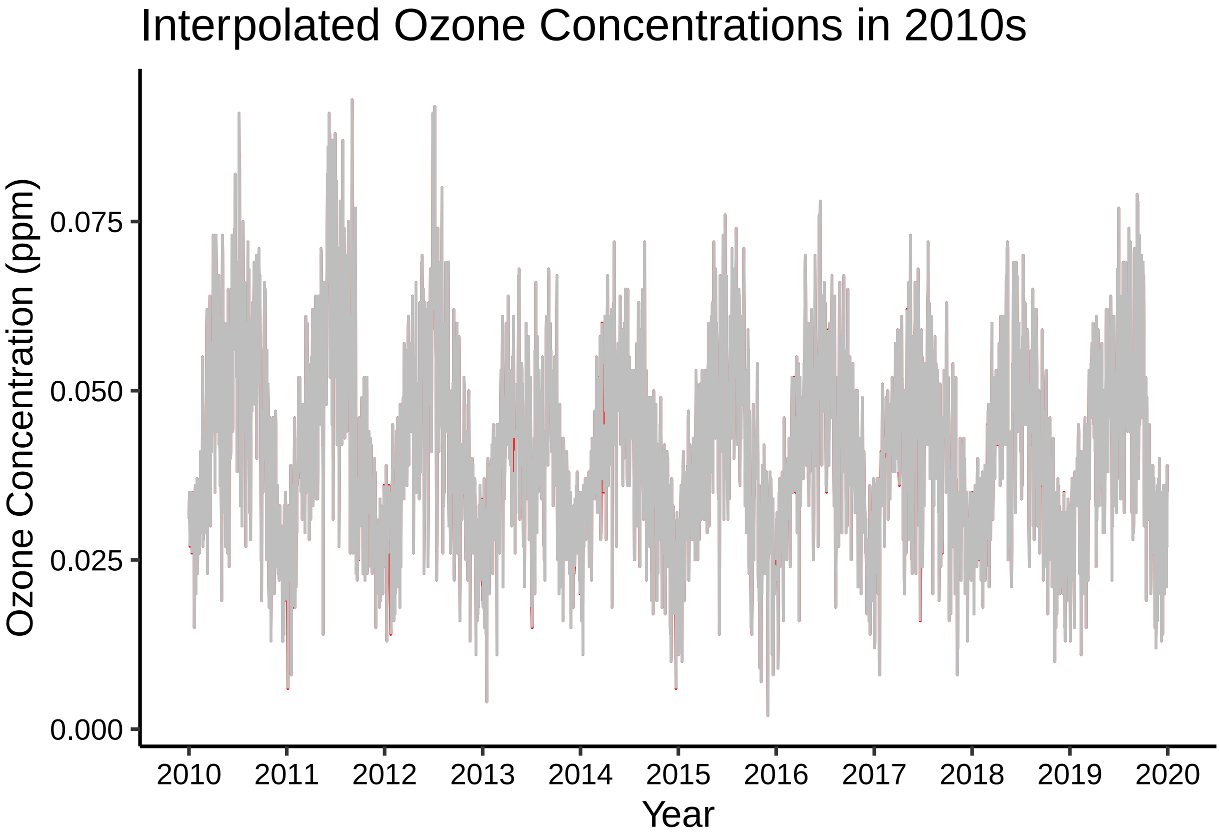
Answer: This plot suggests a slightly declining trend in ozone concentration over time.

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

ggplot(GaringerOzone_clean) +
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration.clean), color = "red") +
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration), color = "gray") +
  labs(x = "Year", y = "Ozone Concentration (ppm)",
       title = "Interpolated Ozone Concentrations in 2010s") +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
  mytheme
```



Answer: Piecewise constant interpolation assumes that the data changes abruptly at each known data point, which might not be appropriate for time series data with continuous changes over time. Spline interpolation, on the other hand, fits a smooth curve through the known data points, which can sometimes introduce unwanted complexity or smooth out important features in the data.

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 <- GaringerOzone %>%
  mutate(year = year(Date), month = month(Date))

GaringerOzone.monthly <- GaringerOzone %>%
  group_by(year, month) %>%
  summarise(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE))

GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(year, month))

head(GaringerOzone.monthly)
```

```
## # A tibble: 6 x 4
## # Groups:   year [1]
##    year month mean_ozone 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
```

