Assignment 8: Time Series Analysis

Joanna Huertas

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
getwd()
```

```
## [1] "C:/Users/joann/Documents/EDA-Spring2023"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.2.3
```

```r
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.2.3
```

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

```r
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.2.3
```

```r
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(dplyr)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.2.3
```

```r
# Set theme
mytheme <- theme_classic(base_size = 14) +
  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
#Import Ozone_TimeSeries
folder_contents <- list.files("./Data/Raw/Ozone_TimeSeries")
folder_contents
```

```
##  [1] "EPAair_O3_GaringerNC2010_raw.csv" "EPAair_O3_GaringerNC2011_raw.csv"
##  [3] "EPAair_O3_GaringerNC2012_raw.csv" "EPAair_O3_GaringerNC2013_raw.csv"
##  [5] "EPAair_O3_GaringerNC2014_raw.csv" "EPAair_O3_GaringerNC2015_raw.csv"
##  [7] "EPAair_O3_GaringerNC2016_raw.csv" "EPAair_O3_GaringerNC2017_raw.csv"
##  [9] "EPAair_O3_GaringerNC2018_raw.csv" "EPAair_O3_GaringerNC2019_raw.csv"
```

```
EPA2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                         stringsAsFactors = TRUE)
EPA2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                         stringsAsFactors = TRUE)
EPA2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                         stringsAsFactors = TRUE)
EPA2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                         stringsAsFactors = TRUE)
EPA2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                         stringsAsFactors = TRUE)
EPA2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                         stringsAsFactors = TRUE)
EPA2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                         stringsAsFactors = TRUE)
EPA2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                         stringsAsFactors = TRUE)
EPA2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                         stringsAsFactors = TRUE)
EPA2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                         stringsAsFactors = TRUE)


GaringerOzone <- rbind(EPA2010, EPA2011, EPA2012, EPA2013, EPA2014, EPA2015, EPA2016,
                       EPA2017, EPA2018, EPA2019)
```

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.

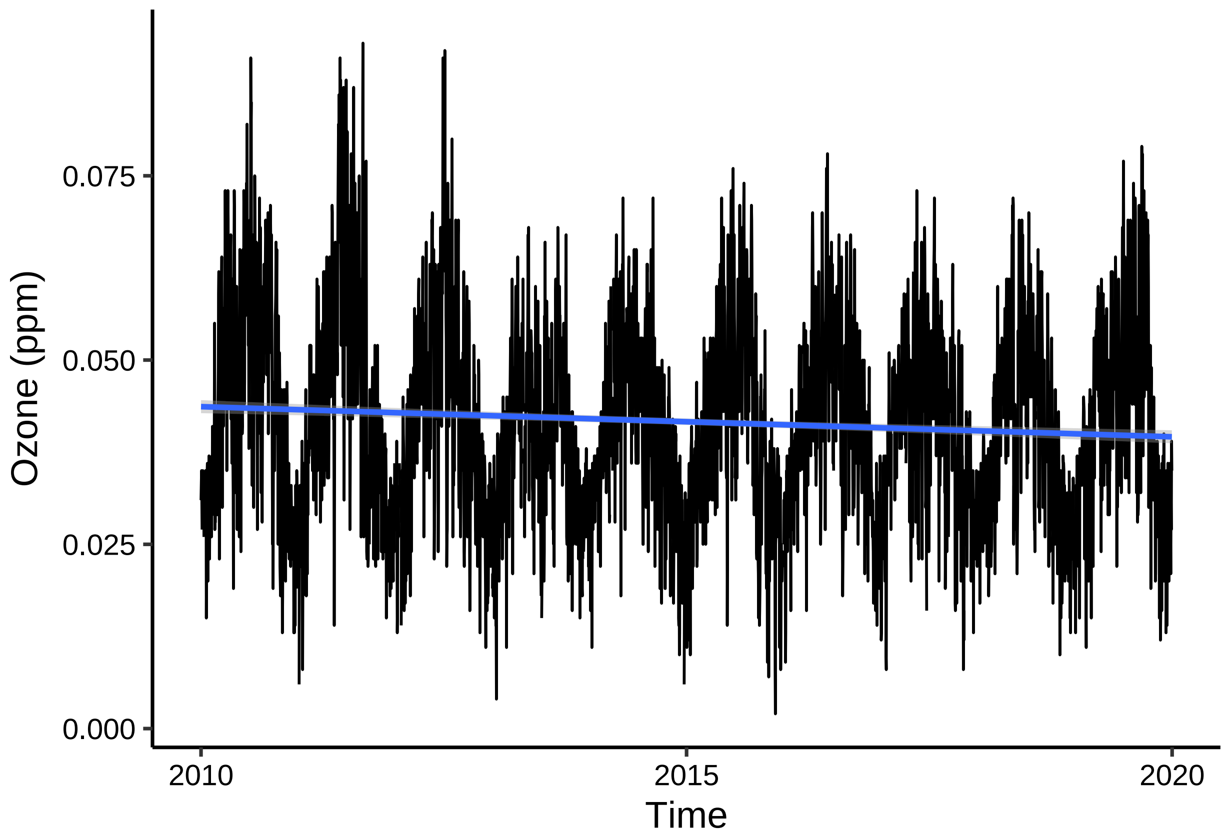
```
#3

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

#4

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

#5

#GaringerOzone.select.1 <- replace(GaringerOzone.select, is.na(GaringerOzone.select$Daily.Max.8.hour.Oz

Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day"))
colnames(Days)[1] <- "Date"

#6

GaringerOzone <- left_join(Days, GaringerOzone.select, 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

ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration
                          )) +
  geom_line() +
  labs(x = "Time", y = expression("Ozone (ppm)"))+
  geom_smooth(method = 'lm')
```

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

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



Answer: Yes, the trend line is slightly downward 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
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

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

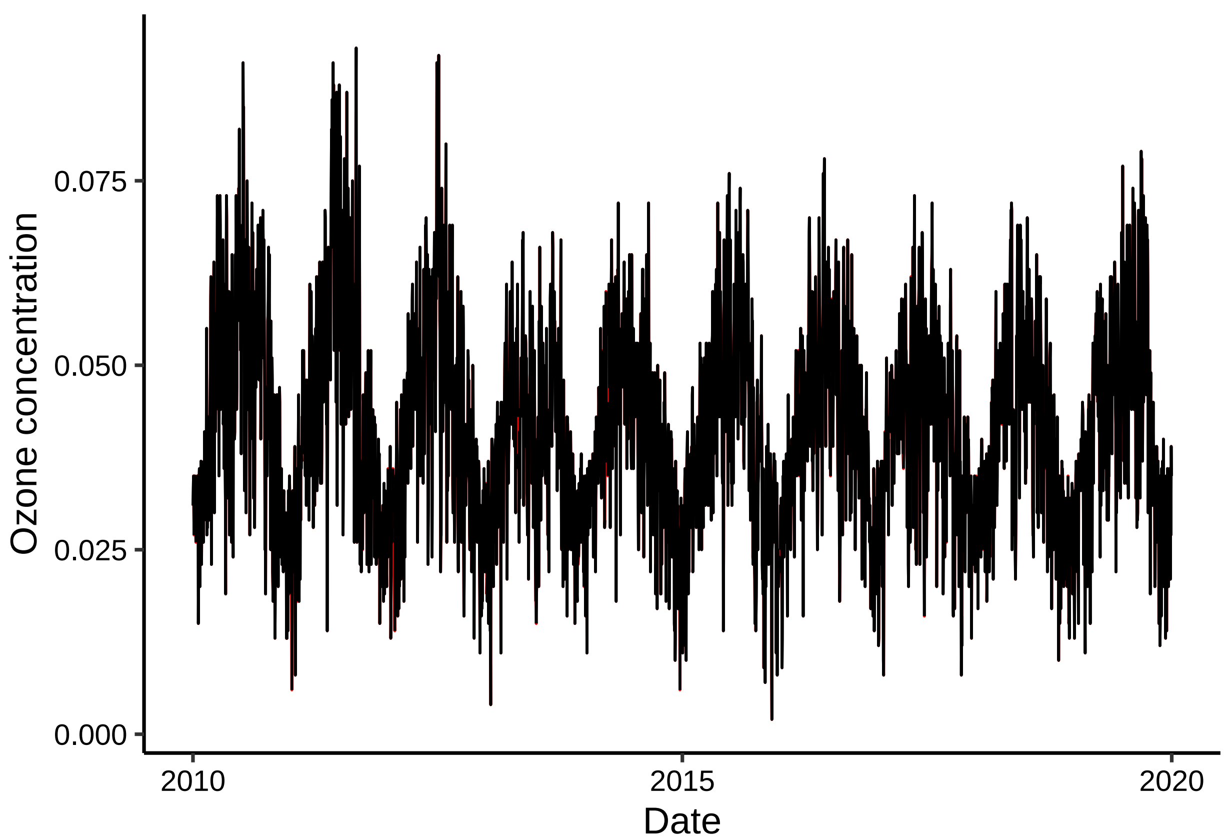
```
summary(GaringerOzone$DAILY_AQI_VALUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2.00   30.00   38.00   41.57   47.00  169.00      63
```

```r
#We have 63 NA's

# replace NAs
GaringerOzone.clean <-
  GaringerOzone %>%
  mutate(Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(GaringerOzone.clean$Ozone.Concentration.clean)
```

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

```r
#NA is gone

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



Answer: yes, linear interpolation is a suitable method for estimating missing daily data for ozone concentration because it assumes a linear relationship between data points, is simple to implement, and is appropriate for continuous and smooth data. Piecewise constant interpolation is not appropriate for this type of data because ozone concentration is unlikely to change abruptly. Spline interpolation may be unnecessary for relatively simple data sets and requires more computational power, it uses cuadratic function to interpolate.

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 %>%
  mutate(year = year(Date),
         month = month(Date, label = TRUE)) %>%
  group_by(year, month) %>%
  summarize(mean_ozone = mean(Ozone.Concentration.clean, na.rm = TRUE))
```

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

```
GaringerOzone.monthly$Date <- as.Date(paste(GaringerOzone.monthly$year,
                                            as.numeric(GaringerOzone.monthly$month),
                                            "01",
                                            sep = "-"))
GaringerOzone.monthly
```

```
## # A tibble: 120 x 4
## # Groups:   year [10]
##     year month mean_ozone Date
##    <dbl> <ord>      <dbl> <date>
##  1  2010 Jan       0.0305 2010-01-01
##  2  2010 Feb       0.0345 2010-02-01
##  3  2010 Mar       0.0446 2010-03-01
##  4  2010 Apr       0.0556 2010-04-01
##  5  2010 May       0.0466 2010-05-01
##  6  2010 Jun       0.0576 2010-06-01
##  7  2010 Jul       0.0578 2010-07-01
##  8  2010 Aug       0.0498 2010-08-01
##  9  2010 Sep       0.0548 2010-09-01
## 10  2010 Oct       0.0435 2010-10-01
## # ... with 110 more rows
```

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_month <- month(first(GaringerOzone.clean$Date))
f_year <- year(first(GaringerOzone.clean$Date))
f_month
```

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

```r
f_year
```

```
## [1] 2010
```

```r
GaringerOzone.daily.ts <- ts(GaringerOzone.clean$Ozone.Concentration.clean,
                   start=c(2010,1),
                   frequency=365)

summary(GaringerOzone.daily.ts)
```

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

```r
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                   start=c(2010,1),
                   frequency=12)
summary(GaringerOzone.monthly.ts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02342 0.03380 0.04335 0.04149 0.04915 0.06623
```

```r
head(GaringerOzone.monthly.ts)
```

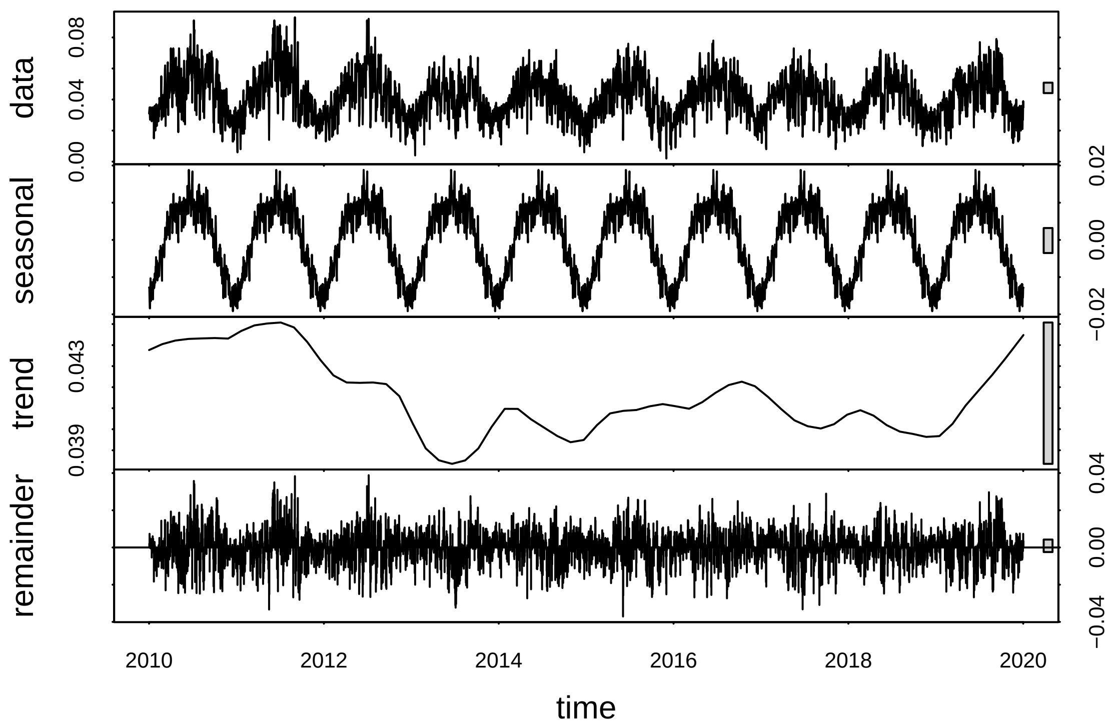
```
##             Jan        Feb        Mar        Apr        May        Jun
## 2010 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
```

```r
GaringerOzone.monthly.ts
```

