Assignment 7: Time Series Analysis

Rachel Gordon - Section 1

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

getwd()#working directory
```

```
## [1] "/Users/rachelgordon/Documents/Duke Spring 2022/Environmental Data Analytics/Environmental_Data_
```

```
library(tidyverse) #load 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(lubridate)#load lubridate
```

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

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

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

```
#install.packages("zoo") #install zoo
library(zoo) #load zoo
```

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

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

```
#install.packages("trend") #install trend
library(trend) #load trend


#building my theme
RGtheme <- theme_gray(base_size = 12) +
  theme(axis.text = element_text(color = "black"))+
  theme(plot.title = element_text(hjust = 0.5))

#setting my theme
theme_set(RGtheme)
```

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 individual datasets
GaringerOzone1 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors = TRUE)
GaringerOzone2 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors = TRUE)
GaringerOzone3 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors = TRUE)
GaringerOzone4 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors = TRUE)
GaringerOzone5 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors = TRUE)
GaringerOzone6 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors = TRUE)
GaringerOzone7 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors = TRUE)
GaringerOzone8 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors = TRUE)
GaringerOzone9 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors = TRUE)
GaringerOzone10 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors = TRUE)

#binding datasets into single dataframe
GaringerOzone <- rbind.data.frame(
  GaringerOzone1,GaringerOzone2,GaringerOzone3,GaringerOzone4,
```

```
  GaringerOzone5,GaringerOzone6,GaringerOzone7,GaringerOzone8,
  GaringerOzone9,GaringerOzone10, stringsAsFactors = TRUE)
```

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
class(GaringerOzone$Date) #checking class of date
```

```
## [1] "factor"
```

```
GaringerOzone$Date <-as.Date(GaringerOzone$Date,
                             format = "%m/%d/%Y") #changing class to date
class(GaringerOzone$Date) #rechecking class
```

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

```
# 4
#wrangling dataset
GaringerOzoneWrangled <-
  GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration,
         DAILY_AQI_VALUE)
summary(GaringerOzoneWrangled)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-03   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2015-01-04   Median :0.04100                      Median : 38.00
##  Mean   :2015-01-01   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-02   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
```

```
sum(is.na(GaringerOzoneWrangled))
```

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

```
# 5
Days <-
as.data.frame(seq.Date(as.Date("2010/01/01"),
                       as.Date("2019/12/31"), "day")) #creating day dataframe

