Assignment 8: Time Series Analysis

Jasmine Papas

Spring 2023

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

```
#1
#load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
```

```
## 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)
library(here)
```

```
## here() starts at /home/guest/R/EDA-Spring2023
```

```
library(Kendall)

#check directory
getwd()
```

```
## [1] "/home/guest/R/EDA-Spring2023"
```

```
#set theme
lab8_theme<- theme_classic(base_size = 12)+
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
theme_set(lab8_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.

```
#2
#uploading files
EPAair_O3_GaringerNC2010<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
EPAair_O3_GaringerNC2011<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
EPAair_O3_GaringerNC2012<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
EPAair_O3_GaringerNC2013<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
EPAair_O3_GaringerNC2014<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
EPAair_O3_GaringerNC2015<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
EPAair_O3_GaringerNC2016<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
EPAair_O3_GaringerNC2017<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
EPAair_O3_GaringerNC2018<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
EPAair_O3_GaringerNC2019<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),

#creating dataset
GaringerOzone<- rbind(EPAair_O3_GaringerNC2010, EPAair_O3_GaringerNC2011, EPAair_O3_GaringerNC2012, EPA
```

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- Set as date
GaringerOzone$Date<- mdy(GaringerOzone$Date)

#4- Wrangle Data set
GaringerOzone_wrangled<- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

#5- Filling in NA
Days<- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31,01-01"), by="1 day"))
colnames(Days)<- "Date"

#6- Creating combined data frame
GaringerOzone<- left_join(Days,GaringerOzone_wrangled)
```

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

Visualize

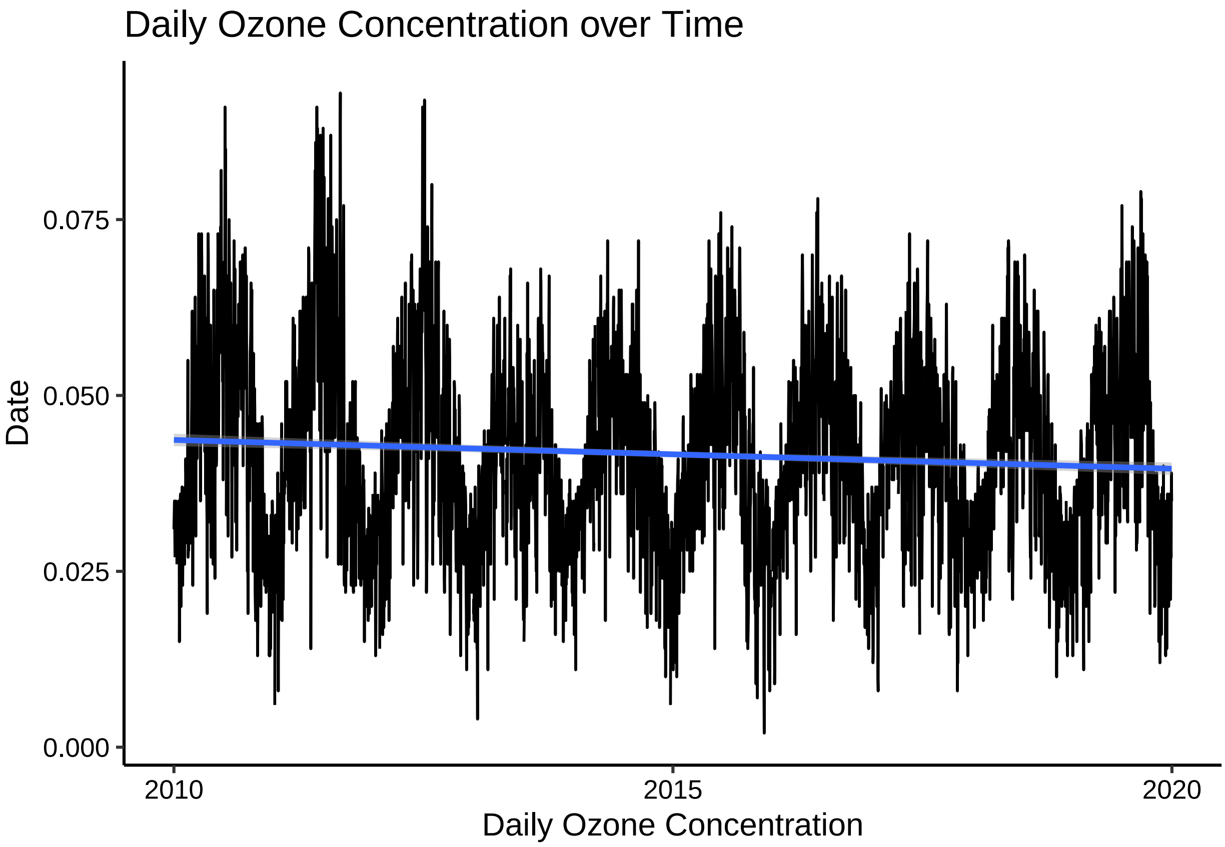
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
GaringerOzone_plot<- ggplot(GaringerOzone, aes(x= Date, y= Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method = lm)+
  labs(x="Daily Ozone Concentration", y= "Date")+
  ggtitle("Daily Ozone Concentration over Time")

