Assignment 7: Time Series Analysis

Karen Thornton

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/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmental_Data_Analy
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(zoo)
```

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

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

```
library(trend)

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

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
EPA2010 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2011 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2012 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2013 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2014 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2015 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2016 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2017 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2018 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen
EPA2019 <- read.csv("/Users/karenthornton/Documents/School/Grad School/Year 1/Semester 2/EDA/Environmen

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

dim(GaringerOzone)
```

```
## [1] 3589   20
```

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"
```

```
# 4
GaringerOzoneWrangled <-
  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"))
colnames(Days) <- c('Date')
view(Days)

# 6

#naming it GaringerOzone2 so it's not confused with the GaringerOzone above

GaringerOzone2 <- left_join(Days, GaringerOzoneWrangled)
```

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

```
dim(GaringerOzone2)
```

```
## [1] 3652    3
```

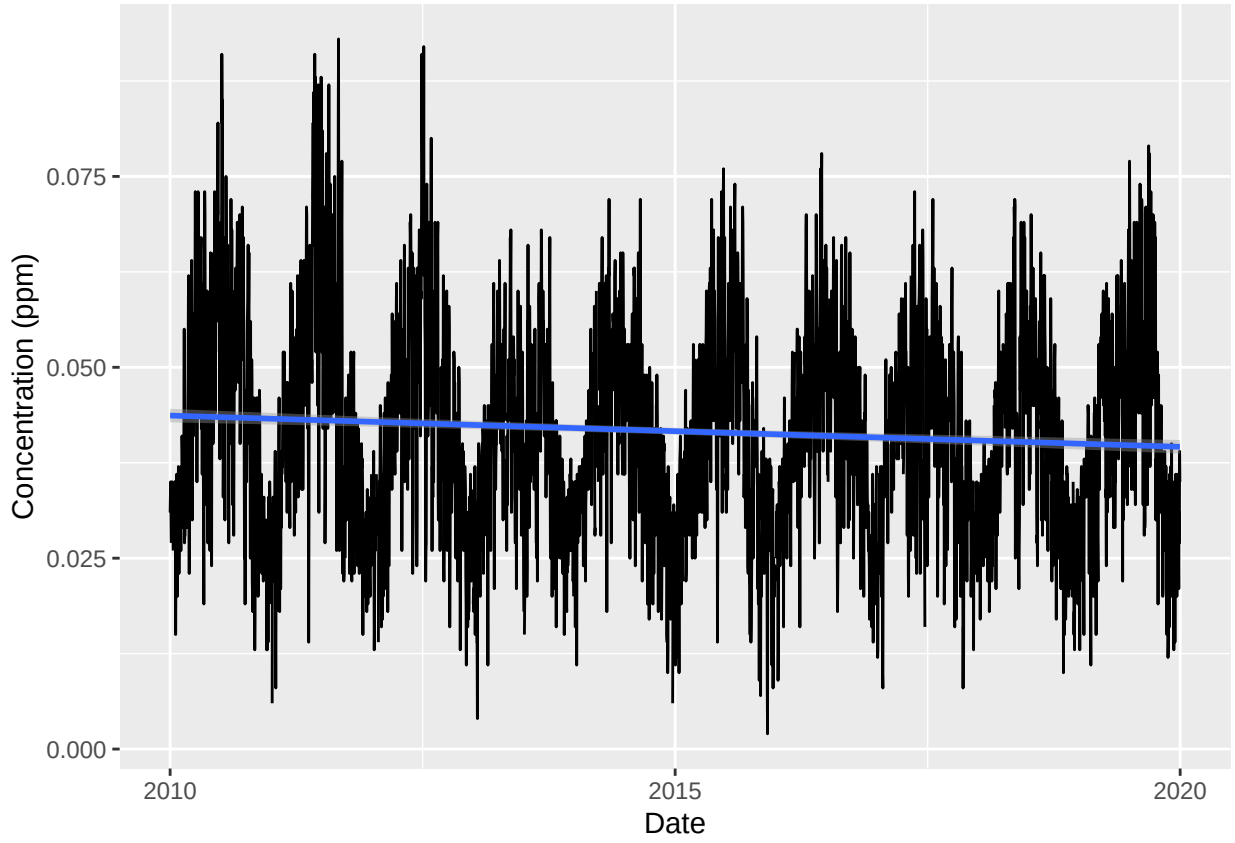
Visualize

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(GaringerOzone2,
       aes(x= Date, y= Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  labs(x = "Date", y="Concentration (ppm)")+
  geom_smooth(method = lm)
```

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

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



Answer: Yes, there seems to be a slight downward trend in 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
GaringerInterpolation <-
  GaringerOzone2 %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(GaringerInterpolation$Daily.Max.8.hour.Ozone.Concentration)
