Assignment 7: Time Series Analysis

Benjamin Culberson

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()
```

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- 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(zoo)
```

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

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

```r
library(trend)

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

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.

```r
#2
EPAair2010 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
           stringsAsFactors = TRUE)

EPAair2011 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
           stringsAsFactors = TRUE)

EPAair2012 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
           stringsAsFactors = TRUE)

EPAair2013 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
           stringsAsFactors = TRUE)

EPAair2014 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
```

```r
          stringsAsFactors = TRUE)

EPAair2015 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
           stringsAsFactors = TRUE)

EPAair2016 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
           stringsAsFactors = TRUE)

EPAair2017 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
           stringsAsFactors = TRUE)

EPAair2018 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
           stringsAsFactors = TRUE)

EPAair2019 <-
  read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
           stringsAsFactors = TRUE)

GaringerOzone <- rbind(EPAair2010, EPAair2011, EPAair2012, EPAair2013, EPAair2014,
                       EPAair2015, EPAair2016, EPAair2017, EPAair2018, EPAair2019)

summary(GaringerOzone)
```

```
##          Date          Source          Site.ID                POC
##  01/01/2010:   1   AQS    :3588   Min.   :371190041   Min.   :1
##  01/02/2010:   1   AirNow:   1   1st Qu.:371190041   1st Qu.:1
##  01/03/2010:   1                  Median :371190041   Median :1
##  01/04/2010:   1                  Mean   :371190041   Mean   :1
##  01/05/2010:   1                  3rd Qu.:371190041   3rd Qu.:1
##  01/07/2010:   1                  Max.   :371190041   Max.   :1
##  (Other)   :3583
##  Daily.Max.8.hour.Ozone.Concentration UNITS       DAILY_AQI_VALUE
##  Min.   :0.00200                      ppm:3589   Min.   :  2.00
##  1st Qu.:0.03200                                 1st Qu.: 30.00
##  Median :0.04100                                 Median : 38.00
##  Mean   :0.04163                                 Mean   : 41.57
##  3rd Qu.:0.05100                                 3rd Qu.: 47.00
##  Max.   :0.09300                                 Max.   :169.00
##
##                Site.Name    DAILY_OBS_COUNT PERCENT_COMPLETE
##  Garinger High School:3589   Min.   : 6.00   Min.   : 35.0
##                              1st Qu.:17.00   1st Qu.:100.0
##                              Median :17.00   Median :100.0
##                              Mean   :16.97   Mean   : 99.8
##                              3rd Qu.:17.00   3rd Qu.:100.0
##                              Max.   :19.00   Max.   :100.0
##
##  AQS_PARAMETER_CODE AQS_PARAMETER_DESC   CBSA_CODE
##  Min.   :44201      Ozone:3589          Min.   :16740
```

```
##  1st Qu.:44201                           1st Qu.:16740
##  Median :44201                           Median :16740
##  Mean   :44201                           Mean   :16740
##  3rd Qu.:44201                           3rd Qu.:16740
##  Max.   :44201                           Max.   :16740
##
##                              CBSA_NAME      STATE_CODE            STATE
##  Charlotte-Concord-Gastonia, NC-SC:3589   Min.   :37   North Carolina:3589
##                                           1st Qu.:37
##                                           Median :37
##                                           Mean   :37
##                                           3rd Qu.:37
##                                           Max.   :37
##
##   COUNTY_CODE          COUNTY     SITE_LATITUDE   SITE_LONGITUDE
##  Min.   :119   Mecklenburg:3589   Min.   :35.24   Min.   :-80.79
##  1st Qu.:119                      1st Qu.:35.24   1st Qu.:-80.79
##  Median :119                      Median :35.24   Median :-80.79
##  Mean   :119                      Mean   :35.24   Mean   :-80.79
##  3rd Qu.:119                      3rd Qu.:35.24   3rd Qu.:-80.79
##  Max.   :119                      Max.   :35.24   Max.   :-80.79
##
```

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

# 5
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by="days"))

Days$Date <- Days$`seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "days")`
Days <- Days %>%
  select(Date)
```

```
# 6
GaringerOzone <- left_join(Days, GaringerOzone_processed)
```

