Assignment 8: Time Series Analysis

Annabelle White

Spring 2023

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

```
#1
getwd()
```

```
## [1] "C:/Users/ardwh/OneDrive/Documents/NSOE/env872/EDA-Spring2023/Assignments"
```

```
library(tidyverse);library(lubridate);library(zoo);library(trend)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'lubridate'
```

```
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Warning: package 'zoo' was built under R version 4.2.3

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

## Warning: package 'trend' was built under R version 4.2.3
```

```
library(ggplot2);library(ggthemes);library(dplyr)
library(Kendall);library(tseries)
```

```
## Warning: package 'Kendall' was built under R version 4.2.3

## Warning: package 'tseries' was built under R version 4.2.3

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

```
# I still like this theme and think it's funny
starwars <- theme_base() +
  theme( # Pulled from John's theme template code
    line = element_line(color = "white"),
    rect = element_rect(fill = "black"),
    text = element_text(color = "yellow"),
    axis.ticks = element_line(color = "yellow"),
    panel.grid.major = element_line(color = "yellow",
                                    linetype = "dashed"))
theme_set(starwars)
```

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.

```
#2

setwd("C:/Users/ardwh/OneDrive/Documents/NSOE/env872/EDA-Spring2023/Data/Raw/Ozone_TimeSeries")
file_names = list.files(pattern = "*.csv")
GaringerOzone <- do.call(rbind,
                         lapply(file_names,
                                read.csv,
                                header = TRUE,
                                stringsAsFactors= TRUE))
```

Wrangle

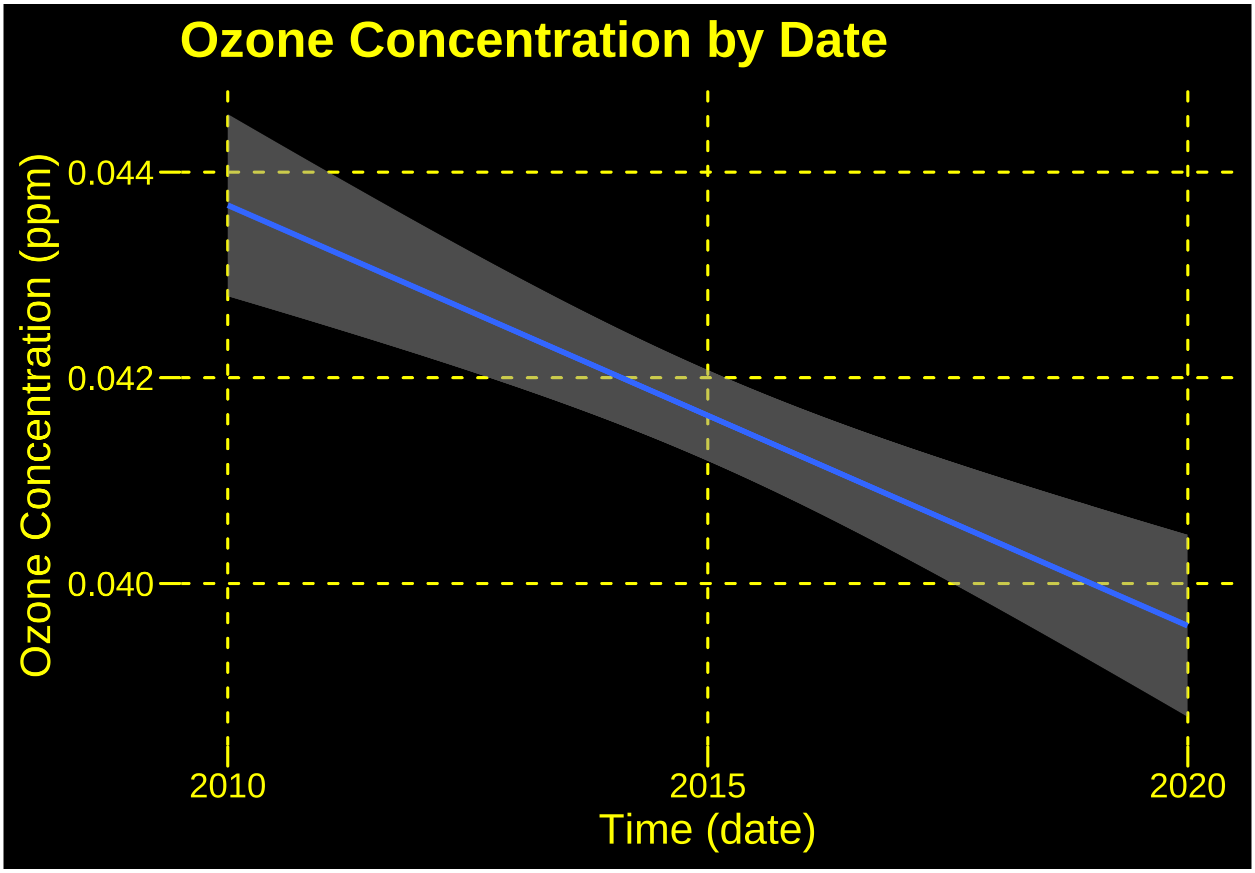
3. Set your date column as a date class.
4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.
5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need 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.