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
f_day <- day(first(GaringerOzone_clean$Date))
f_year <- year(first(GaringerOzone_clean$Date))
GaringerOzone.daily.ts <-
  ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration.clean,
     start=c(f_year,f_day),
     frequency = 365)

f_month <- month(first(GaringerOzone.monthly$Date))
f_year <- year(first(GaringerOzone.monthly$Date))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                                start=c(f_year,f_month),
                                frequency = 12)

str(GaringerOzone.daily.ts)
```

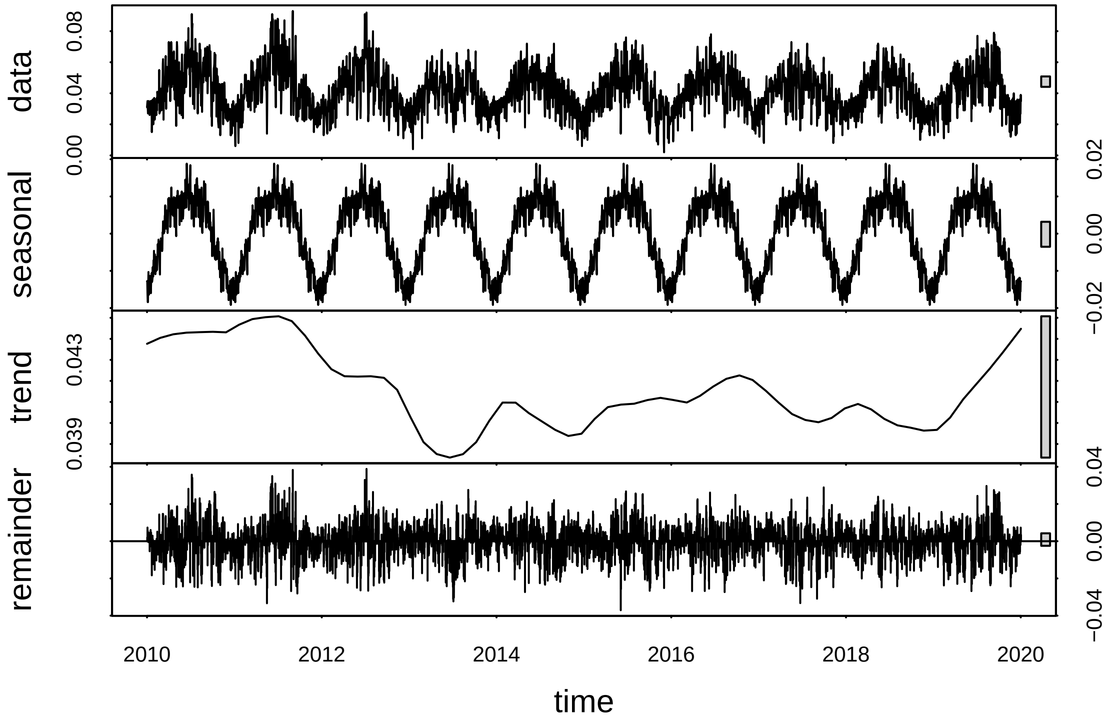
```
##  Time-Series [1:3652] from 2010 to 2020: 0.031 0.033 0.035 0.031 0.027 0.03 0.033 0.035 0.032 0.032
```

