Assignment 7: Time Series Analysis

Katryna Niva

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] "/home/guest/R/EDA-Fall2022"
```

```
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

theme_set(mytheme)

#2
```

```
GH2010 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
GH2011 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
GH2012 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
GH2013 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
GH2014 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
GH2015 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
GH2016 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
GH2017 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
GH2018 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
GH2019 <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <- rbind(GH2010, GH2011, GH2012, GH2013, GH2014, GH2015, GH2016, GH2017, GH2018, GH2019)
```

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

GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4

GaringerOzone2 <- GaringerOzone %>%
  select(Date, `Daily Max 8-hour Ozone Concentration`, DAILY_AQI_VALUE)

# 5

Days <- as.data.frame(seq.Date(from=as.Date("2010-01-01"),to=as.Date("2019-12-31"), by=1))

colnames(Days) <- paste("Date")

# 6

GaringerOzone3 <- left_join(Days, GaringerOzone2, by = c("Date"))
```

Visualize

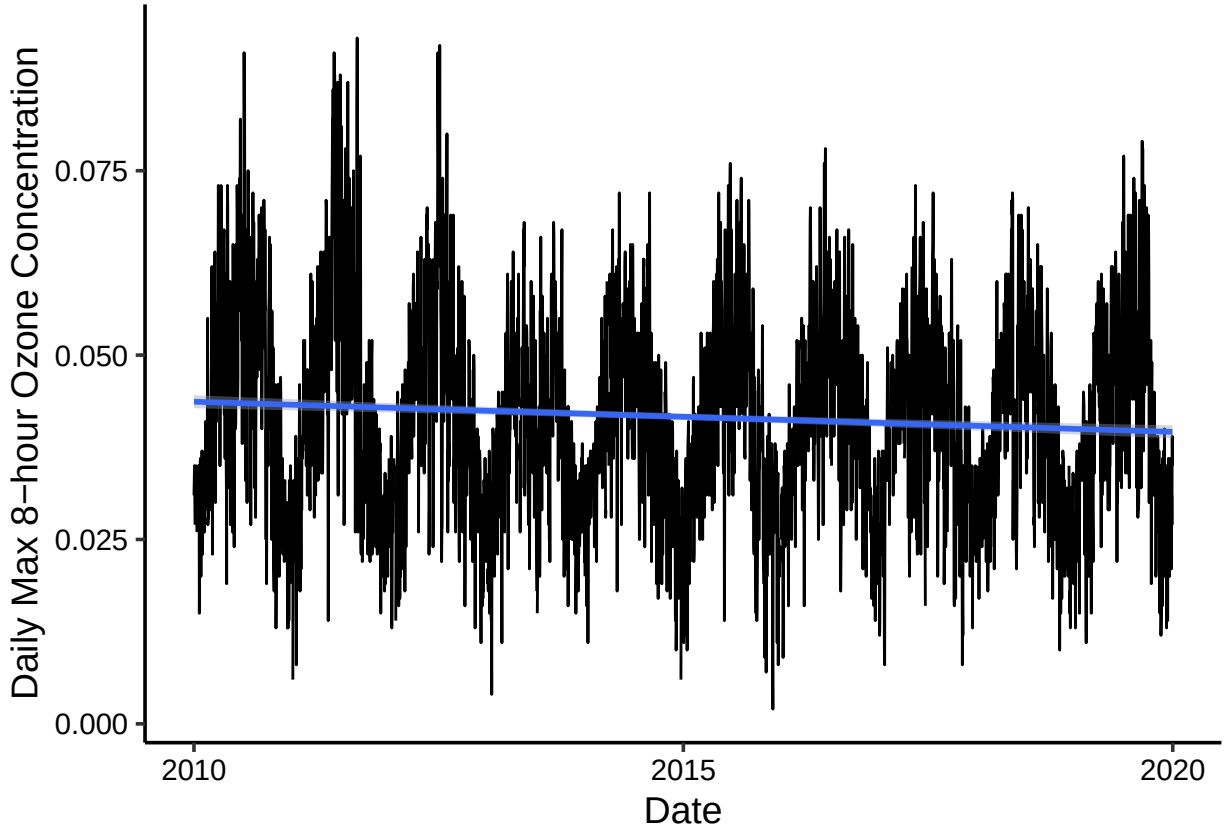
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
```