  colnames(Days) <- "Date" #renaming column to Date
```

```
# 6
#left joining by Date
GaringerOzone <- left_join(Days, GaringerOzoneWrangled)
```

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

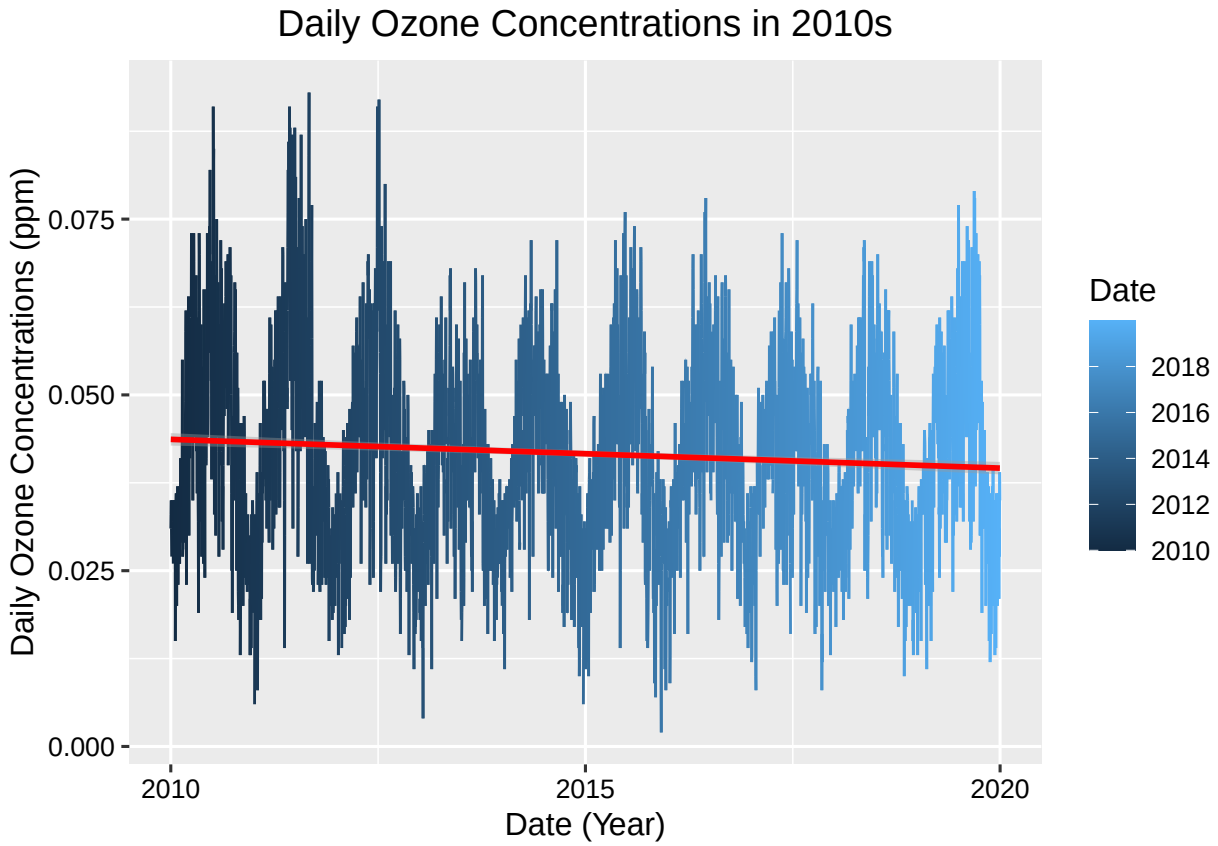
Visualize

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
library(ggplot2)

#line plot of ozone over time
Ozone.Over.Time <-
  ggplot(GaringerOzone, aes(x=Date, y =
                              Daily.Max.8.hour.Ozone.Concentration))+
  geom_line(aes(color=Date))+
  geom_smooth(method = 'lm', color = "red")+
  labs(x = "Date (Year)", y = "Daily Ozone Concentrations (ppm)",
       title = "Daily Ozone Concentrations in 2010s")

print(Ozone.Over.Time)
```

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

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



Answer: Although the smoothed line may indicate there is an overall decreasing trend in the data, the up and down movements of seasonality are masking the overall trend, making it difficult to draw any conclusions about the ozone concentrations over time. Because of the seasonality in the data, we cannot make a clear interpretation of the monotonic trend, and therefore need to conduct a time series analysis.

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
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration) #checking number of N/A's
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
#using linear interpolation to fill in missing data for ozone
GaringerOzone.daily <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration_Clean =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(GaringerOzone.daily$Daily.Max.8.hour.Ozone.Concentration_Clean)
```

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

Answer: The linear interpolation is used because that's the best method to "connect the dots" and insert the data for the missing measurement that falls wthin the previous and next measurement. Since this data seems to be changing over time, we want the linear interpolation instead of piecewise, as piecewise would be giving the exact same value to the measurement made nearest to that date. Additionally, spline interpolation isn't used, as we want a linear and straight line, and spline uses a quadratic function instead.

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
#new monthly data frame
GaringerOzone.monthly <-
  GaringerOzone.daily %>%
  mutate(Month = month(Date),
         Year = year (Date)) %>%
  mutate(Date = my(paste0(Month, "-", Year))) %>%
  dplyr::group_by(Date, Month, Year) %>%
  dplyr::summarise(mean_Ozone = mean(
    Daily.Max.8.hour.Ozone.Concentration_Clean)) %>%
  select(mean_Ozone, Date)
```

