Assignment 8: Time Series Analysis

Sarah Kear

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

```
#1
library(here)
```

```
## here() starts at /Users/sarah/Documents/872_EDA/EDA_Spring2024
```

```
here
```

```
## function (...)
## {
##     .root_env$root$f(...)
## }
## <bytecode: 0x7f88b0a78518>
## <environment: namespace: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   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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)
```

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

```
library(trend)

A08theme <- theme_classic(base_size = 12) +
  theme(
    plot.title = element_text(
      color= 'black',
      size = 12 #decreased font size
    ),
    axis.text = element_text(
      color = "black",
      size = 10
      ),
   axis.title.x = element_text( #Updating x-axis
     color = "black",
     size = 11
   ),
   axis.title.y = element_text( #Updating y-axis
     color = "black",
     size = 11
   ),
    plot.background = element_blank() #removing plot edge/background
   )
theme_set(A08theme)
```

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
O3_Garinger_2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                             stringsAsFactors = TRUE)
O3_Garinger_2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                             stringsAsFactors = TRUE)

GaringerOzone <- rbind(O3_Garinger_2010,
                       O3_Garinger_2011,
                       O3_Garinger_2012,
                       O3_Garinger_2013,
                       O3_Garinger_2014,
                       O3_Garinger_2015,
                       O3_Garinger_2016,
                       O3_Garinger_2017,
                       O3_Garinger_2018,
                       O3_Garinger_2019)
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.

```r
#3
GaringerOzone$Date <- as.Date(
  GaringerOzone$Date,
  format = "%m/%d/%Y")

head(GaringerOzone$Date,
     5)
```

```
## [1] "2010-01-01" "2010-01-02" "2010-01-03" "2010-01-04" "2010-01-05"
```

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

head(GaringerOzone,
     5)
```

```
##         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
```

```r
#5
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,
                           by = "Date")

dim(GaringerOzone)
```

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

```r
head(GaringerOzone, 5)
```

```
##         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
```

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?

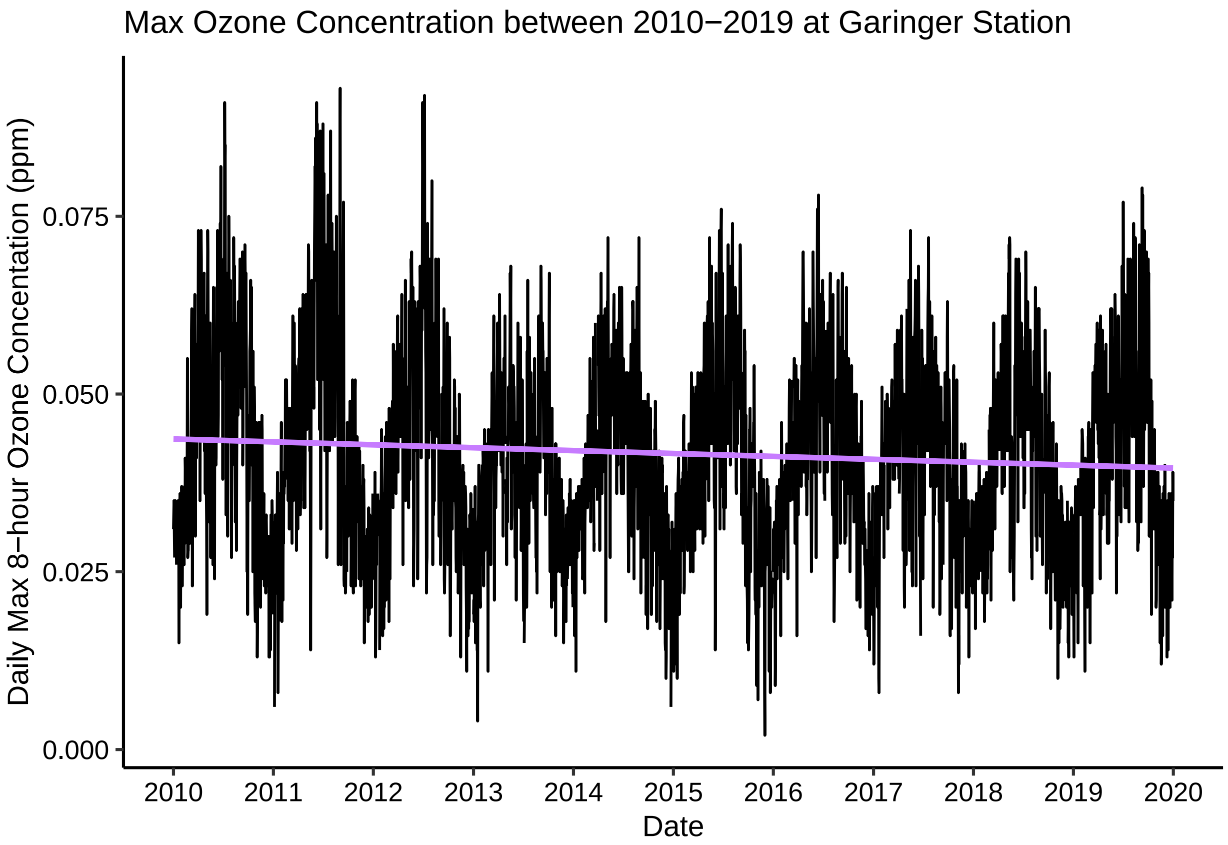
```
#7
plot1 <- GaringerOzone %>%
  ggplot(aes(x = Date,
             y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(size = 0.5) +
  geom_smooth(method = "lm", se = FALSE, col="#c77cff") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  ylab("Daily Max 8-hour Ozone Concentation (ppm)") +
  ggtitle("Max Ozone Concentration between 2010-2019 at Garinger Station") +
  A08theme
```

