Assignment 7: Time Series Analysis

Zhiyuan Chen

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 Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

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

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
getwd()
```

```
## [1] "D:/Rfiles/EDA-Fall2022"
```

```
#install.packages("tidyverse")
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.2
```

```r
#install.packages("lubridate")
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.2.2
```

```r
#install.packages("zoo")
library(zoo)
#install.packages("trend")
library(trend)
#install.packages("Kendall")
library(Kendall)
library(dplyr)
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
#2
EPA2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                    stringsAsFactors = TRUE)
EPA2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                    stringsAsFactors = TRUE)
EPA2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                    stringsAsFactors = TRUE)
EPA2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                    stringsAsFactors = TRUE)
EPA2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                    stringsAsFactors = TRUE)
EPA2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                    stringsAsFactors = TRUE)
EPA2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                    stringsAsFactors = TRUE)
EPA2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                    stringsAsFactors = TRUE)
EPA2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                    stringsAsFactors = TRUE)
EPA2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                    stringsAsFactors = TRUE)
GaringerOzone <- rbind(EPA2010,EPA2011,EPA2012,EPA2013,EPA2014,EPA2015,EPA2016,EPA2017,EPA2018,EPA2019)
```

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")
# 4
GaringerOzone_Ex <-
  GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
# 5
Days <- as.data.frame(seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by="day"))
colnames(Days) = c("Date")
# 6
GaringerOzone <- left_join(Days, GaringerOzone_Ex, by= "Date")
```

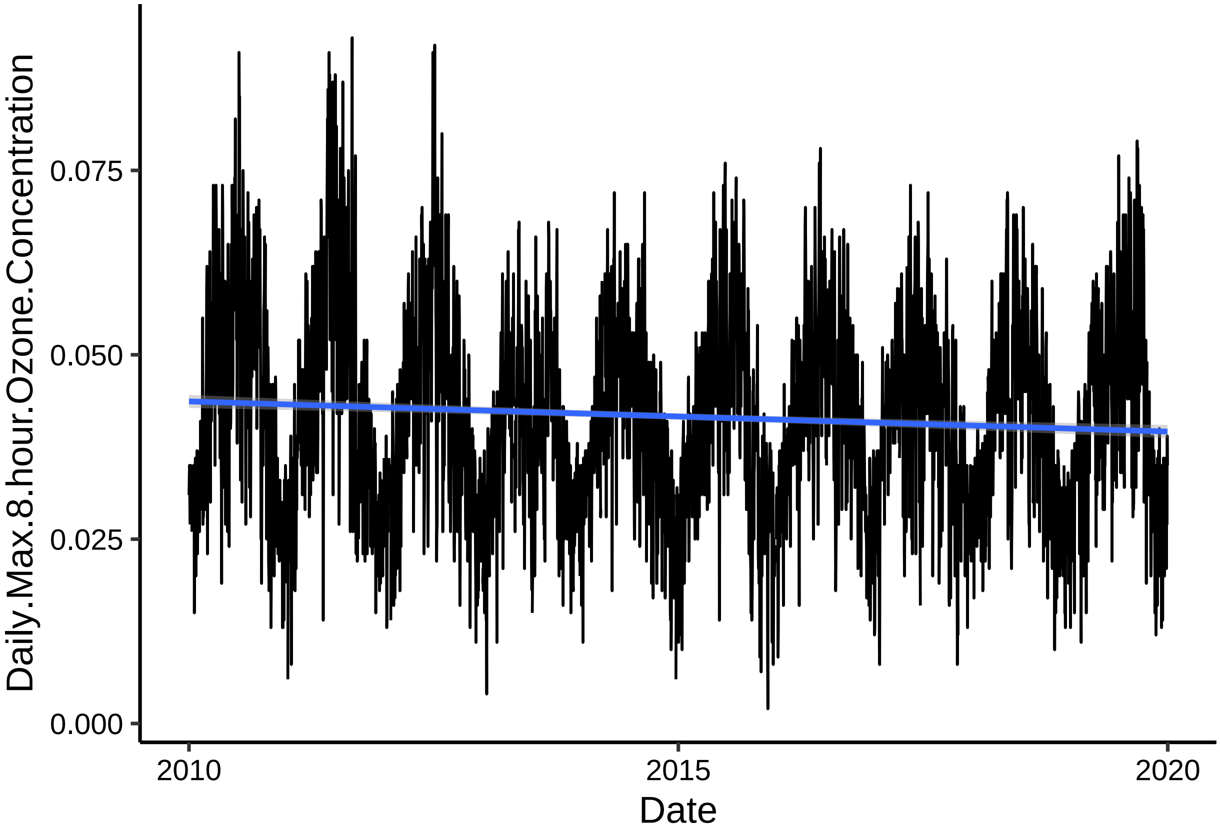
Visualize

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?

```r
#7
ozone_data_plot <-
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  ylab("Daily.Max.8.hour.Ozone.Concentration") +
  xlab("Date")+
  geom_smooth( method = lm )
print(ozone_data_plot)
```