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

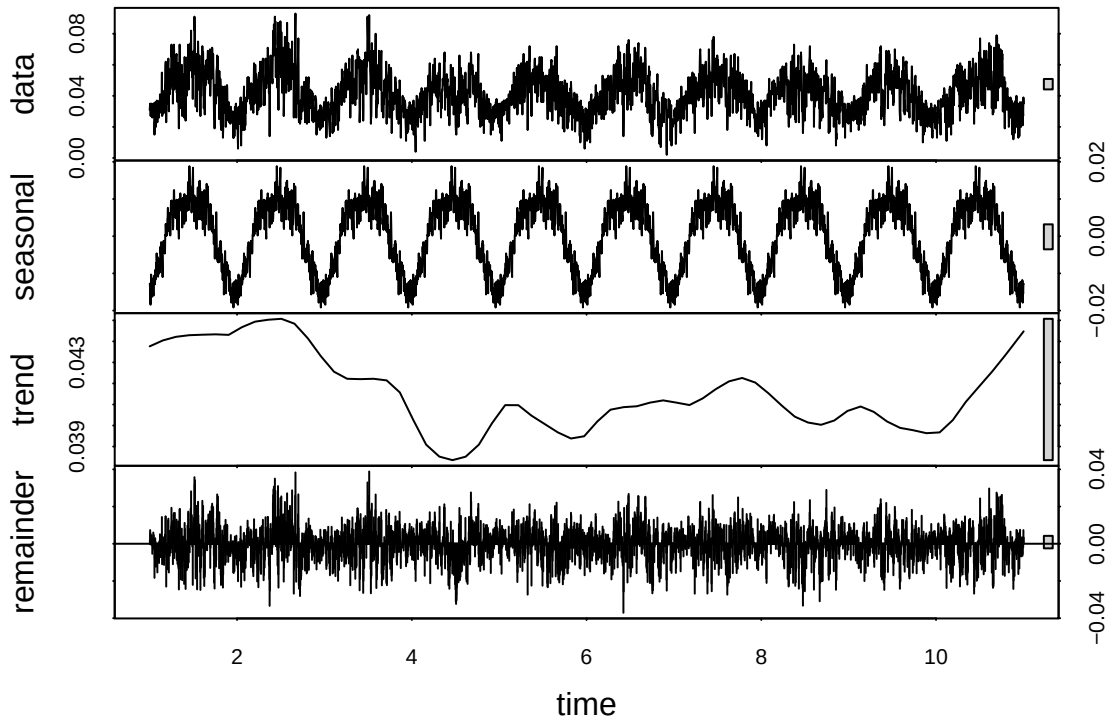
```
## Adding missing grouping variables: `Month`
```

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.

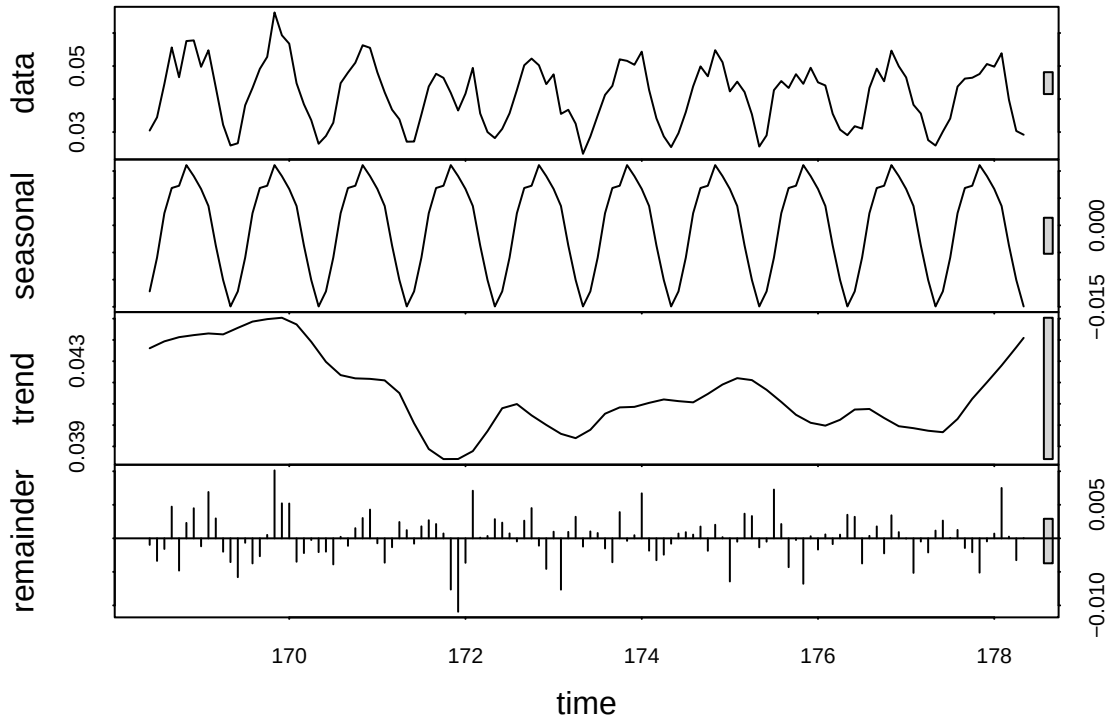
```
#10

#creating first date objects
fdaily <- day(first(GaringerOzone.daily$Date)) #extracting first day of date column
fmonthly <- month(first(GaringerOzone.monthly$Date)) #extracting first month
fyear <- year(first(GaringerOzone.monthly$Date)) #extracting first year

#daily time series object
  GaringerOzone.daily.ts <- ts(
    GaringerOzone.daily$Daily.Max.8.hour.Ozone.Concentration_Clean,
                               start = c(fdaily,fmonthly, fyear), frequency=365)
  #monthly time series object
  GaringerOzone.monthly.ts <- ts(
    GaringerOzone.monthly$mean_Ozone,
                                 start = c(fmonthly,fyear), frequency=12)
```

