Assignment 8: Time Series Analysis

Xuancheng Guo

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()
```

```
## [1] "/home/guest/EDA_Spring2024"
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     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 error
```

```r
library(lubridate)
library(zoo)
```

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

```r
library(trend)

mytheme <- theme_classic(base_size = 14) +
  theme(
    axis.text = element_text(color = "black"),
    legend.position = "top",
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold"),
    plot.margin = margin(10, 10, 10, 10)
  )
```

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
#1
ozone10 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
stringsAsFactors = T)
ozone11 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
stringsAsFactors = T)
ozone12 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
stringsAsFactors = T)
ozone13 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
stringsAsFactors = T)
ozone14 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
stringsAsFactors = T)
ozone15 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
stringsAsFactors = T)
ozone16 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
stringsAsFactors = T)
ozone17 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
stringsAsFactors = T)
ozone18 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
stringsAsFactors = T)
ozone19 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
stringsAsFactors = T)
GaringerOzone <- rbind(ozone10, ozone11, ozone12, ozone13, ozone14, ozone15, ozone16, ozone17, ozone18,
remove(ozone10,ozone11,ozone12,ozone13,ozone14,ozone15,ozone16,ozone17,ozone18,ozone19)
```

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_q4 <- select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "days"))
names(Days) <- "Date"