```

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

Answer: We used a linear interpolation because the concentration is continuous. In order to get from one concentration to the next it has to go through every concentration in between. Using a linear interpolation will get approximate this correctly. Using piecewise would grab the value of the closest concentration, so it would jump in concentrations instead of being continuous. Spline wasn't used because a quadratic equation was not needed.

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 <-
  GaringerInterpolation %>%
  mutate(month = month(Date)) %>%
  mutate(year = year(Date)) %>%
  mutate(month.year = my(paste0(month, "-", year))) %>%
  group_by(month.year, year, month) %>%
  summarise(MeanOzone =
              mean(Daily.Max.8.hour.Ozone.Concentration))
```

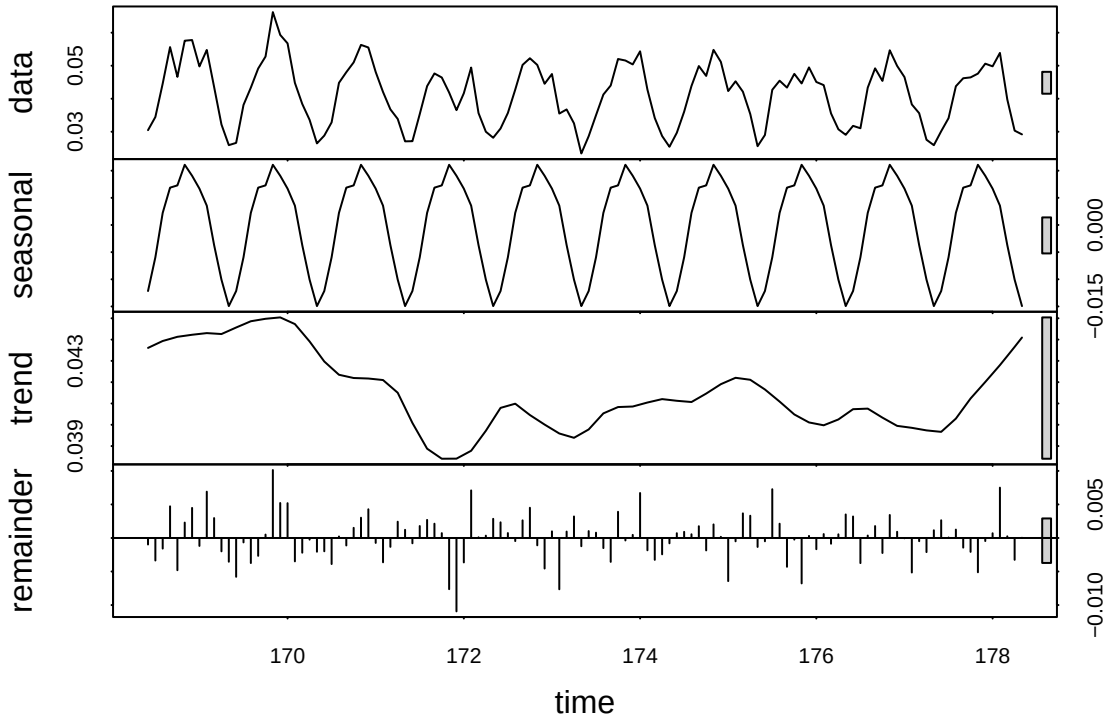
```
## `summarise()` has grouped output by 'month.year', 'year'. You can override using the `.groups` argume
```

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.

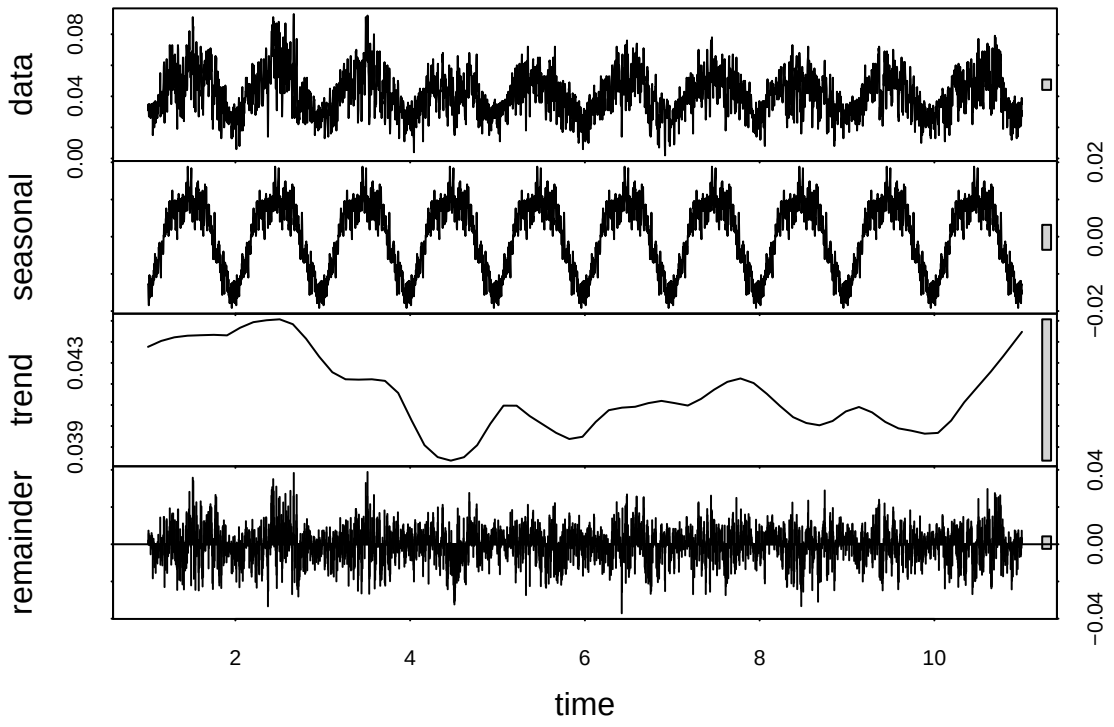
```
#10

