Assignment 8: Time Series Analysis

Shubhangi Gupta

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

```
#checking working directory
getwd()
```

```
## [1] "/home/guest/RStudio Project Folder/EDA_Spring2024"
```

```
#loading packages
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)
library(ggplot2)
library(tseries)
```

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

```r
library(Kendall)

#Setting up ggplot theme
mytheme <- theme_classic(base_size = 14)+
  theme(axis.text = element_text (color = "black"), legend.position = "right")
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.

```r
#Importing the 10 datasets
Ozone2010 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2011 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2012 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2013 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2014 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2015 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2016 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2017 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2018 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2019 <- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                      stringsAsFactors = TRUE)
```

```r
#Combining them into one dataset
Ozone_Raw <- rbind(Ozone2010, Ozone2011, Ozone2012, Ozone2013, Ozone2014, Ozone2015,
                   Ozone2016, Ozone2017, Ozone2018, Ozone2019)
```

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: Convering data column to a date object
Ozone_Raw$Date <- mdy(Ozone_Raw$Date)
glimpse(Ozone_Raw)
```

```
## Rows: 3,589
## Columns: 20
## $ Date                                 <date> 2010-01-01, 2010-01-02, 2010-01-~
## $ Source                               <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS~
## $ Site.ID                              <int> 371190041, 371190041, 371190041, ~
## $ POC                                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.031, 0.033, 0.035, 0.031, 0.027~
## $ UNITS                                <fct> ppm, ppm, ppm, ppm, ppm, ppm, ppm~
## $ DAILY_AQI_VALUE                      <int> 29, 31, 32, 29, 25, 31, 32, 30, 3~
## $ Site.Name                            <fct> Garinger High School, Garinger Hi~
## $ DAILY_OBS_COUNT                      <int> 17, 17, 17, 17, 17, 17, 17, 17, 1~
## $ PERCENT_COMPLETE                     <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE                   <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC                   <fct> Ozone, Ozone, Ozone, Ozone, Ozone~
## $ CBSA_CODE                            <int> 16740, 16740, 16740, 16740, 16740~
## $ CBSA_NAME                            <fct> "Charlotte-Concord-Gastonia, NC-S~
## $ STATE_CODE                           <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
## $ STATE                                <fct> North Carolina, North Carolina, N~
## $ COUNTY_CODE                          <int> 119, 119, 119, 119, 119, 119, 119~
## $ COUNTY                               <fct> Mecklenburg, Mecklenburg, Mecklen~
## $ SITE_LATITUDE                        <dbl> 35.2401, 35.2401, 35.2401, 35.240~
## $ SITE_LONGITUDE                       <dbl> -80.78568, -80.78568, -80.78568, ~
```

```r
# 4: Subsetting the dataset
Ozone_Processed <- Ozone_Raw[,c(1, 5, 7)]
glimpse(Ozone_Processed)
```

```
## Rows: 3,589
## Columns: 3
## $ Date                                 <date> 2010-01-01, 2010-01-02, 2010-01-~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.031, 0.033, 0.035, 0.031, 0.027~
## $ DAILY_AQI_VALUE                      <int> 29, 31, 32, 29, 25, 31, 32, 30, 3~
```

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

```
## Rows: 3,652
## Columns: 1
## $ Date <date> 2010-01-01, 2010-01-02, 2010-01-03, 2010-01-04, 2010-01-05, 2010~
```

```
# 6
GaringerOzone <- left_join(Days, Ozone_Processed, by="Date")
glimpse(GaringerOzone)
```

```
## Rows: 3,652
## Columns: 3
## $ Date                                 <date> 2010-01-01, 2010-01-02, 2010-01-~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.031, 0.033, 0.035, 0.031, 0.027~
## $ DAILY_AQI_VALUE                      <int> 29, 31, 32, 29, 25, NA, 31, 32, 3~
```