# 6
GaringerOzone <- left_join(Days, GaringerOzone_q4)
```

```
## Joining with `by = join_by(Date)`
```

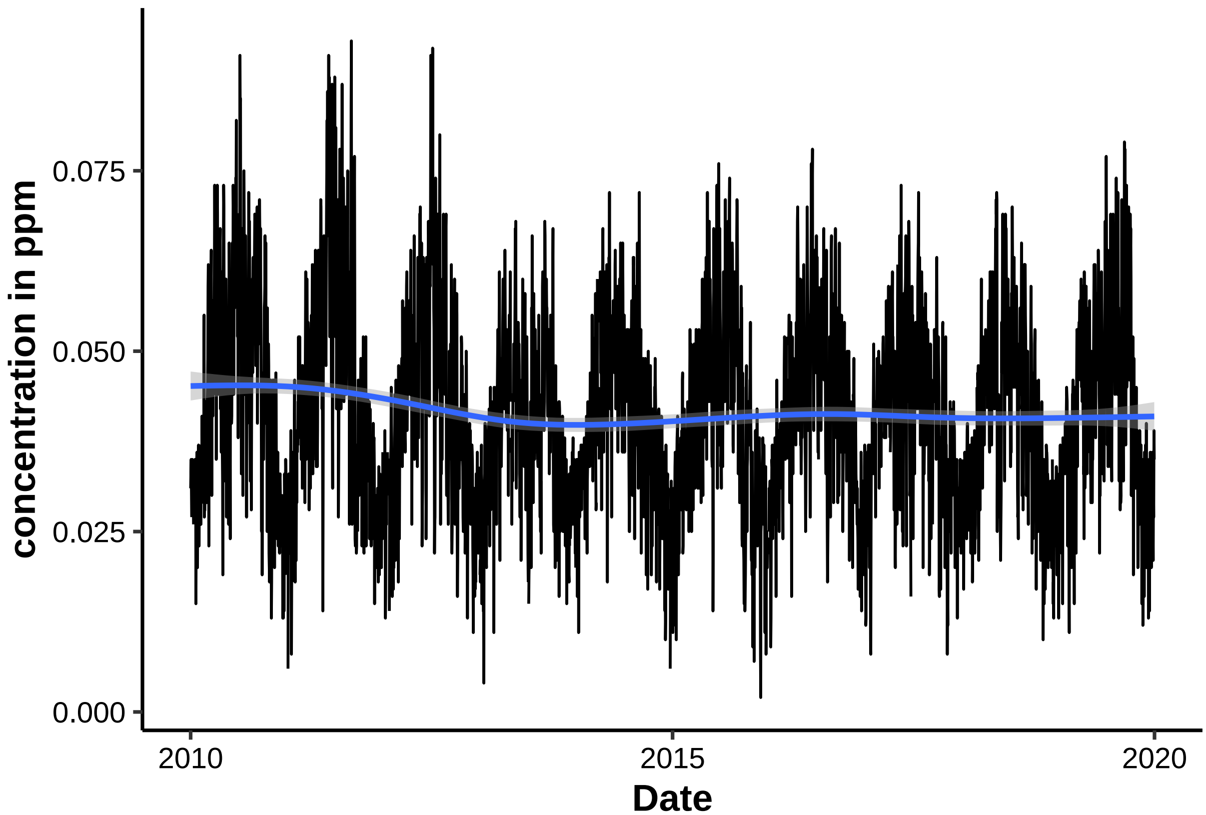
Visualize

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
conc_time_plot <- ggplot(GaringerOzone,
                         aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth() +
  ylab("concentration in ppm") +
  mytheme
print(conc_time_plot)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'

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



Answer:The trend is hard to observe without any further analysis, but it could potentially be decreasing over time.

Time Series Analysis

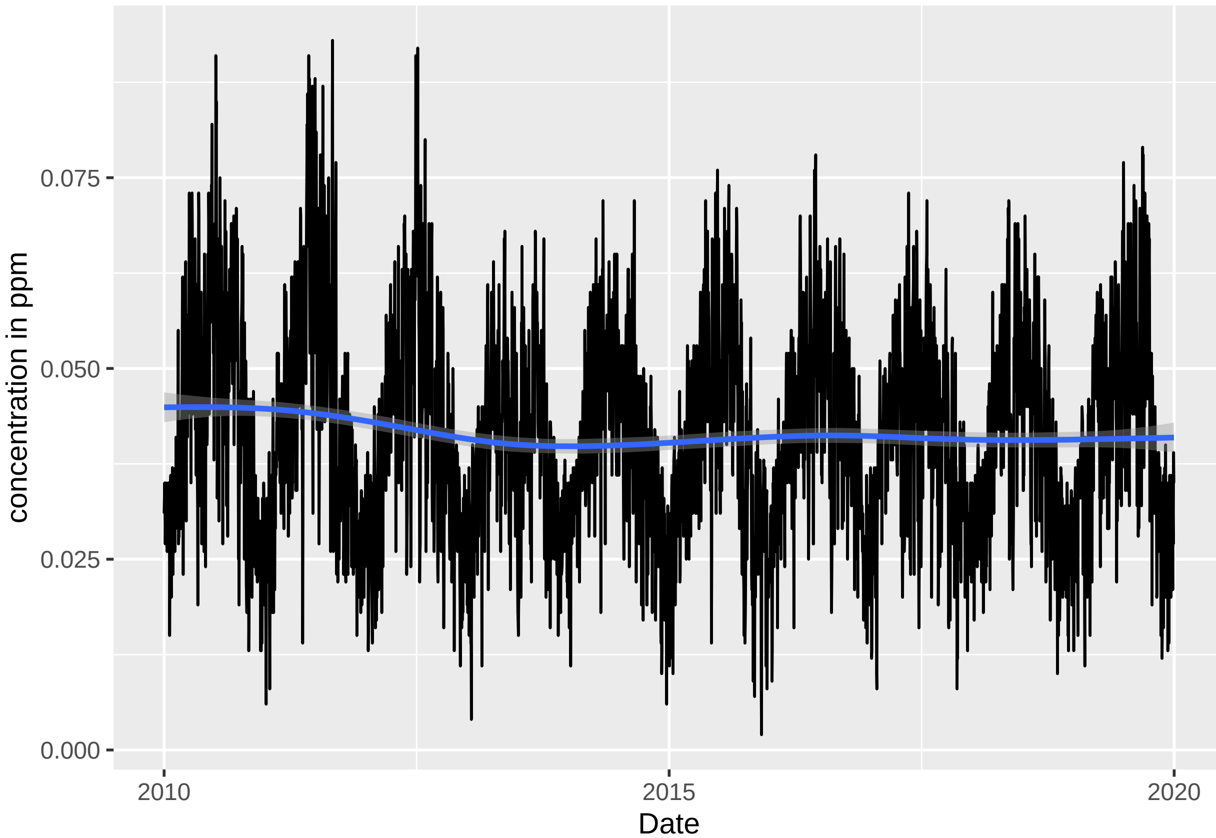
Study question: Have ozone concentrations changed over the 2010s at this station?

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

ggplot(ozone.interpo,
       aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth() +
  ylab("concentration in ppm")
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Answer: Piecewise constant method fills missing values with the nearest non-NA value, which might cause the data to jump up and down and might not be realistic. Spline interpolation uses polynomial functions to connect the data points, resulting in a smoother curve than linear interpolation; but since our trend is seem to be linear, using polynomial can also signify data unexpectly.

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 <- ozone.interpo %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  mutate(Date = my(paste0(Month,"-",Year))) %>%
  group_by(Date) %>%
  mutate(Mean.Ozone.Concentration =
           mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  distinct(Date, Mean.Ozone.Concentration)
GaringerOzone.monthly
```

```
## # A tibble: 120 x 2
## # Groups:   Date [120]
##    Date       Mean.Ozone.Concentration
```

```
##    <date>                         <dbl>
##   1 2010-01-01                   0.0305
##   2 2010-02-01                   0.0345
##   3 2010-03-01                   0.0446
##   4 2010-04-01                   0.0556
##   5 2010-05-01                   0.0466
##   6 2010-06-01                   0.0576
##   7 2010-07-01                   0.0578
##   8 2010-08-01                   0.0498
##   9 2010-09-01                   0.0548
##  10 2010-10-01                   0.0435
## # i 110 more rows
```

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.