```
str(GaringerOzone.monthly.ts)
```

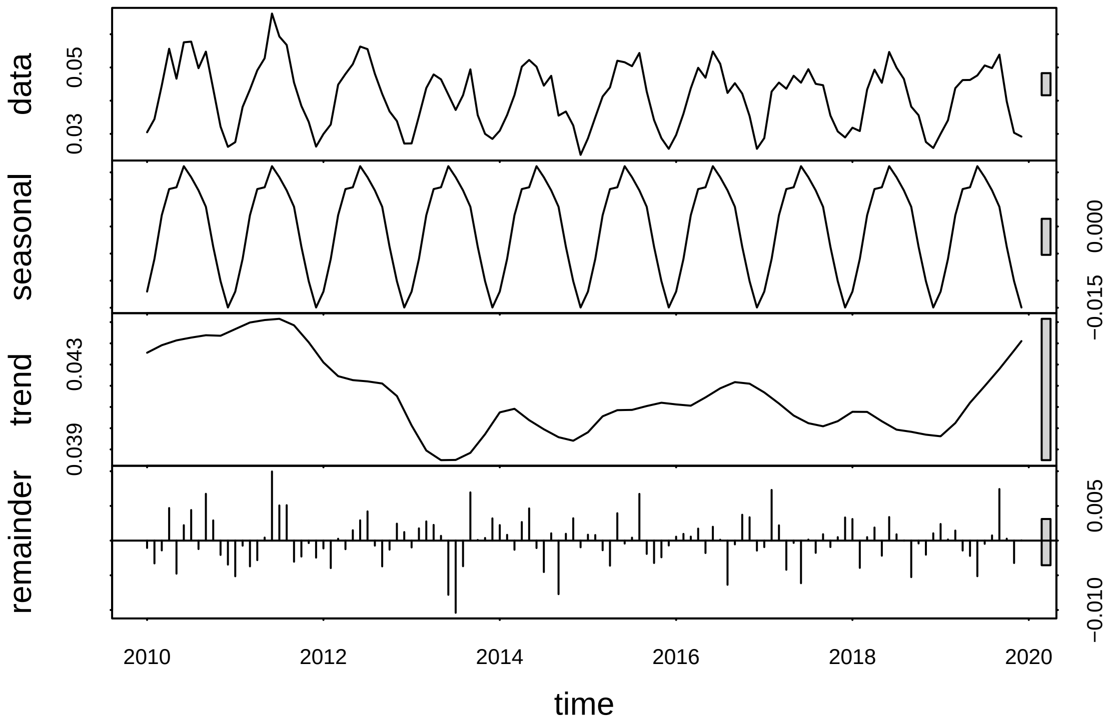
```
##  Time-Series [1:120] from 2010 to 2020: 0.0305 0.0345 0.0446 0.0556 0.0466 ...
```

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

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



```
GaringerOzone.monthly_decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly_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?

```r
#12
GaringerOzone.daily_trend <- Kendall::SeasonalMannKendall(GaringerOzone.daily.ts)
summary(GaringerOzone.daily_trend)
```

```
## Score =  -739 , Var(Score) = 45223.67
## denominator =  16213.86
## tau = -0.0456, 2-sided pvalue =0.00051075
```