```
## 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.
```

```
plot1
```

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

```
## Warning: Removed 63 rows containing non-finite values (‘stat_smooth()‘).
```



Answer: Yes there seems to be a trend overall and within each year. Within each year, the middle of the year seems to have greater Ozone concetrations than the beginning and end of the year, potentially indicating seasonality. The purple trend line suggests that between 2010 and 2019 there has been a decrease in daily Ozone concetration.

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

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

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

Answer: In this case we wouldn't use the piecewise interpolation because piecewise assign the missing values as the values closet to the missing day. This may not be the most approriate as Ozone concetration changes per day. We also wouldn't use spline interpolation because it is a bit more complex than linear interpolation, and the ozone data fits best with a linear function rather than a quadratic function. Linear interpolation is the best because it produces an avaerage of the before and after which best fits as the changes in Ozone concetrations are gradual.

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 %>%
  mutate(year = lubridate::year(Date),
         month = lubridate::month(Date, label = TRUE, abbr = FALSE)) %>%
  group_by(year, month) %>%
  summarise(mean.Ozone.Concentration = mean(Daily.Max.8.hour.Ozone.Concentration,
                                            na.rm = TRUE))
```

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

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(year, month, 1))

head(GaringerOzone.monthly, 5)
```

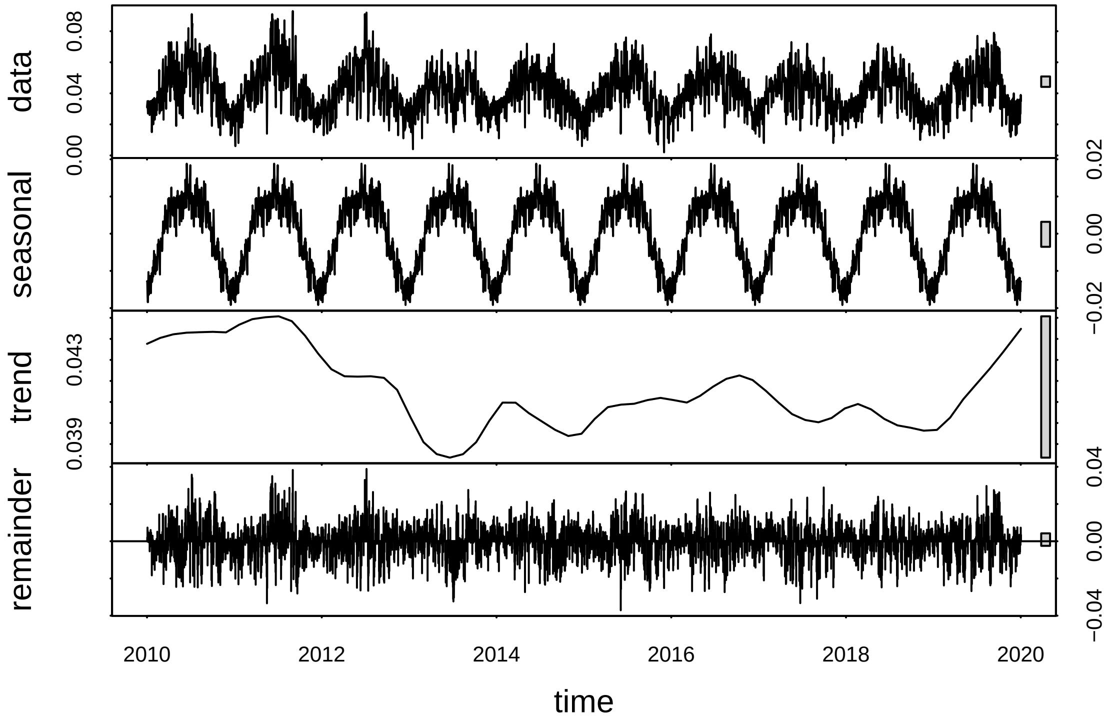
```
## # A tibble: 5 x 4
## # Groups:   year [1]
##    year month    mean.Ozone.Concentration Date      
##   <dbl> <ord>                       <dbl> <date>    
## 1  2010 January                    0.0305 2010-01-01
## 2  2010 February                   0.0345 2010-02-01
## 3  2010 March                      0.0446 2010-03-01
## 4  2010 April                      0.0556 2010-04-01
## 5  2010 May                        0.0466 2010-05-01
```

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.

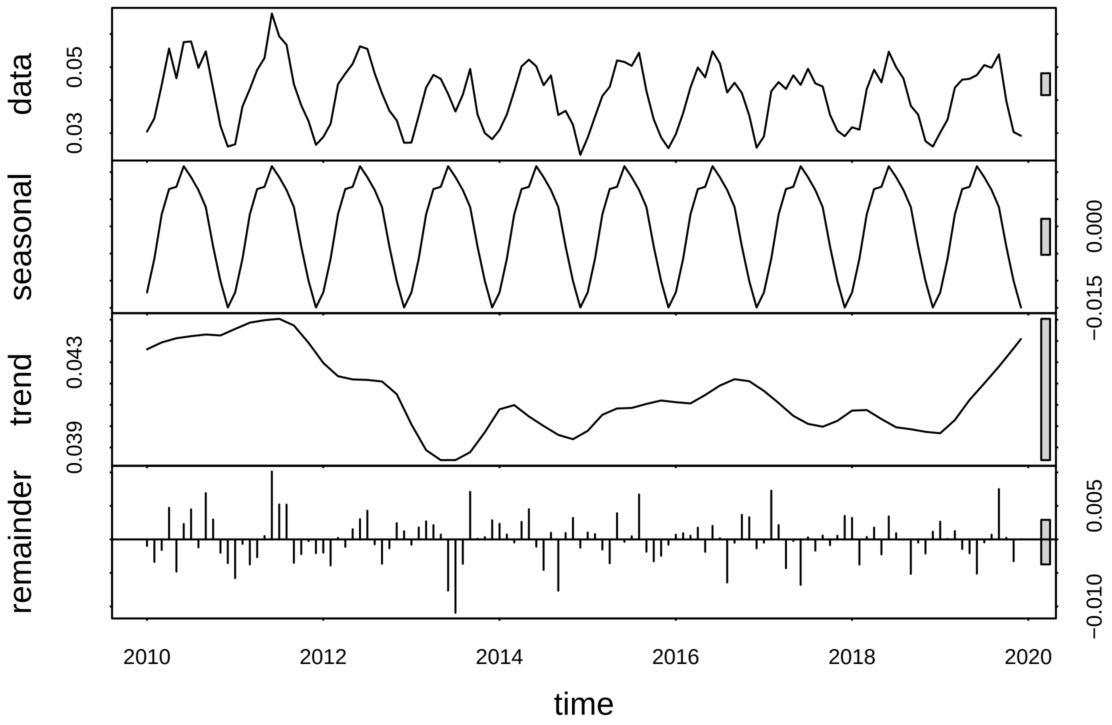
```
#10
f_month <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                              start = c(f_year,f_month),
                              frequency = 365)

