Assignment 7: Time Series Analysis

Dori Rathmell

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
getwd()
```

```
## [1] "/home/guest/R/EDA Fall"
```

```
library(tidyverse)
#install.packages(zoo)
library(zoo)
library(lubridate)
#install.packages('trend')
library(trend)
#install.packages('Kendall')
library(Kendall)
#installing and loading necessary packages


timeseries_plottheme <- theme_gray(base_size = 12) +
  theme(axis.text = element_text(color = "Gray"),
        legend.position = "right")
```

```
theme_set(timeseries_plottheme)
#setting the plot theme
#2

timeseries1 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv')
timeseries2 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv')
timeseries3 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv')
timeseries4 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv')
timeseries5 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv')
timeseries6 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv')
timeseries7 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv')
timeseries8 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv')
timeseries9 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv')
timeseries10 <-
  read.csv('./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv')
#creating all the dataframes

GaringerOzone <- rbind(timeseries1,timeseries2,timeseries3,timeseries4,
                       timeseries5,timeseries6,timeseries7,timeseries8,
                       timeseries9,timeseries10)
#joining all the dataframes into one singular dataframe, 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$Date <- as.Date(GaringerOzone$Date, format = '%m/%d/%Y')
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

```
#setting the Date column to a date class
# 4
GaringerOzoneWrangled <-
```

```
  GaringerOzone %>%
  select('Date', 'Daily.Max.8.hour.Ozone.Concentration', 'DAILY_AQI_VALUE')
#selecting relevant columns


# 5
Days <- as.data.frame(seq(as.Date('2010-01-01'),as.Date('2019-12-31'), 'days'))
colnames(Days) <- c('Date')
#creating a dataframe with a sequence of dates

# 6
#class(GaringerOzoneWrangled$Date)
GaringerOzoneWrangled$Date <- as.Date(GaringerOzoneWrangled$Date,
                                      format = '%m/%d/%Y')
GaringerOzone1 <- left_join(Days, GaringerOzoneWrangled, by = c("Date"))
#combining the wrangled dataframe and the Days dataframe into 1.
```

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?

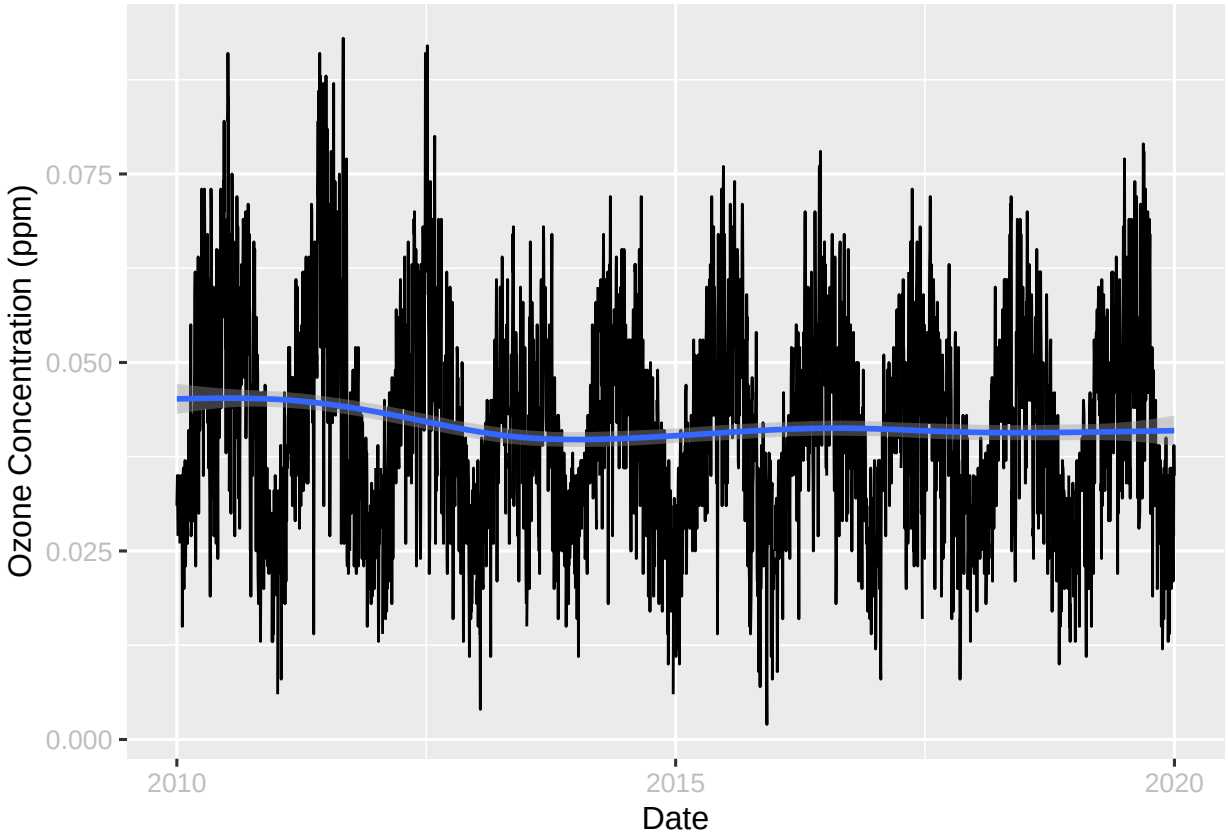
```
#7
lineplot1 <-
  ggplot(GaringerOzone1, aes(x = Date,
                             y = Daily.Max.8.hour.Ozone.Concentration))+
  geom_line(aes(Color = 'black'))+
  geom_smooth(aes(Color = 'blue'))+
  timeseries_plottheme+
  ylab('Ozone Concentration (ppm)')
```

```
## Warning: Ignoring unknown aesthetics: Color
## Ignoring unknown aesthetics: Color
```

```
#creating the initial line plot
print(lineplot1)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

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



Answer:The plot shows a fairly flat line that does not demonstrate a trend in ozone concentration 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(GaringerOzone1$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
```

```
#checking the initial summary
GaringerOzone.clean<-
  GaringerOzone1 %>%
  mutate(Ozone.clean = na.approx(Daily.Max.8.hour.Ozone.Concentration) )
#making the linear interpolation to replace missing data points
summary(GaringerOzone.clean$Ozone.clean)
```