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

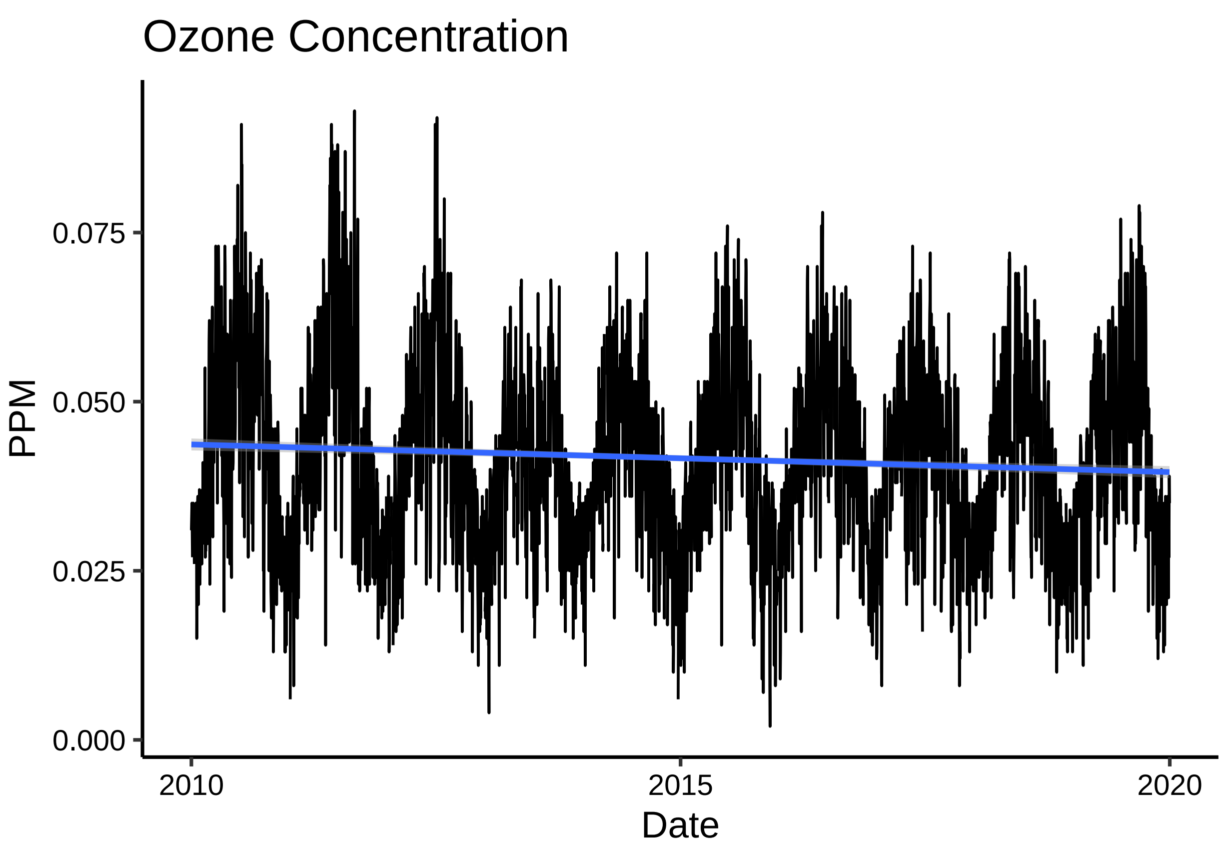
Visualize

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

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

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



Answer: It certainly appears that there is seasonality in the data. There appears to be a small trend, but it somewhat hard to tell given all the noise, but looking at the graph with the trend line, there does appear to be a slight downward progression.

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)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
```

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


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

summary(GaringerOzone)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration.clean
##  Min.   :2010-01-01   Min.   :0.00200
##  1st Qu.:2012-07-01   1st Qu.:0.03200
##  Median :2014-12-31   Median :0.04100
##  Mean   :2014-12-31   Mean   :0.04151
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100
##  Max.   :2019-12-31   Max.   :0.09300
##  DAILY_AQI_VALUE.clean
##  Min.   :  2.00
##  1st Qu.: 30.00
##  Median : 38.00
##  Mean   : 41.41
##  3rd Qu.: 47.00
##  Max.   :169.00
```

Answer: The reason we didn't use a piecewise constant or a spline interpolation is because they would most likey be inaccurate. The data is rather fluid so the piecwise constant will almost certainly not reflect the most accurate estimate of ppm or AQI for the NAs. Furthermore, the spline interolation would use a quadratic function to bridge the gap between two points. A cursory glance at the data does not lend itself to the belief that the data behaves in a quadratic manner.

Simply, if one were to take 3 random sequential points, a linear interpolation consistently looks like the best way to estimate the middle value. The spline interpolation isn't necessarily wrong in all cases, but it appears that it would be mostly inaccurate.

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 %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  group_by(Year, Month) %>%
  summarise(Ozone_mean = mean(Daily.Max.8.hour.Ozone.Concentration.clean))
```

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

```
GaringerOzone.monthly
```

```
## # A tibble: 120 x 3
## # Groups:   Year [10]
##     Year Month Ozone_mean
##    <dbl> <dbl>      <dbl>
##  1  2010     1     0.0305
##  2  2010     2     0.0345
##  3  2010     3     0.0446
##  4  2010     4     0.0556
##  5  2010     5     0.0466
##  6  2010     6     0.0576
##  7  2010     7     0.0578
##  8  2010     8     0.0498
##  9  2010     9     0.0548
## 10  2010    10     0.0435
## # ... with 110 more rows
```

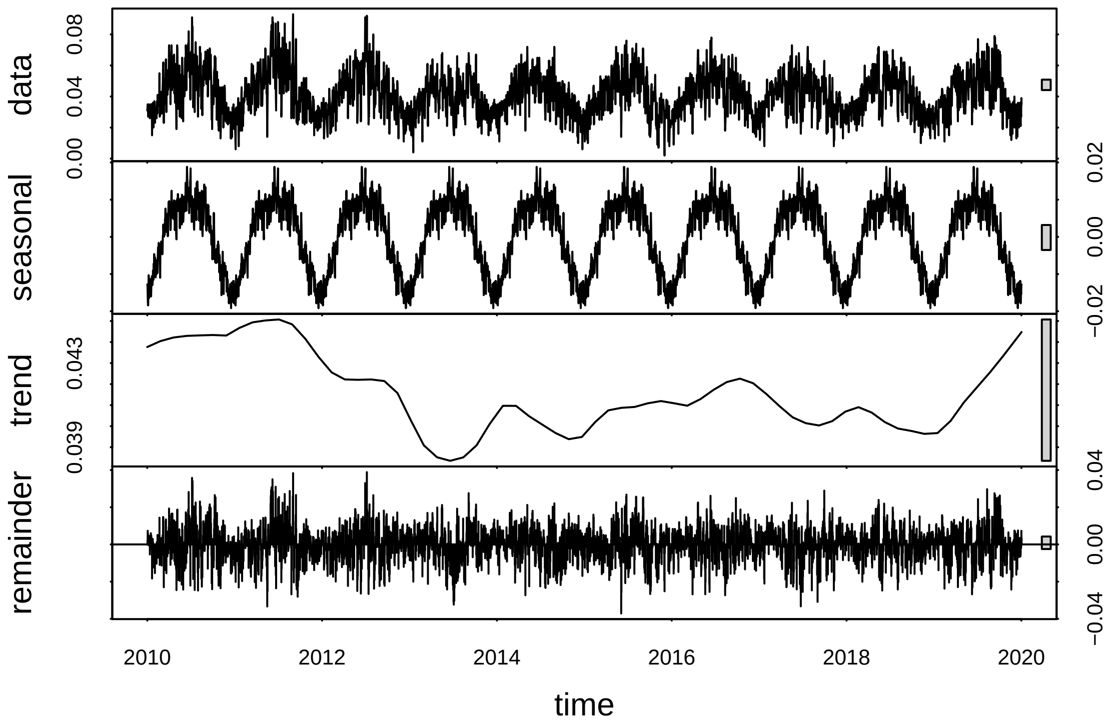
```
GaringerOzone.monthly <-
  GaringerOzone.monthly

