Assignment 7: Time Series Analysis

Jack Alcorn

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
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3

## Warning: package 'ggplot2' was built under R version 4.0.3

## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'tidyr' was built under R version 4.0.3

## Warning: package 'readr' was built under R version 4.0.3

## Warning: package 'purrr' was built under R version 4.0.3

## Warning: package 'dplyr' was built under R version 4.0.3

## Warning: package 'stringr' was built under R version 4.0.3

## Warning: package 'forcats' was built under R version 4.0.3
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.3
```

```
#install.packages("trend")
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.0.4
```

```
#install.packages("zoo")
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.0.4
```

```
#install.packages("Kendall")
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.0.4
```

```
#install.packages("tseries")
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.4
```

```
library(scales)
```

```
## Warning: package 'scales' was built under R version 4.0.3
```

```
getwd()
```

```
## [1] "Z:/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Assignments"
```

```
OZ2010<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2011<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2012<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2013<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2014<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2015<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2016<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2017<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2018<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
OZ2019<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/EPAai
GrangierOzone<-rbind(OZ2010,OZ2011,OZ2012,OZ2013,OZ2014,OZ2015,OZ2016,OZ2017,OZ2018,OZ2019)

OGtheme<-theme_classic(base_size = 14)+
  theme(axis.text = element_text(color = "black"),
        plot.title = element_text(hjust = 0.5),
        legend.position = "right",
        legend.justification ="center",
        legend.background = element_rect(size=0.5, linetype="solid",
                                         colour ="darkblue"))
```

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
GrangierOzone$Date<-as.Date(GrangierOzone$Date, format = "%m/%d/%Y")
# 4
GrangierOZone2<-select(GrangierOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
# 5
summary(GrangierOZone2)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-03   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2015-01-04   Median :0.04100                      Median : 38.00
##  Mean   :2015-01-01   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-02   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
```

```
str(GrangierOZone2)
```

```
## 'data.frame':    3589 obs. of  3 variables:
##  $ Date                                : Date, format: "2010-01-01" "2010-01-02" ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.031 0.033 0.035 0.031 0.027 0.033 0.035 0.032 0.032 0
##  $ DAILY_AQI_VALUE                     : int  29 31 32 29 25 31 32 30 30 28 ...
```

```
Days<-as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), "days"))
names(Days)[1]<-paste("Date")
# 6
GrangierOzone3<-left_join(Days, GrangierOZone2, by = c("Date"))
```

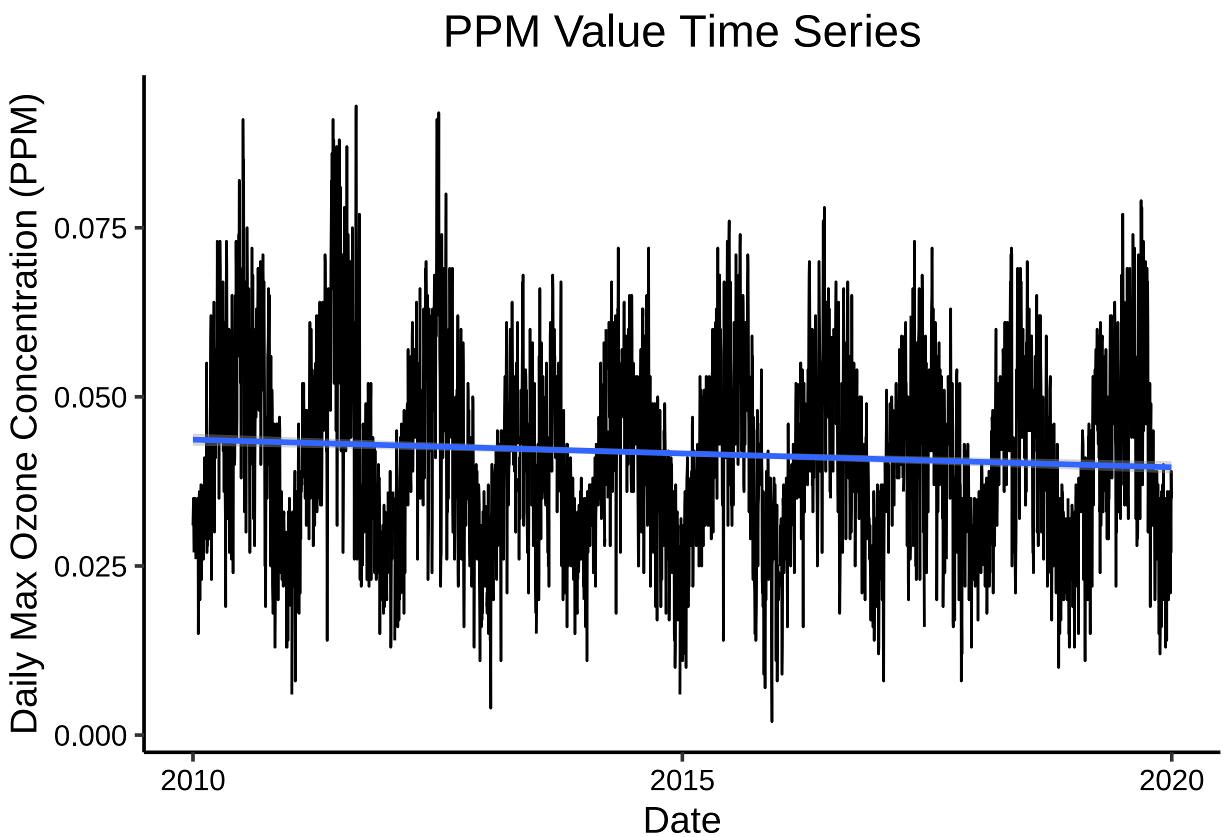
Visualize

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(GrangierOzone3, aes(y=Daily.Max.8.hour.Ozone.Concentration, x=Date))+
  geom_line()+
  geom_smooth(method=lm)+
  labs(title = "PPM Value Time Series")+
  ylab("Daily Max Ozone Concentration (PPM)")+
  OGtheme
```

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

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



Answer:There does not appear to be a discernable trend over time. Potentially there is a slight decrease over time but it look very minor.

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
Grangier_Ozone_clean <-
  GrangierOzone3 %>%
  mutate( Daily_Ozone.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: We used linear because we only have a few missing values in our data. Linear interpolation is a good way to "connect the dots" in small gaps of data.

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
GrangierOzone_monthly<-
  Grangier_Ozone_clean%>%
  mutate(month = month(Date))%>%
  mutate(year = year(Date))%>%
```

```
  group_by(month, year)%>%
  summarize(mean_Ozone=mean(Daily_Ozone.clean))
```

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

```
GrangierOzone_monthly$Date<-ymd(paste(GrangierOzone_monthly$year,GrangierOzone_monthly$month,"1"))
```

