

Assignment 8: Time Series Analysis

Claire Mullaney

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.

```
#Checking working directory and loading packages
```

```
getwd()
```

```
## [1] "/Users/clairemullaney/Desktop/ENV 872/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
```

```
library(lubridate)
```

```
library(zoo)
```

```
library(trend)
```

```
#Setting default ggplot theme
```

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

```
theme_set(deftheme)
```

```
#Importing datasets and placing them into a list
```

```
Ozone_files <- list.files(path = "./Data/Raw/Ozone_TimeSeries/",  
                          pattern="*.csv", full.names=TRUE)
```

```
Ozone_files_named <- list()
```

```
for(i in 1:length(Ozone_files)) {
```

```
Ozone_files_named[[i]] <- assign(paste("GaringerOzone201", i-1 , sep = ""), read.csv(Ozone_files[i]))
}
```

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 2019-12-31 (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.

```
# 2
#Checking column names to decide whether to use join vs. rbind
lapply(Ozone_files_named, colnames)
```

```
## [[1]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[2]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
```

```

## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[3]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[4]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"

```

```

## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[5]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[6]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[7]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"

```

```

## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQI_PARAMETER_CODE"
## [12] "AQI_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[8]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQI_PARAMETER_CODE"
## [12] "AQI_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[9]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQI_PARAMETER_CODE"
## [12] "AQI_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"

```

```
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
##
## [[10]]
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

#Combining datasets into one using rbind_list; rbind_list was used so the dataframes could be bound from

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

```
GaringerOzone <- rbind_list(Ozone_files_named)
```

```
## Warning: 'rbind_list' is deprecated.
```

```
## Use 'bind_rows()' instead.
```

```
## See help("Deprecated")
```

```
## Warning in bind_rows(list_or_dots(...), id = NULL): Unequal factor levels:
## coercing to character
```

```
## Warning in bind_rows(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector
```

```
## Warning in bind_rows(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector
```

```
## Warning in bind_rows(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector
```

```
## Warning in bind_rows(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector
```

```
## Warning in bind_rows_(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector

## Warning in bind_rows_(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector

## Warning in bind_rows_(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector

## Warning in bind_rows_(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector

## Warning in bind_rows_(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector

## Warning in bind_rows_(list_or_dots(...), id = NULL): Unequal factor levels:
## coercing to character

## Warning in bind_rows_(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector

## Warning in bind_rows_(list_or_dots(...), id = NULL): binding character and
## factor vector, coercing into character vector
```

```
# 3
```

```
#Changing date column to date format
```

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

```
# 4
```

```
#Selecting necessary columns
```

```
GaringerOzone.wr <- GaringerOzone %>%
```

```
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
```

```
# 5
```

```
#Creating a days data frame with a column called "Date"
```

```
Days <- as.data.frame(seq(from = as.Date("2010-01-01"), to =
                        as.Date("2019-12-31"), by = 1))
```

```
Days <- rename(Days, "Date" = "seq(from = as.Date(\"2010-01-01\"), to = as.Date(\"2019-12-31\"), by = 1)\"")
```