```
ggplot(GaringerOzone3, aes(x=Date, y=`Daily Max 8-hour Ozone Concentration`)) +
  geom_line()+
  geom_smooth(method="lm")
```

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

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



Answer: There is a small trend downward seen by the slight downward slope but it does not seem very significant.

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

GaringerOzone4 <-
  GaringerOzone3 %>%
  mutate(OzoneConc.clean = zoo::na.approx(`Daily Max 8-hour Ozone Concentration`))
```

Answer: The piecewise constant option would not be as ideal because it only takes data from one of the neighboring points. Since there is such strong seasonality instead of quantized ozone concentrations, it is better to consider both neighboring points and fill in the missing point at a place in between. Spline interpolation would also not be as ideal because most changes in ozone

concentration are due to a constant input and so a linear slope is the best way to approximate this change. Using the spline interpolation would use quadratic formulas which would imply a relationship that is not there in this case.

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 <- GaringerOzone4 %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date))%>%
  mutate(NewDate = my(paste(Month,"-",Year)))%>%
  group_by(NewDate)%>%
  summarise(MeanMonthlyOzone = mean(OzoneConc.clean))
```

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_day <- day(first(GaringerOzone4$Date))
f_month <- month(first(GaringerOzone4$Date))
f_year <- year(first(GaringerOzone4$Date))

f2_month <- month(first(GaringerOzone.monthly$Date))
```

```
## Warning: Unknown or uninitialised column: `Date`.
```

```
## Warning: tz(): Don't know how to compute timezone for object of class NULL;
## returning "UTC". This warning will become an error in the next major version of
## lubridate.
```

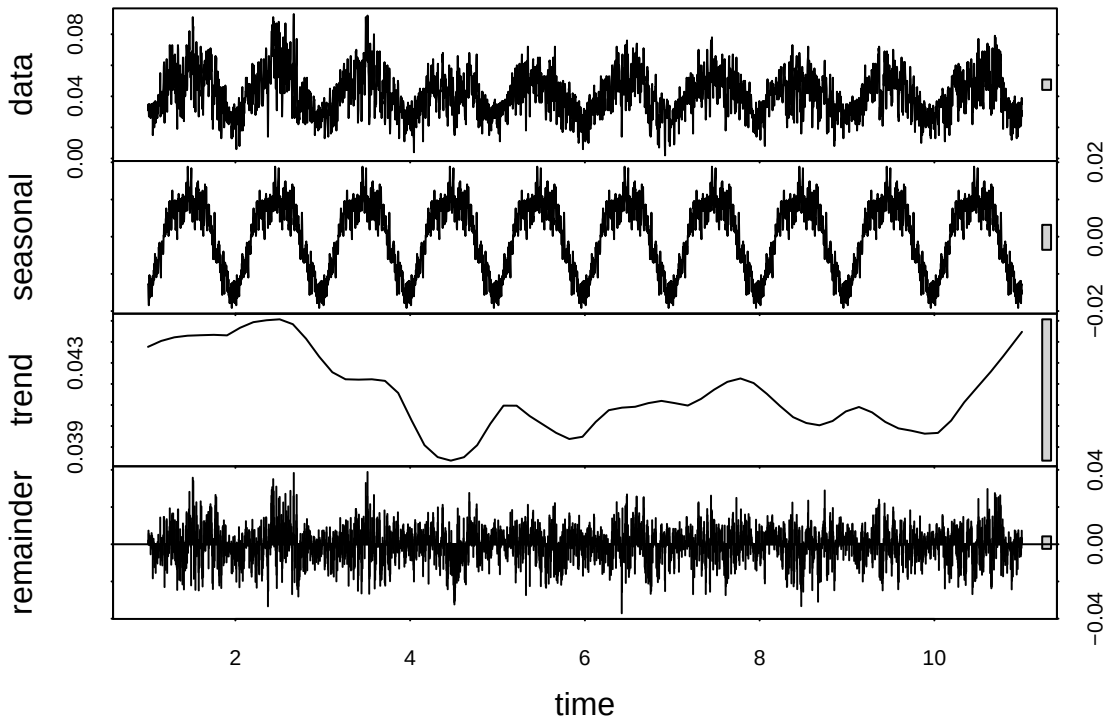
```
f2_year <- year(first(GaringerOzone.monthly$Date))
```

```
## Warning: Unknown or uninitialised column: `Date`.
## tz(): Don't know how to compute timezone for object of class NULL; returning "UTC". This warning wil
```

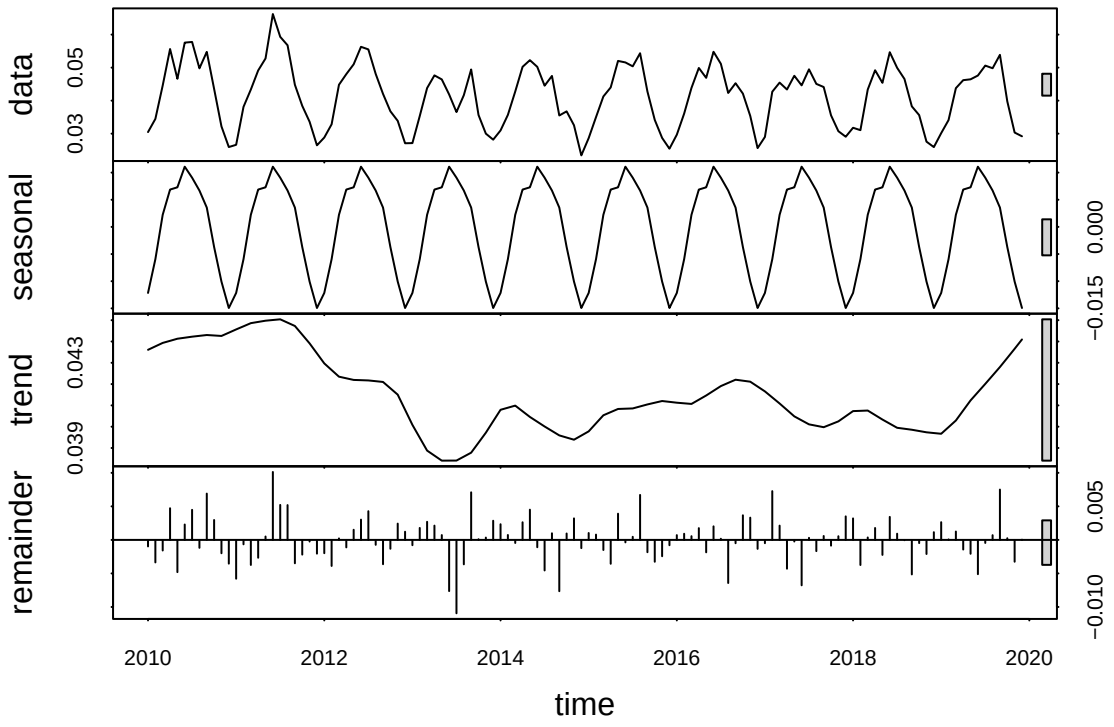
```
GaringerOzone.daily.ts <- ts(GaringerOzone4$OzoneConc.clean, frequency = 365, start = c(f_day, f_month,

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

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

```
#11

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?

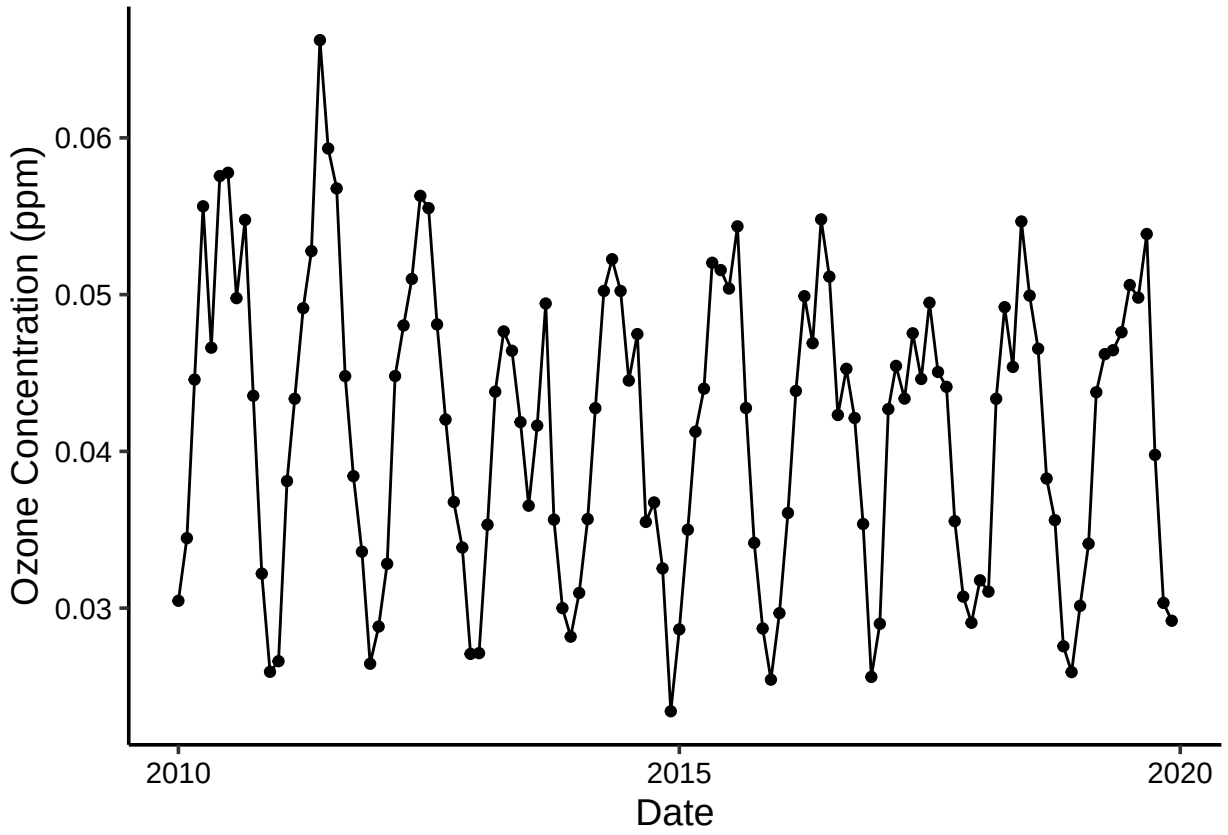
```
#12

GaringerOzone.monthly.MK <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

print(GaringerOzone.monthly.MK)
```

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

Answer: The seasonal Mann-Kendall is the best option because; 1) the ozone concentration has strong seasonality. This is seen in the decomposed plot from Q11 with the very small bar next to the seasonality plot. 2)The Mann-Kendall test is non-parametric and the large bar on the trend indicates that you should select a non-parametric test.

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

ggplot(GaringerOzone.monthly, aes(x=NewDate, y=MeanMonthlyOzone))+
  geom_line()+
  geom_point()+
  xlab("Date")+
  ylab("Ozone Concentration (ppm)")
```



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.

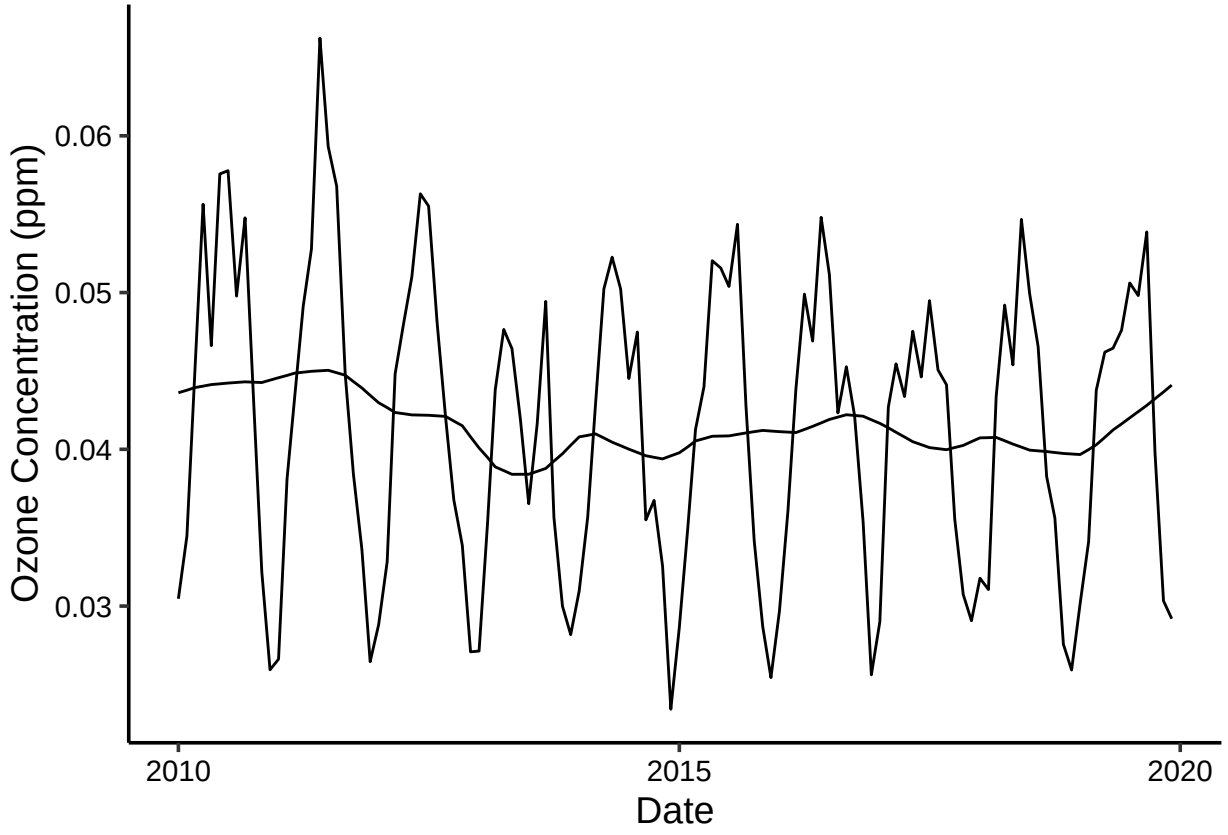
    Answer: The research question was: Have ozone concentrations changed over the 2010s at this station? We have concluded that ozone concentrations have not changed in a statistically relavent way in the 2010s at this station. The MK test yielded a 2-sided p-value of 0.0467 which is considerably higher than 0.05 when doubled as you must do with a 2-sided p-value. The S value is -77 which means that the ozone level is decreasing, just not in a significant way.

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.Components <- as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

GaringerOzone.monthly.Components2 <- mutate(GaringerOzone.monthly.Components, Observed = GaringerOzone.m


Q15 <- ggplot(GaringerOzone.monthly.Components2)+
  geom_line(aes(x=Date, y=Observed))+
  geom_line(aes(x=Date, y=trend))+
  xlab("Date")+
  ylab("Ozone Concentration (ppm)")
```

```
print(Q15)
```




```
#16

GaringerOzone.monthly.trend.ts <- ts(GaringerOzone.monthly.Components2$trend, frequency = 12, start = c

GaringerOzone.monthly.trend.MK <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.trend.ts)

print(GaringerOzone.monthly.trend.MK)
```

```
## tau = -0.304, 2-sided pvalue =2.291e-05
```

Answer: After removing the seasonality, the decrease in ozone over time is statistically significant. The p-value for the MK test is far lower than 0.05 at 2.291e-5. The S value is also more negative at -164 which indicates there is a greater trend in revealed in decreasing ozone. The statistical trend is clearly much cleaner without the seasonality.