Assignment 8: Time Series Analysis

Kim Myers

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., "Salk_A06_GLMs_Week1.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 3 at 1:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme
- 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). Call these GaringerOzone201*, with the star filled in with the appropriate year in each of ten cases.

```
getwd()
```

```
## [1] "C:/Users/Temp/Documents/Duke/S20/DataAnalytics/Environmental_Data_Analytics_2020/Assignments"
```

```
pacman::p_load(tidyverse,lubridate,zoo,trend) #load packages at once
GaringerOzone2010 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
GaringerOzone2011 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
GaringerOzone2012 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
GaringerOzone2013 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
GaringerOzone2014 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
GaringerOzone2015 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
GaringerOzone2016 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
GaringerOzone2017 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
GaringerOzone2018 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
GaringerOzone2019 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")
```

Wrangle

2. Combine your ten datasets into one dataset called GaringerOzone. Think about whether you should use a join or a row bind.
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 2019-12-13 (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 comine 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.

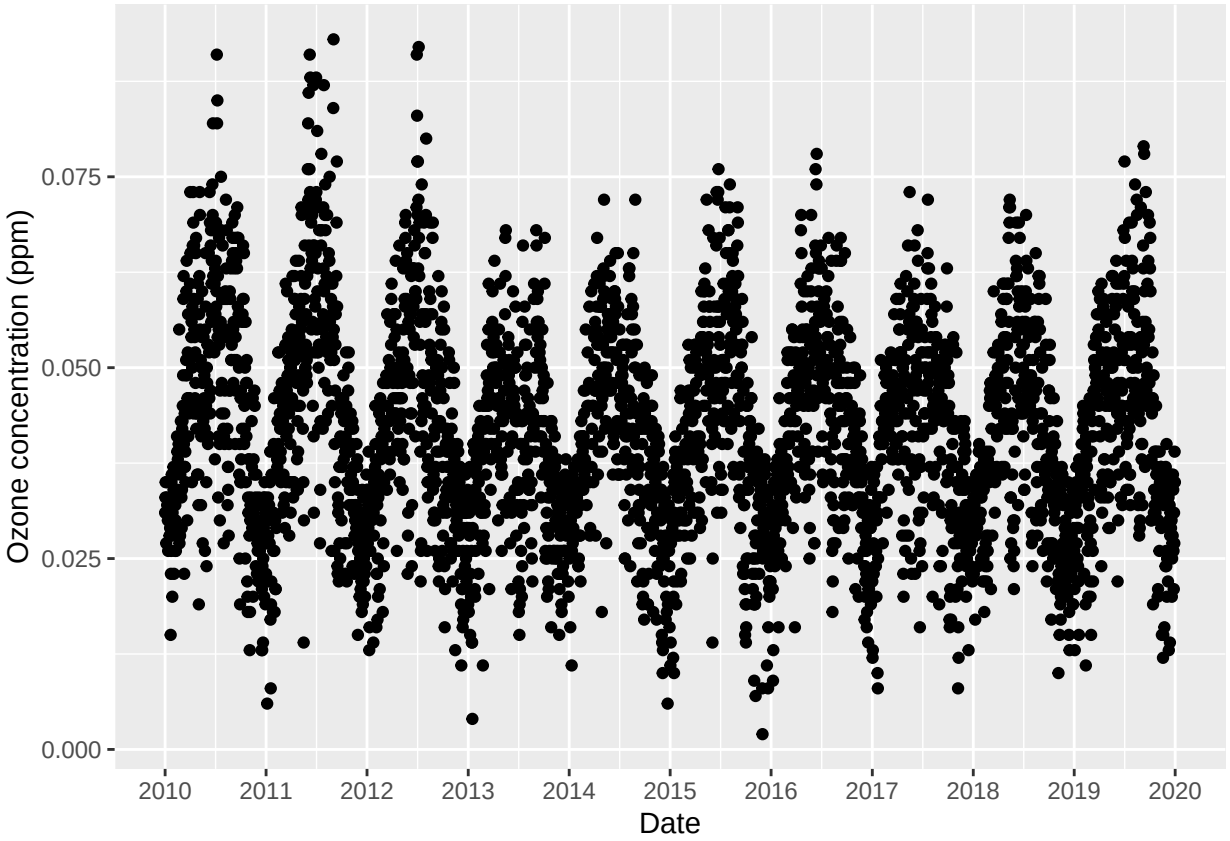
```
# 2
GaringerOzone <- rbind(GaringerOzone2010,GaringerOzone2011,GaringerOzone2012,
                       GaringerOzone2013,GaringerOzone2014,GaringerOzone2015,
                       GaringerOzone2016,GaringerOzone2017,GaringerOzone2018,
                       GaringerOzone2019)

# 3
GaringerOzone$Date <- as.Date(GaringerOzone$Date,format="%m/%d/%Y")

# 4
GaringerOzone <- 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"),"days"))
colnames(Days) <- "Date"

# 6
GaringerOzone <- left_join(Days,GaringerOzone,"Date")
```

Visualize

7. Create a ggplot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly.

```
ggplot(GaringerOzone) +
  geom_point(aes(x=Date,y=Daily.Max.8.hour.Ozone.Concentration)) +
  scale_x_date(date_breaks = "year", date_labels =  "%Y") +
  labs(y="Ozone concentration (ppm)")
```

```
## Warning: Removed 63 rows containing missing values (geom_point).
```



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?