f_month1 <- month(first(GaringerOzone.monthly$Date))
f_year1 <- year(first(GaringerOzone.monthly$Date))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.Ozone.Concentration,
                   start=c(f_year1,f_month1),
                   frequency=12)
```

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?

```
#12
GaringerOzone.monthly.trend <- trend::smk.test(GaringerOzone.monthly.ts)
GaringerOzone.monthly.trend
```

```
##
##  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(GaringerOzone.monthly.trend)
```

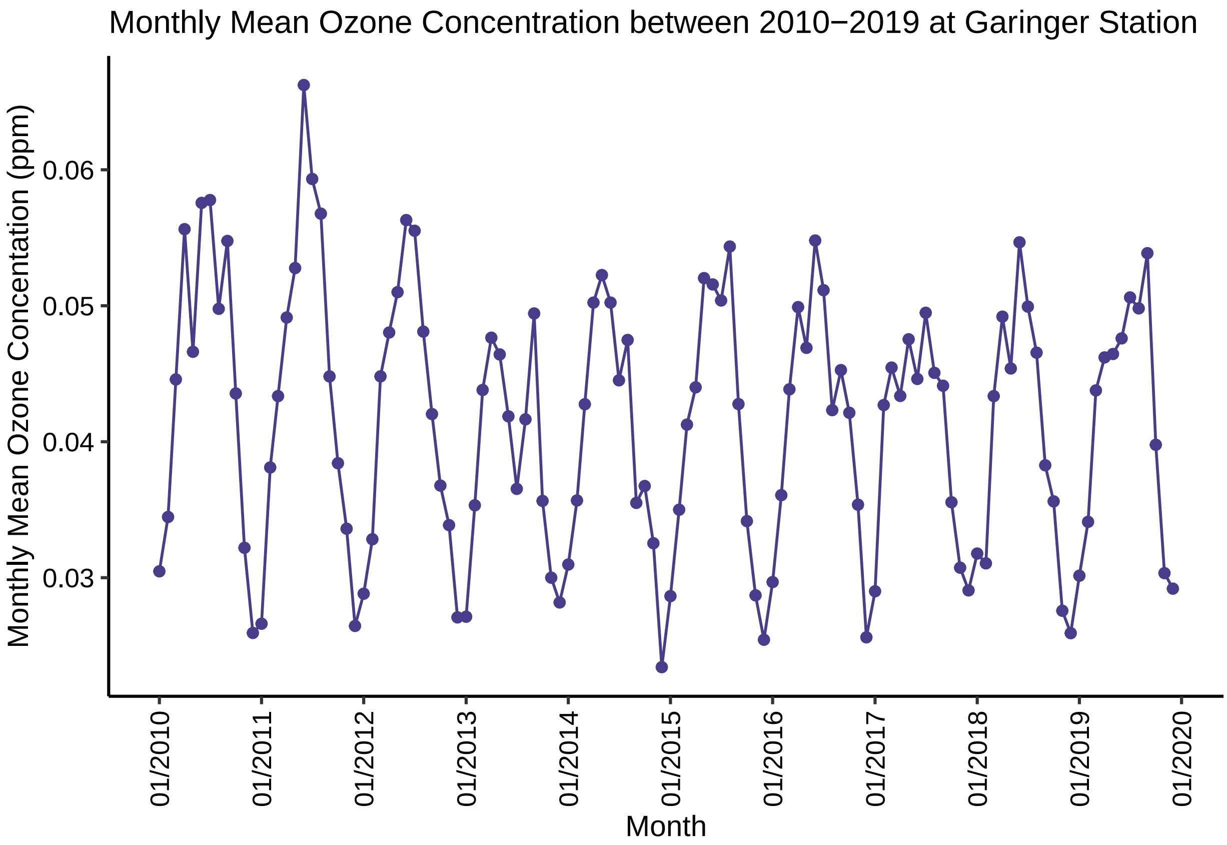
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
```

```
##
## 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: The seasonal Mann-Kendall test is the most appropriate because it evaluates to see if there is a monotonic trend in Ozone concentration and also accounts for the that seasonality of Ozone concentration throughout the 2010s.

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
plot2 <-ggplot(GaringerOzone.monthly,
               aes(x = Date,
                   y = mean.Ozone.Concentration)) +
  geom_point(color = "darkslateblue") +
  geom_line(color = "darkslateblue") +
  scale_x_date(date_breaks = "1 year", date_labels = "%m/%Y") +
  ylab("Monthly Mean Ozone Concentation (ppm)") +
  xlab("Month") +
  ggtitle("Monthly Mean Ozone Concentration between 2010-2019 at Garinger Station") +
  A08theme +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=.5))

print(plot2)
```



