Assignment 7: Time Series Analysis

David Amanfu

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

```
getwd()
```

```
## [1] "/Users/davidamanfu/Desktop/Duke MPP/Environ Data /Environmental_Data_Analytics_2022/Assignments
```

```
knitr::opts_knit$set(root.dir = "~/Desktop/Duke MPP/Environ Data /Environmental_Data_Analytics_2022/")
```

```
#1
library(agricolae)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(cowplot)
library(extrafont)
```

```
## Registering fonts with R
```

```
library(extrafontdb)
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggpubr'
```

```
## The following object is masked from 'package:cowplot':
##
##     get_legend
```

```
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
```

```
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
library(hrbrthemes)
library(Kendall)
library(lubridate)
```

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

```
## The following object is masked from 'package:cowplot':
##
##     stamp
```

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

```
library(tidyverse)
```

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

```
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::stamp()       masks cowplot::stamp()
## x lubridate::union()       masks base::union()
```

```
library(trend)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(zoo)
```

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

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

```
AmanfuTheme2 <- theme_ipsum()+
  theme(legend.position = "bottom",
        legend.key = element_rect(fill = "white", colour = "black"),legend.direction = "horizontal",
        legend.title = element_text(face = "bold"))
theme_set(AmanfuTheme2)
```

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
O3NC2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors =
O3NC2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors =
O3NC2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors =
O3NC2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors =
O3NC2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors =
O3NC2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors =
O3NC2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors =
O3NC2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors =
O3NC2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors =
O3NC2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors =


GaringerOzone <- bind_rows(O3NC2010,O3NC2011,O3NC2012,O3NC2013,O3NC2014,O3NC2015,O3NC2016,O3NC2017,O3NC
```

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

# 4
GaringerOzoneF <- 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) <-"Date"

# 6
GaringerOzone <- left_join(Days,GaringerOzoneF,by="Date")
# left_join(days,GaringerOzoneF,by = "Date")
colnames(GaringerOzone) <- c("Date", "Ozone","AQI")
```