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

```
#11

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




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

#running seasonal Mann-Kendall
GaringerOzone.monthly.trend1 <- Kendall::SeasonalMannKendall(
  GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.trend1)
```

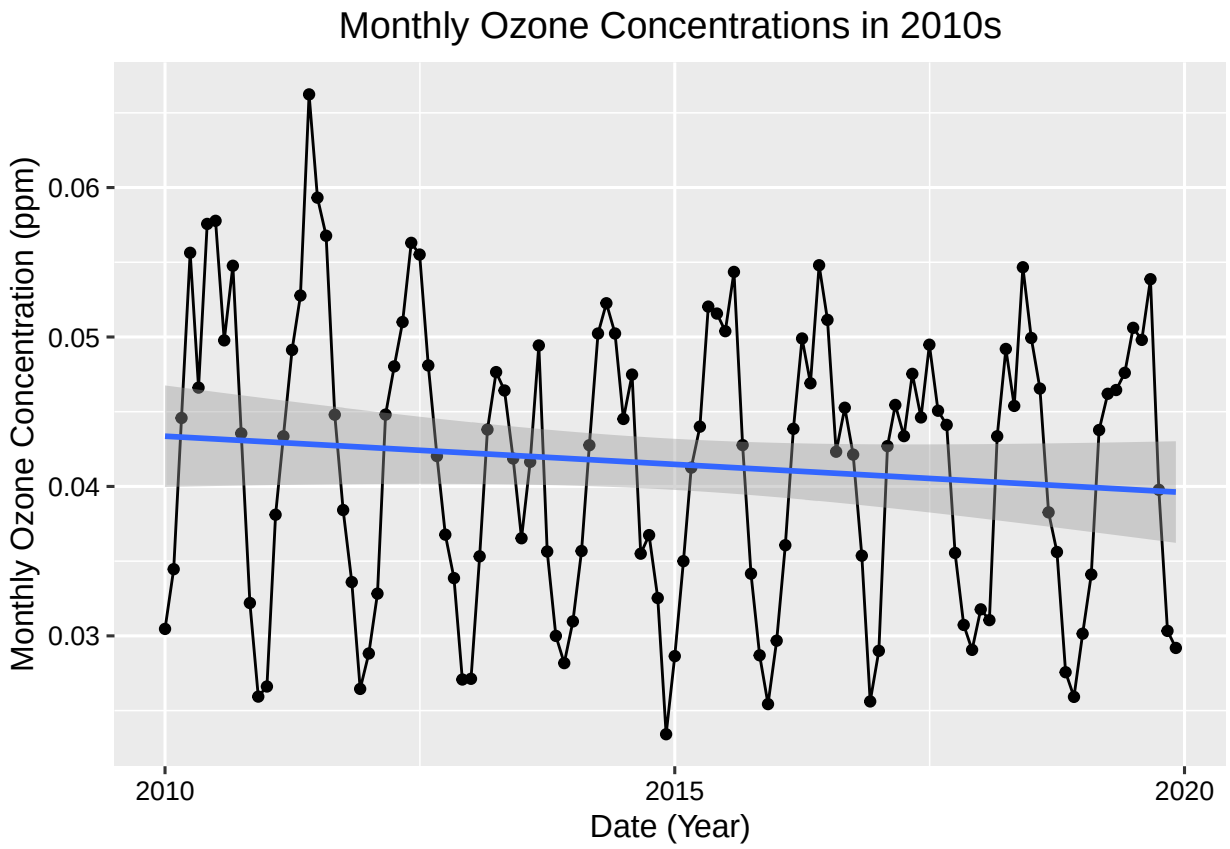
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann-Kendall is most appropriate as this data is seasonal, and the seasonal Mann-Kendall is the only monotonic trend analysis test that can handle seasonal data.

