Assignment 7: Time Series Analysis

Tasha Griffiths

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
#1 set up session
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```
library(trend)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(Kendall)

getwd()
```

```
## [1] "C:/Users/Tasha Griffiths/Documents/Duke Year 1/Spring 22 Classes/Environmental Data Analytics/G
```

```
#2 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.

```
#2 download 10 datasets
Ozone.2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                       stringsAsFactors = TRUE)
Ozone.2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                         stringsAsFactors = TRUE) #got to be a better way!

#import all at once instead
library(data.table)
#reset working directory to file
setwd("./Data/Raw/Ozone_TimeSeries/")
#import all cvs in the folder
ozone.10.files <- list.files(pattern = "*.csv")
#combine all
ozone_dataset = do.call(rbind, lapply(ozone.10.files, fread))
#remove file
rm(Ozone.10.files)
```

```
## Warning in rm(Ozone.10.files): object 'Ozone.10.files' not found
```

```
#turn into a dataframe
ozone_combined_dataset <- as.data.frame(unclass(ozone_dataset))

#reset working directory back to original
setwd("../")
getwd()
```

```
## [1] "C:/Users/Tasha Griffiths/Documents/Duke Year 1/Spring 22 Classes/Environmental Data Analytics/G
```

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 Set date to date format
ozone_combined_dataset$Date <- as.Date(ozone_combined_dataset$Date,
                                       format = "%m/%d/%Y")
class(ozone_combined_dataset$Date)
```

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

```
# 4 select a subset of the columns
ozone_combined_dataset_clean <- ozone_combined_dataset %>% select(Date, Daily.Max.8.hour.Ozone.Concentra

# 5 fill in missing NA days
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"), by = "days"))
colnames(Days) <- "Date"

# 6 left_join to combine clean data and date
GaringerOzone <- left_join(Days, ozone_combined_dataset_clean)
```

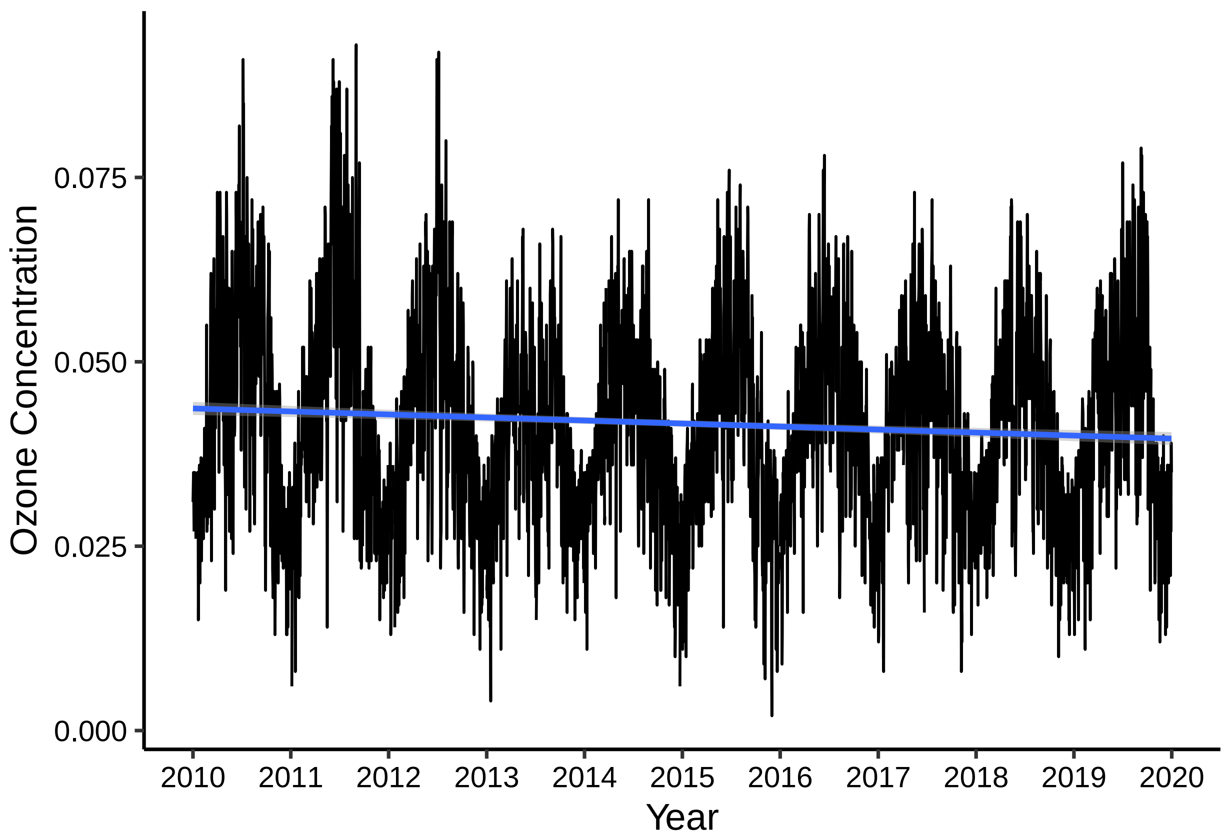
```
## Joining, by = "Date"
```

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?

```
#7 visualize data

ozone.line.plot <- ggplot(GaringerOzone,
                          aes(y = Daily.Max.8.hour.Ozone.Concentration,
                              x = Date), size = 0.25) +
  geom_line() +
  geom_smooth(method = "lm") +
  scale_x_date(date_breaks = "years", date_labels = "%Y") +
  labs( x= "Year", y = "Ozone Concentration")

print(ozone.line.plot)
```

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

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



Answer: Looking at the plot, you can see a slight downward trend in daily ozone concenration over time. This indicates that there is some type of relationship over time and it appears that different seasons may play an impact on concentration peaks and valleys.

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?

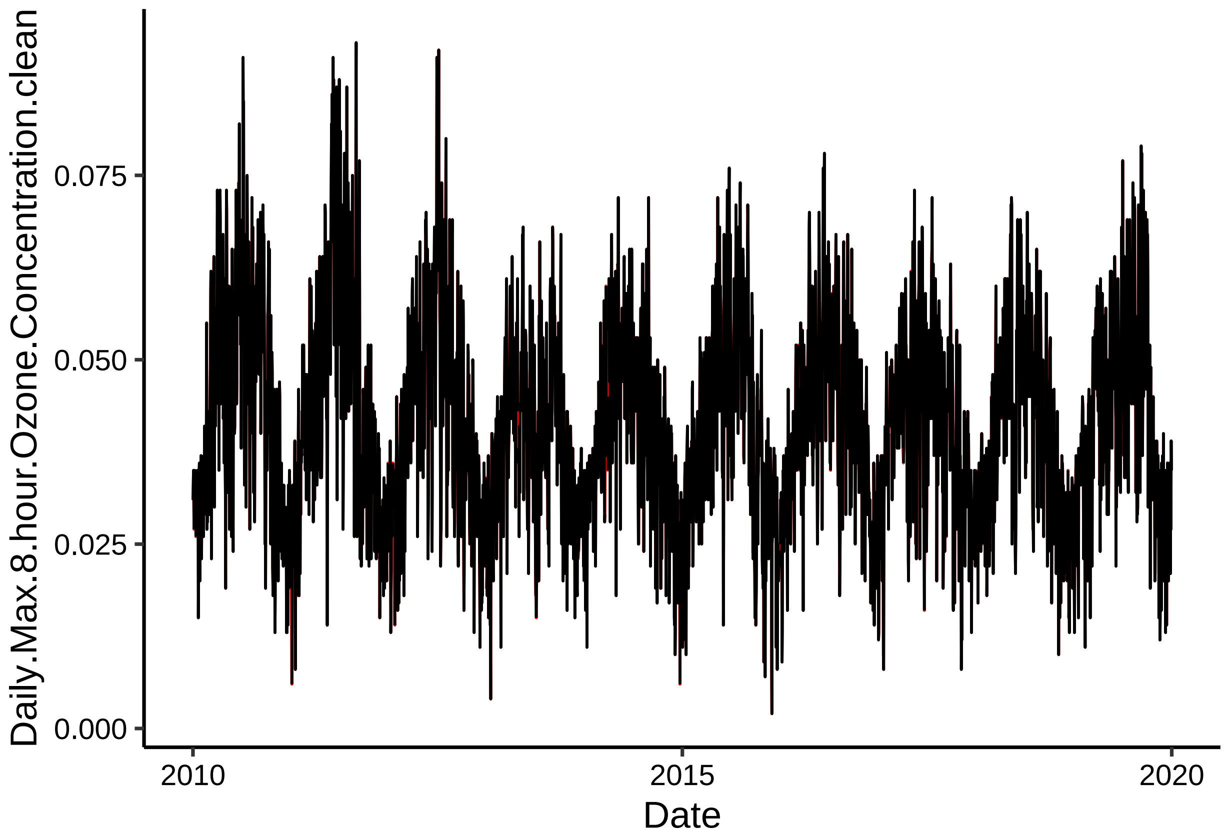
```
#8 fill in missing daily data with linear method
GaringerOzoneClean <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentratio

summary(GaringerOzoneClean)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
##  Daily.Max.8.hour.Ozone.Concentration.clean
##  Min.   :0.00200
##  1st Qu.:0.03200
##  Median :0.04100
##  Mean   :0.04151
##  3rd Qu.:0.05100
##  Max.   :0.09300
##
```

```
#repeat for AQI
GaringerOzoneClean2 <-
  GaringerOzone %>%
  mutate(DAILY_AQI_VALUE.clean = zoo::na.approx(DAILY_AQI_VALUE))

summary(GaringerOzoneClean2)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
##  DAILY_AQI_VALUE.clean
##  Min.   :  2.00
##  1st Qu.: 30.00
##  Median : 38.00
##  Mean   : 41.41
##  3rd Qu.: 47.00
##  Max.   :169.00
##
```

```r
#plotting to see how its interpolated
ggplot(GaringerOzoneClean) +
  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 = "black")
```



Answer: The linear interpolation made the most sense for the missing daily data, since they were very short periods of missing data. If we were missing years at a time, it would make more sense to use a different interpolation method. THe picewise constant is helpful if you want to assume that missing data is equal to a neighbor, but since we are missing just a daily data - we know that it will fall somewhere between the previous and next measurement so the linear method makes the most sense. The spline method uses a quadratic function, which is also not necessary for this data set as the linear straight line connections are what we want.

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)

```r
#9 new dataframe for aggregated daily to monthly data
GaringerOzone.monthly <- GaringerOzoneClean %>%
  mutate(month = month(Date)) %>%
  mutate(year = year(Date)) %>%
```

```
  group_by(year, month) %>%
  summarise(mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration.clean)) %>%
  mutate(Date = my(paste0(month, "--", year))) %>%
  select(Date, mean_Ozone)
```

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

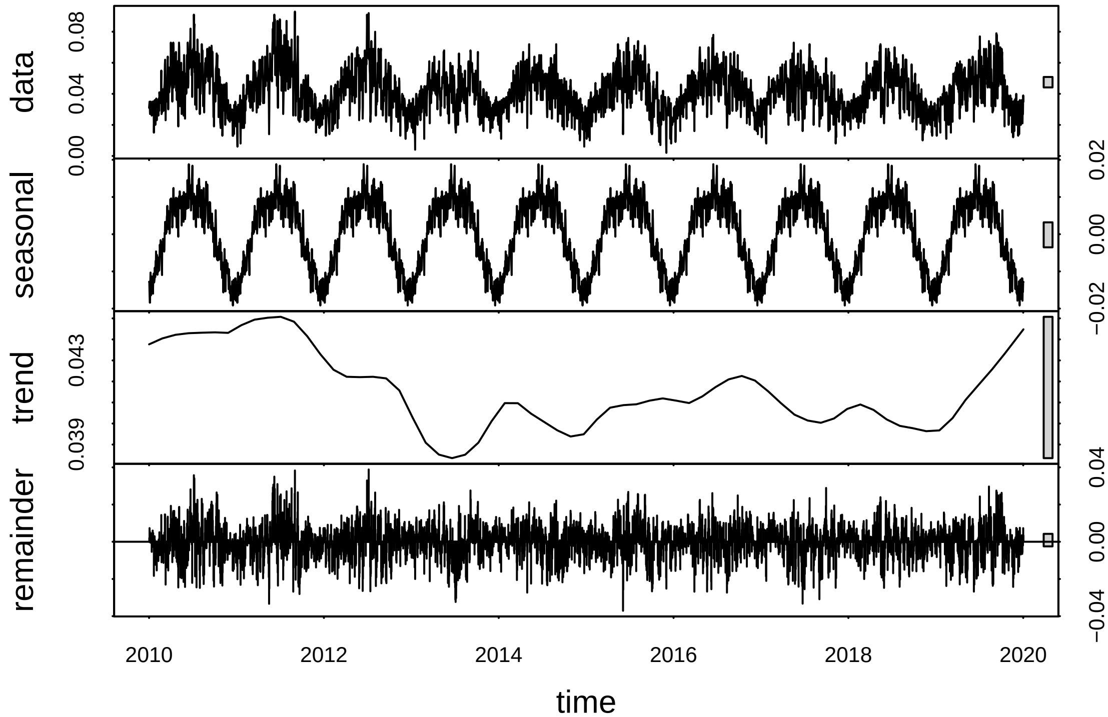
```
## Adding missing grouping variables: `year`
```

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.

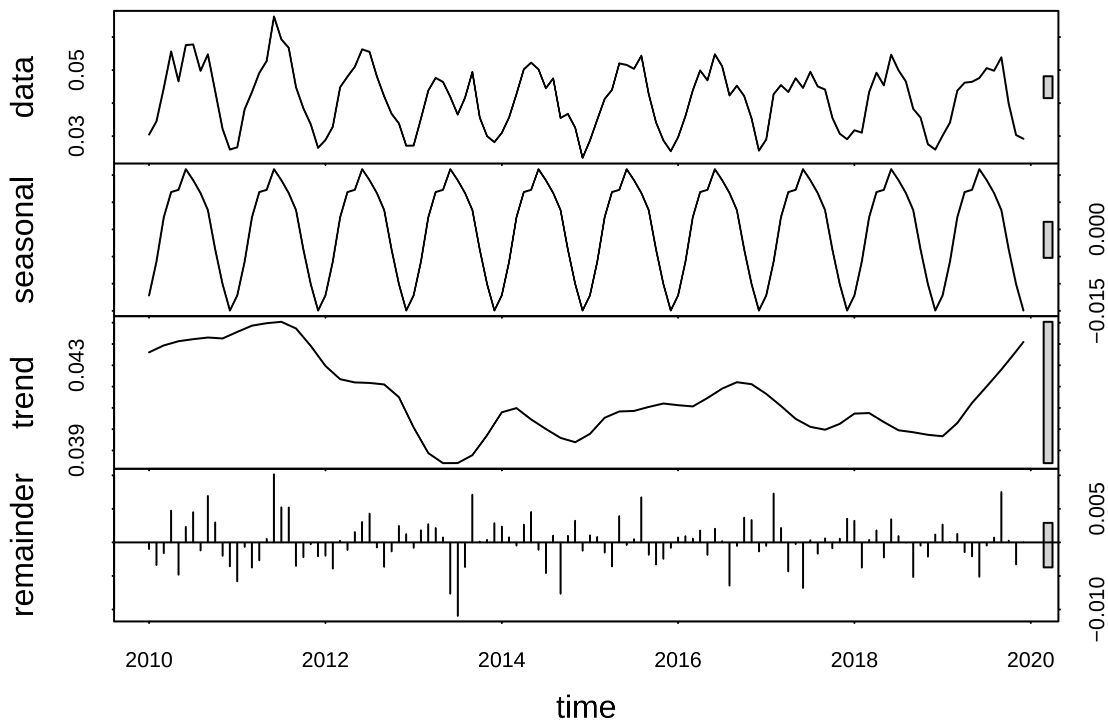
```
#10 create a daily and monthly separate timeseries
GaringerOzone.daily.ts <-
  ts(GaringerOzoneClean$Daily.Max.8.hour.Ozone.Concentration.clean,
     start = c(2010,01,01), frequency = 365)

GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$mean_Ozone, start = c(2010,01,01), frequency = 12)
```

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

```
#11 decompose the two timeseries and check plots
GaringerOzone.daily.ts.decomp <- stl(GaringerOzone.daily.ts,
                                     s.window = "periodic")
plot(GaringerOzone.daily.ts.decomp)
```




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




```
#it makes sense that the daily decomp has thicker lines, since it has much more data points than the ag
```

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 monotonic trend analysis both Kendall and smk(seasonal)
monthly.ozone.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(monthly.ozone.trend)
```

```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

```
monthly.ozone.trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
summary(monthly.ozone.trend2)
```

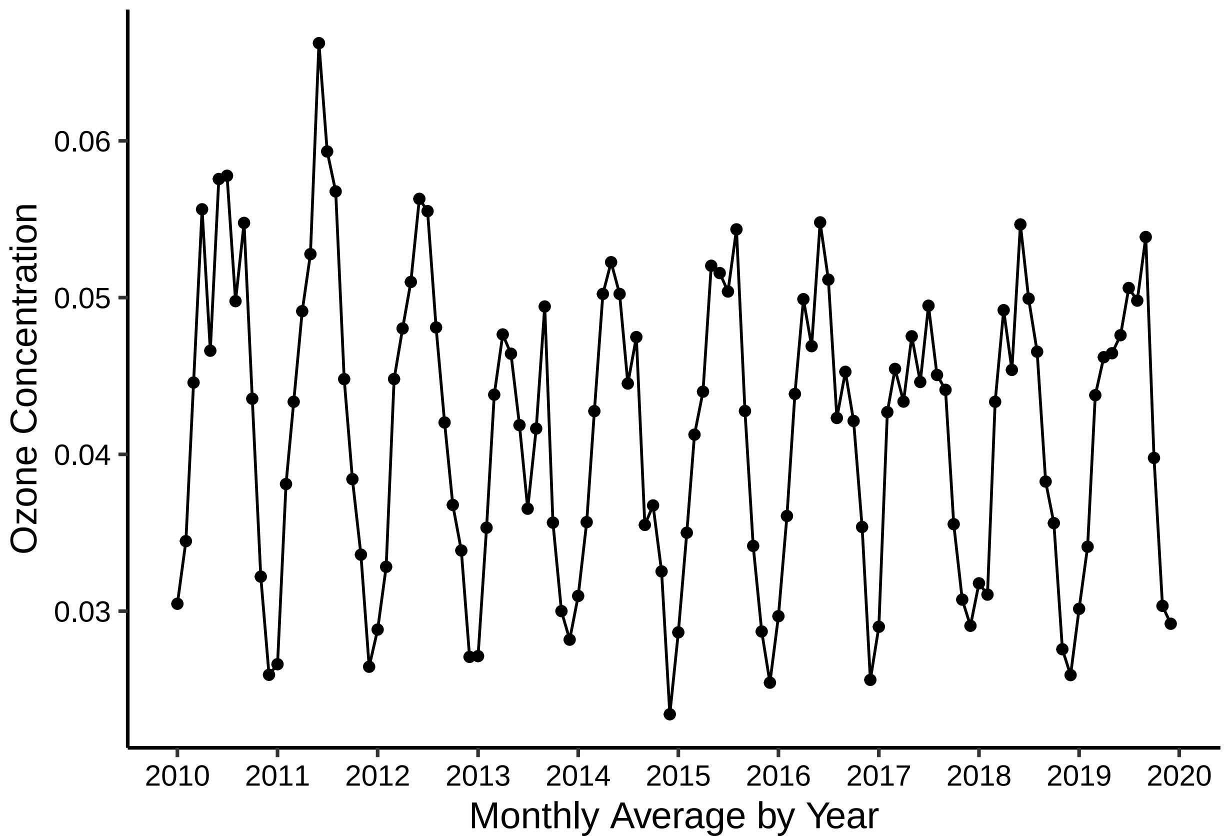
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
```

```
## 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 general Mann-Kendall test assumes no seasonality - however, it is clear from the visualizations that there is some seasonal impact on the daily ozone concentrations, so it makes sense to use the SeasonalMann-Kendall test instead. That way the cyclical nature of the data can be accounted for in the trend test. These tests are for monotonic movement where the variable consistently increases or decreases over time.

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 visualize data over time
plot.mean.monthly.ozone <- ggplot(GaringerOzone.monthly,
                                  aes(y = mean_Ozone, x = Date),
                                  size = 0.25) +
  geom_line() +
  geom_point() +
  #geom_smooth(method = "lm") +
  scale_x_date(date_breaks = "years", date_labels = "%Y") +
  labs(x = "Monthly Average by Year", y = "Ozone Concentration")
print(plot.mean.monthly.ozone)
```



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: Our study question was: Have ozone concentrations changed over the 2010s at this station? The answer is yes, since we were able to rejct the null hypothesis based on the results of our Mann-Kendall seasonal test. The pvalue for the seasonal kendall test is just barely less than .05 (it is .046724) meaning that we can reject the null hypothesis of no monotonic trend and we can see a trend is present.

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 remove the seasonal component from data
GaringerOzone.monthly.noseason <-
  as.data.frame(GaringerOzone.monthly.ts.decomp$time.series[,1:3])

GaringerOzone.monthly.noseason <- mutate(GaringerOzone.monthly.noseason,
         Observed = GaringerOzone.monthly$mean_Ozone,
         Date = GaringerOzone.monthly$Date)

#turn it in to a time series so we can use trend tests
```

```r
GaringerOzone.monthly.noseason.ts <-
  ts(GaringerOzone.monthly.noseason$trend,
     start = c(2010,01,01), frequency = 12)


#16 run the Mann Kendall test
noseason.kendall.test <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.noseason.ts)
summary(noseason.kendall.test)
```

```
## Score =  -164 , Var(Score) = 1500
## denominator =  540
## tau = -0.304, 2-sided pvalue =2.291e-05
```

```r
#fix directory so we can knit
setwd("C:/Users/Tasha Griffiths/Documents/Duke Year 1/Spring 22 Classes/Environmental Data Analytics/Gi
getwd()
```

```
## [1] "C:/Users/Tasha Griffiths/Documents/Duke Year 1/Spring 22 Classes/Environmental Data Analytics/Gi
```

> Answer: When we subtract the seasonal component from our data, we see a significant impact on the Mann-Kendall test's p-value. The pvalue changes from .o4 to 2.291e-5. This means that when seasonality is removed, we can even more strongly reject the null hypothesis of no monotonic trend and conclude that there is a trend from 2010 to 2020.