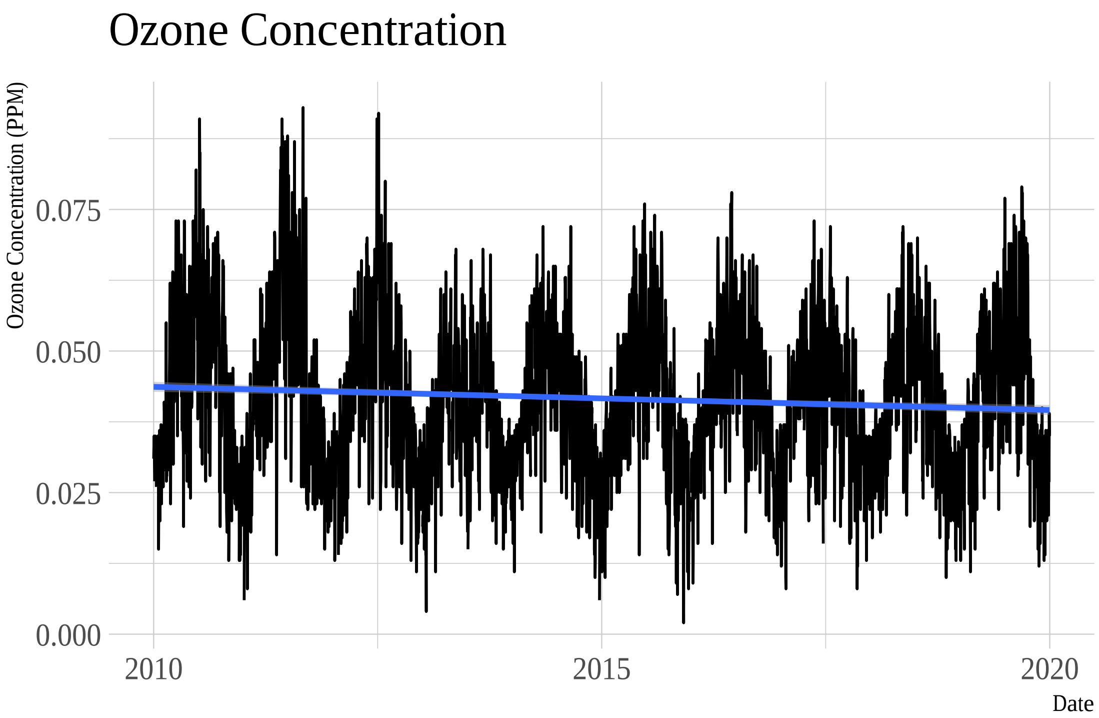
Visualize

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
Ozone_plot <- ggplot(GaringerOzone,aes(x=Date,y= Ozone))+
              labs(title= "Ozone Concentration", y="Ozone Concentration (PPM)")+
              geom_line()+
              geom_smooth(method ="lm")
Ozone_plot
```

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

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



Answer: The smoothed line suggests a decrease in Ozone in PPM over time, however slight.

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
GaringerOzoneInterp <- GaringerOzone %>%
  mutate(Ozone = na.approx(Ozone))

# Ozone_plot2 <- ggplot(GaringerOzoneInterp,aes(x=Date,y= Ozone))+
#                labs(title= "Ozone Concentration", y="Ozone Concentration (PPM)",subtitle = "With Linea
#                geom_line()+
#                geom_smooth(method ="lm")
# Ozone_plot2
# Ozone_plot
```

Answer:
Using (piecewise constant) nearest neighbor would leave us with a lot of jagged data, and a spline interpolation uses a cubic polynomial, which would also presumably work, but for simplicity sake and because we are assessing seasonal data, relying on linear interpolation to maintain seasonality probably will give us the least variant noise in our analyses.

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)

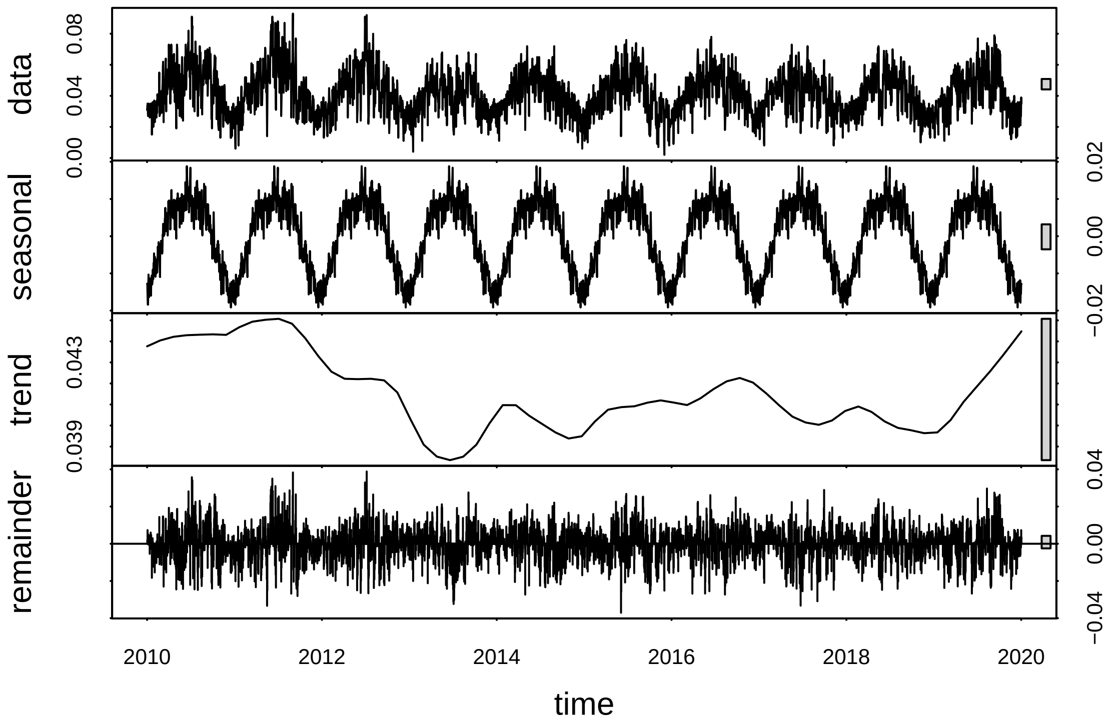
```
#9
GaringerOzone.monthly.filter <- GaringerOzoneInterp %>%
  mutate(Year = year(Date),Month = month(Date),Day = 01, .before= "Ozone")
GaringerOzone.monthly.filter <- GaringerOzone.monthly.filter %>%
  mutate(MonthDate= make_date(Year,Month,Day), .before="Ozone") %>%
  select(Date,MonthDate,Ozone,AQI)
GaringerOzone.monthly <- GaringerOzone.monthly.filter %>%
  group_by(MonthDate)%>%
  summarize(meanOzone = mean(Ozone))
```