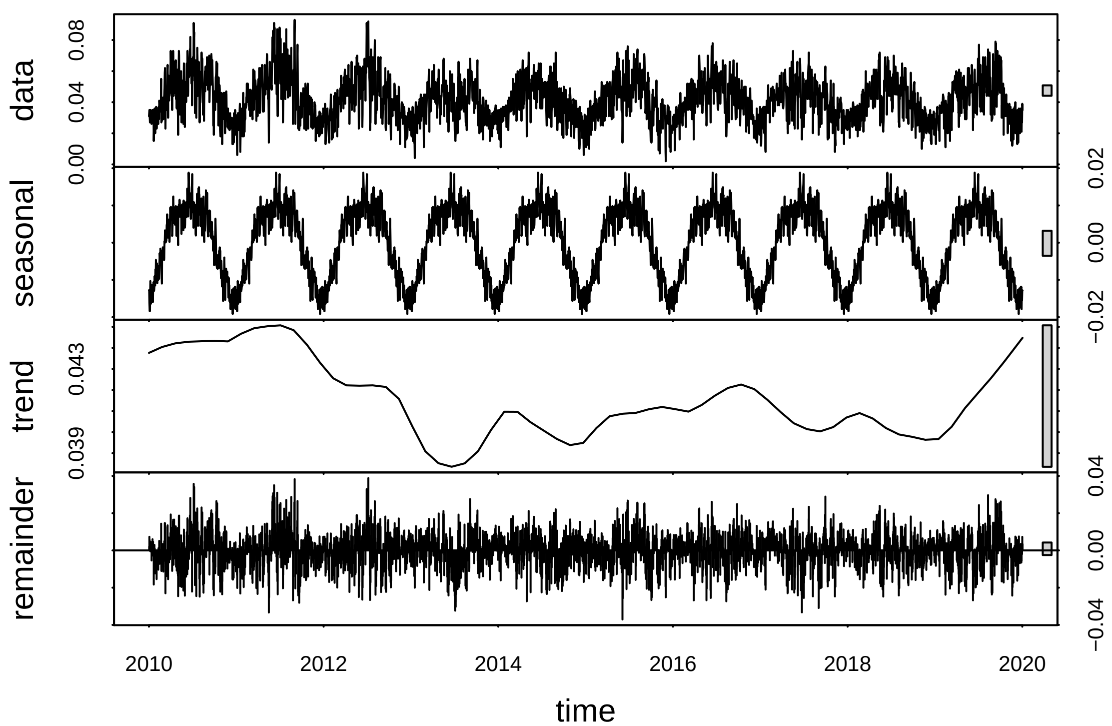
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```
#10
GrangierOzone.daily.ts<-ts(Grangier_Ozone_clean$Daily_Ozone.clean, start=c(2010,1), frequency=(365))
GrangierOzone_monthly.ts<-ts(GrangierOzone_monthly$mean_Ozone, start=c(2010,1), frequency = (12))
```

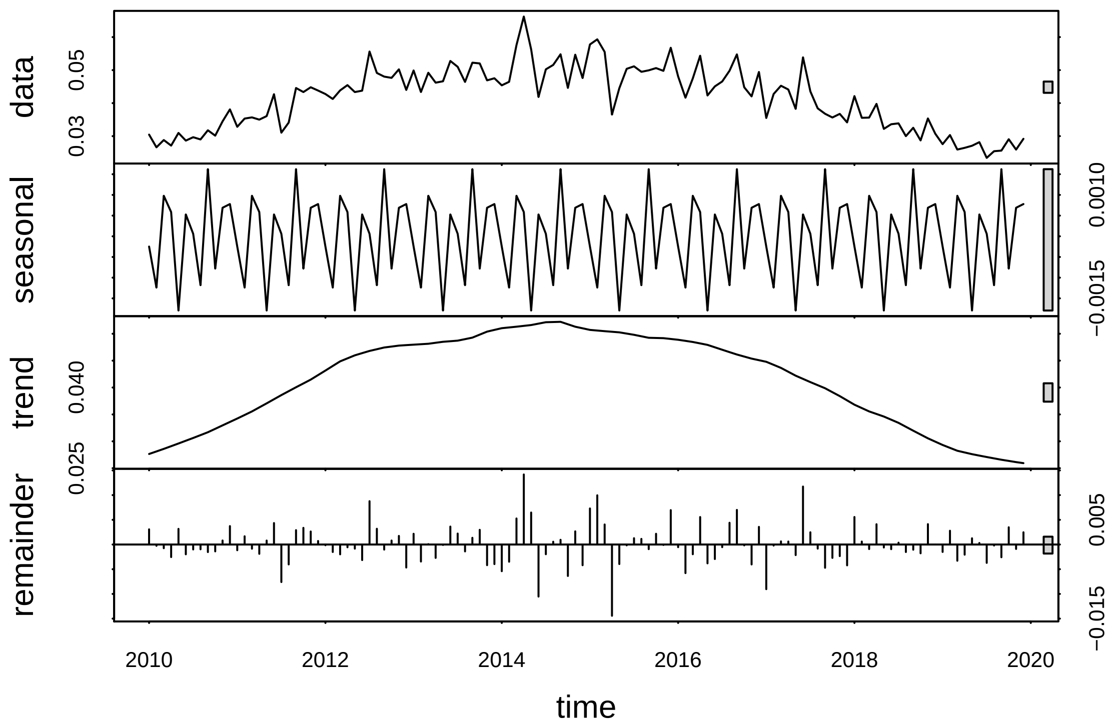
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11
GrangierOzone.daily.decomposed<-stl(GrangierOzone.daily.ts, s.window = "periodic")
GrangierOzone.monthly.decomposed<-stl(GrangierOzone_monthly.ts, s.window = "periodic")

plot(GrangierOzone.daily.decomposed)
```




```
plot(GrangierOzone.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
monthly_ozone_trend<- Kendall::SeasonalMannKendall(GrangierOzone_monthly.ts)
monthly_ozone_trend
```

```
## tau = -0.1, 2-sided pvalue =0.16323
```

```
summary(monthly_ozone_trend)
```

```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

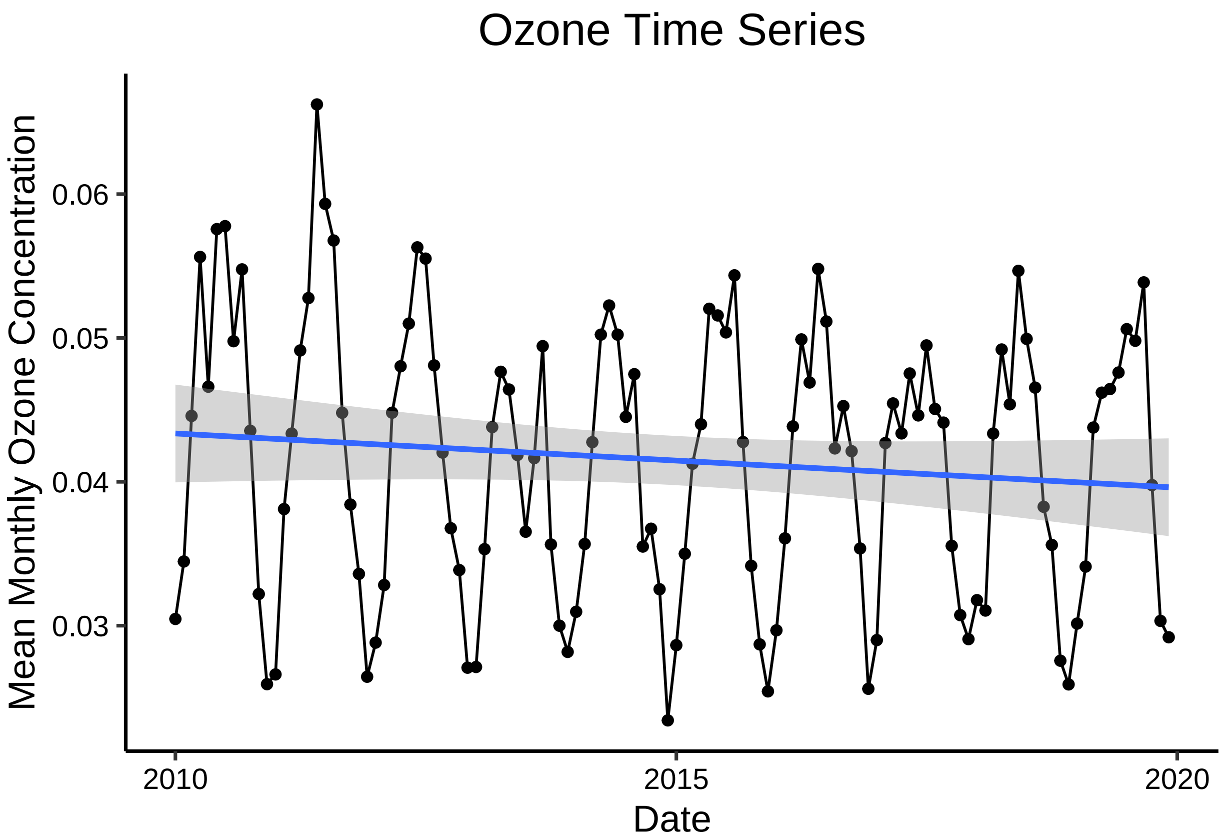
Answer: Our data has seasonality to it. Ozone levels increase during the summer and decrease during the winter. THe Seasonal Mann Knedall test will account for those seasonal trends in the data.

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
GrangierOzone_monthly_plot <-
ggplot(GrangierOzone_monthly, aes(x = Date, y = mean_Ozone)) +
  geom_point() +
  geom_line()+
  ylab("Mean Monthly Ozone Concentration") +
  xlab("Date")+
```

```
  labs(title = "Ozone Time Series")+
  geom_smooth( method = lm )+
  OGtheme
print(GrangierOzone_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: For our research, the null hypothesis would be that the time series data is stationary and the alternative hypothesis is that the time series data is not stationary and thus has a positive or negative trend. Our results conclude that we would accept the null hypothesis. This is becuase our p-value is greater than .05 (pvalue = 0.16).

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
GrangierOzone.monthly.components<-
  as.data.frame(GrangierOzone.monthly.decomposed$time.series[,2:3])



#16
```

```
GrangierOzone.monthly.comp.ts<-ts(GrangierOzone.monthly.components$trend)
monthly_nonseasonal_trend<- Kendall::MannKendall(GrangierOzone.monthly.comp.ts)
monthly_nonseasonal_trend
```

```
## tau = -0.13, 2-sided pvalue =0.035101
```

```
summary(monthly_nonseasonal_trend)
```

```
## Score =  -930 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.13, 2-sided pvalue =0.035101
```

> Answer: The results of the Mann Kendall test show that we would reject the null hypthosesis and accept the alternative hypothesis that there is not a stationary trend in the data (pvalue = .035). This is different that the seasonal kendall results in which we accepted the null hypthosesis.