```
# 6
```

```
#Joining the data frames and checking the dimensions
```

```
GaringerOzone <- left_join(Days, GaringerOzone.wr)
```

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

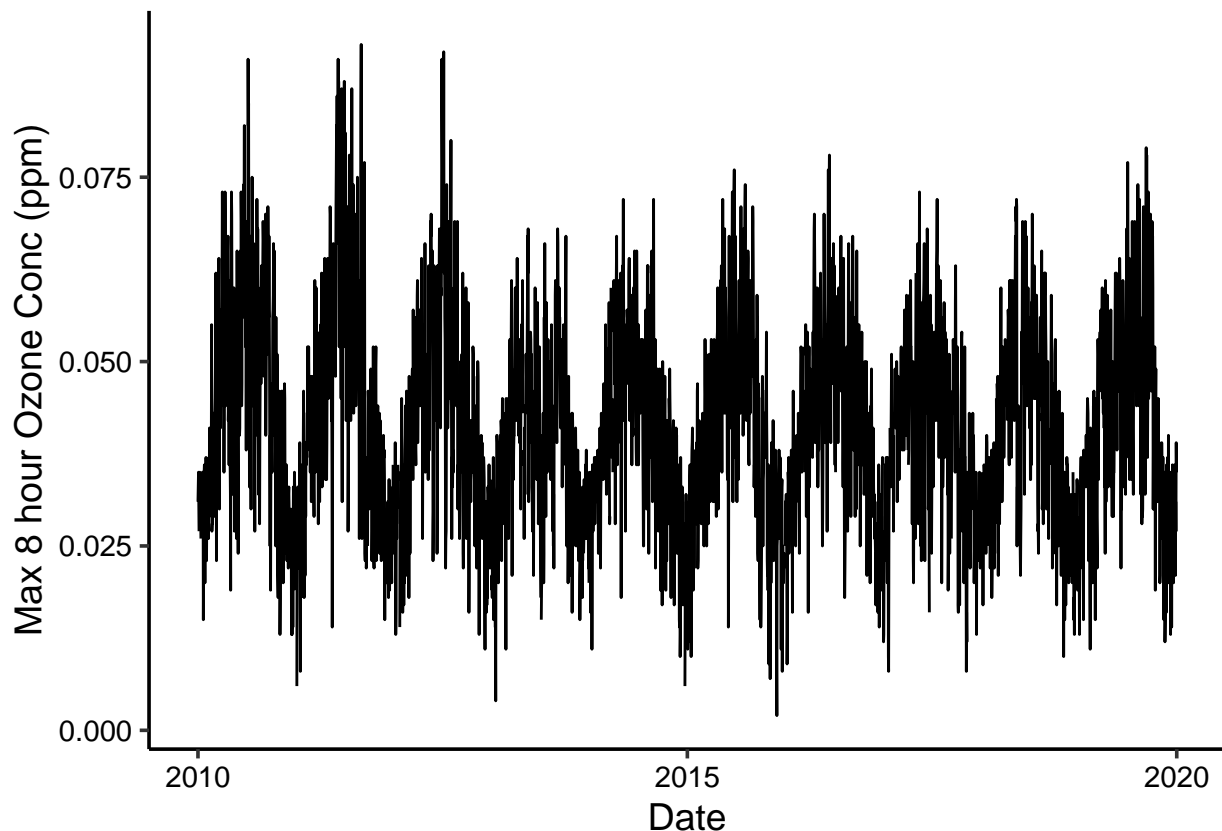
```
dim(GaringerOzone)
```

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

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.

```
#Plotting ozone concentrations over time  
ggplot(data = GaringerOzone,  
       aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +  
  geom_line() +  
  labs(y = "Max 8 hour Ozone Conc (ppm)")
```



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: ???

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: ???

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
#Linear interpolation
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)

# 9
#Adding Month and Year columns, grouping the df by month/year, and summarizing ozone concentrations by month/year
GaringerOzone.monthly <- GaringerOzone %>%
mutate(Year = year(Date), Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(Mean.Ozone.Conc = mean(Daily.Max.8.hour.Ozone.Concentration))

#Adding a new Date column that lists each month/year as a complete date including the first day of the month
GaringerOzone.monthly$Date <- seq(from = as.Date("2010-01-01"), to =
  as.Date("2019-12-01"), by = "month")

# 10
#Creating a time series
GaringerOzone.monthly.ts <- ts(
  GaringerOzone.monthly$Mean.Ozone.Conc, frequency = 12,
  start = c(2010, 1, 1), end = c(2019, 12, 1))

# 11
#Running a time series analysis (smk)
GaringerOzone.monthly.trend <- smk.test(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.trend)
```

```
##
## Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##
```

	S	varS	tau	z	Pr(> z)	
## Season 1:	S = 0	15	125	0.333	1.252	0.21050
## Season 2:	S = 0	-1	125	-0.022	0.000	1.00000
## Season 3:	S = 0	-4	124	-0.090	-0.269	0.78762
## Season 4:	S = 0	-17	125	-0.378	-1.431	0.15241
## Season 5:	S = 0	-15	125	-0.333	-1.252	0.21050
## Season 6:	S = 0	-17	125	-0.378	-1.431	0.15241
## Season 7:	S = 0	-11	125	-0.244	-0.894	0.37109
## Season 8:	S = 0	-7	125	-0.156	-0.537	0.59151
## Season 9:	S = 0	-5	125	-0.111	-0.358	0.72051
## Season 10:	S = 0	-13	125	-0.289	-1.073	0.28313
## Season 11:	S = 0	-13	125	-0.289	-1.073	0.28313
## Season 12:	S = 0	11	125	0.244	0.894	0.37109