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

```
#checking the summary after interpolation
```

Answer: A linear interpolation means the missing data is assumed to fall inbetween the nearest

two data points, while spline takes a quadratic function and piecewise assumes that the missing value is the same as the missing points nearest neighbor. We used a linear interpolation so as to not change the mean of the overall monthly data, spline and piecewise would be close but would change the value of the mean on a monthly level.

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(Month = month(Date))%>%
  mutate(Year = year(Date)) %>%
  group_by(Year, Month) %>%
  summarize(meanozone = mean(Ozone.clean)) %>%
              mutate(month_year = my(paste(Month, "-", Year)))
```

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

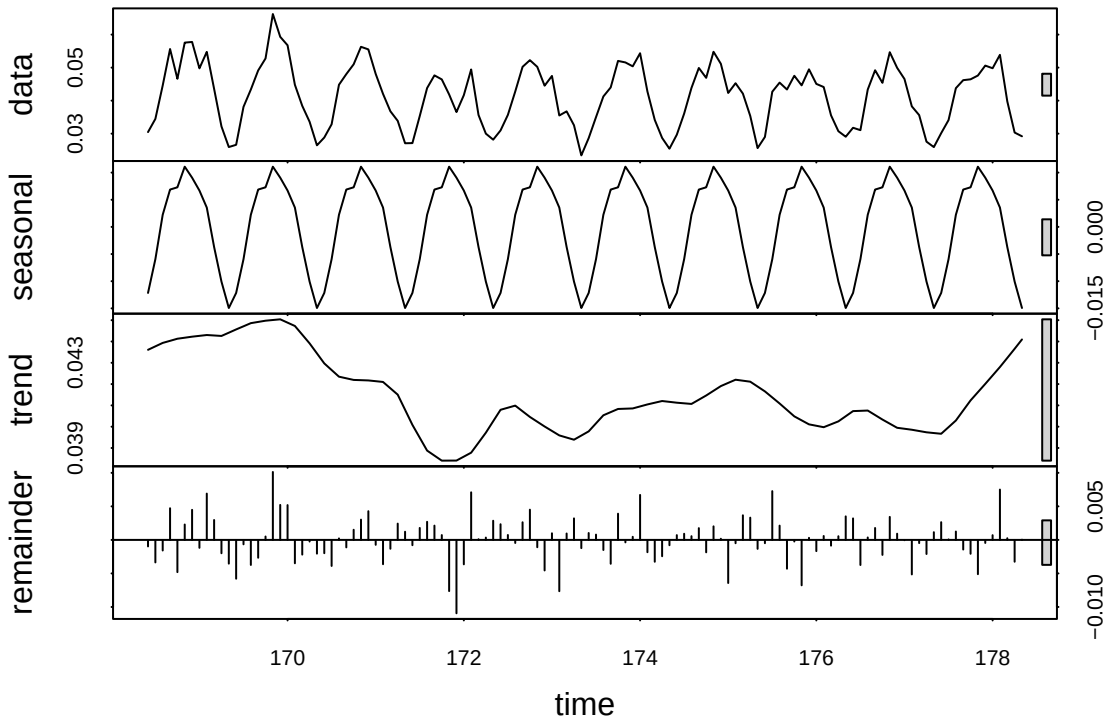
```
#Running a pipe to find the mean concentrations of ozone by month
```

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.

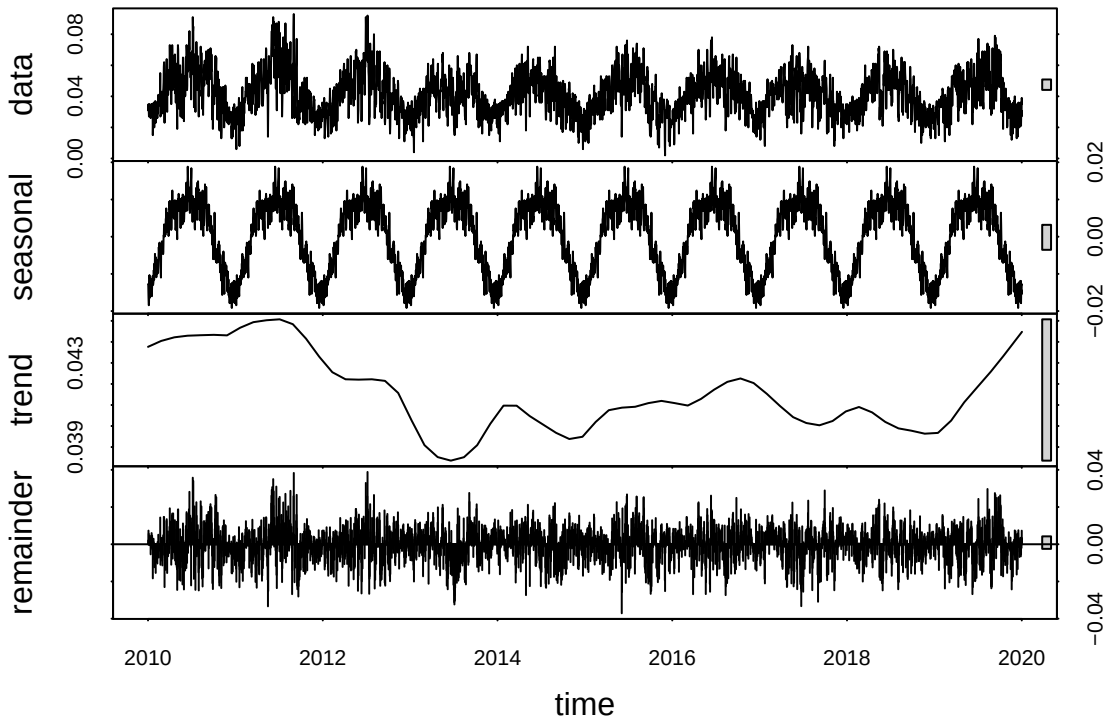
```
#10
f_month <- month(first(GaringerOzone.clean$Date))
f_year <- year(first(GaringerOzone.clean$Date))
f_day<- day(first(GaringerOzone.clean$Date))
f_month2 <- first(GaringerOzone.monthly$Month)
f_year2 <- first(GaringerOzone.monthly$Year)
#creating the starting values for the time series

GaringerOzone.daily.ts <- ts(GaringerOzone.clean$Ozone.clean,
                             start=c(f_year,f_month, f_day),frequency=365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanozone,
                               start=c(f_month2, f_year2),frequency = 12)
#making the actual timeseries
```

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

```
#11
Garingerdecomp.monthly <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
Garingerdecomp.daily <-stl(GaringerOzone.daily.ts,s.window = "periodic")
#decomposing both monthly and daily timeseries
plot(Garingerdecomp.monthly)
```




```
plot(Garingerdecomp.daily)
```




```
#plotting both sets of timeseries data
```

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
Garinger_SMK <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(Garinger_SMK)
```