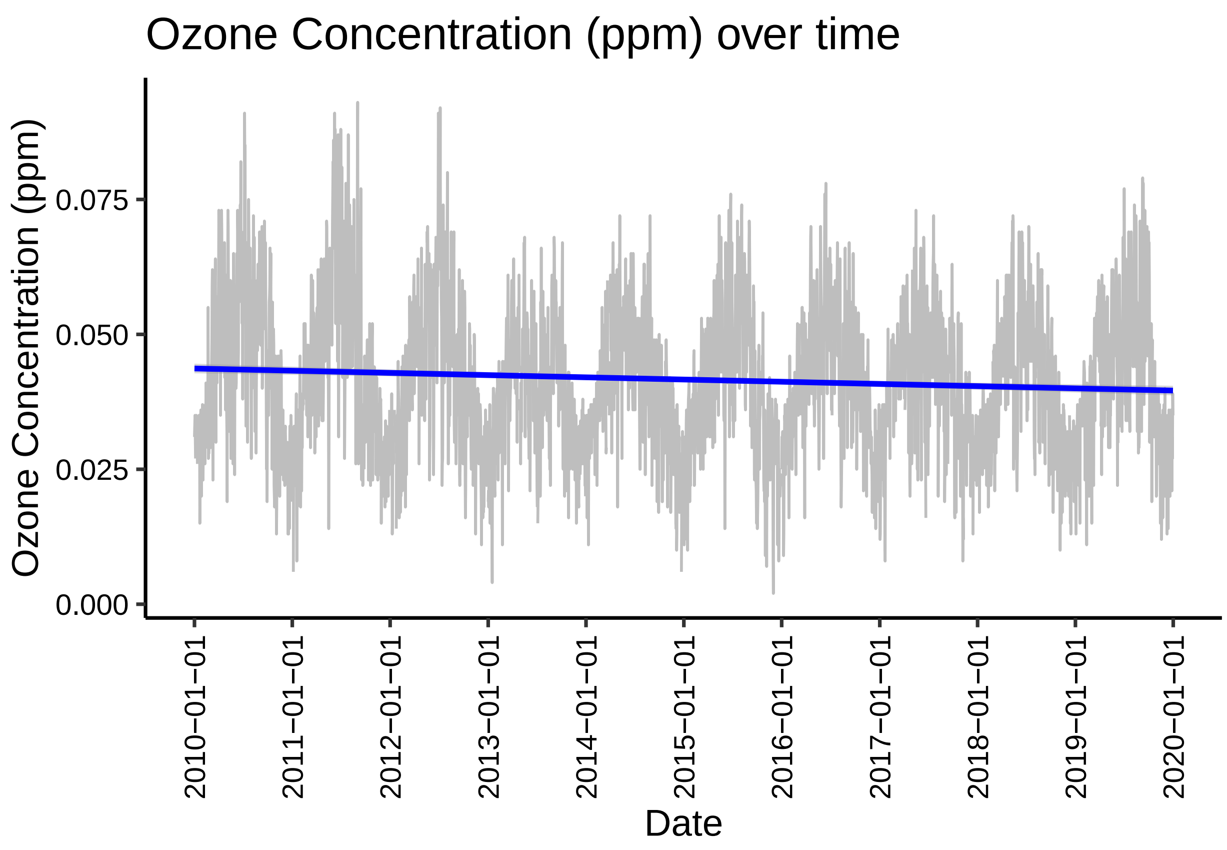
Visualize

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(GaringerOzone, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line(color="gray")+
  scale_x_date(breaks = "1 year")+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  xlab("Date")+
  ylab("Ozone Concentration (ppm)")+
  ggtitle("Ozone Concentration (ppm) over time")+
  geom_smooth(color = "blue", method = "lm")
```

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

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



Answer: The data shows a very strong seasonal trend where it peaks at its highest and lowest values every six months. The overall linear trend shows a mild but not very significant decline.

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

#Checking if it worked
glimpse(GaringerOzone)
```

```
## Rows: 3,652
## Columns: 4
## $ Date                                              <date> 2010-01-01, 2010-01~
## $ Daily.Max.8.hour.Ozone.Concentration              <dbl> 0.031, 0.033, 0.035,~
## $ DAILY_AQI_VALUE                                   <int> 29, 31, 32, 29, 25, ~
## $ Daily.Max.8.hour.Ozone.Concentration_Interpolated <dbl> 0.03100, 0.03300, 0.~
```

```r
sum(is.na(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration))
```

```
## [1] 63
```

```r
sum(is.na(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration_Interpolated))
```

```
## [1] 0
```

```r
#Removing the ozone concentration column that doesn't have the interpolated values
GaringerOzone <- GaringerOzone[,c(1,3,4)]
#Renaming the columns for ease
colnames(GaringerOzone) <- c("Date", "DailyAQI", "DailyOzoneConcentration")
#Checking dataset
glimpse(GaringerOzone)
```

```
## Rows: 3,652
## Columns: 3
## $ Date                    <date> 2010-01-01, 2010-01-02, 2010-01-03, 2010-01-0~
## $ DailyAQI                <int> 29, 31, 32, 29, 25, NA, 31, 32, 30, 30, 28, 24~
## $ DailyOzoneConcentration <dbl> 0.03100, 0.03300, 0.03500, 0.03100, 0.02700, 0~
```

Answer: We don't use piecewise constant because that simply copies the previous/ next data point, which we know is not accurate in this case, since there is a seasonal trend in the data and consecutive datapoints are unlikely to be the same. Spline is not used because this dataset is not non-linear and so a quadratic function is not appropriate. Linear interpolation is the most appropriate since it is seasonal linear data and so the average of the previous and next datapoints of a missing data point is likely to be a good estimate.

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
#Calculating monthly means
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Year = lubridate::year(Date), Month = lubridate::month(Date)) %>%
  group_by(Year, Month)%>%
  summarise(Monthly_Mean_Ozone = mean(DailyOzoneConcentration))
```

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

```r
#Creating a new date column for month-year and adding the monthly mean ozone values to it.
DateMonthly <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "months"))
colnames(DateMonthly) <- "Date_Monthly"
GaringerOzone.monthly <- cbind(GaringerOzone.monthly, DateMonthly)

#Only retaining the Date_Monthly date column and removing the separate Year and Month columns
GaringerOzone.monthly <- GaringerOzone.monthly[,c(3,4)]
head(GaringerOzone.monthly)
```

```
## # A tibble: 6 x 2
##   Monthly_Mean_Ozone Date_Monthly
##                <dbl> <date>
## 1             0.0305 2010-01-01
## 2             0.0345 2010-02-01
## 3             0.0446 2010-03-01
## 4             0.0556 2010-04-01
## 5             0.0466 2010-05-01
## 6             0.0576 2010-06-01
```

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.