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

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



Answer: Yes. My plot suggests a downward 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
#head(GaringerOzone)
#summary(GaringerOzone)
GaringerOzone_clean <-
  GaringerOzone %>%
  mutate( Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: If we use a piecewise constant interpolation, any missing data are assumed to be equal to the measurement made nearest to that date (could be earlier or later), which is not good for our data. Spline interpolation incurs a smaller error than linear interpolation. But in this case, the error could be neglected. Also, the global nature of the basis functions leads to ill-conditioning.

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_clean %>%
  mutate(GaringerOzone_clean, year = year(Date), month = month(Date)) %>%
  group_by(year, month) %>%
  dplyr::summarise(mean_ozone_concentrations = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

```
Date <- as.data.frame(seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-01"), by="month"))
colnames(Date) = c("Date")
GaringerOzone.monthly <- cbind(Date, GaringerOzone.monthly)
```

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.

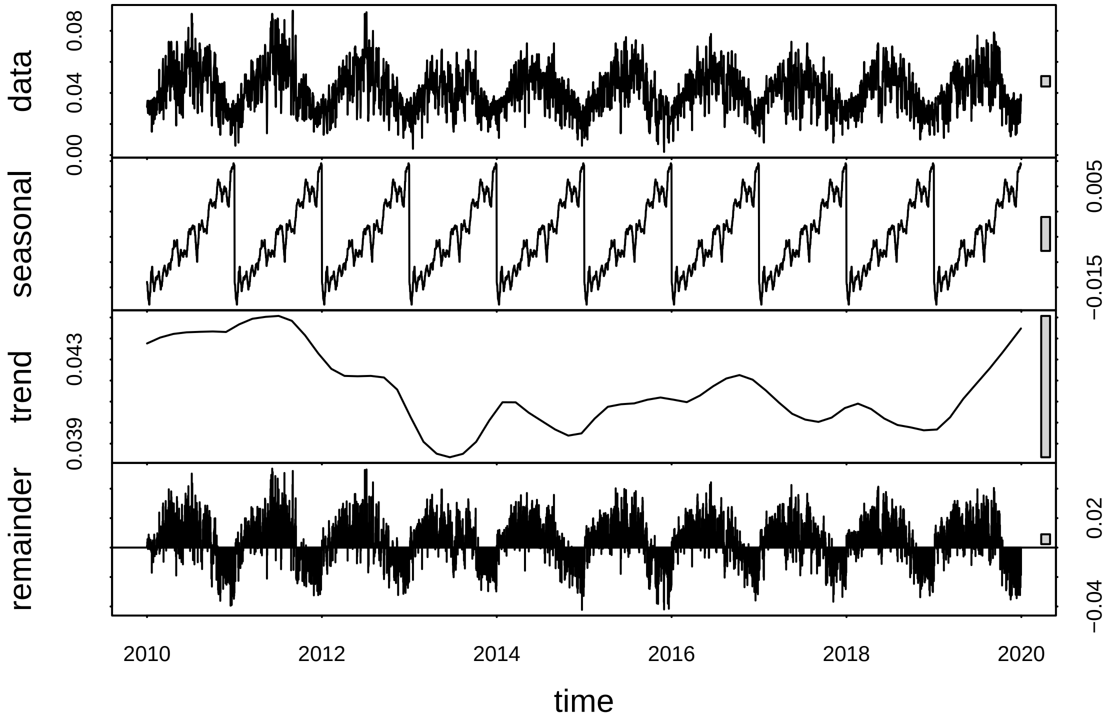
```
#10
f_year <- year(first(GaringerOzone_clean$Date))
f_month <- month(first(GaringerOzone_clean$Date))
f_day <- day(first(GaringerOzone_clean$Date))
GaringerOzone.daily.ts<-
  ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration,
          start = c(f_year,f_month,f_day),
          frequency = 365.25)
#GaringerOzone.daily.ts

GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$mean_ozone_concentrations,
     start = c(2010,1),
     frequency = 12)
GaringerOzone.monthly.ts
```

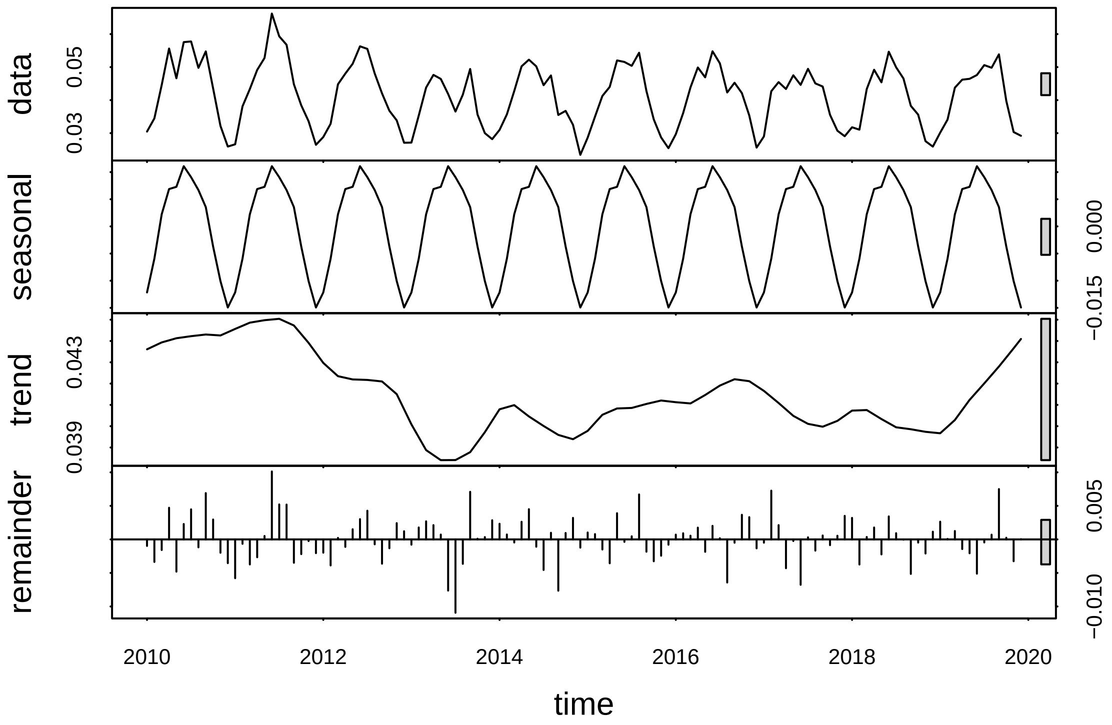
```
##             Jan        Feb        Mar        Apr        May        Jun
## 2010 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
## 2011 0.02661290 0.03810714 0.04335484 0.04913333 0.05277419 0.06623333
## 2012 0.02882258 0.03282759 0.04480645 0.04803333 0.05100000 0.05630000
## 2013 0.02712903 0.03532143 0.04380645 0.04765000 0.04641935 0.04186667
## 2014 0.03096774 0.03567857 0.04275806 0.05023333 0.05225806 0.05023333
## 2015 0.02864516 0.03500000 0.04125806 0.04400000 0.05203226 0.05156667
## 2016 0.02967742 0.03606897 0.04385484 0.04990000 0.04690323 0.05480000
## 2017 0.02900000 0.04269643 0.04545161 0.04336667 0.04753226 0.04461667
## 2018 0.03177419 0.03105357 0.04335484 0.04920000 0.04538710 0.05466667
## 2019 0.03014516 0.03410714 0.04377419 0.04620000 0.04645161 0.04760000
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2010 0.05777419 0.04977419 0.05476667 0.04354839 0.03220000 0.02593548
## 2011 0.05932258 0.05677419 0.04480000 0.03841935 0.03360000 0.02645161
## 2012 0.05551613 0.04809677 0.04203333 0.03677419 0.03386667 0.02708065
## 2013 0.03653226 0.04164516 0.04943333 0.03564516 0.03000000 0.02817742
## 2014 0.04451613 0.04748387 0.03550000 0.03674194 0.03253333 0.02341935
## 2015 0.05038710 0.05435484 0.04276667 0.03416129 0.02870000 0.02543548
## 2016 0.05114516 0.04232258 0.04526667 0.04212903 0.03536667 0.02561290
## 2017 0.04948387 0.04506452 0.04411667 0.03554839 0.03073333 0.02906452
## 2018 0.04993548 0.04654839 0.03826667 0.03561290 0.02756667 0.02591935
## 2019 0.05061290 0.04980645 0.05386667 0.03977419 0.03033333 0.02919355
```