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

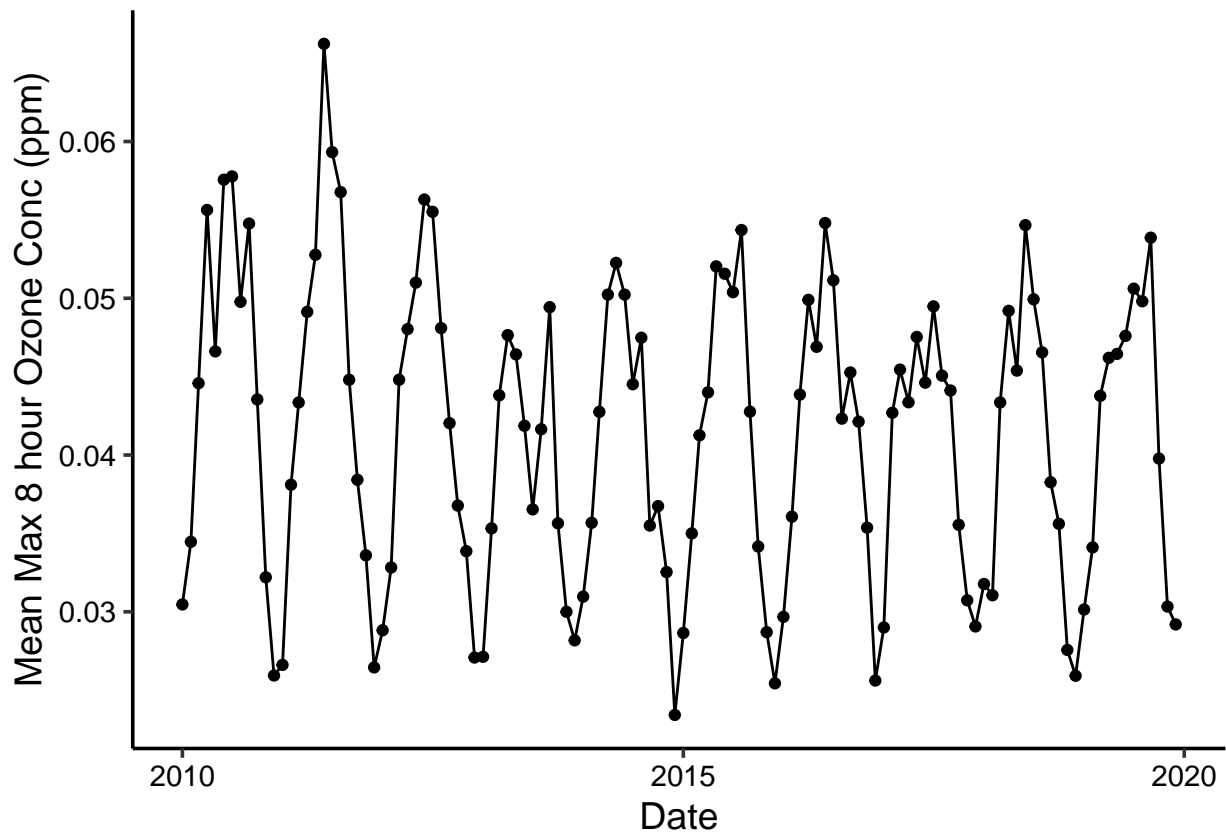
```
# 12
```

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

```
## [1] -0.0002044163
```

```
# 13
```

```
ggplot(data = GaringerOzone.monthly,  
  aes(x = Date, y = Mean.Ozone.Conc)) +  
  geom_line() +  
  geom_point() +  
  labs(y = "Mean Max 8 hour Ozone Conc (ppm)")
```



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: