Assignment 7: Time Series Analysis

Natasha Jacob

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 Checking working directory
getwd()
```

```
## [1] "/Users/natashajacob/Desktop/EDA872/Environmental_Data_Analytics_2022"
```

```
#Loading in the required packages
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   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

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

```
library(lubridate)
```

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

## The following objects are masked from 'package:base':
##
```

```
##     date, intersect, setdiff, union
```

```
library(zoo)
```

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

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

```
library(trend)

#Setting the ggplot theme
mytheme <- theme_classic(base_size = 12) +
  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 Importing datasets
EPA_2010 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2011 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2012 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2013 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2014 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2015 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2016 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2017 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2018 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger
EPA_2019 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_Garinger

# Merging the ten datasets into one data frame using rbind
EPA_merged <- rbind(EPA_2010, EPA_2011, EPA_2012, EPA_2013,
                    EPA_2014, EPA_2015, EPA_2016, EPA_2017, EPA_2018, EPA_2019)

# Checking dimensions
dim(EPA_merged)
```

```
## [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.

```
# 3 Setting the date column as a date class
EPA_merged$Date <- as.Date(EPA_merged$Date, format = "%m/%d/%Y")
class(EPA_merged$Date)
```

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

```
# 4 Selecting the required columns
EPA_ozone <- select(EPA_merged, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5 Generating a daily dataset
Days <- as.data.frame(seq(as.Date("2010/01/01"), as.Date("2019/12/31"), "days"))
names(Days)[names(Days) == 'seq(as.Date("2010/01/01"), as.Date("2019/12/31"), "days")'] <- 'Date'

# 6 Combining the dataframes and checking dimensions
GaringerOzone <- left_join(Days, EPA_ozone)
```

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

```
dim(GaringerOzone)
```

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

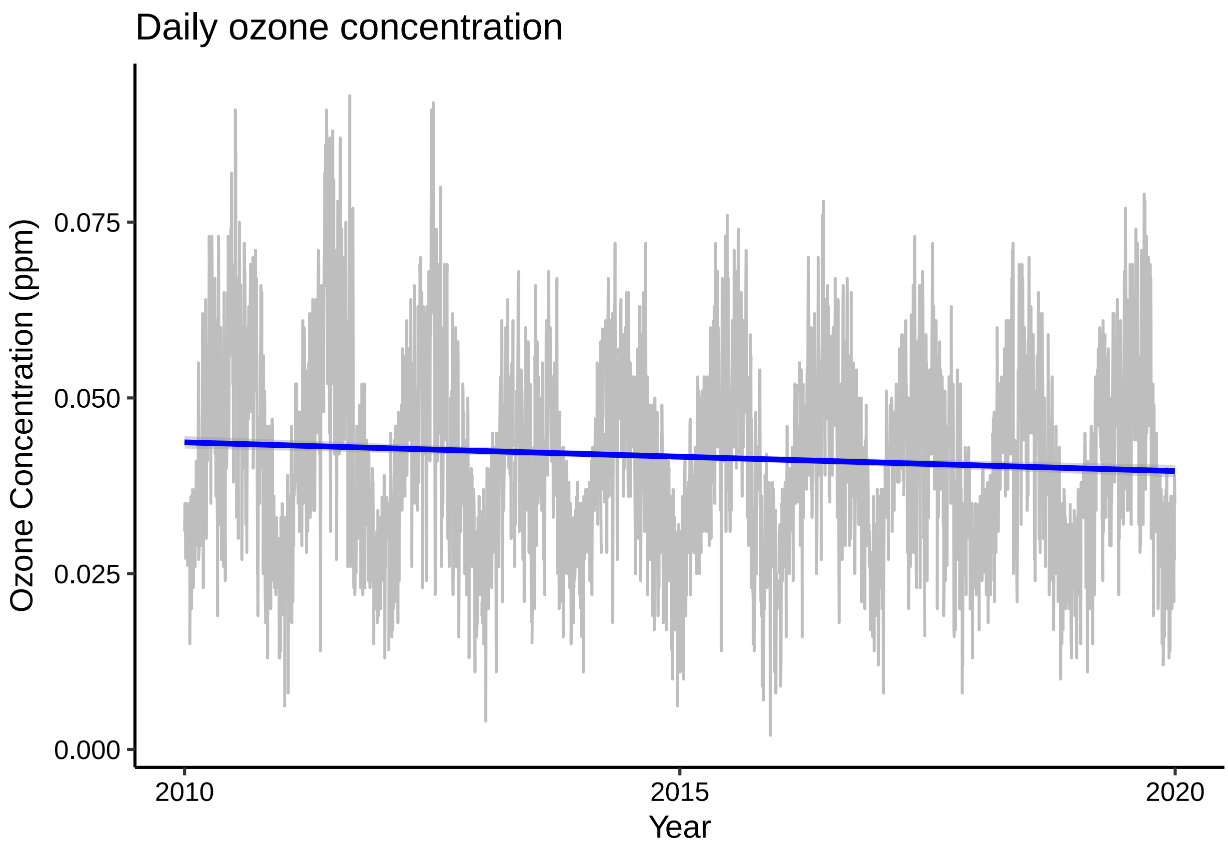
Visualize

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 Creating a line plot
OzoneConcentration <- ggplot(GaringerOzone,
                                  aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(color = "gray") +
  geom_smooth(method = lm, color = "blue") +
  ylab(expression("Ozone Concentration (ppm)")) +
  xlab(expression("Year")) +
  ggtitle("Daily ozone concentration") +
  mytheme
print(OzoneConcentration)
```

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

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



Answer: A slight decrease in ozone concentration over time can be observed from our plot. Hence, a trend can be observed.

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?

```
#Checking the summary of the dataset with NA's
summary(GaringerOzone)
```

```
##       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
```

```
#8 Using a linear interpolation to fill in missing daily data
GaringerOzone_clean <-
  GaringerOzone %>%
  mutate(Ozone.Concentration.Clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

#Note that the NA is gone
summary(GaringerOzone_clean)
```

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

Answer: A linear interpolation was used to fill the missing daily data since daily max 8 hour ozone concentration is a seasonal variation and the missing values were spread out in the dataset. The missing values did not follow a pattern.

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 Creating a new dataframe containing aggregated data
GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(Month = month(Date),
         Year = year(Date)) %>%
  mutate(Date = my(paste0(Month, "-", Year))) %>%
  dplyr::group_by(Date, Month, Year) %>%
  dplyr::summarise(OzoneMean = mean(Ozone.Concentration.Clean))
```

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

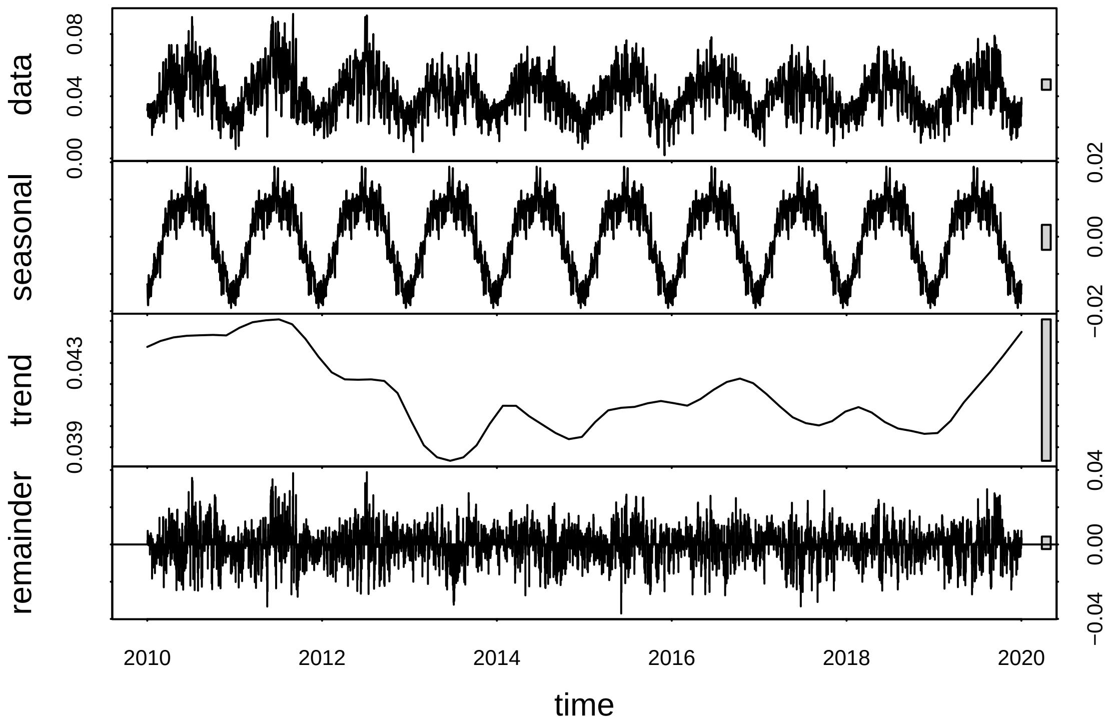
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```
#10 Generating a time series object based on daily observations
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Ozone.Concentration.Clean, start = c(2010, 1),
                             frequency = 365)
# Generating a time series object based on monthly average ozone observations
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$OzoneMean, start = c(2010,1),
                               frequency = 12)
```

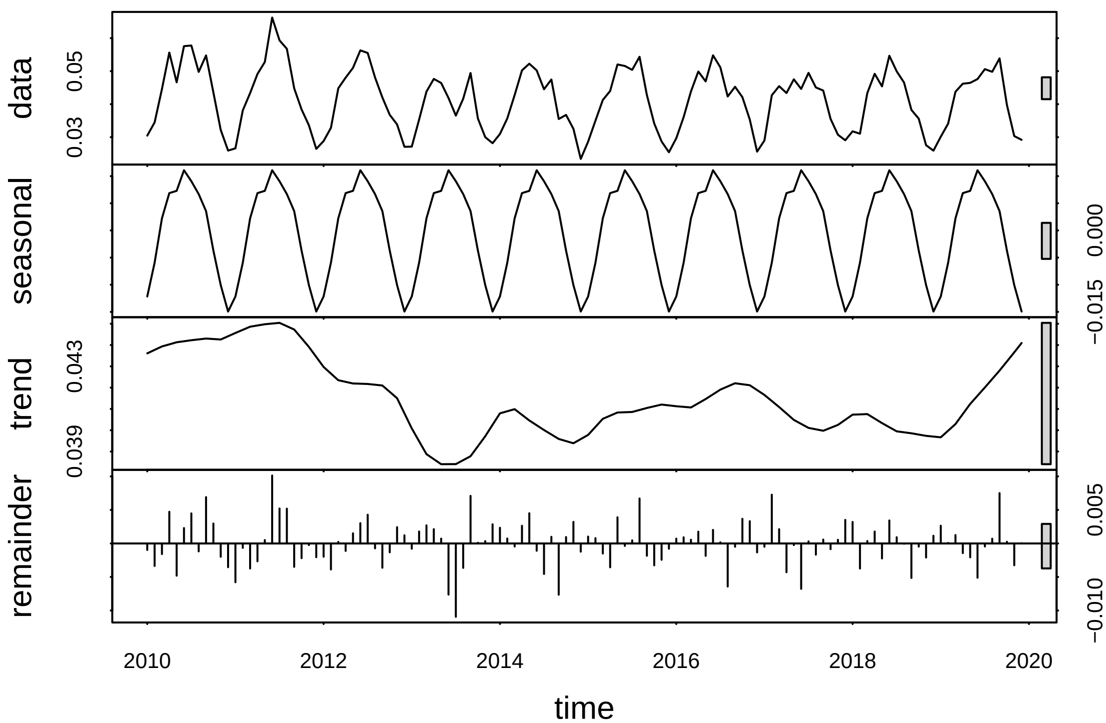
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11 Decomposing the daily and monthly time series objects and plotting the components

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




```
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 Running a monotonic trend analysis

Monthly_Ozone_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
```

```r
Monthly_Ozone_trend1
```

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

```r
summary(Monthly_Ozone_trend1)
```

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

```r
Monthly_Ozone_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
Monthly_Ozone_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
```

```r
summary(Monthly_Ozone_trend2)
```

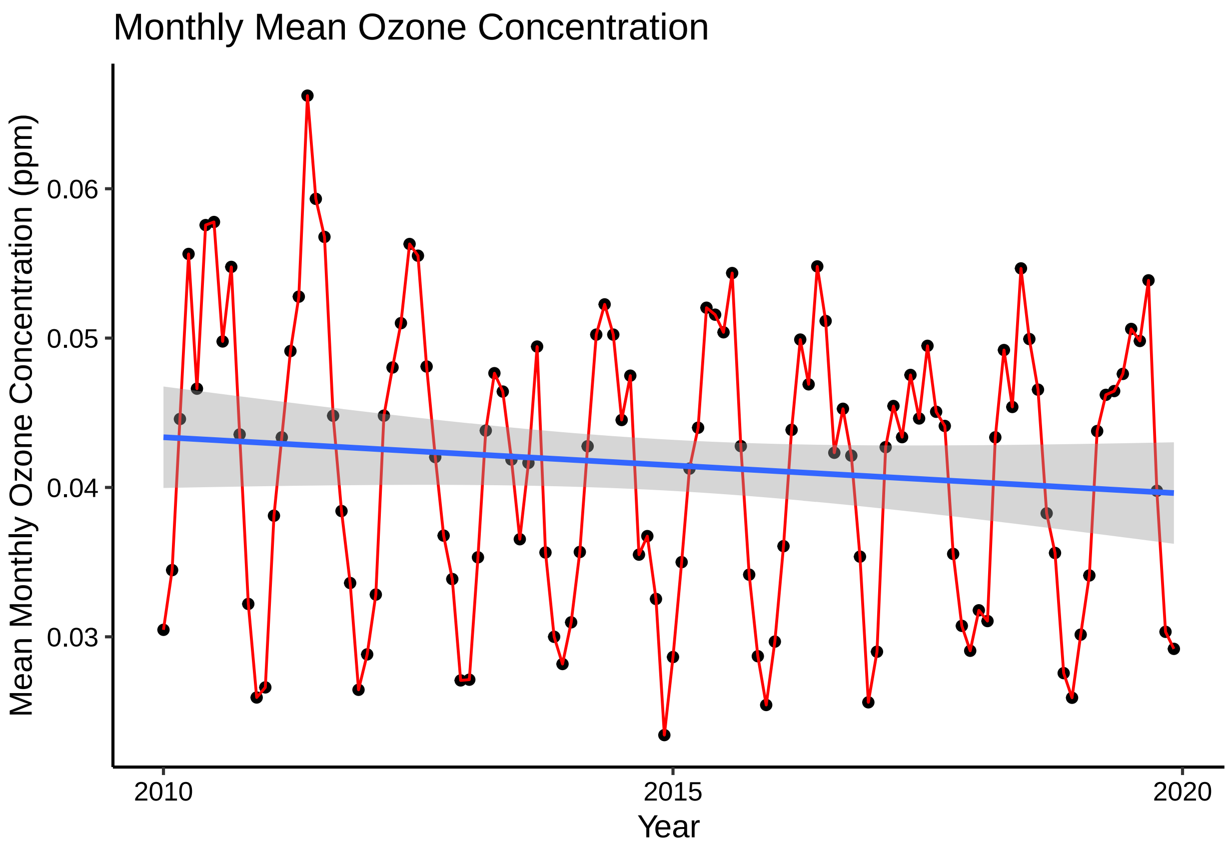
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                        S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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 seasonal Mann-Kendall test was used in this case since the montly ozone series is seasonal data i.e. an up and down repetitive movement occuring periodically can be seen.

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 Creating a plot depicting mean monthly ozone concentrations over time
monthly_ozone_conc_plot <-
  ggplot(GaringerOzone.monthly, aes(x = Date, y = OzoneMean)) +
  geom_point() +
  geom_line(color = "red") +
  ylab("Mean Monthly Ozone Concentration (ppm)") +
  xlab("Year") +
  ggtitle("Monthly Mean Ozone Concentration") +
  geom_smooth(method = lm)
print(monthly_ozone_conc_plot)
```

```
## `geom_smooth()` using 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: The graph shows a slight decline in monthly mean ozone concentration over the years. However, the results from the Seasonal Mann-Kendall test (trend 1) provies a p value almost equal to 0.05 (tau = -0.143, 2-sided pvalue =0.046724). Hence, we accept the null hypothesis that our data is stationary and conclude that there is no trend. From the results of the second Seasonal Mann-Kendall test (trend 2), we can observe a p value almost equal to 0.05 (z = -1.963, p-value = 0.04965). From the results of the summary we can see that there is not much variation in S values i.e, there is no strong increase or decrease. Note that there are no significant p values in the summary. From this, we can conclude that there is no significant trend.

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 Subtracting the seasonal component from the monthly time series object

GaringerOzone.monthly_components <- as.data.frame(GaringerOzone.monthly_Decomposed$time.series[,1:3])
GaringerOzone.monthly_components <- select(GaringerOzone.monthly_components, trend, remainder)
GaringerOzone.monthly_components <- mutate(GaringerOzone.monthly_components,
                                           Observed = GaringerOzone.monthly$OzoneMean,
                                           date = GaringerOzone.monthly$Date)



#16 Running the Mann Kendall test on the non-seasonal Ozone monthly series

GaringerOzone.monthly_components_ts <- ts(GaringerOzone.monthly_components$Observed, start = c(2010,1),
                                          frequency = 12)
GaringerOzone.monthly_components_trend1 <- Kendall::MannKendall(GaringerOzone.monthly_components_ts)
GaringerOzone.monthly_components_trend1
```

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

```
summary(GaringerOzone.monthly_components_trend1)
```

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

```
GaringerOzone.monthly_components_trend2 <- trend::smk.test(GaringerOzone.monthly_components_ts)
GaringerOzone.monthly_components_trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly_components_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_components_trend2)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly_components_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 Mann Kendall test on the non-seasonal Ozone monthly series for trend 1 previdesa result of tau = -0.143, 2-sided pvalue =0.046724. Since the pvalue is almost equal to 0.05, we can say that there is no trend and that the series is stationary. From the summary of trend 2 we can see that there is not much variation in S values i.e, there is no strong increase or decrease. Note that none of the pvalues are significant as well. The smk test provides a result of z = -1.963, p-value = 0.04965 (pvalue almost equal to 0.05) indicating that there is no trend.

The Man-Kendall results from both the seasonal and non-seasonal ozone monthly series seems to be similar, indicating that there is no trend.