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: There is a present negative trend in Ozone concentrations changing over the 2010s at the Garinger High School station (Score = -77, z = -1.963, 2-sided p-value = 0.04965). Since the p=value is less than 0.05, we can reject the null hypothesis that there is no trend. As the score is -77, we conclude that there is a negative trend in Ozone concentration throughout the 2010s. This decreasing trend is also illustrated in the line graph as there is a gradual decrease in average monthly Ozone concentrations per year.

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
Monthy.O3.Components <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])

Monthy.O3.Components <- mutate(Monthy.O3.Components,
        Observed = GaringerOzone.monthly$mean.Ozone.Concentration,
        Date = GaringerOzone.monthly$Date)

head(Monthy.O3.Components, 5)
```

```
##       seasonal      trend     remainder   Observed       Date
## 1 -0.012164159 0.04360892 -0.0009770197 0.03046774 2010-01-01
## 2 -0.005945745 0.04377124 -0.0033612105 0.03446429 2010-02-01
## 3  0.002231834 0.04393356 -0.0015847518 0.04458065 2010-03-01
## 4  0.006878411 0.04403138  0.0047235448 0.05563333 2010-04-01
## 5  0.007292088 0.04412919 -0.0048083781 0.04661290 2010-05-01
```

```
monthly.subtract.ts <- GaringerOzone.monthly.ts - Monthy.O3.Components$seasonal

#16
monthly.O3.trend1 <- trend::mk.test(monthly.subtract.ts)
monthly.O3.trend1
```

```
##
##  Mann-Kendall trend test
##
## data:  monthly.subtract.ts
## 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: Both monthly series have a p-value of less than 0.05, meaning that both monthly series, seasonal or not, can reject the null hypothesis. There is statistically significant trend in both seasonal and non-seasonal series. Both also have a negative score, indicating a negative trend in either series. The non-seasonal monthly series has a p-value far less than the seasonal series (p = 0.00754), indicating it has a greater statistical significanc than the seasonal series. Furthermore, the non-seasonal series has a far higher score of -1,179. This means that is has a more significant montonic, negative trend than the seasonal series. By removing the seasonality from the monthly series, we have a far more statistically significant and monotonic negative trend in Ozone concentration at the Garinger High School station during the 2010s.