GaringerOzone.daily.ts <-
  ts(GaringerInterpolation$Daily.Max.8.hour.Ozone.Concentration,
     start = c(1,1,2010), frequency = 365)


GaringerOzone.monthly.ts  <-
  ts(GaringerOzone.monthly$MeanOzone,
     start = c(1,2010), frequency = 12)

#check these to see if they are correct! do i need to "print" them
```

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

```
#11

monthly_decomposed <-
  stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(monthly_decomposed)
```




```
daily_decomposed <-
  stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(daily_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
trend::smk.test(GaringerOzone.monthly.ts)
```

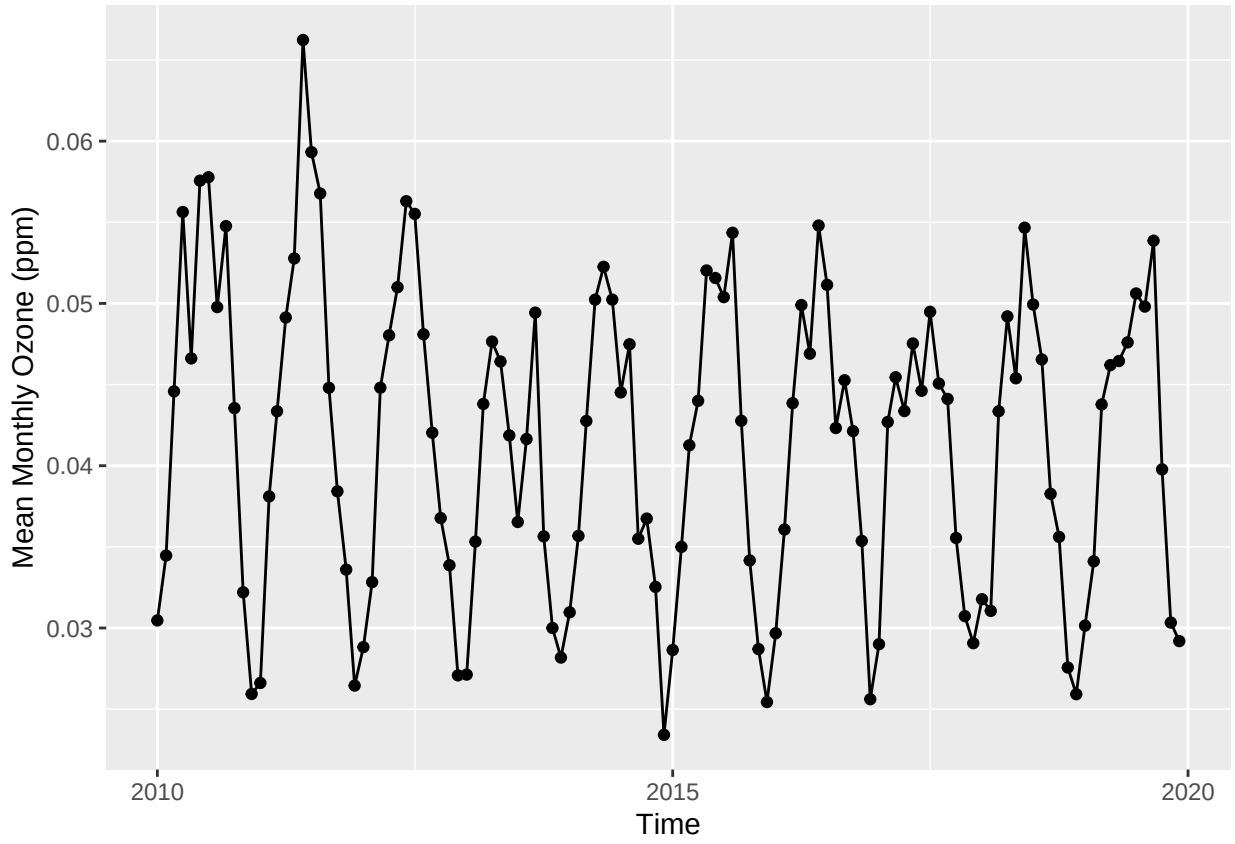
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

```
#p-value is 0.04965
```

Answer: We use the seasonal Mann-Kendall because the data is seasonal and non-parametric. This data is nonparametric because we are evaluating it for the whole population, not a certain parameter. It is seasonal because it goes up and down in a cycle 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
ggplot(GaringerOzone.monthly,
       aes(x= month.year, y = MeanOzone))+
  geom_point()+
  geom_line()+
  labs(y= "Mean Monthly Ozone (ppm)", x= "Time")
```



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: Even with seasonality, the time series is still statistically significant (ozone decreasing over time). The p-value is 0.04965 which is less than 0.05, meaning the ozone has changed significantly since 2010.

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.

```r
#15
Garinger_Components <-
  monthly_decomposed$time.series[,2]+
```

```
  monthly_decomposed$time.series[,3]


#16

trend::mk.test(Garinger_Components)
```

```
##
##  Mann-Kendall trend test
##
## data:  Garinger_Components
## z = -2.672, n = 120, p-value = 0.00754
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.179000e+03  1.943657e+05 -1.651376e-01
```

```
#p-value is 0.00754
```

Answer: The p-value for the Mann-Kendall test (p-value = 0.00754) was a lot smaller (more statistically significant) than the Seasonal Mann-Kendall test (p-value = 0.04965). This shows that if you take the seasonality out of the equation the data is stil statistically significant. The ozone is still decreasing with time.