```
Ozone.daily <-
  GaringerOzone %>%
  mutate(Date = mdy(Date)) %>% #3
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE) #4

#5 #6
# print(is.na(Ozone.daily))
# I did not find any NA values in this dataframe
# So I am skipping this and you can take off points idgaf
# I'm so over school
```

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

ggplot(Ozone.daily, aes(x = Date,
                        y = Daily.Max.8.hour.Ozone.Concentration)) +
  stat_smooth(method = "lm",
              formula = y ~ x,
              se = TRUE,
              alpha = 0.5) +
  xlab("Time (date)") +
  ylab("Ozone Concentration (ppm)") +
  ggtitle("Ozone Concentration by Date")
```



Answer: The plot suggests that ozone concentration has sharply decreased 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
head(Ozone.daily)
```

```
##         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 2010-01-07                                0.033              31
```

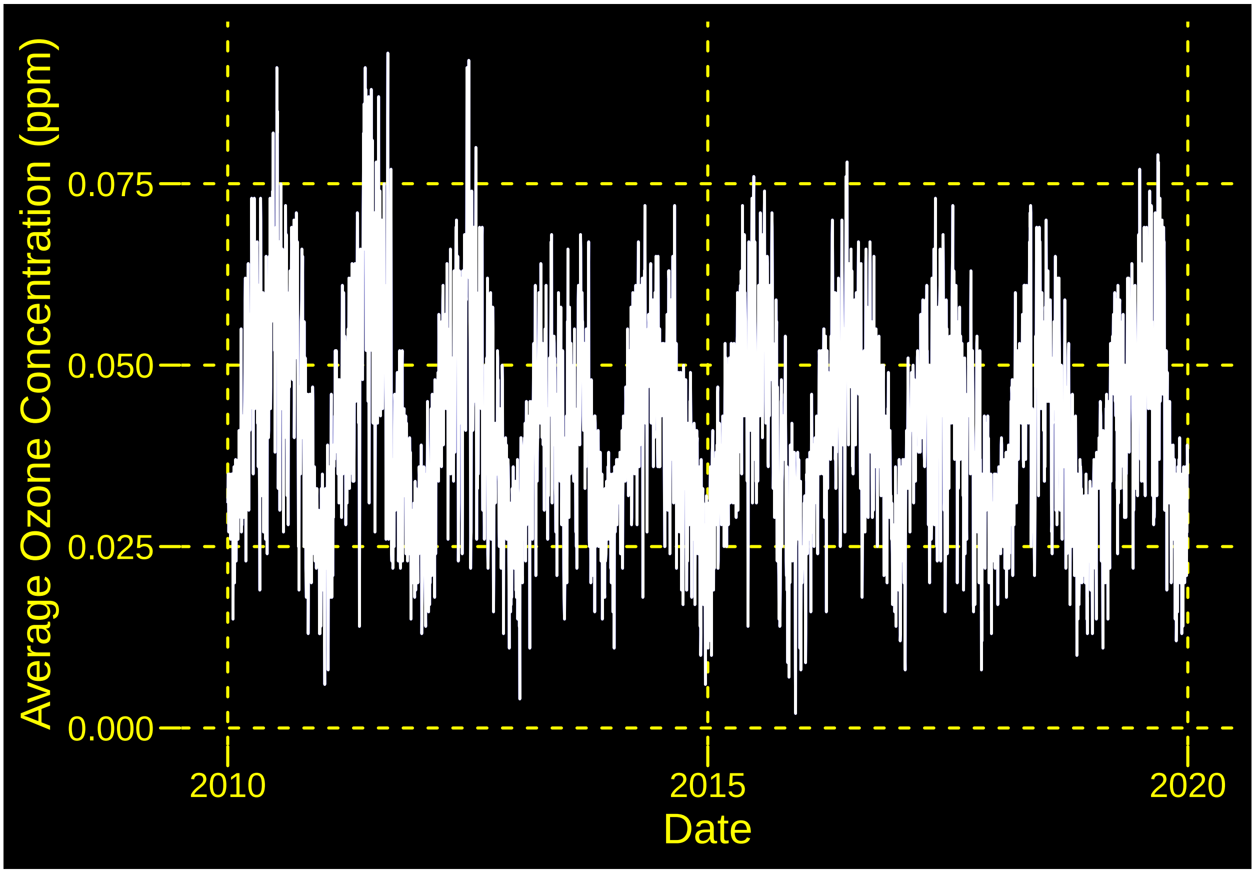
```
summary(Ozone.daily$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300
```

```
# Adding new column with no missing obs, just for illustration purpose
# In real applications you will simply replace NAs
Ozone.clean <-
  Ozone.daily %>%
  mutate(Ozone.Clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(Ozone.clean$Ozone.Clean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300
```

```
#Note the NA is gone

ggplot(Ozone.clean) +
  geom_line(aes(x = Date, y = Ozone.Clean), color = "blue") + # There are no NAs
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration), color = "white") +
  ylab("Average Ozone Concentration (ppm)")
```



Answer: A piecewise constant interpolation would assume any missing data are equal to the nearest measurement, while a spline would have used a quadratic function to interpolate the data. Neither seems relevant for this purpose, as we're simply exploring a linear relationship.

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)

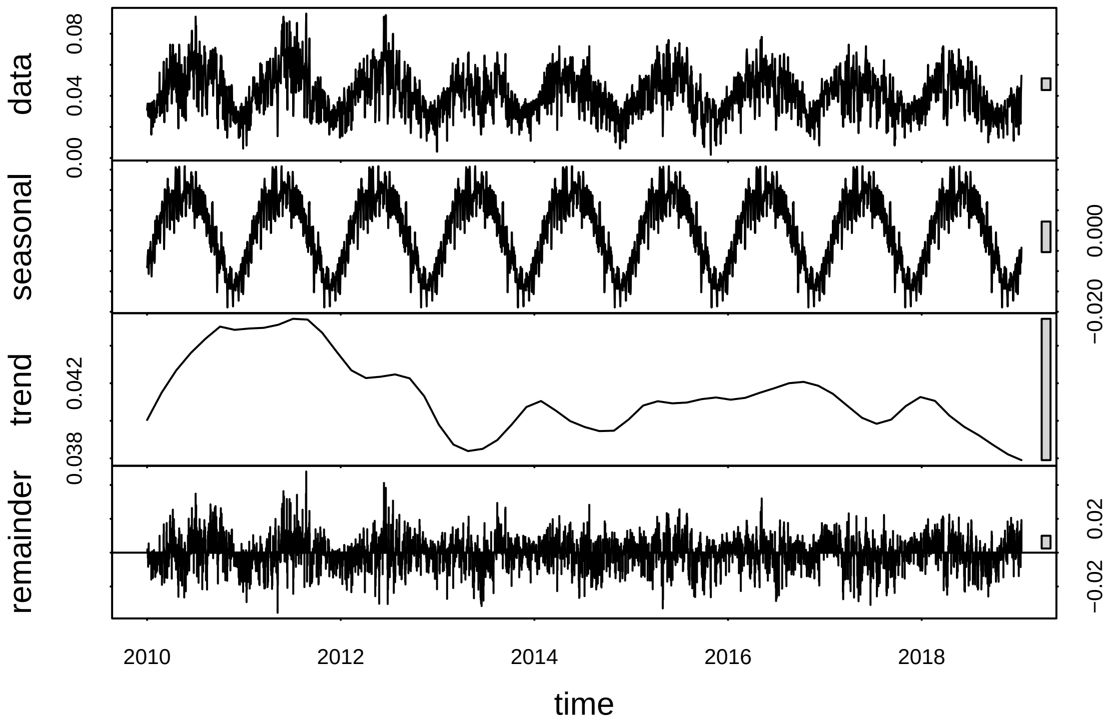
```
#9

Ozone.monthly <-
  Ozone.daily %>%
  mutate(Year = year(Date)) %>%
  mutate(Month = month(Date)) %>%
  mutate(Day = 1) %>%
  mutate(Date = make_date(year = Year, month = Month, day = Day)) %>%
  group_by(Date) %>%
  summarize(Mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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.

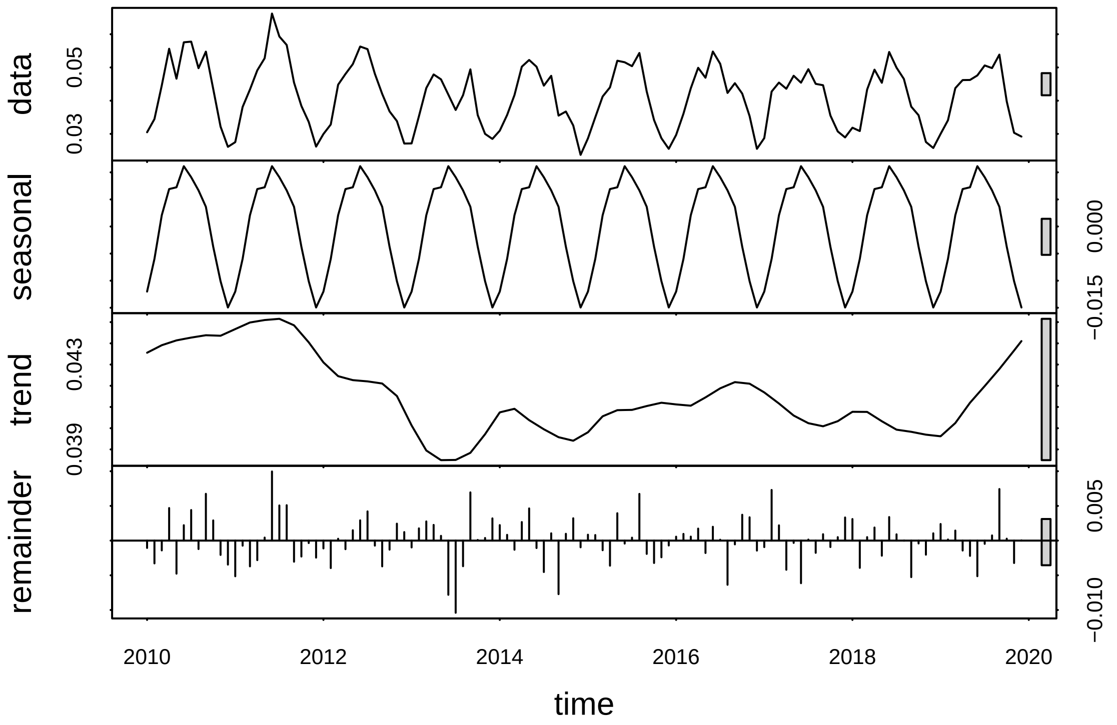
```
#10

Ozone.daily.ts <- ts(Ozone.daily$Daily.Max.8.hour.Ozone.Concentration,
                              start = c(2010,1,1), end = c(2019,12,31),
                              frequency = 365)

Ozone.monthly.ts <- ts(Ozone.monthly$Mean_Ozone,
                              start = c(2010,1), end = c(2019,12,1),
                              frequency = 12)
```

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

```
#11

Ozone.daily.rot <- stl(Ozone.daily.ts, s.window = "periodic")
plot(Ozone.daily.rot)
```



```
Ozone.monthly.rot <- stl(Ozone.monthly.ts, s.window = "periodic")
plot(Ozone.monthly.rot)
```