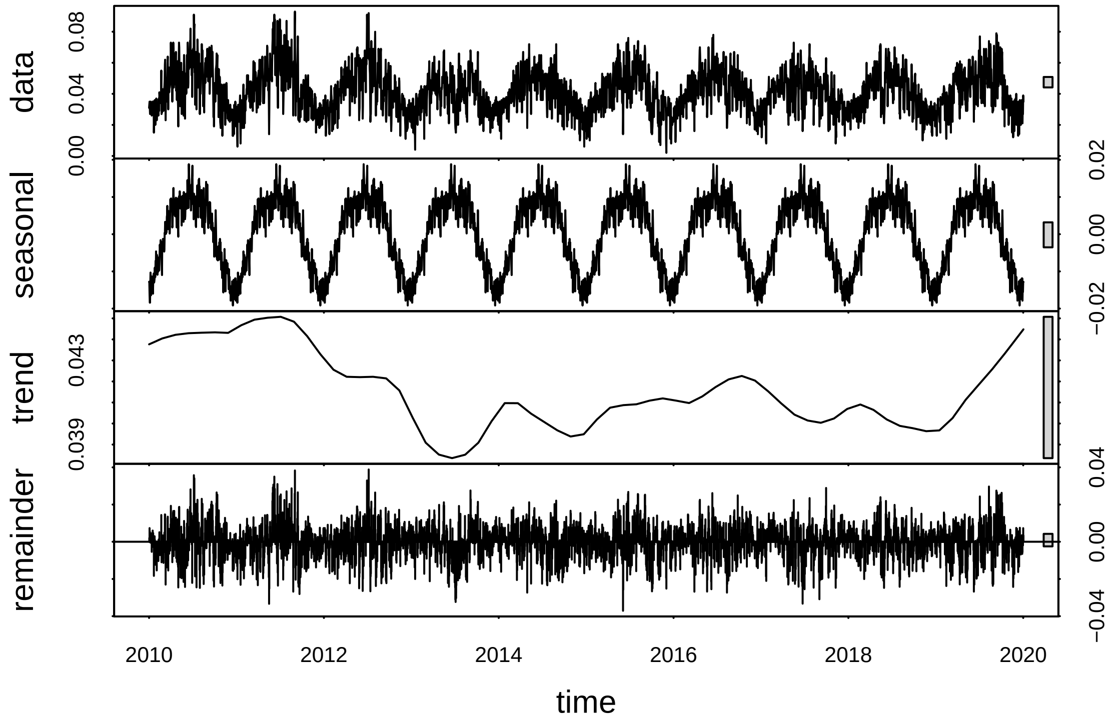
```
#10
# Daily
GaringerOzone.daily.ts <-
  ts(ozone.interpo$Daily.Max.8.hour.Ozone.Concentration,
     start = c(2010,1),
     frequency = 365)

# Monthly
GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$Mean.Ozone.Concentration,
     start = c(2010,1),
     frequency = 12)
```