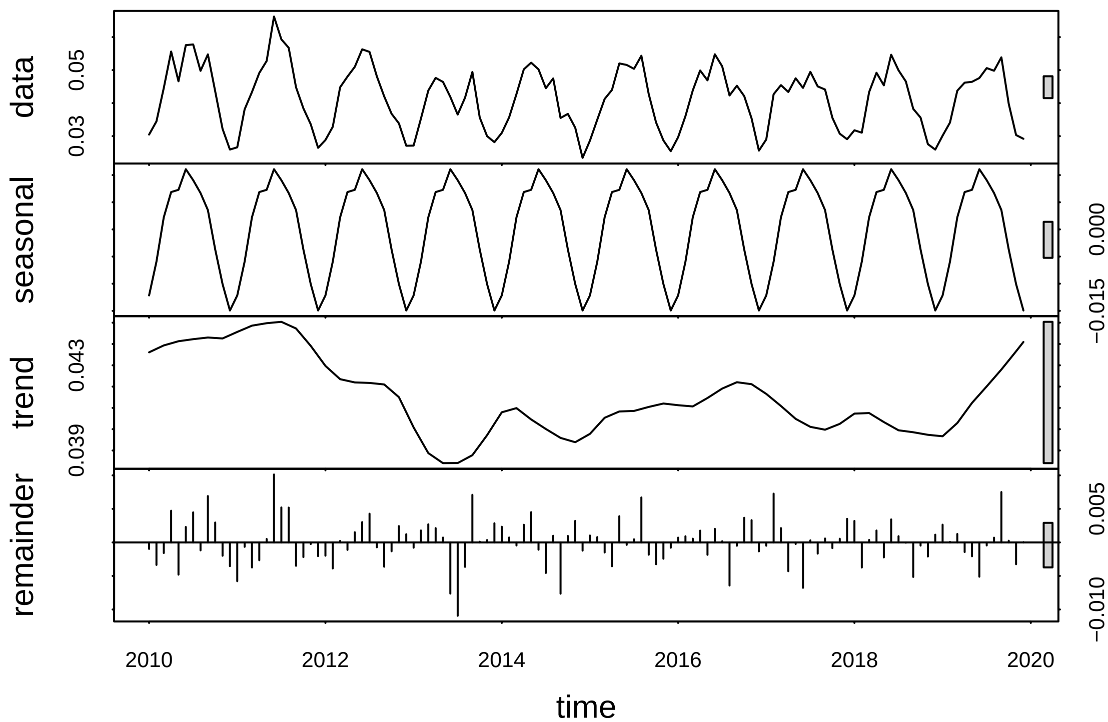
```
##             Jan        Feb        Mar        Apr        May        Jun
## 2010 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
## 2011 0.02661290 0.03810714 0.04335484 0.04913333 0.05277419 0.06623333
## 2012 0.02882258 0.03282759 0.04480645 0.04803333 0.05100000 0.05630000
## 2013 0.02712903 0.03532143 0.04380645 0.04765000 0.04641935 0.04186667
## 2014 0.03096774 0.03567857 0.04275806 0.05023333 0.05225806 0.05023333
## 2015 0.02864516 0.03500000 0.04125806 0.04400000 0.05203226 0.05156667
## 2016 0.02967742 0.03606897 0.04385484 0.04990000 0.04690323 0.05480000
## 2017 0.02900000 0.04269643 0.04545161 0.04336667 0.04753226 0.04461667
## 2018 0.03177419 0.03105357 0.04335484 0.04920000 0.04538710 0.05466667
## 2019 0.03014516 0.03410714 0.04377419 0.04620000 0.04645161 0.04760000
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2010 0.05777419 0.04977419 0.05476667 0.04354839 0.03220000 0.02593548
## 2011 0.05932258 0.05677419 0.04480000 0.03841935 0.03360000 0.02645161
## 2012 0.05551613 0.04809677 0.04203333 0.03677419 0.03386667 0.02708065
## 2013 0.03653226 0.04164516 0.04943333 0.03564516 0.03000000 0.02817742
## 2014 0.04451613 0.04748387 0.03550000 0.03674194 0.03253333 0.02341935
## 2015 0.05038710 0.05435484 0.04276667 0.03416129 0.02870000 0.02543548
## 2016 0.05114516 0.04232258 0.04526667 0.04212903 0.03536667 0.02561290
## 2017 0.04948387 0.04506452 0.04411667 0.03554839 0.03073333 0.02906452
## 2018 0.04993548 0.04654839 0.03826667 0.03561290 0.02756667 0.02591935
## 2019 0.05061290 0.04980645 0.05386667 0.03977419 0.03033333 0.02919355
```

