Assignment 8: Time Series Analysis

Ina Liao

Fall 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

```r
library(tidyverse)
library(lubridate)
library(trend)
library(zoo)
library(Kendall)
library(tseries)
library(here)
library(ggplot2)
library(ggthemes)
here()
```

```
## [1] "/Users/inaliao/Desktop/EDE_Fall2023"
```

```r
plot_theme<-theme_calc()+
  theme(
    #plot background
    plot.background=element_rect(color="gray"),

    #plot title
    plot.title=element_text(color="black",hjust=0.5,vjust=1),

    #axis labels
    axis.text=element_text(color="black"),

    #legend
    legend.key=element_rect(color="white"),
    legend.background=element_rect(color="white"),
    legend.position="right")

theme_set(plot_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

EPAair_2010<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"), stringsAsFact
EPAair_2011<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"), stringsAsFact
EPAair_2012<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"), stringsAsFact
EPAair_2013<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"), stringsAsFact
EPAair_2014<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"), stringsAsFact
EPAair_2015<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"), stringsAsFact
EPAair_2016<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"), stringsAsFact
EPAair_2017<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"), stringsAsFact
EPAair_2018<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"), stringsAsFact
EPAair_2019<-read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"), stringsAsFact
GaringerOzone<-rbind(EPAair_2010,EPAair_2011,EPAair_2012,EPAair_2013,EPAair_2014,EPAair_2015,EPAair_201
#GaringerOzone
```

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$Date<-mdy(GaringerOzone$Date)

# 4
colnames(GaringerOzone)
```

```
##  [1] "Date"
##  [2] "Source"
##  [3] "Site.ID"
##  [4] "POC"
##  [5] "Daily.Max.8.hour.Ozone.Concentration"
##  [6] "UNITS"
##  [7] "DAILY_AQI_VALUE"
##  [8] "Site.Name"
##  [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

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

# 5
# filling missing days with NA
# create a sequence as a data frame from 2010-01-01 until 2019-12-31
seq_date<-as.character(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "days"))
df_date<-as.data.frame(seq_date)
# rename column name
colnames(df_date)<-"Date"
df_date$Date<-ymd(df_date$Date)

# 6
GaringerOzone<-left_join(df_date,GaringerOzone,by="Date")
# notes: the order of left_join matters

#rename column names
GaringerOzone<-GaringerOzone%>%
  rename(Ozone=Daily.Max.8.hour.Ozone.Concentration)%>%
  rename(AQI=DAILY_AQI_VALUE)
```

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=Ozone))+
  geom_line()+
  geom_smooth(method=lm,color="blue")+
  labs(x="",y="ozone concentration (ppm)")
```

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

```
# missing data is removed in this plot
```

Answer:The plot does suggest a clear trend in ozone concentration over time. Further testing is needed to determine whether the data follows a trend.