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
Garinger.Ozone.monthly.plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_Ozone)) +
  geom_point() +
  geom_line() +
  labs(x = "Date (Year)", y = "Monthly Ozone Concentration (ppm)",
       title = "Monthly Ozone Concentrations in 2010s") +
  geom_smooth( method = lm )
print(Garinger.Ozone.monthly.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: As the research question is asking if the ozone concentrations have changed over the 2010s at this station, the null hypothesis is that they did not change, and the alternative hypothesis is that ozone concentrations have changed. The results of the seasonal Mann Kendall statistical test show that the pvalue is less than .05, indicating that the null hypothesis should be rejected, and ozone concentrations have changed over the 2010s. Additionally, since the tau is negative, this shows that the ozone concentrations are decreasing over time. (Score = -77 , Var(Score) = 1499, denominator = 539.4972, tau = -0.143, 2-sided pvalue =0.046724)

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 seasonal component
GaringerOzone.monthly.ts_Components <-
  as.data.frame(
    GaringerOzone.monthly.decomp$time.series[,2:3])

#adding trend + remainder together into Observed column
GaringerOzone.monthly.ts_Components <- mutate(
  GaringerOzone.monthly.ts_Components,
        Observed = GaringerOzone.monthly.ts_Components$trend +
    GaringerOzone.monthly.ts_Components$remainder,
  Date = GaringerOzone.monthly$Date)

#16


#creating first year and month object
fmonthly.trend2 <- month(first(
  GaringerOzone.monthly.ts_Components$Date)) #extracting first month
fyear.trend2 <- year(first(
  GaringerOzone.monthly.ts_Components$Date)) #extracting first year

#creating new monthly time series for 'Observed' column
GaringerOzone.monthly.trend2.ts <- ts(
  GaringerOzone.monthly.ts_Components$Observed,
                                start = c(
                                  fmonthly.trend2,fyear.trend2), frequency=12)

#running non-seasonal Mann Kendall
GaringerOzone.monthly.trend2 <-Kendall::MannKendall(
  GaringerOzone.monthly.trend2.ts)
summary(GaringerOzone.monthly.trend2)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The Mann Kendall test gives a pvalue of .008 and a tau of -0.165. Like the seasonal Mann

Kendall, we would reject the null hypothesis since the p-value is less than .05 and can conclude that the ozone concentrations are changing over time since the 2010s. It is important to note that compared to the seasonal Mann Kendall, this p-value is much smaller, showing that there is stronger evidence in favor of the alternative hypothesis when we remove the seasonal component. Additionally, the tau is also negative in this test and greater in absolute value compared to the seasonal Mann Kendall, so without the seasonal component, ozone concentrations are shown to be decreasing even more compared to the season Mann Kendall (Score = -1179 , Var(Score) = 194365.7, denominator = 7139.5, tau = -0.165, 2-sided pvalue =0.0075402)