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

```
#11
#decompose
GaringerOzone.daily_decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic") #frequency 12
plot(GaringerOzone.daily_decomp)
```




```
GaringerOzone.monthly_decomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic") #frequency 12, peri
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

# Run SMK test
GaringerOzone.monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
GaringerOzone.monthly_trend
```

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

```
summary(GaringerOzone.monthly_trend)
```

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

```
GaringerOzone.monthly_trend1 <- trend::smk.test(GaringerOzone.monthly.ts)
# Inspect results
GaringerOzone.monthly_trend1
```

```
##
##  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
```

```
summary(GaringerOzone.monthly_trend1)
```

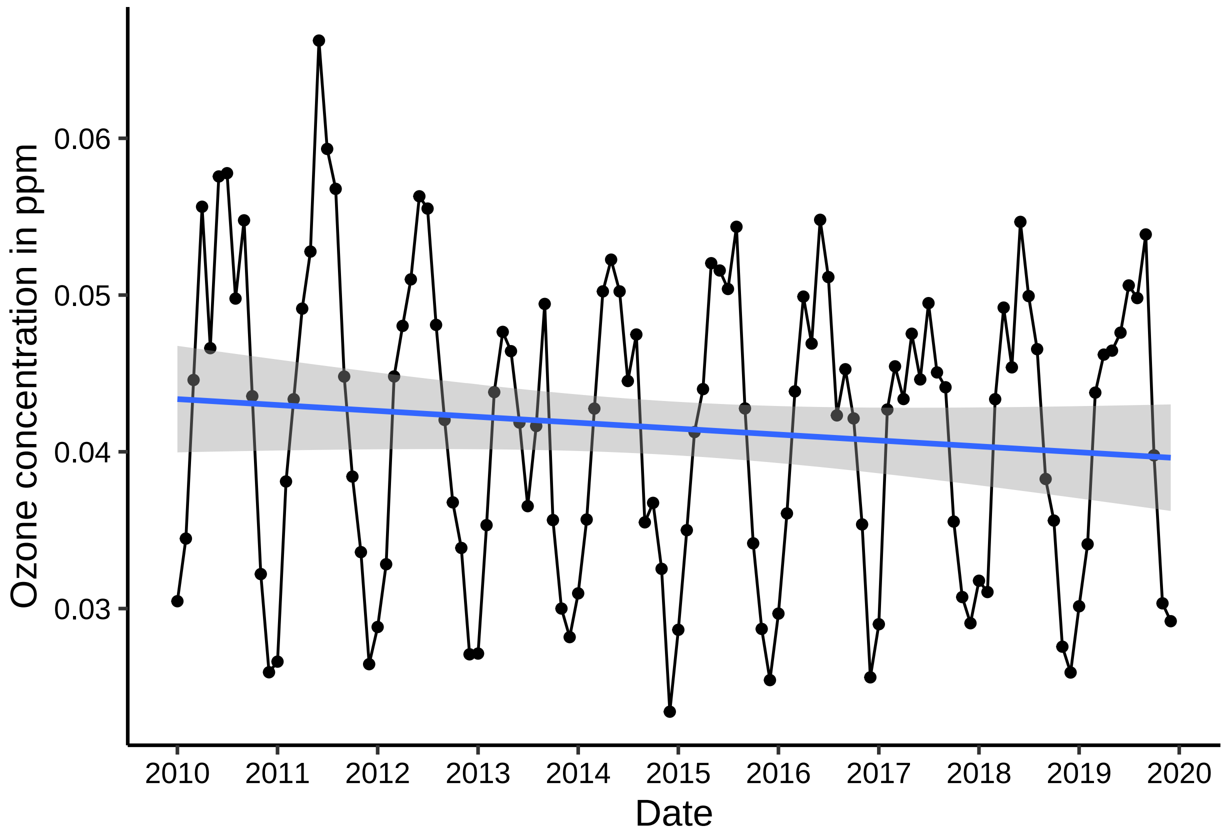
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                      S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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: Because Mann-Kendall is appropiate for seasonal data, and the monthly Ozone series suggests 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

#Visualization
GaringerOzone.monthly_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
  geom_point() +
  geom_line() +
  scale_x_date(date_labels = "%Y", date_breaks  ="1 year")+
  ylab("Ozone concentration in ppm") +
  geom_smooth( method = lm )
print(GaringerOzone.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: in the first method of SeasonalMannKendall, we notice the p-value is less than 0.05, so we are going to reject the null hypothesis that the data is stationary. In the second method, we notice S is high in most cases, which indicates stronger tendency of decrease, and in other cases the trend is stationary but overall the trend is decrease. The plot shows a negative slope.

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

# We can extract the components and turn them into data frames
GaringerOzone.monthly.ts_Components <- as.data.frame(GaringerOzone.monthly_decomp$time.series[,1:3])

GaringerOzone.monthly.ts_Components <- mutate(GaringerOzone.monthly.ts_Components,
        Observed = GaringerOzone.monthly$mean_ozone,
        Date = GaringerOzone.monthly$Date)
```

```r
GaringerOzone.monthly.ts_Components
```

```
##        seasonal      trend     remainder   Observed       Date
## 1   -0.012164159 0.04360892 -9.770197e-04 0.03046774 2010-01-01
## 2   -0.005945745 0.04377124 -3.361210e-03 0.03446429 2010-02-01
## 3    0.002231834 0.04393356 -1.584752e-03 0.04458065 2010-03-01
## 4    0.006878411 0.04403138  4.723545e-03 0.05563333 2010-04-01
## 5    0.007292088 0.04412919 -4.808378e-03 0.04661290 2010-05-01
## 6    0.011093186 0.04417744  2.296036e-03 0.05756667 2010-06-01
## 7    0.009063964 0.04422570  4.484533e-03 0.05777419 2010-07-01
## 8    0.006696219 0.04426488 -1.186904e-03 0.04977419 2010-08-01
## 9    0.003558522 0.04430406  6.904084e-03 0.05476667 2010-09-01
## 10  -0.003703722 0.04428190  2.970213e-03 0.04354839 2010-10-01
## 11  -0.010065266 0.04425973 -1.994463e-03 0.03220000 2010-11-01
## 12  -0.014935333 0.04441318 -3.542364e-03 0.02593548 2010-12-01
## 13  -0.012164159 0.04456663 -5.789569e-03 0.02661290 2011-01-01
## 14  -0.005945745 0.04471362 -6.607351e-04 0.03810714 2011-02-01
## 15   0.002231834 0.04486062 -3.737611e-03 0.04335484 2011-03-01
## 16   0.006878411 0.04491823 -2.663307e-03 0.04913333 2011-04-01
## 17   0.007292088 0.04497584  5.062612e-04 0.05277419 2011-05-01
## 18   0.011093186 0.04500876  1.013138e-02 0.06623333 2011-06-01
## 19   0.009063964 0.04504168  5.216935e-03 0.05932258 2011-07-01
## 20   0.006696219 0.04488493  5.193046e-03 0.05677419 2011-08-01
## 21   0.003558522 0.04472818 -3.486698e-03 0.04480000 2011-09-01
## 22  -0.003703722 0.04431994 -2.196859e-03 0.03841935 2011-10-01
## 23  -0.010065266 0.04391169 -2.464284e-04 0.03360000 2011-11-01
## 24  -0.014935333 0.04344140 -2.054455e-03 0.02645161 2011-12-01
## 25  -0.012164159 0.04297111 -1.984368e-03 0.02882258 2012-01-01
## 26  -0.005945745 0.04265981 -3.886474e-03 0.03282759 2012-02-01
## 27   0.002231834 0.04234850  2.261130e-04 0.04480645 2012-03-01
## 28   0.006878411 0.04227170 -1.116781e-03 0.04803333 2012-04-01
## 29   0.007292088 0.04219490  1.513008e-03 0.05100000 2012-05-01
## 30   0.011093186 0.04218339  3.023424e-03 0.05630000 2012-06-01
## 31   0.009063964 0.04217188  4.280289e-03 0.05551613 2012-07-01
## 32   0.006696219 0.04213627 -7.357190e-04 0.04809677 2012-08-01
## 33   0.003558522 0.04210067 -3.625862e-03 0.04203333 2012-09-01
## 34  -0.003703722 0.04180209 -1.324173e-03 0.03677419 2012-10-01
## 35  -0.010065266 0.04150350  2.428429e-03 0.03386667 2012-11-01
## 36  -0.014935333 0.04079457  1.221408e-03 0.02708065 2012-12-01
## 37  -0.012164159 0.04008564 -7.924444e-04 0.02712903 2013-01-01
## 38  -0.005945745 0.03948151  1.785660e-03 0.03532143 2013-02-01
## 39   0.002231834 0.03887739  2.697225e-03 0.04380645 2013-03-01
## 40   0.006878411 0.03864154  2.130054e-03 0.04765000 2013-04-01
## 41   0.007292088 0.03840568  7.215879e-04 0.04641935 2013-05-01
## 42   0.011093186 0.03840759 -7.634105e-03 0.04186667 2013-06-01
## 43   0.009063964 0.03840949 -1.094120e-02 0.03653226 2013-07-01
## 44   0.006696219 0.03859429 -3.645346e-03 0.04164516 2013-08-01
## 45   0.003558522 0.03877908  7.095727e-03 0.04943333 2013-09-01
## 46  -0.003703722 0.03924804  1.008392e-04 0.03564516 2013-10-01
## 47  -0.010065266 0.03971700  3.482635e-04 0.03000000 2013-11-01
## 48  -0.014935333 0.04025520  2.857553e-03 0.02817742 2013-12-01
## 49  -0.012164159 0.04079340  2.338505e-03 0.03096774 2014-01-01
## 50  -0.005945745 0.04089116  7.331607e-04 0.03567857 2014-02-01
```

```
## 51   0.002231834 0.04098892 -4.626858e-04 0.04275806 2014-03-01
## 52   0.006878411 0.04072307  2.631857e-03 0.05023333 2014-04-01
## 53   0.007292088 0.04045722  4.508761e-03 0.05225806 2014-05-01
## 54   0.011093186 0.04023313 -1.092979e-03 0.05023333 2014-06-01
## 55   0.009063964 0.04000904 -4.556871e-03 0.04451613 2014-07-01
## 56   0.006696219 0.03980012  9.875369e-04 0.04748387 2014-08-01
## 57   0.003558522 0.03959119 -7.649716e-03 0.03550000 2014-09-01
## 58  -0.003703722 0.03948933  9.563285e-04 0.03674194 2014-10-01
## 59  -0.010065266 0.03938746  3.211137e-03 0.03253333 2014-11-01
## 60  -0.014935333 0.03958361 -1.228925e-03 0.02341935 2014-12-01
## 61  -0.012164159 0.03977976  1.029557e-03 0.02864516 2015-01-01
## 62  -0.005945745 0.04015746  7.882878e-04 0.03500000 2015-02-01
## 63   0.002231834 0.04053515 -1.508922e-03 0.04125806 2015-03-01
## 64   0.006878411 0.04068343 -3.561840e-03 0.04400000 2015-04-01
## 65   0.007292088 0.04083171  3.908462e-03 0.05203226 2015-05-01
## 66   0.011093186 0.04084443 -3.709540e-04 0.05156667 2015-06-01
## 67   0.009063964 0.04085716  4.659715e-04 0.05038710 2015-07-01
## 68   0.006696219 0.04095089  6.707727e-03 0.05435484 2015-08-01
## 69   0.003558522 0.04104462 -1.836479e-03 0.04276667 2015-09-01
## 70  -0.003703722 0.04112503 -3.260018e-03 0.03416129 2015-10-01
## 71  -0.010065266 0.04120544 -2.440169e-03 0.02870000 2015-11-01
## 72  -0.014935333 0.04116586 -7.950401e-04 0.02543548 2015-12-01
## 73  -0.012164159 0.04112628  7.153002e-04 0.02967742 2016-01-01
## 74  -0.005945745 0.04109729  9.174244e-04 0.03606897 2016-02-01
## 75   0.002231834 0.04106829  5.547094e-04 0.04385484 2016-03-01
## 76   0.006878411 0.04126622  1.755371e-03 0.04990000 2016-04-01
## 77   0.007292088 0.04146414 -1.853004e-03 0.04690323 2016-05-01
## 78   0.011093186 0.04168507  2.021745e-03 0.05480000 2016-06-01
## 79   0.009063964 0.04190600  1.752015e-04 0.05114516 2016-07-01
## 80   0.006696219 0.04205585 -6.429485e-03 0.04232258 2016-08-01
## 81   0.003558522 0.04220570 -4.975523e-04 0.04526667 2016-09-01
## 82  -0.003703722 0.04215909  3.673669e-03 0.04212903 2016-10-01
## 83  -0.010065266 0.04211247  3.319460e-03 0.03536667 2016-11-01
## 84  -0.014935333 0.04188444 -1.336205e-03 0.02561290 2016-12-01
## 85  -0.012164159 0.04165641 -4.922514e-04 0.02900000 2017-01-01
## 86  -0.005945745 0.04137218  7.269994e-03 0.04269643 2017-02-01
## 87   0.002231834 0.04108795  2.131829e-03 0.04545161 2017-03-01
## 88   0.006878411 0.04078631 -4.298051e-03 0.04336667 2017-04-01
## 89   0.007292088 0.04048466 -2.444947e-04 0.04753226 2017-05-01
## 90   0.011093186 0.04029711 -6.773633e-03 0.04461667 2017-06-01
## 91   0.009063964 0.04010956  3.103435e-04 0.04948387 2017-07-01
## 92   0.006696219 0.04004253 -1.674231e-03 0.04506452 2017-08-01
## 93   0.003558522 0.03997549  5.826512e-04 0.04411667 2017-09-01
## 94  -0.003703722 0.04011236 -8.602550e-04 0.03554839 2017-10-01
## 95  -0.010065266 0.04024923  5.493661e-04 0.03073333 2017-11-01
## 96  -0.014935333 0.04049011  3.509737e-03 0.02906452 2017-12-01
## 97  -0.012164159 0.04073099  3.207362e-03 0.03177419 2018-01-01
## 98  -0.005945745 0.04074429 -3.744973e-03 0.03105357 2018-02-01
## 99   0.002231834 0.04075759  3.654164e-04 0.04335484 2018-03-01
## 100  0.006878411 0.04054622  1.775371e-03 0.04920000 2018-04-01
## 101  0.007292088 0.04033485 -2.239840e-03 0.04538710 2018-05-01
## 102  0.011093186 0.04014142  3.432062e-03 0.05466667 2018-06-01
## 103  0.009063964 0.03994799  9.235311e-04 0.04993548 2018-07-01
## 104  0.006696219 0.03990239 -5.021935e-05 0.04654839 2018-08-01
```

```
## 105  0.003558522 0.03985679 -5.148642e-03 0.03826667 2018-09-01
## 106 -0.003703722 0.03979642 -4.797964e-04 0.03561290 2018-10-01
## 107 -0.010065266 0.03973606 -2.104123e-03 0.02756667 2018-11-01
## 108 -0.014935333 0.03970143  1.153255e-03 0.02591935 2018-12-01
## 109 -0.012164159 0.03966681  2.642510e-03 0.03014516 2019-01-01
## 110 -0.005945745 0.03997833  7.455584e-05 0.03410714 2019-02-01
## 111  0.002231834 0.04028985  1.252505e-03 0.04377419 2019-03-01
## 112  0.006878411 0.04076189 -1.440300e-03 0.04620000 2019-04-01
## 113  0.007292088 0.04123392 -2.074399e-03 0.04645161 2019-05-01
## 114  0.011093186 0.04162100 -5.114182e-03 0.04760000 2019-06-01
## 115  0.009063964 0.04200807 -4.591295e-04 0.05061290 2019-07-01
## 116  0.006696219 0.04240444  7.057916e-04 0.04980645 2019-08-01
## 117  0.003558522 0.04280081  7.507331e-03 0.05386667 2019-09-01
## 118 -0.003703722 0.04322863  2.492802e-04 0.03977419 2019-10-01
## 119 -0.010065266 0.04365646 -3.257856e-03 0.03033333 2019-11-01
## 120 -0.014935333 0.04409543  3.345012e-05 0.02919355 2019-12-01
```

```r
# Subtract the seasonal component from the original time series
GaringerOzone.monthly_no_seasonality <- GaringerOzone.monthly.ts -
  GaringerOzone.monthly_decomp$time.series[, "seasonal"]

#16
GaringerOzone.monthly_trend1 <- Kendall::MannKendall(GaringerOzone.monthly_no_seasonality)
# Inspect results
GaringerOzone.monthly_trend1
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```r
summary(GaringerOzone.monthly_trend1)
```

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

> Answer: In this case, the p-value obtained from the Seasonal Mann Kendall test on the complete ozone monthly series is 0.046, which suggests that there is a significant trend in the series, but this trend may be due to seasonality. On the other hand, the p-value obtained from the non-seasonal ozone monthly series is 0.0075402, which suggests that there is a significant trend in the series even after the seasonal component has been removed. Since 0.0075402 is less than 0.046, the first one is way more significant. Comparing both p-values, the trend detected by the Seasonal Mann Kendall test on the complete series may be partly explained by seasonality.