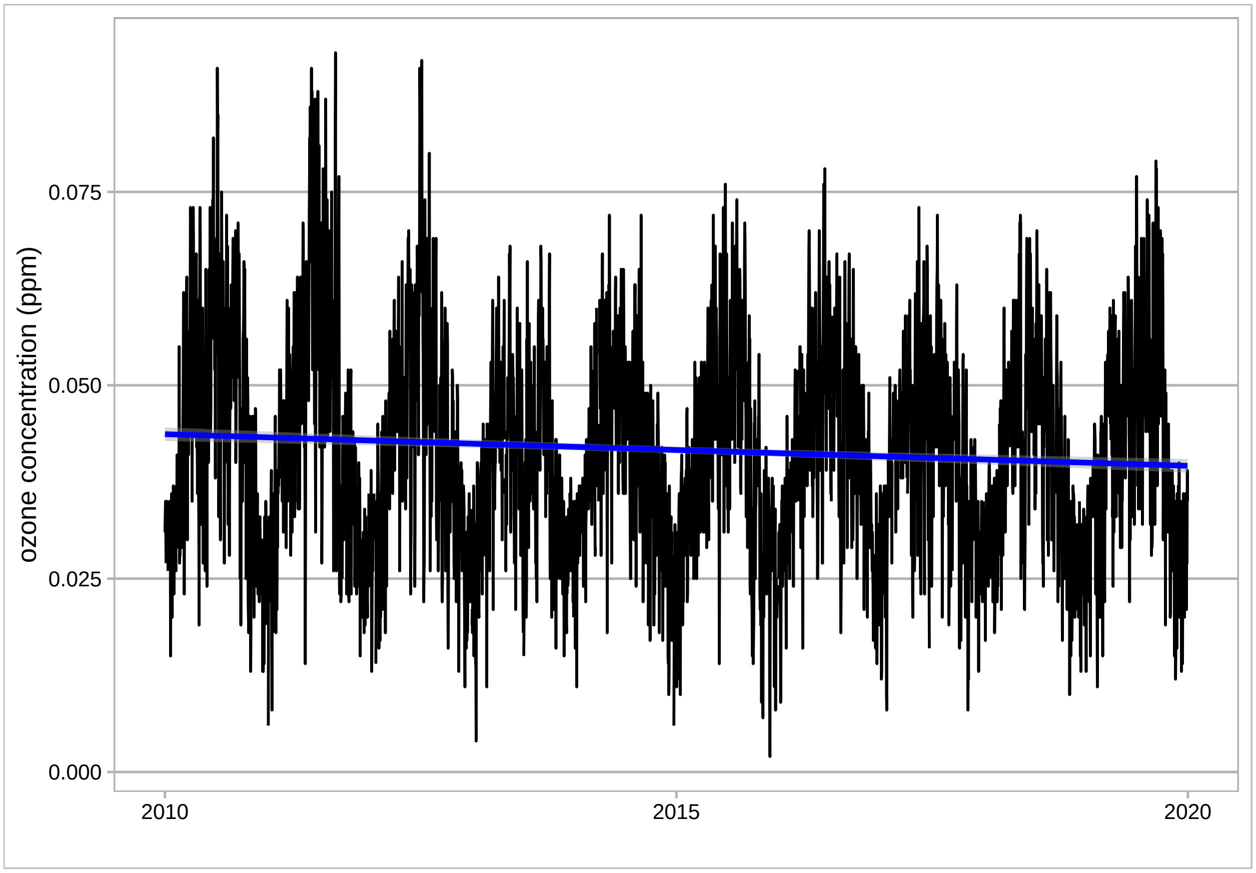


Figure 1: Ozone Concentrations over Time

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
# check missing observations
summary(GaringerOzone$Ozone)
```

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

```
# add a new column that replace NA by interpolation
GaringerOzone.clear<-GaringerOzone%>%
  mutate(Ozone_clean=zoo::na.approx(Ozone))
summary(GaringerOzone$Ozone_clear)
```

```
## Length  Class   Mode 
##      0   NULL   NULL
```

```
# use the red line to highlight the missing data
ggplot(GaringerOzone.clear) +
  geom_line(aes(x = Date, y = Ozone_clean), color = "red") +
  geom_line(aes(x = Date, y = Ozone), color = "black") +
  labs(x="",y="ozone concentration (ppm)")
```

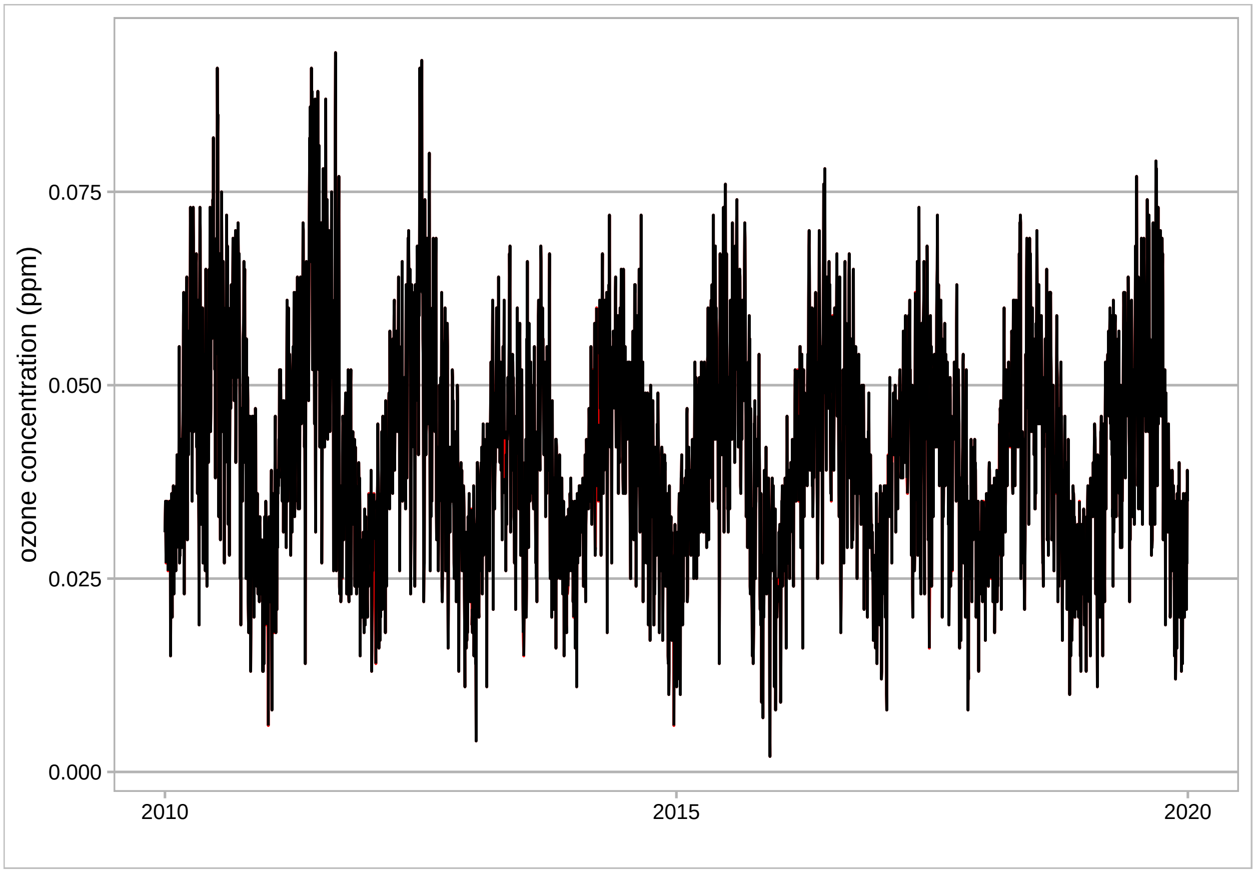


Figure 2: Ozone Concentrations over Time

Answer: The ozone concentration data exhibit a relatively smooth and continuous trend over the time; in addition, the data only have a short period of missing data, and thus, linear interpolation may provide a better approximation than piecewise constant and spline interpolation.

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)

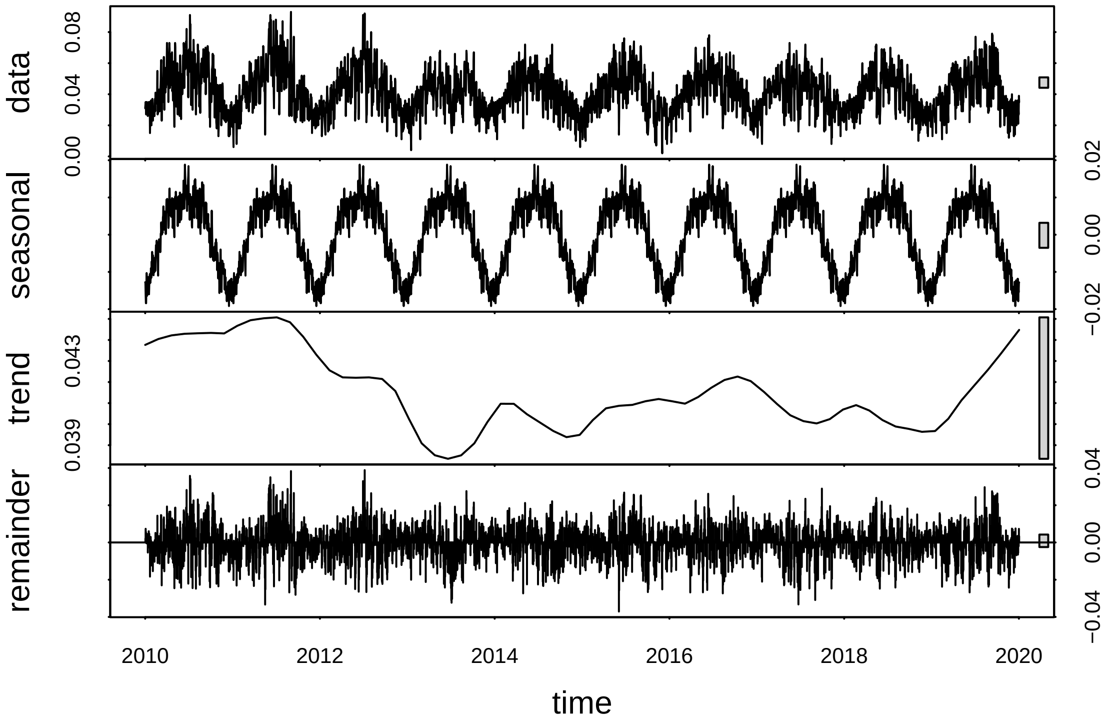
```
#9
GaringerOzone.monthly <- GaringerOzone.clear %>%
  separate(Date, c("Year","Month","Day"),"-") %>%
  group_by(Year,Month) %>%
  summarize(Ozone_clean=mean(Ozone_clean)) %>%
  mutate(Date=my(paste0(Month,"-",Year))) %>%
  select(Date,Ozone_clean)
```

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.

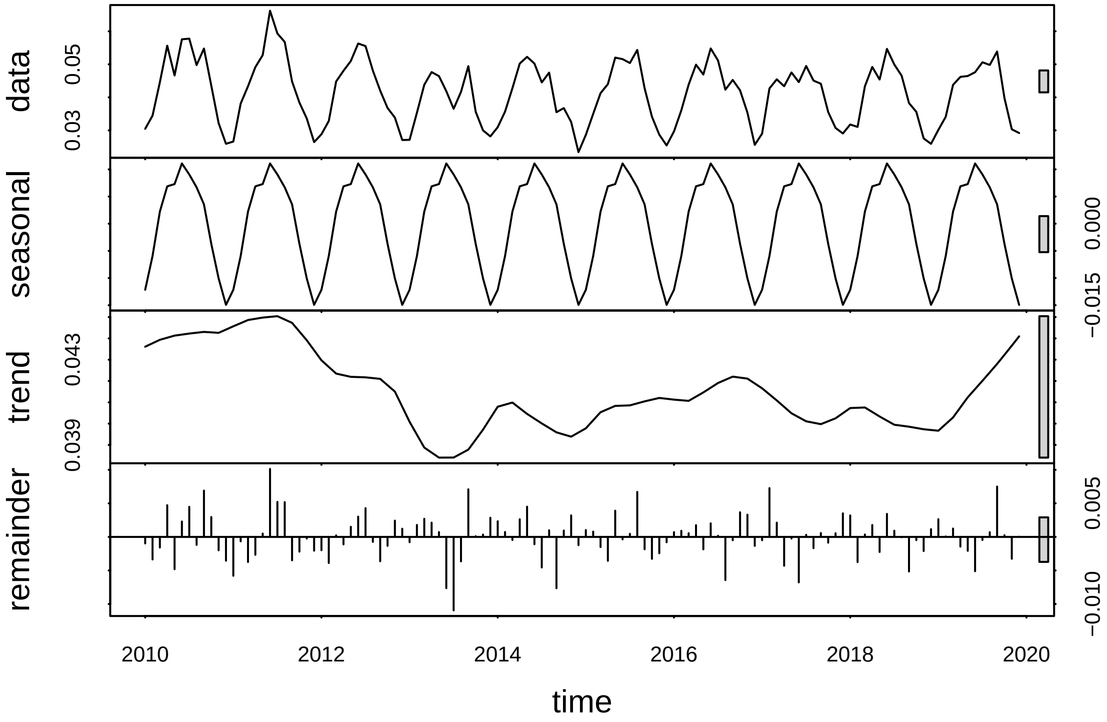
```
#10

# daily time series

# specific the first month and year
f_month_daily<-month(first(GaringerOzone.clear$Date))
f_year_daily<-year(first(GaringerOzone.clear$Date))
# create time series objects
GaringerOzone.daily.ts<-ts(GaringerOzone.clear$Ozone_clean,start=c(f_year_daily,f_month_daily),frequenc


# monthly time series

# specific the first month and year
f_month_monthly<-month(first(GaringerOzone.monthly$Date))
f_year_monthly<-year(first(GaringerOzone.monthly$Date))
# create time series objects
GaringerOzone.monthly.ts<-ts(GaringerOzone.monthly$Ozone_clean,start=c(f_year_monthly,f_month_monthly),
```

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

```
#11

# daily time series
# decompose
GaringerOzone.daily.decompose<-stl(GaringerOzone.daily.ts,s.window="periodic")
# plot
plot(GaringerOzone.daily.decompose)
```



```
# monthly time series
# decompose
GaringerOzone.monthly.decompose<-stl(GaringerOzone.monthly.ts,s.window="periodic")
# plot
plot(GaringerOzone.monthly.decompose)
```



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

# run Seasonal Mann-Kendall test
# Ho: data is stationary
# Ha: data follow a trend

Ozone_trend<- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
Ozone_trend_season<- trend::smk.test(GaringerOzone.monthly.ts)
summary(Ozone_trend)
```