print(GaringerOzone_plot)
```

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

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



Answer: The plot suggests that the ozone levels area slightly decreasing over the span of approximately 10 years.

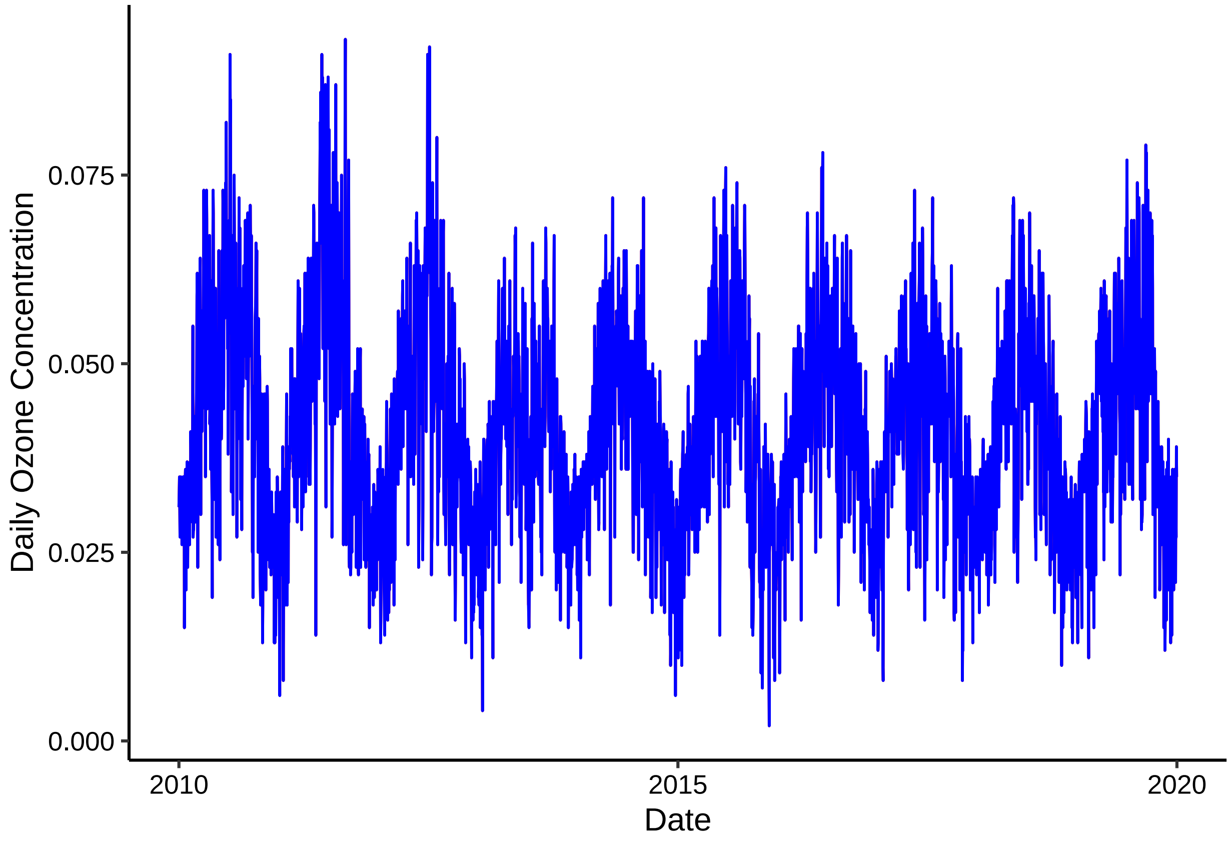
Time Series Analysis

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_clean <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentrati

ggplot(GaringerOzone_clean)+
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration), color = "red")+
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration.clean), color = "blue")+
  ylab("Daily Ozone Concentration")
```