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

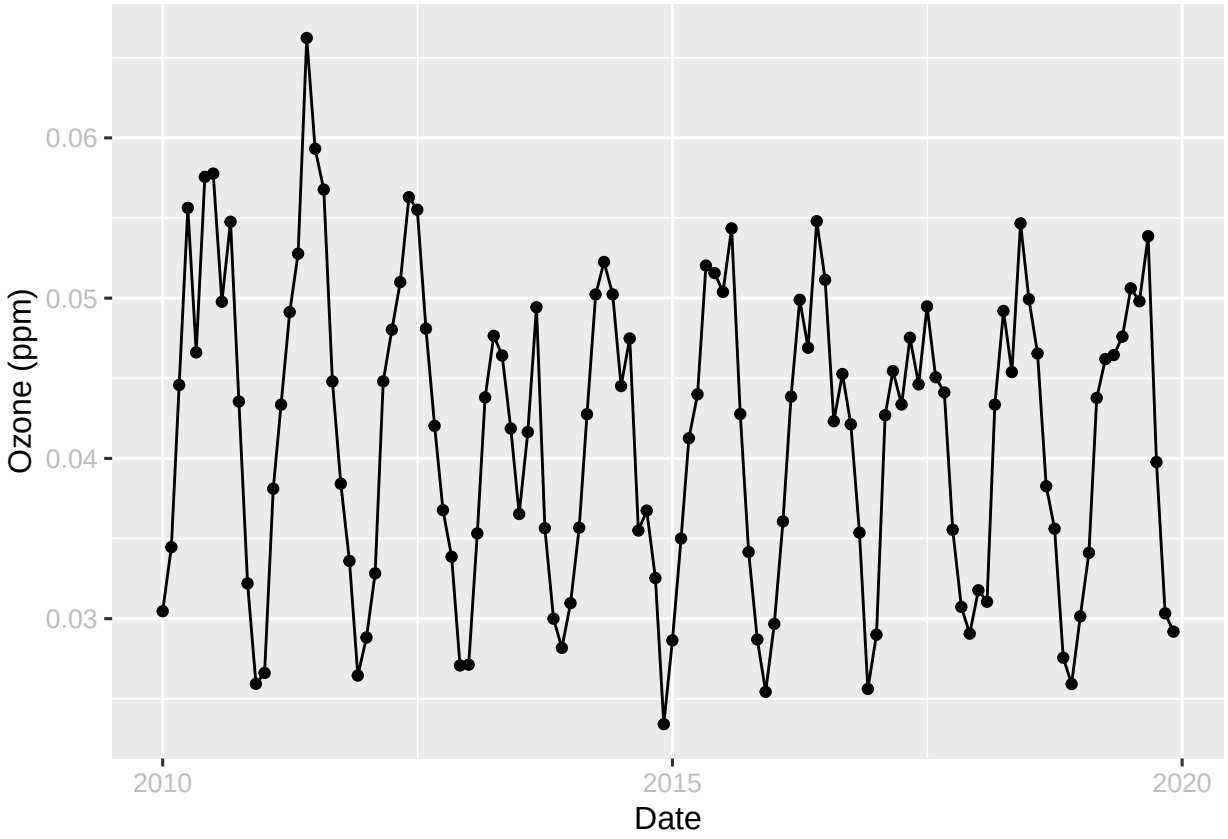
```
#running the seasonal mann kendall test
```

Answer: Because we are looking at the data on a seasonal level, we are evaluating ozone levels throughout the year on a monthly level and therefore examining trends in 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
ozoneOverTime_plot<-
  ggplot(GaringerOzone.monthly, aes(x = month_year, y = meanozone))+
  geom_point()+
  geom_line()+
  timeseries_plottheme+
  ylab('Ozone (ppm)')+
  xlab('Date')
```

```
#creating a plot of mean ozone over time

print(ozoneOverTime_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: The graph shows that there is a seasonal variation in ozone and that since 2010 the amplitude in ozone levels has decreased. In the early 2010's the mean ozone peaked at around .065ppm but in later years it never surpassed .055ppm (Score = -77, Var(score)= 1499, denominator = 539.4972, tau = -0.143, 2 sided 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
Garinger_extracted <- as.data.frame(Garingerdecomp.monthly$time.series[,2:3])

Garinger_extracted<- mutate(Garinger_extracted,
        Observed = GaringerOzone.monthly$meanozone,
        Date = GaringerOzone.monthly$month_year)
#erasing the seasonal component from the data
#16
Garinger_extracted.mk <- MannKendall(GaringerOzone.monthly.ts)
```

```
summary(Garinger_extracted.mk)
```

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

```
#running a (non seasonal) mann kendall test on the data.
```

Answer: The 2- sided P value for the non-seasonal Garinger ozone is .33732 while the 2-sided P value for the seasonal Garinger ozone levels was .0467. The P value for the seasonal data indicates that that dataset is more significant than the non-seasonal data.