   Answer: Linear interpolation makes the most sense in this context because concentration data is temporally autocorrelated. In other words, the most likely value on a given day is between the prior day and the proceeding day. The piecewise interpolation would not reflect this relationship and the spline interpolation would not be necessary as as the day to day relationship is likely not a quadratic function.

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)

10. Generate a time series called GaringerOzone.monthly.ts, with a monthly frequency that specifies the correct start and end dates.

11. Run a time series analysis. In this case the seasonal Mann-Kendall is most appropriate; why is this?

    Answer: We are using the seasonal Mann-Kendall analysis because the data appears to have a cyclical trend and the linear regression, Mann-Kendall, and modified Mann-Kendall analyses do not account for this.

12. To figure out the slope of the trend, run the function `sea.sens.slope` on the time series dataset.

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. No need to add a line for the seasonal Sen's slope; this is difficult to apply to a graph with time as the x axis. Edit your axis labels accordingly.

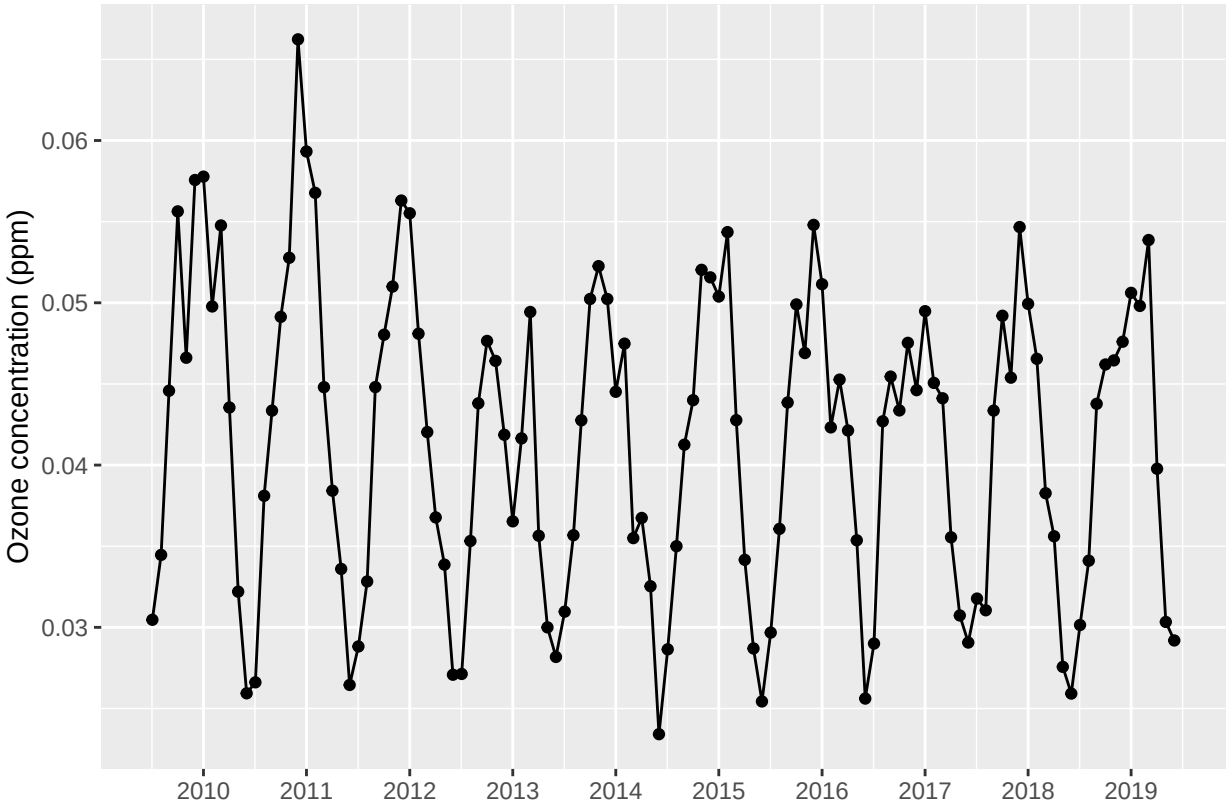
```
# 8
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Co

# 9
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(month=month(Date),year=year(Date))
GaringerOzone.monthly$DayYear <- as.Date(paste(GaringerOzone.monthly$year,GaringerOzone.monthly$month,1
          format = "%Y-%m-%d")

# 10
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration, frequency =

# 11
GaringerSeries <- smk.test(GaringerOzone.monthly.ts)
GaringerSeries
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = 8.4714, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  328 1490
```

```
# 12
sea.sens.slope(GaringerOzone.monthly.ts)
```

```
## [1] 0.003236111
```

```
# 13
GaringerOzone.mean.monthly <- GaringerOzone.monthly %>%
  group_by(month,year) %>%
  summarise(O3conc = mean(Daily.Max.8.hour.Ozone.Concentration))

GaringerOzone.mean.monthly$date <- as.Date(paste(factor(GaringerOzone.mean.monthly$year),factor(Garinge
          format = "%Y-%m-%d")

ggplot(GaringerOzone.mean.monthly, aes(x=date,y=O3conc)) +
  geom_point() +
  geom_line() +
  labs(y="Ozone concentration (ppm)",x="") +
  scale_x_date(date_breaks = "12 months", date_labels =  "%Y")
```



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: My results show that there was a significant seasonal trend of ozone concentrations from January 2010 through December 2019 (Seasonal Mann-Kendall test, z=8.4714, pivalue<0.0001). This trend tends to peak at the beginning of the year and valley in the middle of the year. From month to month, the change in concentration is roughly 0.003 ppm.