Answer: We don't want to use a peicewise constant because it will just assume that the values are similar to the nearest neighbor value. Meanwhile, the spline interpolation uses a quadratic function to determine the line and that would cause too drastic of a value change for our graph. Instead, we just want it to fill in the NAs with data that will continue in a line and in the range we prefer.

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)

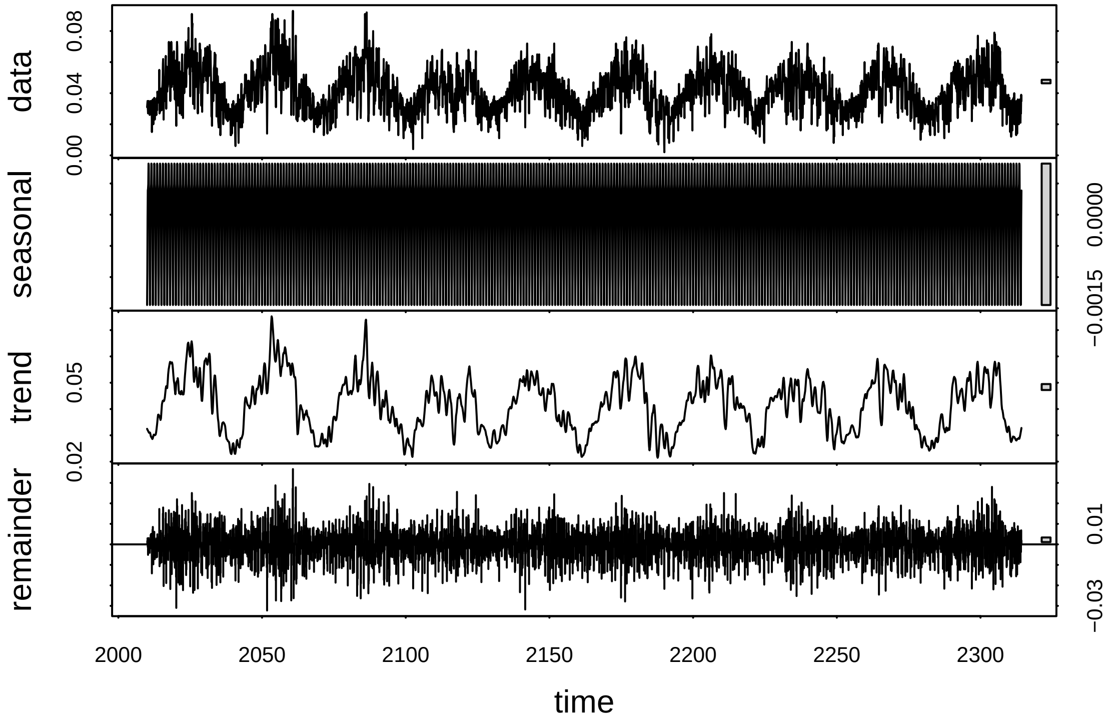
```
#9
GaringerOzone.monthly<- GaringerOzone_clean %>%
  separate(Date,c("Null", "Month", "Null2")) %>%
  rename(Year = Null) %>%
  mutate( Date = my(paste0(Month,"-",Year))) %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration.clean, DAILY_AQI_VALUE)
```

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.