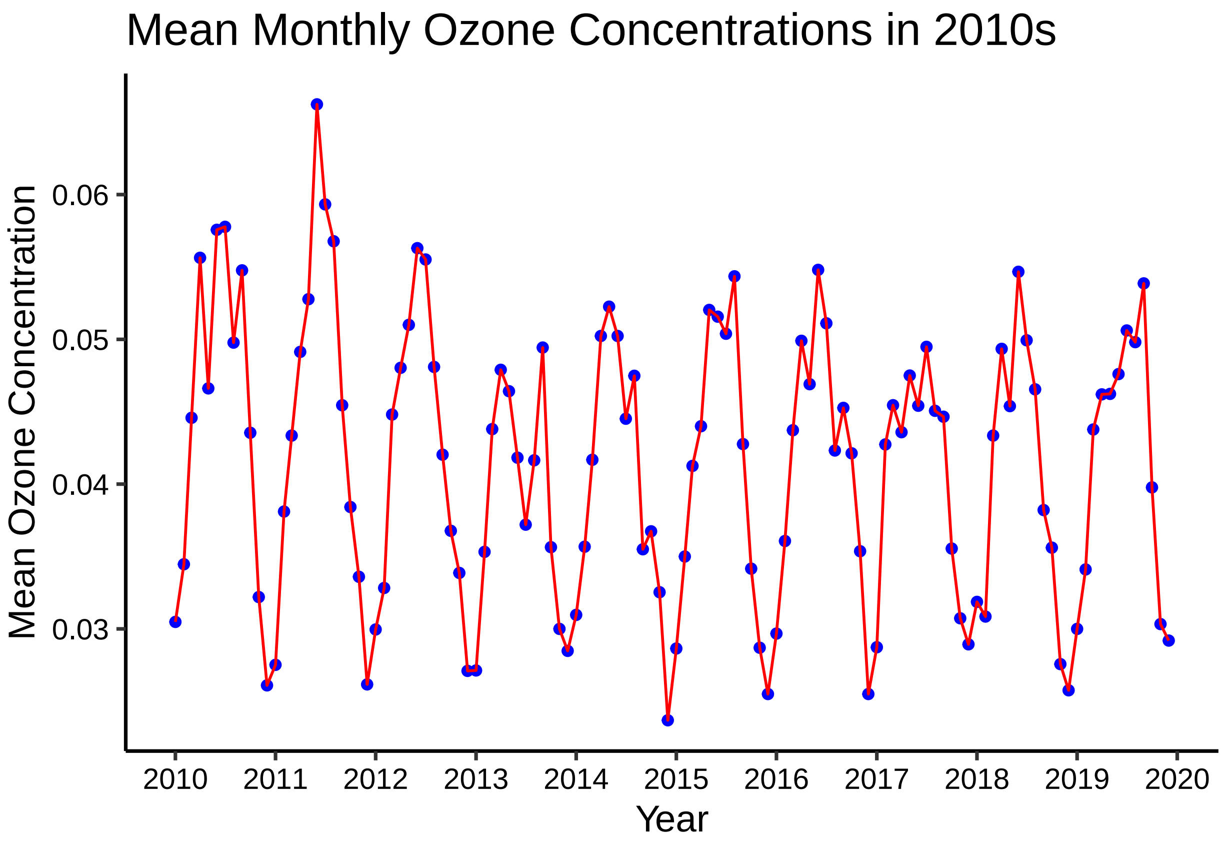
```r
GaringerOzone.monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly_trend)
```

```
## Score =  -88 , Var(Score) = 1498
## denominator =  538.9944
## tau = -0.163, 2-sided pvalue =0.022986
```

Answer: Monthly ozone concentrations often exhibit seasonal patterns due to various factors such as weather, temperature, and atmospheric conditions. These seasonal variations can introduce autocorrelation and complicate trend analysis. The SMK test adjusts for this seasonality by considering the data's periodic nature.

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
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
  geom_point(color = "blue") +
  geom_line(color = "red") +
  labs(x = "Year", y = "Mean Ozone Concentration",
       title = "Mean Monthly Ozone Concentrations in 2010s") +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
  mytheme
```



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: Based on the plot depicting mean monthly ozone concentrations over the 2010s at this station, we observe a decreasing trend in ozone concentrations over time. This observation aligns with the statistical test results, which indicate a statistically significant negative monotonic trend in the monthly ozone concentrations (tau = -0.163, p-value = 0.022986). Therefore, it appears that ozone concentrations have indeed changed over the 2010s at this station, showing a decreasing trend throughout the decade.

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
Garinger_monthly_non_seasonal_component <-
  GaringerOzone.monthly_decomp$time.series[, "remainder"]

#16
Garinger_monthly_non_seasonal_trend <-
  Kendall::SeasonalMannKendall(Garinger_monthly_non_seasonal_component)
summary(Garinger_monthly_non_seasonal_trend)
```

```
## Score =  36 , Var(Score) = 1500
## denominator =  540
## tau = 0.0667, 2-sided pvalue =0.35262
```

Answer:

- Seasonal Mann-Kendall Test on the Complete Series: The negative tau value (-0.163) indicates a moderate negative monotonic trend in the complete series after accounting for seasonality. The p-value (0.022986) suggests that this trend is statistically significant at the conventional significance level (e.g., 0.05).
- Non-Seasonal Mann-Kendall Test on the Complete Series: The positive tau value (0.0667) suggests a weak positive monotonic trend in the complete series after removing seasonality. However, the p-value (0.35262) indicates that this trend is not statistically significant at the conventional significance level.
- This difference highlights the importance of accounting for seasonality in trend analysis, as it can mask or exaggerate underlying trends.