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

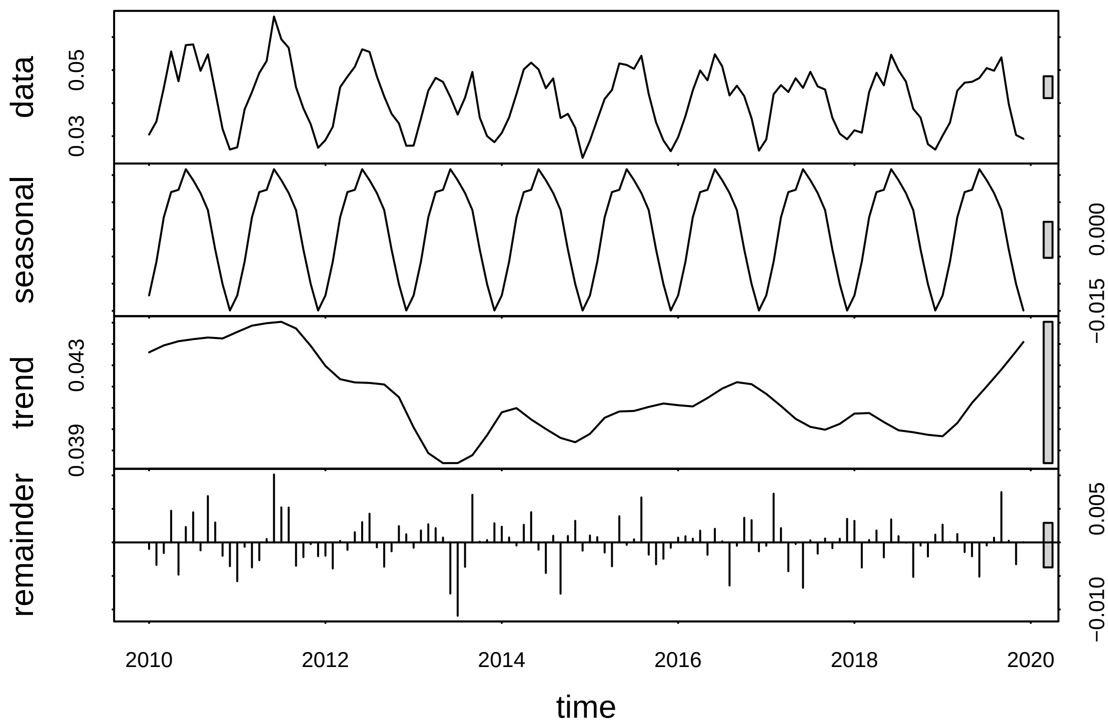
```
#11
# Daily
GaringerOzone.daily.decomposed <-
  stl(GaringerOzone.daily.ts, s.window = "periodic")

plot(GaringerOzone.daily.decomposed)
```




```r
# Monthly
GaringerOzone.Monthly.Decomposed <-
  stl(GaringerOzone.monthly.ts, s.window = "periodic")

plot(GaringerOzone.Monthly.Decomposed)
```



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
Monthly.Trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
Monthly.Trend
```

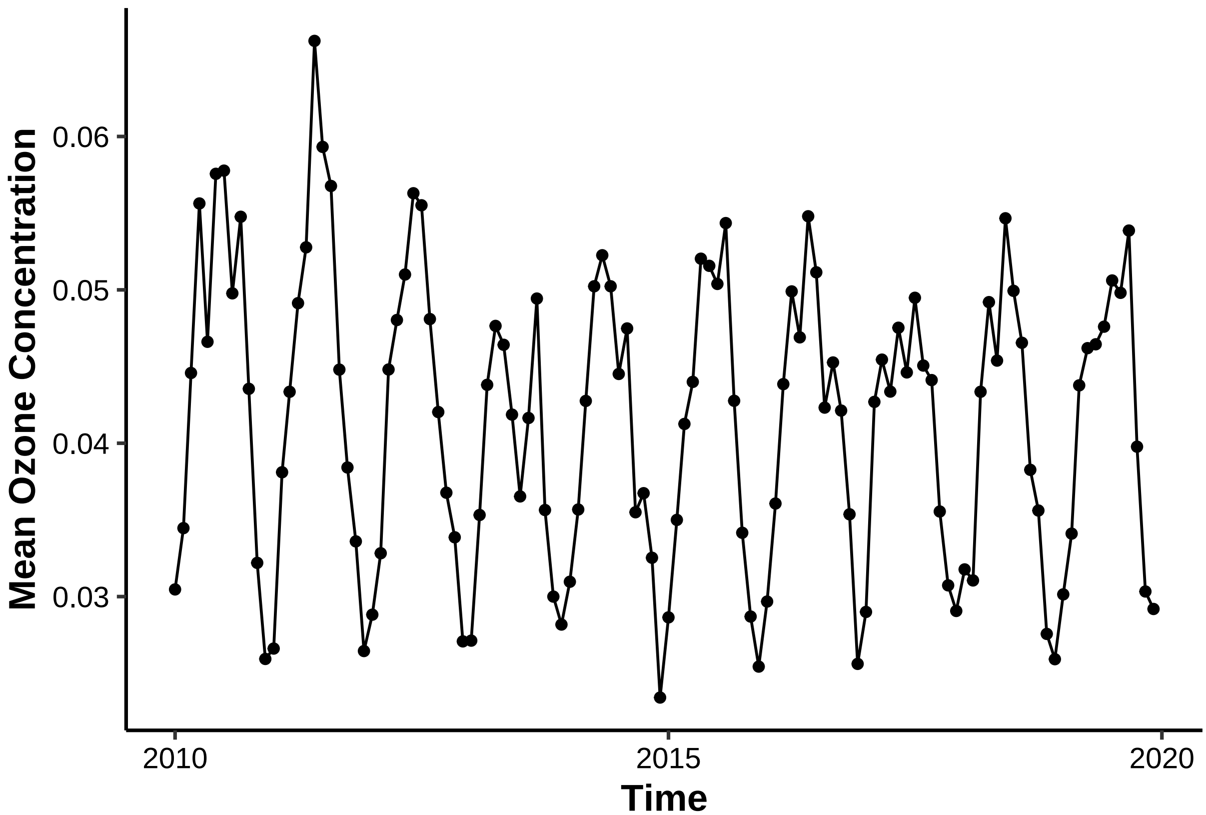
```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Seasonal MK test, different from classical MK test, account for the seasonal trend of the time-series. These trends might not be account for if other tests are performed.