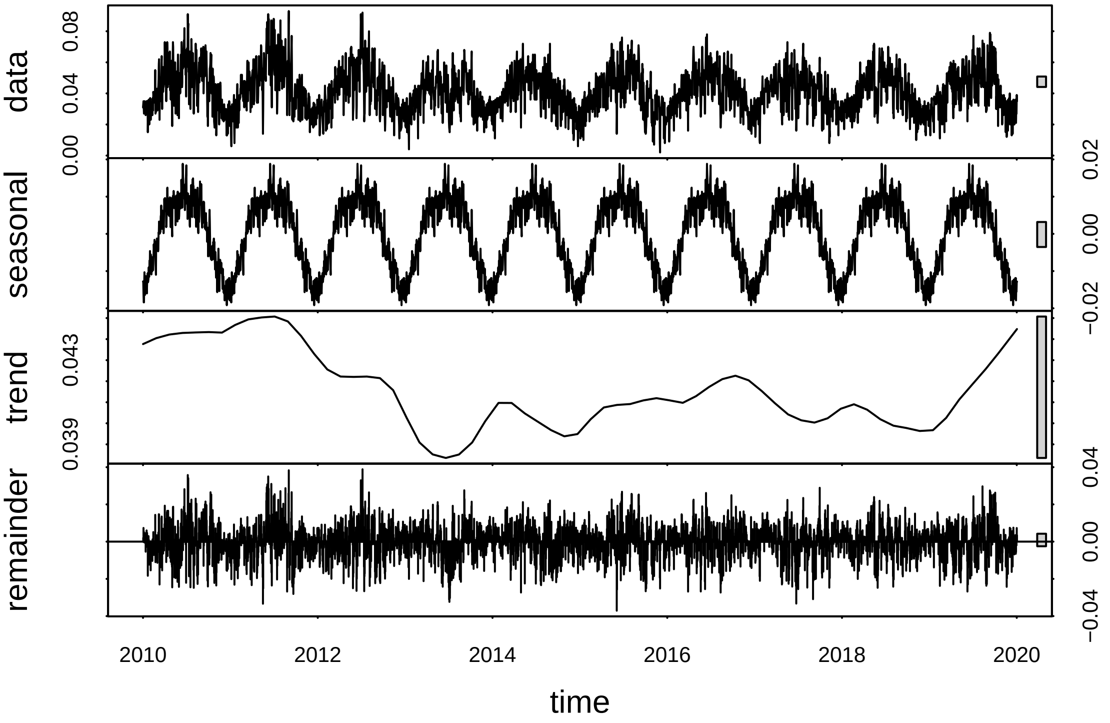
```
#10
GaringerOzone.daily.ts<- ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration.clean, start = c(20
GaringerOzone.monthly.ts<- ts(GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration.clean, start =
```

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

```
#11
GaringerOzone.monthly.decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly.decomp)
```



```
GaringerOzone.daily.decomp<- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.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
Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
```