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.

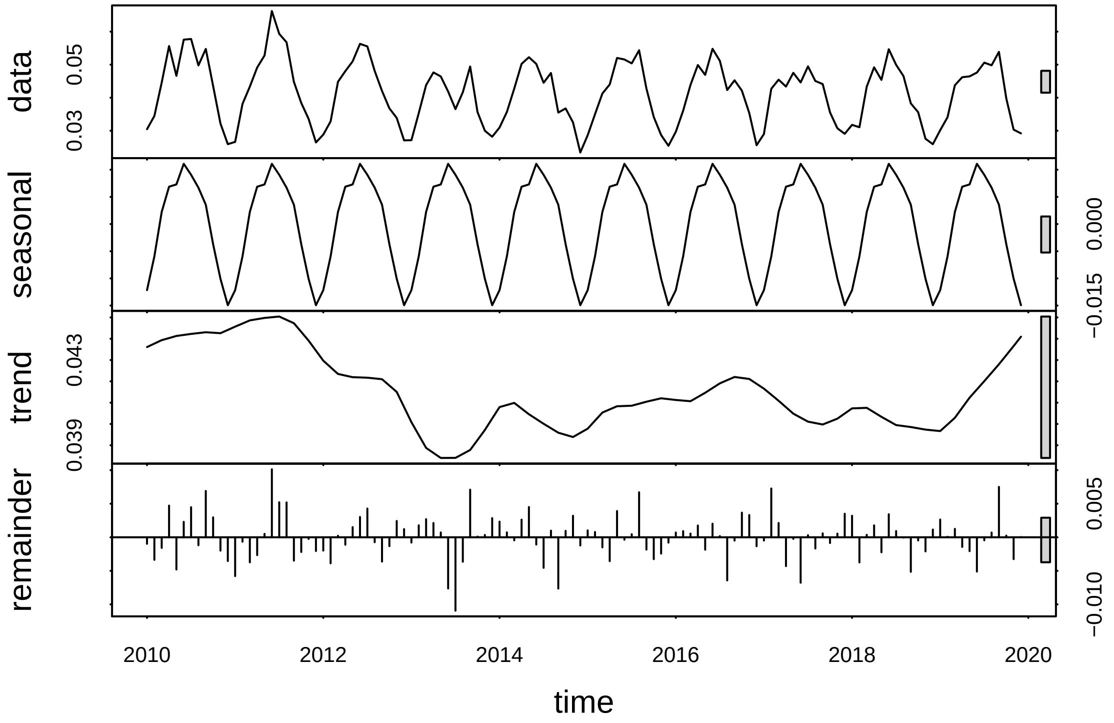
```
#10
GaringerOzone.daily.ts = ts(GaringerOzone.monthly.filter$Ozone,start=c(2010,01),frequency=365)
GaringerOzone.monthly.ts = ts(GaringerOzone.monthly$meanOzone,start=c(2010,01),frequency=12)
```

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

```
#11
Garinger_Daily <- stl(GaringerOzone.daily.ts,s.window="periodic")
Garinger_Monthly <- stl(GaringerOzone.monthly.ts,s.window="periodic")
plot(Garinger_Daily)
```



```
plot(Garinger_Monthly)
```



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_SKendall_M <- SeasonalMannKendall(GaringerOzone.monthly.ts)
Garinger_SMK_M <- smk.test(GaringerOzone.monthly.ts, alternative ="less")
# Garinger_Spearman_M <- cor.test(x=GaringerOzone.monthly.ts,y=GaringerOzone.monthly$MonthDate,method="
# Garinger_Kendall_M <- Kendall(GaringerOzone.monthly.ts)
Garinger_Dick_M <- adf.test(GaringerOzone.monthly.ts, alternative = "s")
```

```
## Warning in adf.test(GaringerOzone.monthly.ts, alternative = "s"): p-value
## smaller than printed p-value
```

```
# Garinger_Kendall_M
# summary(Garinger_Kendall_M)

Garinger_SKendall_M
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
summary(Garinger_SKendall_M)
```

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

```
Garinger_SMK_M
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.02483
## alternative hypothesis: true S is less than 0
## sample estimates:
##    S varS
##  -77 1499
```

```
summary(Garinger_SMK_M)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: less
##
## Statistics for individual seasons
##
## H0
##                        S varS    tau      z   Pr(<z)
## Season 1:   S >= 0   15  125  0.333  1.252 0.894751
## Season 2:   S >= 0   -1  125 -0.022  0.000 0.500000
## Season 3:   S >= 0   -4  124 -0.090 -0.269 0.393808
## Season 4:   S >= 0  -17  125 -0.378 -1.431 0.076203 .
## Season 5:   S >= 0  -15  125 -0.333 -1.252 0.105249
## Season 6:   S >= 0  -17  125 -0.378 -1.431 0.076203 .
## Season 7:   S >= 0  -11  125 -0.244 -0.894 0.185547
## Season 8:   S >= 0   -7  125 -0.156 -0.537 0.295753
## Season 9:   S >= 0   -5  125 -0.111 -0.358 0.360257
## Season 10:   S >= 0 -13  125 -0.289 -1.073 0.141565
## Season 11:   S >= 0 -13  125 -0.289 -1.073 0.141565
## Season 12:   S >= 0  11  125  0.244  0.894 0.814453
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
# Garinger_Spearman_M
# summary(Garinger_Spearman_M)
#
Garinger_Dick_M
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  GaringerOzone.monthly.ts
## Dickey-Fuller = -8.6413, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

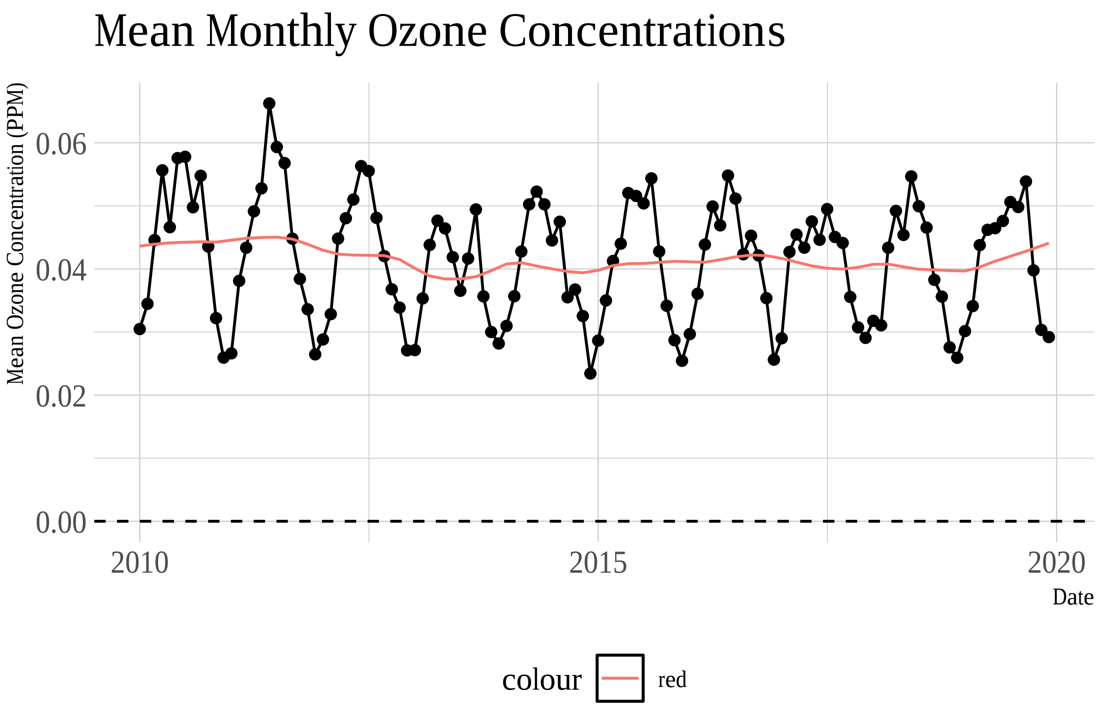
```
# summary(Garinger_Dick_M)
```

Answer:
As noted in the lesson, our data represents a time series with a seasonality component, the ozone levels generally follow an annual pattern. The three other tests commented out in the code are not applicable to seaonal data, unless we remove the seasonality from our data. It is interesting however to see differences in results between the smk.test and the SeasonalMannKendall protocols.

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
GMonthly <- as.data.frame(Garinger_Monthly$time.series[,1:3])
MonthlyOzone <- GaringerOzone.monthly %>%
  mutate(Seasonal = GMonthly$seasonal, Trend=GMonthly$trend, Remainder=GMonthly$remainder,.after= "mean
Garinger_Monthly_plot <- ggplot(MonthlyOzone,aes(x=MonthDate))+
  geom_line(aes(y=meanOzone))+
  geom_point(aes(y=meanOzone))+
  geom_line(aes(y=Trend,color="red"))+
  geom_hline(yintercept = 0, lty = 2)+
  labs(title = "Mean Monthly Ozone Concentrations",y="Mean Ozone Concentration (PPM)",x="Date")

Garinger_Monthly_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: We are plotting ozone concentrations over a period of 10 years, and attempting to establish whether there is a seasonally-adjusted trend – meaning that there is statistical deviation from an established norm or baseline over time. As such, we plot the raw data (in black, the line plot with dotted points), and the results of our time series decomposition, displaying the seasonally adjusted trend (the red line). While we observe general seasonal peaks and valleys throughout each year, when analyzed with a Sasonal Mann-Kendall (SMK) test we can see that there is year-to-year variation. Without reading the statistical outpu, the trend line, in red, seems quite static in comparison, suggesting there is no variation. The output from these tests show a different story, where statistically, there is a distinct change, and we can reject our null hypothesis that there is no difference in trend for ozone concentrations. Using the SeasonalMannKendall() function we find a p value of 0.0467, less than our 95% confidence interval, which would imply that we cannot say there is no trend. Alternatively, there is a change in ozone trend over time. The smk.test() (with an alternative hypothesis of Ha<0) function backs this up, finding a p-value of 0.024, which lends additional statistical support to our theory.