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
monthly_ozone_plot <- ggplot(GaringerOzone.monthly,
                             aes(x = Date, y = Mean.Ozone.Concentration)) +
  geom_point() +
  geom_line() +
  xlab("Time") +
  ylab("Mean Ozone Concentration") +
  mytheme

print(monthly_ozone_plot)
```



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 Kendall's tau value of minus 0.143 suggests a slight downward trend in the monthly ozone concentrations.The p-value is a bit tricky since it's below but very close to our level of significance which is 0.05, meaning there's a low probability that the observed trend could have occurred by chance.

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
# Create new df and paste origional df together
GaringerOzone.Components <-
  as.data.frame(GaringerOzone.Monthly.Decomposed$time.series) %>%
  mutate(observed = GaringerOzone.monthly$Mean.Ozone.Concentration,
         date = GaringerOzone.monthly$Date)

# Subtract seasonal components
GaringerOzone.Nonseasonal <- GaringerOzone.monthly.ts -
  GaringerOzone.Components$seasonal
```

```r
GaringerOzone.Nonseasonal
```

```
##             Jan        Feb        Mar        Apr        May        Jun
## 2010 0.04263190 0.04041003 0.04234881 0.04875492 0.03932081 0.04647348
## 2011 0.03877706 0.04405289 0.04112300 0.04225492 0.04548211 0.05514015
## 2012 0.04098674 0.03877333 0.04257462 0.04115492 0.04370791 0.04520681
## 2013 0.03929319 0.04126717 0.04157462 0.04077159 0.03912727 0.03077348
## 2014 0.04313190 0.04162432 0.04052623 0.04335492 0.04496598 0.03914015
## 2015 0.04080932 0.04094575 0.03902623 0.03712159 0.04474017 0.04047348
## 2016 0.04184158 0.04201471 0.04162300 0.04302159 0.03961114 0.04370681
## 2017 0.04116416 0.04864217 0.04321978 0.03648826 0.04024017 0.03352348
## 2018 0.04393835 0.03699932 0.04112300 0.04232159 0.03809501 0.04357348
## 2019 0.04230932 0.04005289 0.04154236 0.03932159 0.03915952 0.03650681
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2010 0.04871023 0.04307797 0.05120815 0.04725211 0.04226527 0.04087082
## 2011 0.05025862 0.05007797 0.04124148 0.04212308 0.04366527 0.04138695
## 2012 0.04645216 0.04140056 0.03847481 0.04047792 0.04393193 0.04201598
## 2013 0.02746829 0.03494894 0.04587481 0.03934888 0.04006527 0.04311275
## 2014 0.03545216 0.04078765 0.03194148 0.04044566 0.04259860 0.03835469
## 2015 0.04132313 0.04765862 0.03920815 0.03786501 0.03876527 0.04037082
## 2016 0.04208120 0.03562636 0.04170815 0.04583275 0.04543193 0.04054824
## 2017 0.04041991 0.03836830 0.04055815 0.03925211 0.04079860 0.04399985
## 2018 0.04087152 0.03985217 0.03470815 0.03931662 0.03763193 0.04085469
## 2019 0.04154894 0.04311023 0.05030815 0.04347792 0.04039860 0.04412888
```

```r
#16
Nonseasonal.trend <- Kendall::MannKendall(GaringerOzone.Nonseasonal)
Nonseasonal.trend
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

> Answer: Both tests indicate a negative trend in ozone concentrations, as shown by the negative tau values (minus 0.143 vs. minus 0.165). This suggests a consistent downward trend across both analyses, and the nonseasonal trend is stronger. The p-value is considerably lower after subtracting the seasonal component (0.0075402) compared to the original series. This suggests a higher level of statistical confidence of the existing of a trend.