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

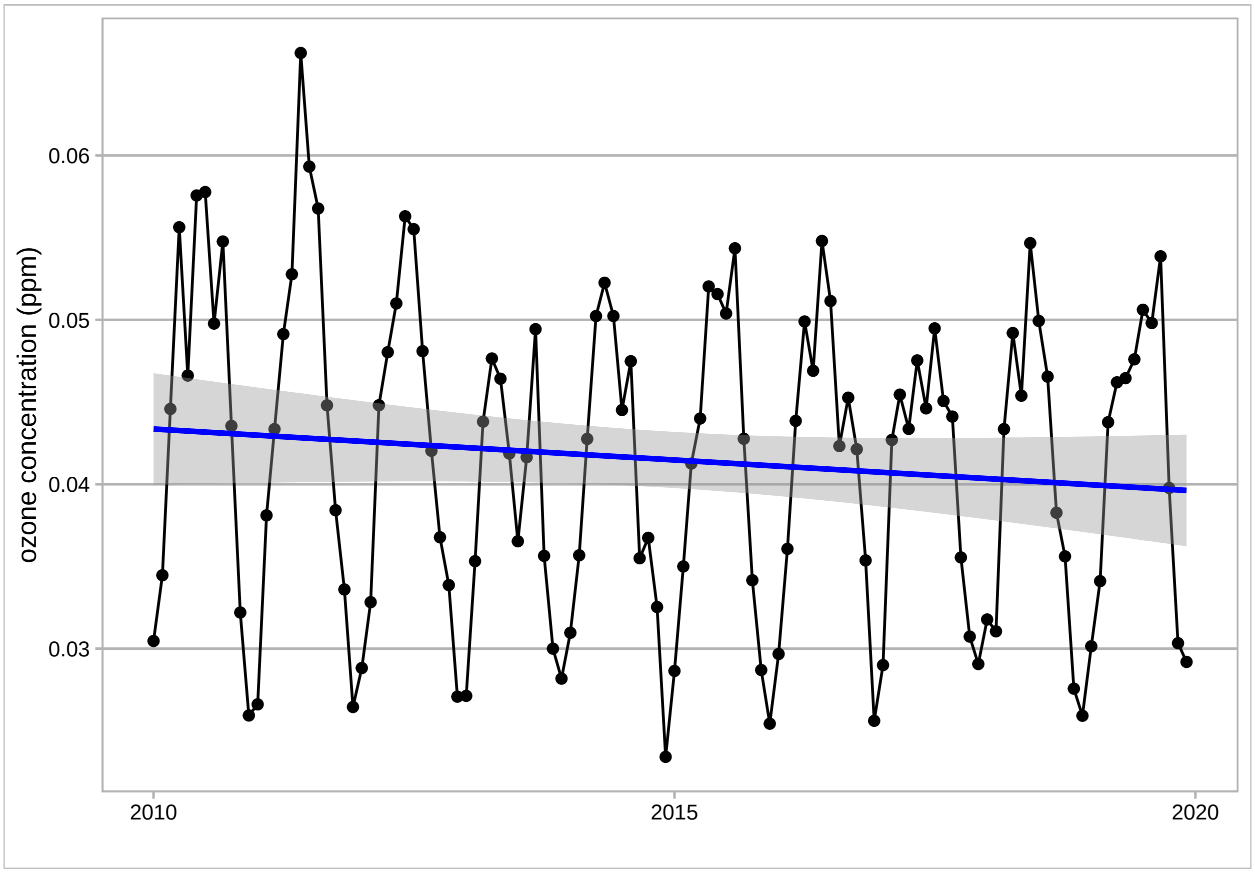
```r
# do not reject Ho (p-value>0.01), meaning that the data does not follow a trend
summary(Ozone_trend_season)
```

```
##
##  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 plot suggests that the ozone concentration data might have seasonality, so applying seasonal Mann-Kendall to determine if the data is stationary is appropriate. We can also remove the seasonality in the dataset, and apply other tests to analyze the underlying trend over time.

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
plot_Ozone<-ggplot(GaringerOzone.monthly,aes(x=Date,y=Ozone_clean))+
  geom_point()+
  geom_line()+
  geom_smooth(method=lm,color="blue")+
  labs(x=" ",y="ozone concentration (ppm)")
print(plot_Ozone)
```



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 ozone concentration data does not have a statistically significant trend over 2010 at the station (p-bvalue>0.01).

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

# remove seasonality
# extract the components of monthly time sereis and turn it into data frame
colnames(GaringerOzone.monthly.decompose$time.series)
```

```
## [1] "seasonal"  "trend"     "remainder"
```

```
Ozone_monthly_components<-as.data.frame(GaringerOzone.monthly.decompose$time.series[,2:3])

#add date and observed to components data frame
Ozone_monthly_components<-Ozone_monthly_components%>%
  mutate(Ozone_clean=GaringerOzone.monthly$Ozone_clean,
         Date=GaringerOzone.monthly$Date)

#16
```

```
# create another monthly time series
# specific the first month and year
f_month_monthly_wo<-month(first(Ozone_monthly_components$Date))
f_year_monthly_wo<-year(first(Ozone_monthly_components$Date))
# create time series objects
Ozone_monthly_components.ts<-ts(Ozone_monthly_components$Ozone_clean,start=c(f_year_monthly,f_month_mont

# Mann-Kendall
Ozone_MannKendall_all<- Kendall::MannKendall(Ozone_monthly_components.ts)
summary(Ozone_MannKendall_all)
```

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

```
#reject Ho (p-value>0.1)
```

Answer: The ozone concentration data does not have a statistically significant trend over 2010 at the station (p-bvalue>0.01), consistent with the results of the Seasonal Mann-Kendall test.