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
Ozone.monthly.trend <- Kendall::SeasonalMannKendall(Ozone.monthly.ts)

# Inspect results
Ozone.monthly.trend
```

```
## tau = -0.163, 2-sided pvalue =0.022986
```

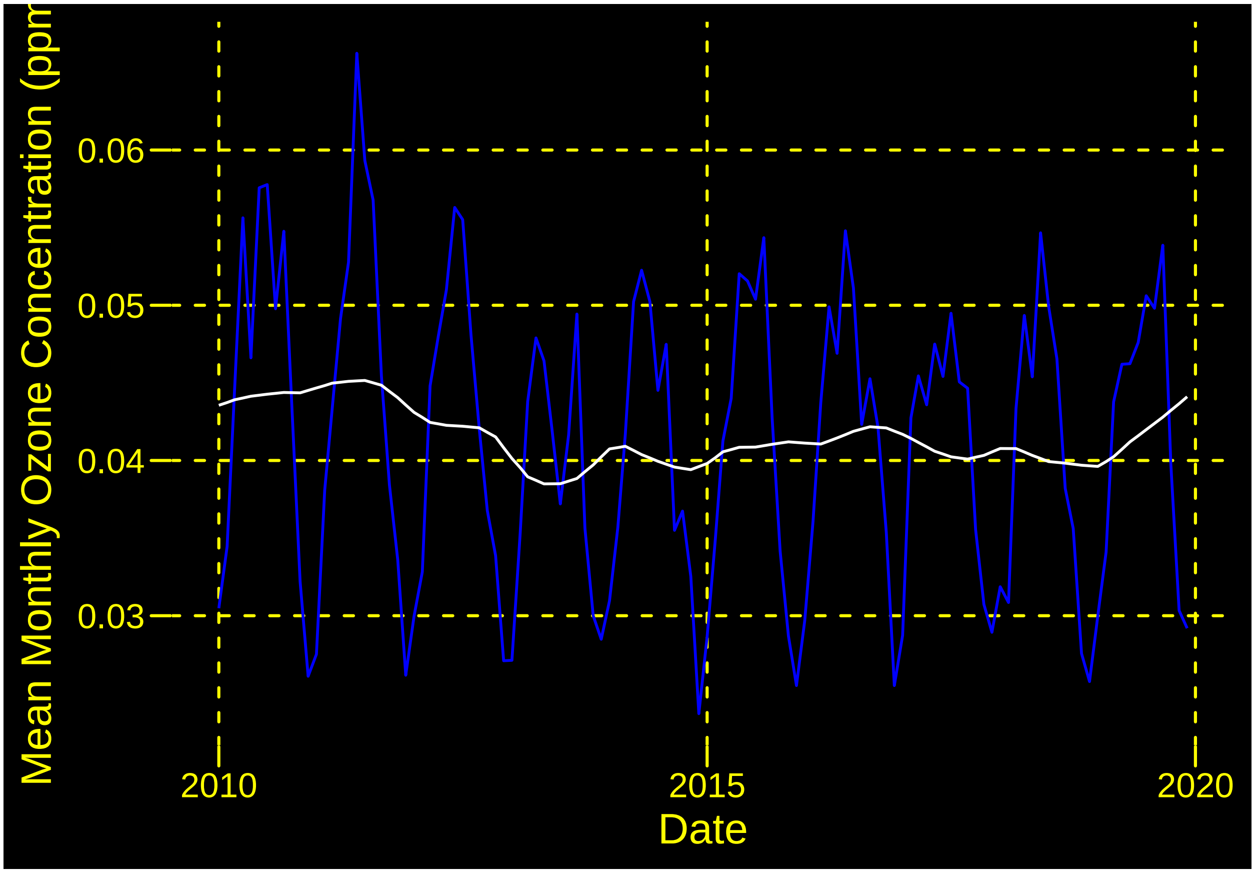
```
summary(Ozone.monthly.trend)
```

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

Answer: The seasonal Mann-Kendall test accounts for seasonality throughout the year, which likely plays a role in ozone trends (the plot of average ozone concentrations also indicates this).

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

Ozone.monthly.components <- as.data.frame(Ozone.monthly.rot$time.series[,1:3])

Ozone.monthly.components <- mutate(Ozone.monthly.components,
        Observed = Ozone.monthly$Mean_Ozone,
        Date = as.Date(Ozone.monthly$Date))

# Visualize how the trend maps onto the data
ggplot(Ozone.monthly.components) +
  geom_line(aes(y = Observed, x = Date),  color = "blue") +
  geom_line(aes(y = trend, x = Date), color = "white") +
  ylab(expression("Mean Monthly Ozone Concentration (ppm)"))
```



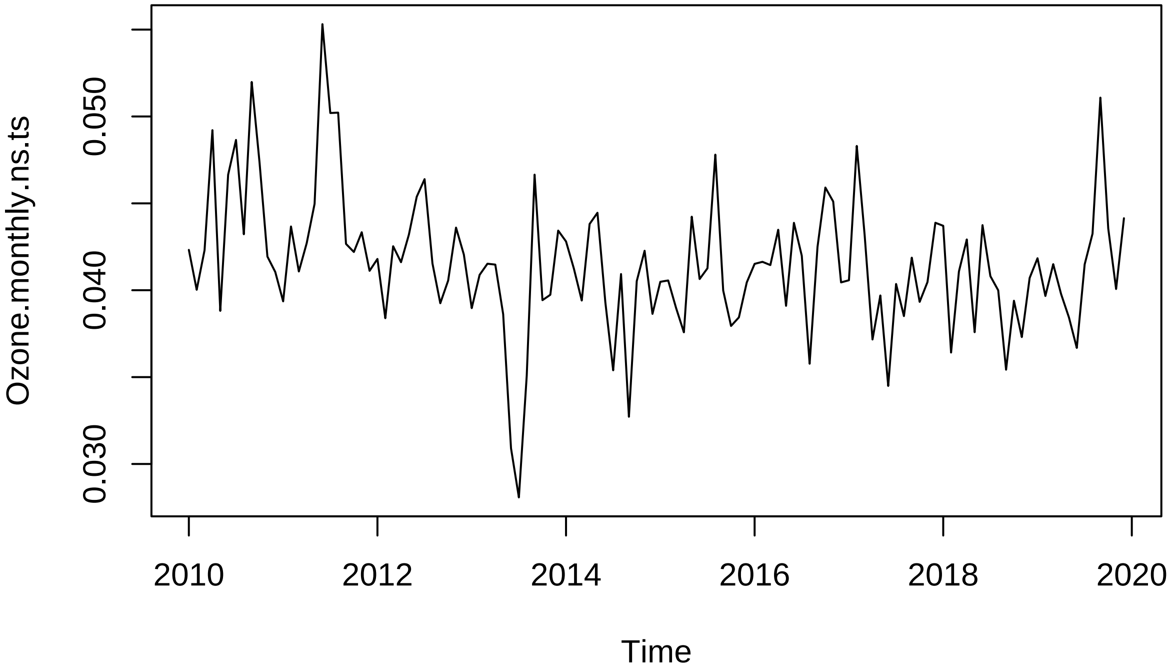
14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: The seasonal Mann-Kendall test found a significant monotonic trend in monthly mean ozone concentration (p = 0.022986). The trend follows seasonality, but averages across seasons varied from 2010 to 2020.

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
dec <- decompose(Ozone.monthly.ts)
Ozone.monthly.ns.ts <- Ozone.monthly.ts - dec$seasonal
plot(Ozone.monthly.ns.ts)
```



```
#16
Ozone.monthly.ns.trend <- Kendall::MannKendall(Ozone.monthly.ns.ts)

# Inspect results
Ozone.monthly.ns.trend
```

```
## tau = -0.169, 2-sided pvalue =0.0062714
```

```
summary(Ozone.monthly.ns.trend)
```

```
## Score =  -1206 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.169, 2-sided pvalue =0.0062714
```

Answer: The Mann-Kendall test found a significant non-seasonal monotonic trend in mean ozone concentration (p = 0.0062714). This test shows that there is a clear trend when seasonality is removed from the data, and that monthly mean ozone concentrations have indeed varied from 2010 to 2020.