```r
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$DailyOzoneConcentration,
               start = c(year(first(GaringerOzone$Date)), month(first(GaringerOzone$Date))),
               frequency = 365)
glimpse(GaringerOzone.daily.ts)
```

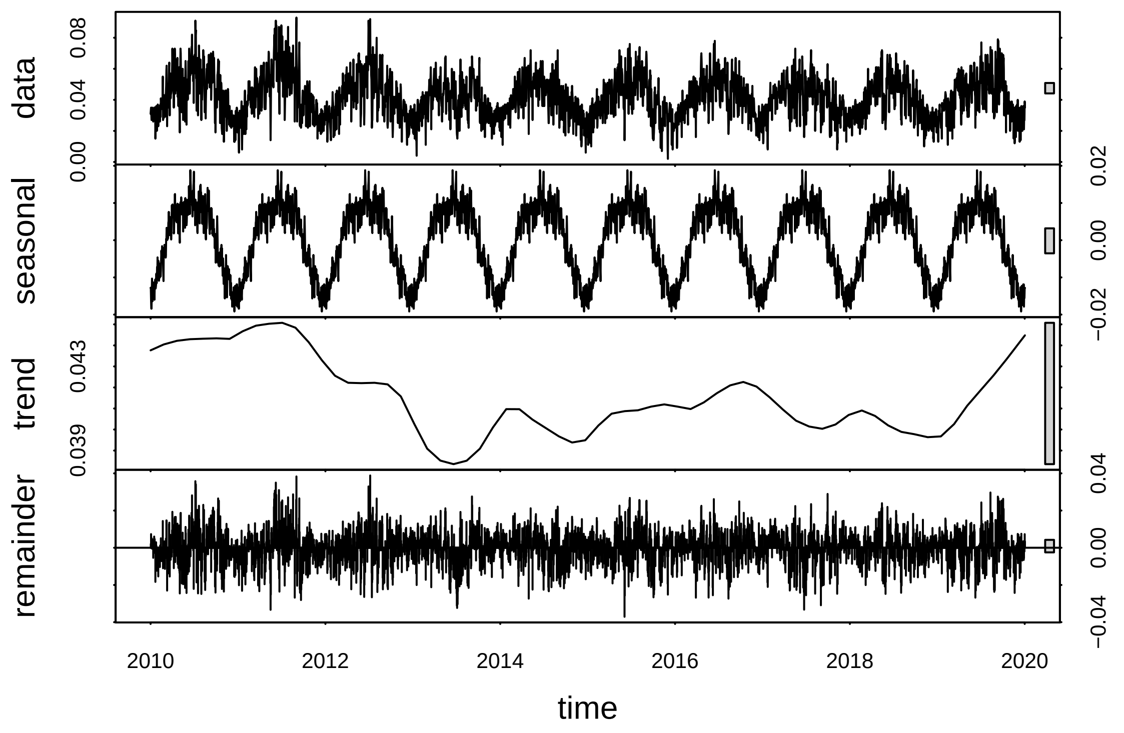
```
##  Time-Series [1:3652] from 2010 to 2020: 0.031 0.033 0.035 0.031 0.027 0.03 0.033 0.035 0.032 0.032
```

```r
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Monthly_Mean_Ozone, frequency = 12,
              start = c(year(first(GaringerOzone$Date)), month(first(GaringerOzone$Date))))
glimpse(GaringerOzone.monthly.ts)
```

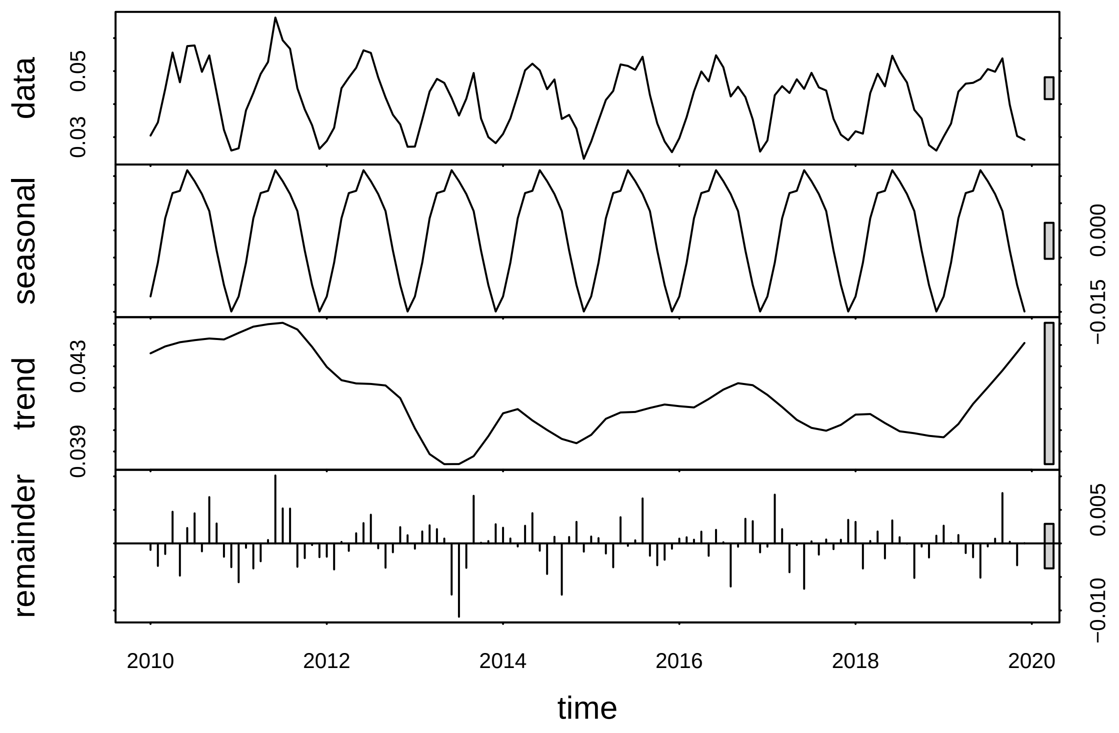
```
##  Time-Series [1:120] from 2010 to 2020: 0.0305 0.0345 0.0446 0.0556 0.0466 ...
```

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

```r
#11
GaringerOzone.daily.ts_decomposed <- stl(GaringerOzone.daily.ts, s.window="periodic")
plot(GaringerOzone.daily.ts_decomposed)
```



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

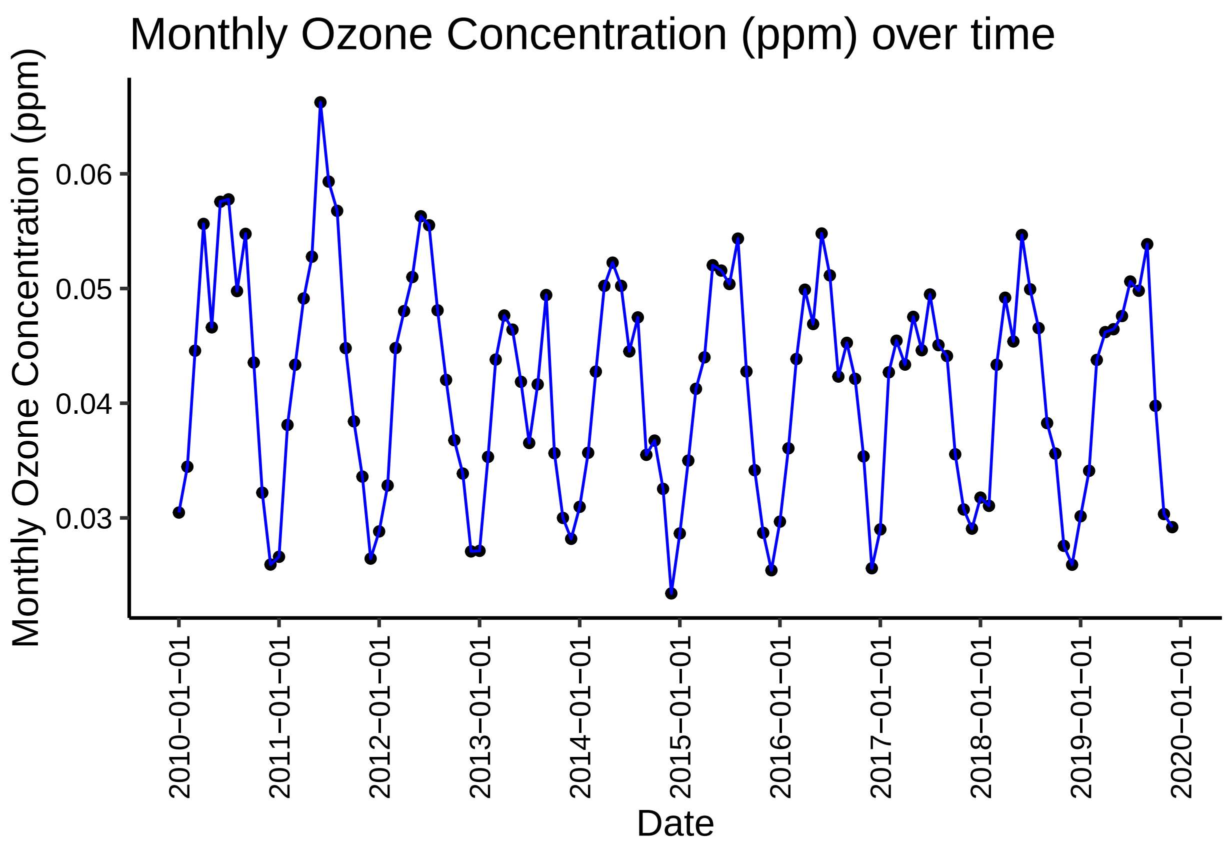
```r
#12
summary(SeasonalMannKendall(GaringerOzone.monthly.ts))
```

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

Answer: Seasonal Mann Kendall is the only appropriate test for this dataset set since it is seasonal data whereas all the other tests can only be used for data with no seasonality

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.

```r
# 13
ggplot(GaringerOzone.monthly, aes(x=Date_Monthly, y=Monthly_Mean_Ozone))+
  geom_point()+
  geom_line(color = "blue")+
  scale_x_date(breaks = "1 year")+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  xlab("Date")+
  ylab("Monthly Ozone Concentration (ppm)")+
  ggtitle("Monthly Ozone Concentration (ppm) over time")
```



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: While considering the monthly mean data, the null hypothesis is that the data is iid (stationary), while the alternative hypothesis is that there is a trend in the data. The seasonal mann kendall test conducts this test on seasonal data and finds that that there is a small significant negative trend in the data. Since the p-value = 0.046 < 0.05, we reject the null hypothesis indicating that there is a trend in the data. The tau value represents the correlation of the datapoints with time which in this case being -0.143 indicates a slight negative trend. The score = -77 also indicates that the magnitude of the trend which is negative and thus declining. However, given that the p-value is very close to the threshold of 5% and the tau value is close to 0, these results should be validated with other tests.

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.deseasoned <- as.data.frame(GaringerOzone.monthly.ts_decomposed$time.series[,1:3]
GaringerOzone.monthly.deseasoned <- mutate(GaringerOzone.monthly.deseasoned,
        Observed = GaringerOzone.monthly$Monthly_Mean_Ozone,
        Date = GaringerOzone.monthly$Date_Monthly)
glimpse(GaringerOzone.monthly.deseasoned)
```

```
## Rows: 120
## Columns: 5
## $ seasonal  <dbl> -0.012164159, -0.005945745, 0.002231834, 0.006878411, 0.0072~
## $ trend     <dbl> 0.04360892, 0.04377124, 0.04393356, 0.04403138, 0.04412919, ~
## $ remainder <dbl> -0.0009770197, -0.0033612105, -0.0015847518, 0.0047235448, -~
## $ Observed  <dbl> 0.03046774, 0.03446429, 0.04458065, 0.05563333, 0.04661290, ~
## $ Date      <date> 2010-01-01, 2010-02-01, 2010-03-01, 2010-04-01, 2010-05-01,~
```

```
#The "trend" column of the decomposed dataset in Qn 11 represents the deseasoned data
#ie original data minus the seasonal component
GaringerOzone.monthly.trend <- GaringerOzone.monthly.ts_decomposed$time.series[,2]

#16

summary(SeasonalMannKendall(GaringerOzone.monthly.trend))
```

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

Answer: The SMK test gives a more robust answer this time around. The p-value is actually statistically significant as it is considerably lower than 0.05, the tau value (ie correlation) rises in magnitude to -0.304 ie the negative trend is stronger, and the S value (score) also more than doubles in value to -164 indicating a stronger negative trend.