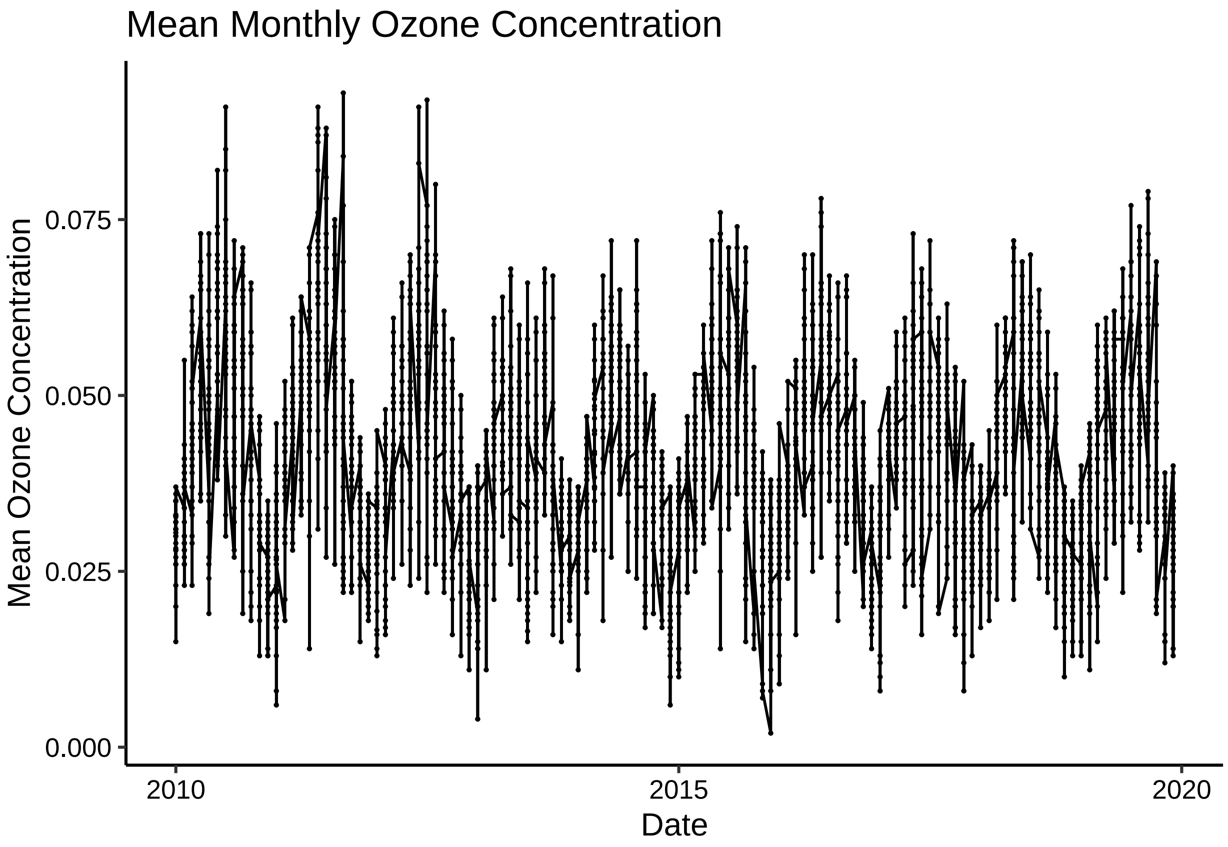
```
## tau = -0.0408, 2-sided pvalue =0.00027247
```

Answer: We wanted to use a seasonal Mann-Kendall test because it is the only tests that allowed us to factor in the seasonality of the data. It also compares the monthly data to each other over the years. For example, January data from 2010 is compared to January data from 2011, 2012, 2013, etc.

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
GaringerOzone.monthly.plot<- ggplot(GaringerOzone.monthly, aes(x= Date, y= Daily.Max.8.hour.Ozone.Conce
  geom_line()+
  geom_point(size = 0.25)+
  ylab("Mean Ozone Concentration")+
  ggtitle("Mean Monthly Ozone Concentration")

print(GaringerOzone.monthly.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: According to our Seasonal Mann-Kendall test, the ozone concentration is decreasing across the years. This was supported by the negative S value (-22362)that we recieved which means that the trend is decreasing. This can also be seen by drawing a trend line over the line plot. We also got a p-value of <0.5 which tells us that there is a monotonic trend in our data.

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_components<- as.data.frame(GaringerOzone.monthly.ts)

GaringerOzone_Components <- mutate(GaringerOzone_components,
        Observed =GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration.clean,
        Date = GaringerOzone.monthly$Date)

GaringerOzone_Components_ts<- ts(GaringerOzone_Components$Observed, start = c(2010, 1), frequency = 12)
#16

Kendall::MannKendall(GaringerOzone_Components_ts)
```

```
## tau = -0.0401, 2-sided pvalue =0.0003176
```

Answer: Both MannKendall tests showed that the ozone concentration levels are decreasing over time. For the nonseasonal MannKendall test, we got an S value of -264863 which is significantly larger than the seasonal MannKendall implying that the nonseasonal test showed a steeper rate of decrease in ozone concentration.