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")
#GaringerOzone.daily_decomp
plot(GaringerOzone.daily_decomp)
```



```
GaringerOzone.monthly_decomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
#GaringerOzone.monthly_decomp
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
ozone_data_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
ozone_data_trend1
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
summary(ozone_data_trend1)
```

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

```
ozone_data_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
ozone_data_trend2
```

```
##
##  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(ozone_data_trend2)
```

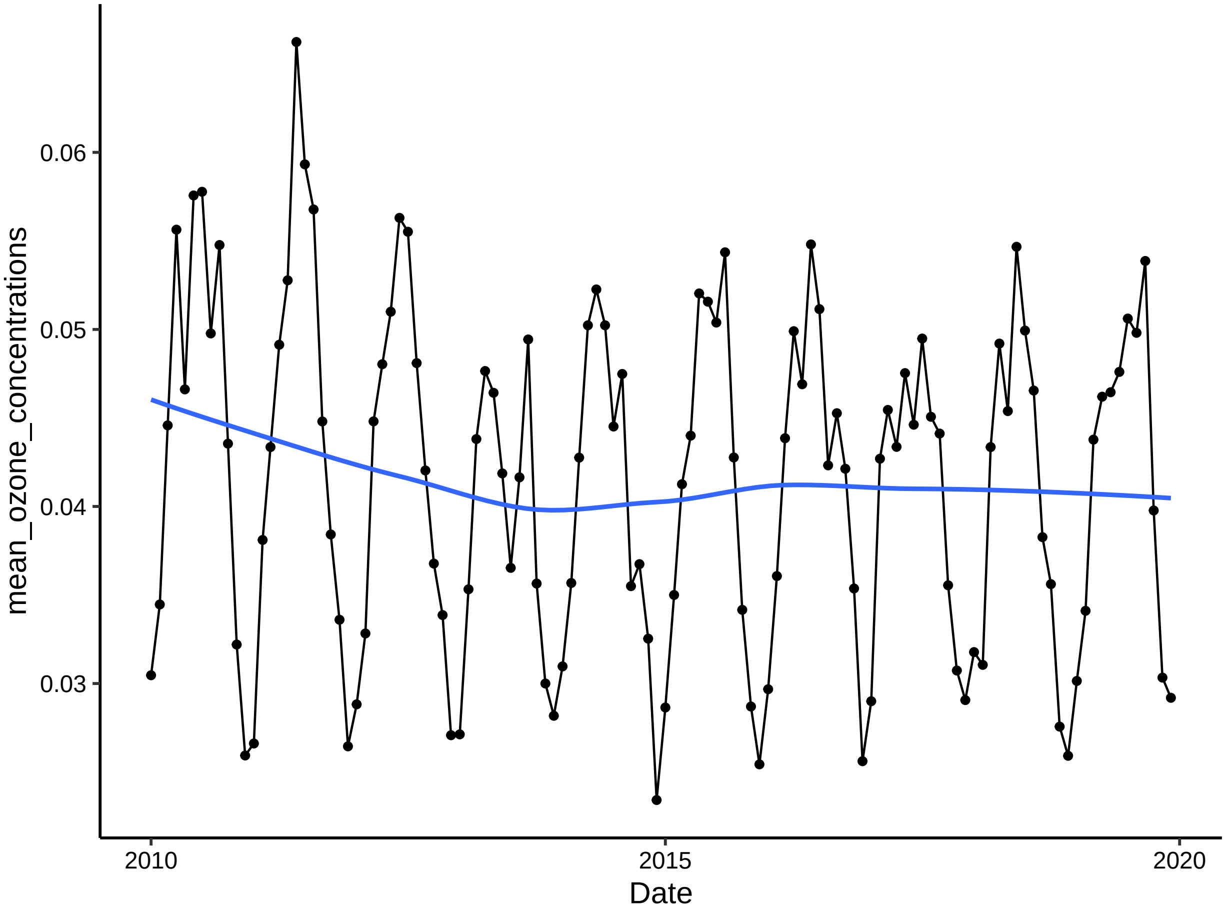
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                      S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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: Because the data has seasonality and is non-parametric.

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_data_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone_concentrations)) +
  geom_point() +
  geom_line() +
  ylab("mean_ozone_concentrations") +
  xlab("Date")+
  geom_smooth(se = FALSE)
print(ozone_data_plot)
```