Months <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-1"), by = "months"))
Months$Date <- Months$`seq(as.Date("2010-01-01"), as.Date("2019-12-1"), by = "months")`
Months <- Months %>%
  select(Date)

GaringerOzone.monthly <- cbind(GaringerOzone.monthly, Months)
```

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.

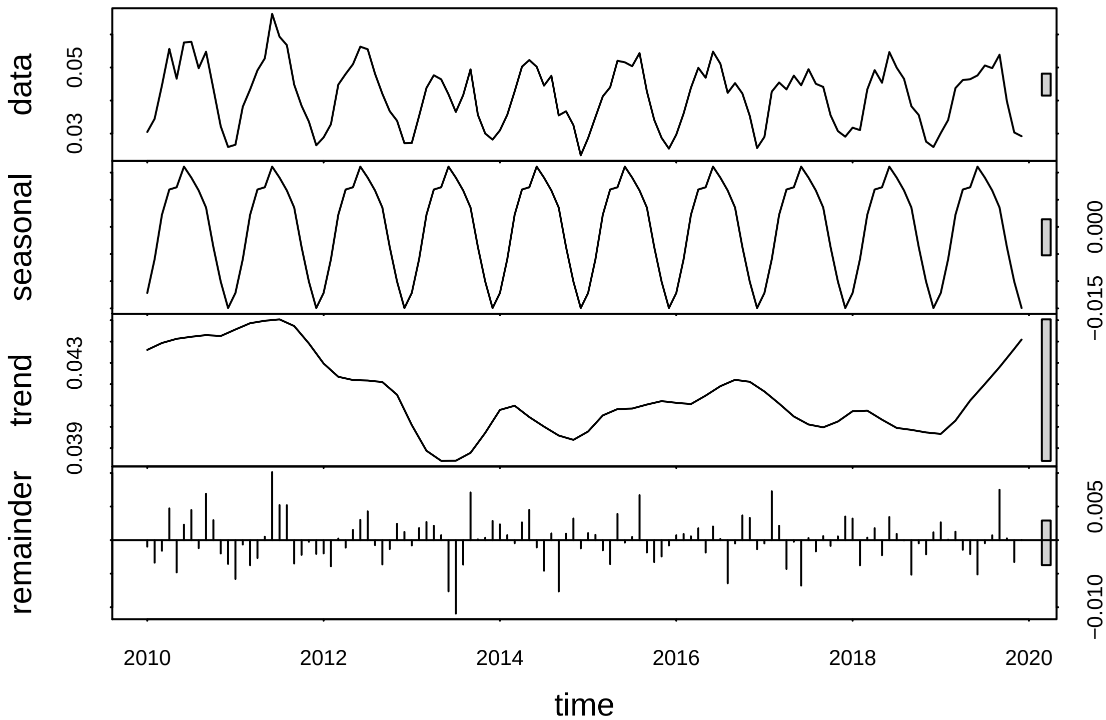
```r
#10
f_day_ozone <- day(first(GaringerOzone$Date))
f_month_ozone <- month(first(GaringerOzone.monthly$Date))
f_year_ozone <- year(first(GaringerOzone.monthly$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration.clean,
                   start=c(f_year_ozone,f_month_ozone,f_day_ozone),
                   frequency=365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Ozone_mean,
                               start = c(f_year_ozone,f_month_ozone), frequency = 12)
```

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

```r
#11
daily_Ozone_decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(daily_Ozone_decomp)
```




```r
monthly_Ozone_decomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
plot(monthly_Ozone_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?

```r
#12
monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

monthly_trend
```

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

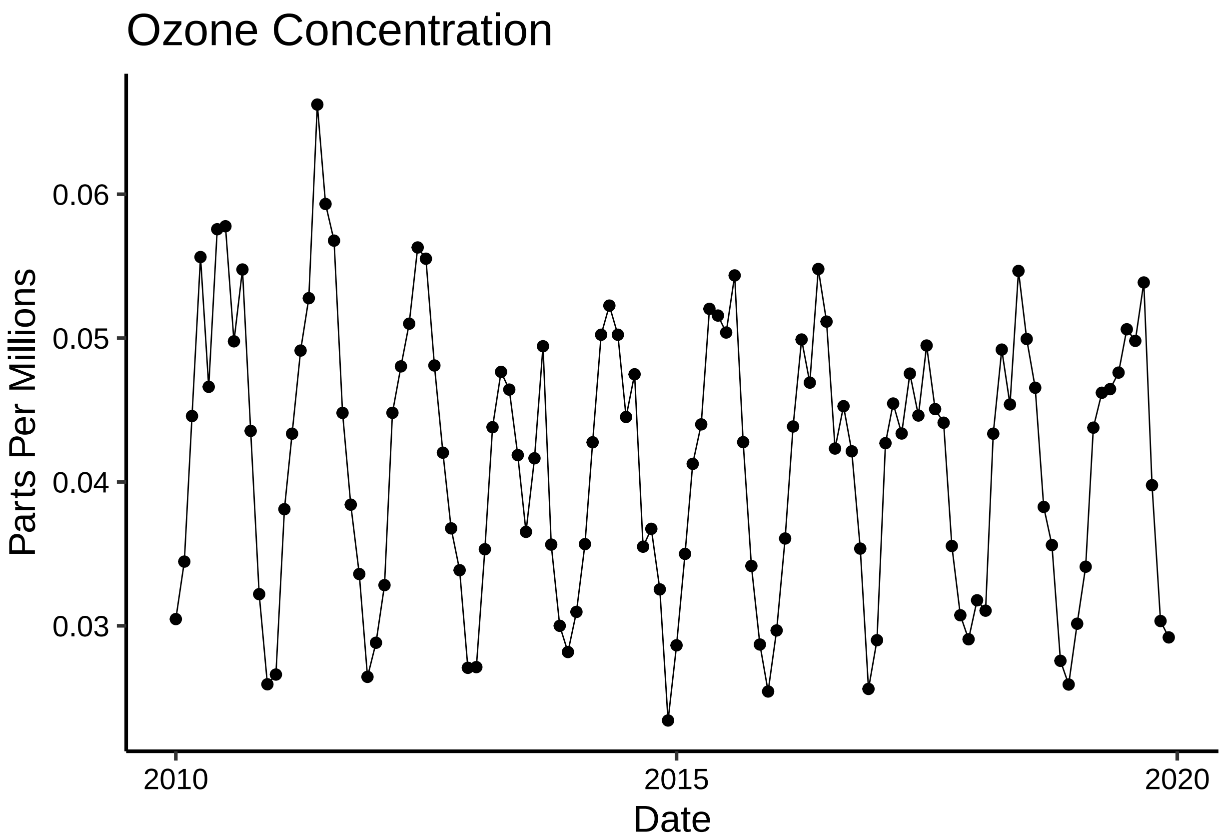
```r
summary(monthly_trend)
```

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

> Answer: THe seasonal Mann-Kendall stationarity test is most appropriate in this case because the decomposition of the monthly time series Ozone object clearly shows a seasonal component. If we ran a non-seasonal Mann-Kendall test, we may find there to be a trend with higher statistical significance than there really is, or no trend at all when there is one. After running the seasonal Mann-kendall, it does appear that the test has found there to be a statistically significant trend.