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
MonthlyOzone <- MonthlyOzone %>%
  mutate(SznAdj = MonthlyOzone$meanOzone-MonthlyOzone$Seasonal)


#16
SeasonalOzone.monthly.ts <- ts(MonthlyOzone$SznAdj,start=c(2010,01),frequency=12)
AdjGaringer_Kendall <- Kendall(x=MonthlyOzone$MonthDate, y=SeasonalOzone.monthly.ts)
AdjGaringer_Kendall
```

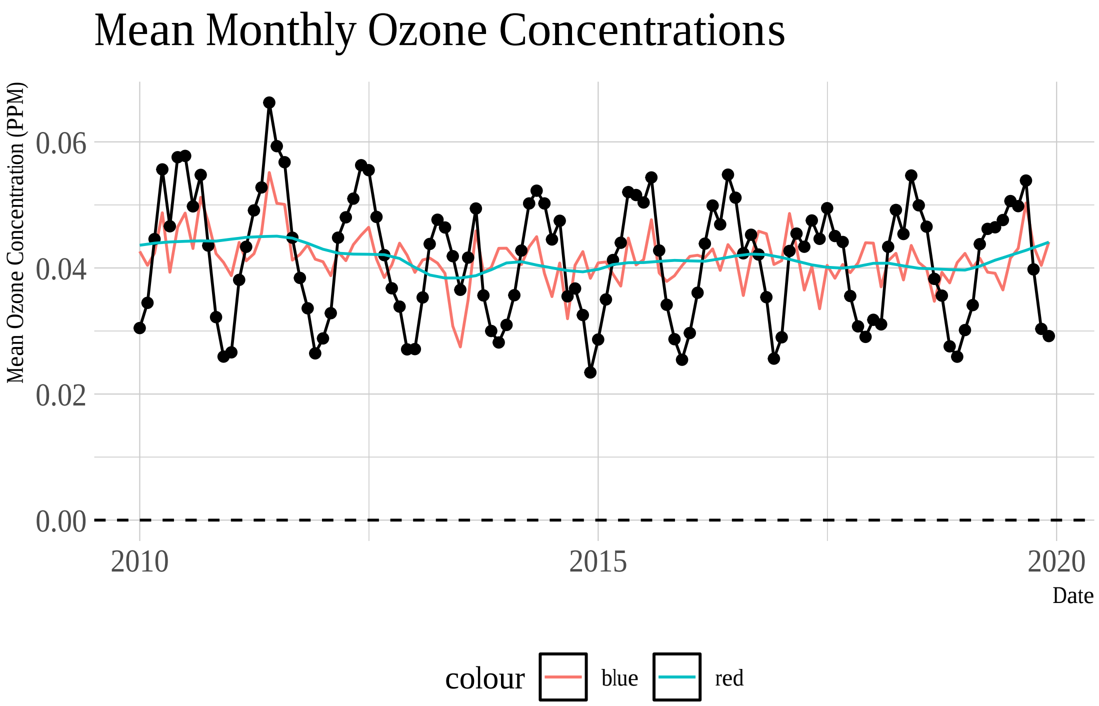
```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
summary(AdjGaringer_Kendall)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
Garinger_Monthly_plot2 <- ggplot(MonthlyOzone,aes(x=MonthDate))+
  geom_line(aes(y=SznAdj,color="blue"))+
  geom_point(aes(y=meanOzone))+
  geom_line(aes(y=meanOzone))+
  geom_line(aes(y=Trend,color="red"))+
  geom_hline(yintercept = 0, lty = 2)+
  labs(title = "Mean Monthly Ozone Concentrations",y="Mean Ozone Concentration (PPM)",x="Date")

Garinger_Monthly_plot2
```



Answer: I am not sure if I am interpreting this correctly but it appears that the seasonal adjusted trend , when analyzed through Mann Kendall, actually shows there is no monotonic trend, contrary to our seasonal Mann Kendall analysis above. Whether that's because I did this wrong or because its fundamentally different I'm a bit unclear.