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



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: Ozone concentrations have changed over the 2010s at this station. We could see a decreasing trend on Ozone concentrations in general.From 2010 to 2013, Ozone concentrations kept decreasing and reached the bottom in 2013. Then from 2014 to 2016, Ozone concentrations increased a little. After that, Ozone concentrations were almost the same from 2017 to 2019. In addition, the p-value of the Seasonal Mann Kendall test(0.046724) is smaller than 0.05, which also means Ozone concentrations have changed over the 2010s at this station.

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
GaringerOzone.monthly.ts_COMPONENTS <- as.data.frame(GaringerOzone.monthly_decomp$time.series[,2:3])
GaringerOzone.monthly.ts_COMPONENTS <-
  mutate(GaringerOzone.monthly.ts_COMPONENTS,
        Observed =GaringerOzone.monthly$mean_ozone_concentrations,
        Date = GaringerOzone.monthly$Date)
```

```
nonseasonal.GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly.ts_COMPONENTS$Observed,
           start = c(2010,1),
           frequency = 12)
nonseasonal.GaringerOzone.monthly.ts
```

```
##             Jan        Feb        Mar        Apr        May        Jun
## 2010 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
## 2011 0.02661290 0.03810714 0.04335484 0.04913333 0.05277419 0.06623333
## 2012 0.02882258 0.03282759 0.04480645 0.04803333 0.05100000 0.05630000
## 2013 0.02712903 0.03532143 0.04380645 0.04765000 0.04641935 0.04186667
## 2014 0.03096774 0.03567857 0.04275806 0.05023333 0.05225806 0.05023333
## 2015 0.02864516 0.03500000 0.04125806 0.04400000 0.05203226 0.05156667
## 2016 0.02967742 0.03606897 0.04385484 0.04990000 0.04690323 0.05480000
## 2017 0.02900000 0.04269643 0.04545161 0.04336667 0.04753226 0.04461667
## 2018 0.03177419 0.03105357 0.04335484 0.04920000 0.04538710 0.05466667
## 2019 0.03014516 0.03410714 0.04377419 0.04620000 0.04645161 0.04760000
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2010 0.05777419 0.04977419 0.05476667 0.04354839 0.03220000 0.02593548
## 2011 0.05932258 0.05677419 0.04480000 0.03841935 0.03360000 0.02645161
## 2012 0.05551613 0.04809677 0.04203333 0.03677419 0.03386667 0.02708065
## 2013 0.03653226 0.04164516 0.04943333 0.03564516 0.03000000 0.02817742
## 2014 0.04451613 0.04748387 0.03550000 0.03674194 0.03253333 0.02341935
## 2015 0.05038710 0.05435484 0.04276667 0.03416129 0.02870000 0.02543548
## 2016 0.05114516 0.04232258 0.04526667 0.04212903 0.03536667 0.02561290
## 2017 0.04948387 0.04506452 0.04411667 0.03554839 0.03073333 0.02906452
## 2018 0.04993548 0.04654839 0.03826667 0.03561290 0.02756667 0.02591935
## 2019 0.05061290 0.04980645 0.05386667 0.03977419 0.03033333 0.02919355
```

```
#16
ozone_data_trend3 <- MannKendall(nonseasonal.GaringerOzone.monthly.ts)
ozone_data_trend3
```

```
## tau = -0.0594, 2-sided pvalue =0.33732
```

```
summary(ozone_data_trend3)
```

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

Answer: The p-value of Mann Kendall test(0.33732) on the non-seasonal Ozone monthly series is larger than 0.05, which means Ozone concentrations have not changed over the 2010s at this station. While the p-value of the seasonal Mann-Kendall test(0.046724) on the complete series is smaller than 0.05, which means Ozone concentrations have changed over the 2010s at this station. And the tau of Mann Kendall test(-0.0594) is larger than the tau of the seasonal Mann-Kendall test(-0.143). Also, the Var(Score) and denominator of Mann Kendall test(194364.7 and 7139 respectively) are larger than those of the seasonal Mann-Kendall test(1499 and 539.4972 respectively). In addition, the score of Mann Kendall test(-424) is smaller than that of the seasonal Mann-Kendall test(-77).