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
ggplot(GaringerOzone.monthly) +
  geom_line(aes(y = Ozone_mean, x = Date),  size = 0.25) +
  geom_point(aes(y= Ozone_mean, x = Date)) +
  labs(y = "Parts Per Millions", x = "Date", title = "Ozone Concentration" )
```



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: We are interested in determining whether or not there is a trend in our Ozone Concentration Data over time. From this graph here, it certainly appears that there is significant seasonality, but whether or not a trend exists is unclear (it's hard to judge). However, after running a seasonal Mann-Kendall test we reject the null hypothesis of no statistically significant trend at the 0.05 level (p -value = 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
ozone_components <- as.data.frame(monthly_Ozone_decomp$time.series[,1:3])
ozone_components
```

```
##         seasonal      trend     remainder
## 1   -0.012164159 0.04360892 -9.770197e-04
## 2   -0.005945745 0.04377124 -3.361210e-03
## 3    0.002231834 0.04393356 -1.584752e-03
## 4    0.006878411 0.04403138  4.723545e-03
## 5    0.007292088 0.04412919 -4.808378e-03
```

```
## 6    0.011093186 0.04417744  2.296036e-03
## 7    0.009063964 0.04422570  4.484533e-03
## 8    0.006696219 0.04426488 -1.186904e-03
## 9    0.003558522 0.04430406  6.904084e-03
## 10  -0.003703722 0.04428190  2.970213e-03
## 11  -0.010065266 0.04425973 -1.994463e-03
## 12  -0.014935333 0.04441318 -3.542364e-03
## 13  -0.012164159 0.04456663 -5.789569e-03
## 14  -0.005945745 0.04471362 -6.607351e-04
## 15   0.002231834 0.04486062 -3.737611e-03
## 16   0.006878411 0.04491823 -2.663307e-03
## 17   0.007292088 0.04497584  5.062612e-04
## 18   0.011093186 0.04500876  1.013138e-02
## 19   0.009063964 0.04504168  5.216935e-03
## 20   0.006696219 0.04488493  5.193046e-03
## 21   0.003558522 0.04472818 -3.486698e-03
## 22  -0.003703722 0.04431994 -2.196859e-03
## 23  -0.010065266 0.04391169 -2.464284e-04
## 24  -0.014935333 0.04344140 -2.054455e-03
## 25  -0.012164159 0.04297111 -1.984368e-03
## 26  -0.005945745 0.04265981 -3.886474e-03
## 27   0.002231834 0.04234850  2.261130e-04
## 28   0.006878411 0.04227170 -1.116781e-03
## 29   0.007292088 0.04219490  1.513008e-03
## 30   0.011093186 0.04218339  3.023424e-03
## 31   0.009063964 0.04217188  4.280289e-03
## 32   0.006696219 0.04213627 -7.357190e-04
## 33   0.003558522 0.04210067 -3.625862e-03
## 34  -0.003703722 0.04180209 -1.324173e-03
## 35  -0.010065266 0.04150350  2.428429e-03
## 36  -0.014935333 0.04079457  1.221408e-03
## 37  -0.012164159 0.04008564 -7.924444e-04
## 38  -0.005945745 0.03948151  1.785660e-03
## 39   0.002231834 0.03887739  2.697225e-03
## 40   0.006878411 0.03864154  2.130054e-03
## 41   0.007292088 0.03840568  7.215879e-04
## 42   0.011093186 0.03840759 -7.634105e-03
## 43   0.009063964 0.03840949 -1.094120e-02
## 44   0.006696219 0.03859429 -3.645346e-03
## 45   0.003558522 0.03877908  7.095727e-03
## 46  -0.003703722 0.03924804  1.008392e-04
## 47  -0.010065266 0.03971700  3.482635e-04
## 48  -0.014935333 0.04025520  2.857553e-03
## 49  -0.012164159 0.04079340  2.338505e-03
## 50  -0.005945745 0.04089116  7.331607e-04
## 51   0.002231834 0.04098892 -4.626858e-04
## 52   0.006878411 0.04072307  2.631857e-03
## 53   0.007292088 0.04045722  4.508761e-03
## 54   0.011093186 0.04023313 -1.092979e-03
## 55   0.009063964 0.04000904 -4.556871e-03
## 56   0.006696219 0.03980012  9.875369e-04
## 57   0.003558522 0.03959119 -7.649716e-03
## 58  -0.003703722 0.03948933  9.563285e-04
## 59  -0.010065266 0.03938746  3.211137e-03
```

```
## 60  -0.014935333 0.03958361 -1.228925e-03
## 61  -0.012164159 0.03977976  1.029557e-03
## 62  -0.005945745 0.04015746  7.882878e-04
## 63   0.002231834 0.04053515 -1.508922e-03
## 64   0.006878411 0.04068343 -3.561840e-03
## 65   0.007292088 0.04083171  3.908462e-03
## 66   0.011093186 0.04084443 -3.709540e-04
## 67   0.009063964 0.04085716  4.659715e-04
## 68   0.006696219 0.04095089  6.707727e-03
## 69   0.003558522 0.04104462 -1.836479e-03
## 70  -0.003703722 0.04112503 -3.260018e-03
## 71  -0.010065266 0.04120544 -2.440169e-03
## 72  -0.014935333 0.04116586 -7.950401e-04
## 73  -0.012164159 0.04112628  7.153002e-04
## 74  -0.005945745 0.04109729  9.174244e-04
## 75   0.002231834 0.04106829  5.547094e-04
## 76   0.006878411 0.04126622  1.755371e-03
## 77   0.007292088 0.04146414 -1.853004e-03
## 78   0.011093186 0.04168507  2.021745e-03
## 79   0.009063964 0.04190600  1.752015e-04
## 80   0.006696219 0.04205585 -6.429485e-03
## 81   0.003558522 0.04220570 -4.975523e-04
## 82  -0.003703722 0.04215909  3.673669e-03
## 83  -0.010065266 0.04211247  3.319460e-03
## 84  -0.014935333 0.04188444 -1.336205e-03
## 85  -0.012164159 0.04165641 -4.922514e-04
## 86  -0.005945745 0.04137218  7.269994e-03
## 87   0.002231834 0.04108795  2.131829e-03
## 88   0.006878411 0.04078631 -4.298051e-03
## 89   0.007292088 0.04048466 -2.444947e-04
## 90   0.011093186 0.04029711 -6.773633e-03
## 91   0.009063964 0.04010956  3.103435e-04
## 92   0.006696219 0.04004253 -1.674231e-03
## 93   0.003558522 0.03997549  5.826512e-04
## 94  -0.003703722 0.04011236 -8.602550e-04
## 95  -0.010065266 0.04024923  5.493661e-04
## 96  -0.014935333 0.04049011  3.509737e-03
## 97  -0.012164159 0.04073099  3.207362e-03
## 98  -0.005945745 0.04074429 -3.744973e-03
## 99   0.002231834 0.04075759  3.654164e-04
## 100  0.006878411 0.04054622  1.775371e-03
## 101  0.007292088 0.04033485 -2.239840e-03
## 102  0.011093186 0.04014142  3.432062e-03
## 103  0.009063964 0.03994799  9.235311e-04
## 104  0.006696219 0.03990239 -5.021935e-05
## 105  0.003558522 0.03985679 -5.148642e-03
## 106 -0.003703722 0.03979642 -4.797964e-04
## 107 -0.010065266 0.03973606 -2.104123e-03
## 108 -0.014935333 0.03970143  1.153255e-03
## 109 -0.012164159 0.03966681  2.642510e-03
## 110 -0.005945745 0.03997833  7.455584e-05
## 111  0.002231834 0.04028985  1.252505e-03
## 112  0.006878411 0.04076189 -1.440300e-03
## 113  0.007292088 0.04123392 -2.074399e-03
```

```
## 114  0.011093186 0.04162100 -5.114182e-03
## 115  0.009063964 0.04200807 -4.591295e-04
## 116  0.006696219 0.04240444  7.057916e-04
## 117  0.003558522 0.04280081  7.507331e-03
## 118 -0.003703722 0.04322863  2.492802e-04
## 119 -0.010065266 0.04365646 -3.257856e-03
## 120 -0.014935333 0.04409543  3.345012e-05
```

```r
monthly_ozone_deseasoned_ts <- GaringerOzone.monthly.ts - ozone_components$seasonal

#16
monthly_trend2 <- Kendall::MannKendall(monthly_ozone_deseasoned_ts)

monthly_trend2
```

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

```r
summary(monthly_trend2)
```

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

> Answer: After deseasoning the Ozone Monthly Series and running a non-seasonal Mann-Kendall test, we find a statistically significant trend as well. This test actually rejects the null hypothesis of stationarity with much higher confidence as well (p-value = 0.00754)