Assignment 7: Time Series Analysis

Student Name

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] "C:/Users/dgp20/Documents/ENV 872/Environmental_Data_Analytics_2022/Assignments"
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.1.1
## -- 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)
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.1.3
```

```
garrettTheme <- theme(legend.position = "right", axis.text = element_text(color = "green", size = 10),
                  legend.title = element_text(size = 15, color = "orange"))
```

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
ozone10 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv', stringsAsFactors =
ozone11 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv', stringsAsFactors =
ozone12 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv', stringsAsFactors =
ozone13 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv', stringsAsFactors =
ozone14 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv', stringsAsFactors =
ozone15 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv', stringsAsFactors =
ozone16 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv', stringsAsFactors =
ozone17 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv', stringsAsFactors =
ozone18 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv', stringsAsFactors =
ozone19 <- read.csv('../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv', stringsAsFactors =

#Uploading all individually to combine into one dataset:
GaringerOzone <- rbind(ozone10, ozone11, ozone12, ozone13, ozone14, ozone15, ozone16, ozone17, ozone18,
```

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")

# 4
#Creating a processed dataset with the above columns
GarOz.processed <- select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
#Creating df "Days"
Days <- as.data.frame(seq(as.Date("2010/1/1"), as.Date("2019/12/31"), "years"))

#renaming column
colnames(Days) <- ('Date')

# 6
#Joining processed dataset and Days df
GarOz.joined <- left_join(GarOz.processed, Days)
```

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

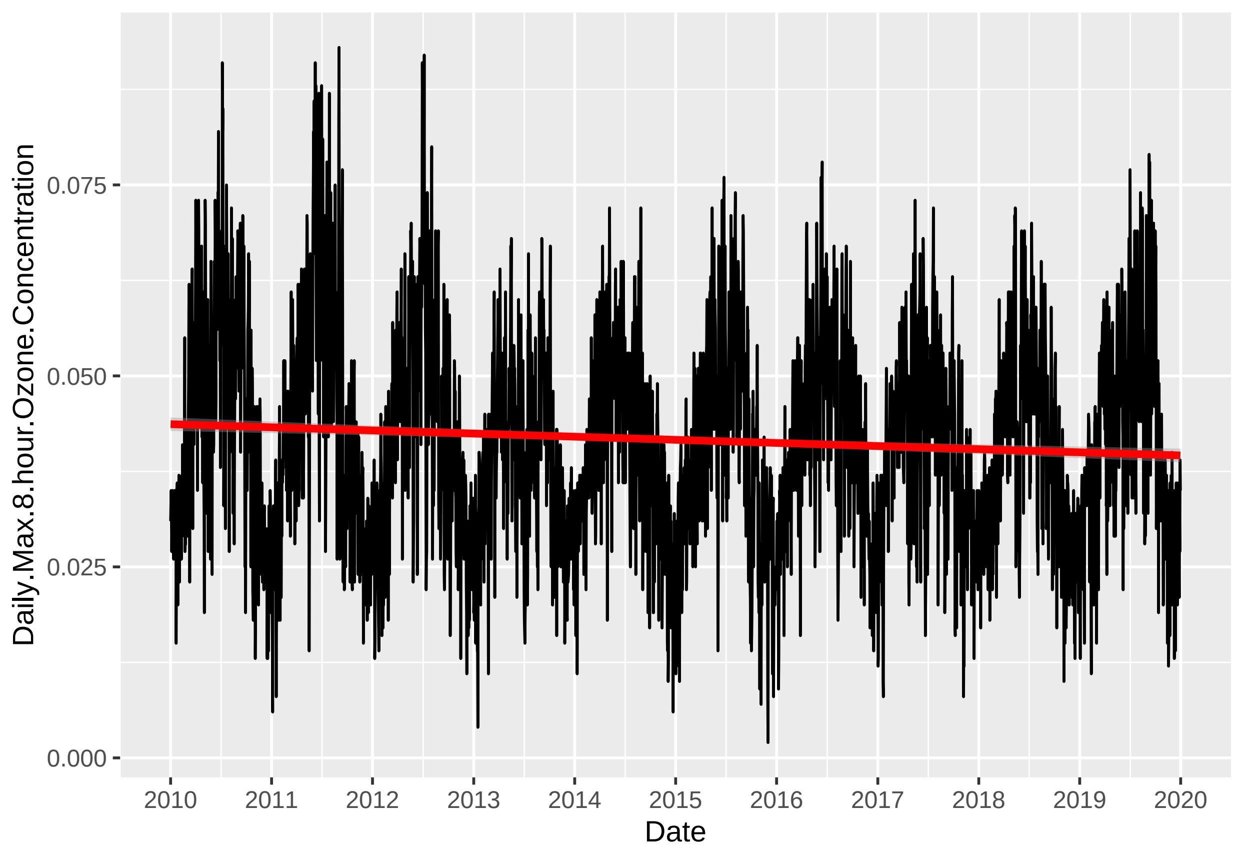
NOTE- I hope it's OK that I titled this joined dataframe something different than in the instructions above-I already had a dataframe called GaringerOzone, so I wanted to avoid confusion and keep the raw data untouched just in case.

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
#Formatting axes to just above and below min/max values, and including all years on x axis.

ggplot(GarOz.joined, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = 'lm', size = 1.35, color = 'red') +
  scale_x_date(date_labels="%Y",date_breaks  ="1 year")
```

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



```
ylim(0.00150, 0.0950)
```

```
## <ScaleContinuousPosition>
##  Range:
##  Limits: 0.0015 -- 0.095
```

> Answer: This suggests that ozone concentrations tend to fluctuate in seasonal trends- usually fairly high in the middle of the year around summer and lower in the winter. This trend is consistent year-to-year, enough so that the overall trend line is fairly flat. There is some downward slope in the smoothed trend line, suggesting that ozone has been slowly and slightly decreasing since 2010 despite these regular fluctuations. However, given the flatness of this line, it does not seem that ozone levels have not changed dramatically 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(GarOz.joined)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-03   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2015-01-04   Median :0.04100                      Median : 38.00
##  Mean   :2015-01-01   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-02   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
```

```
GarOz.interpolate <-
  GarOz.joined %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

> Answer: Given the trends observed in the line graph in the previous chunk, a linear "connect the dots" approach works better than a piecewise approach here. It may be safer to assume that ozone is constantly fluctuating, and therefore uring a line between the next and previous measurements will work better than assuming a missing measurement is equal to the one closest to it. A linear measurement better preserves the patterns we can already observe. Similarly, using a quadratic function through a splice interpolation may be overly complex and give answers that do not fit the linear model that we can observe.

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 <- GarOz.processed %>%
  mutate(Month = month(Date),
         Year = year(Date)) %>%
  mutate(Month_Year = my(paste0(Month, "-", Year))) %>%
  dplyr ::group_by(Month_Year, Month, Year) %>%
  dplyr::summarize(meanOzone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

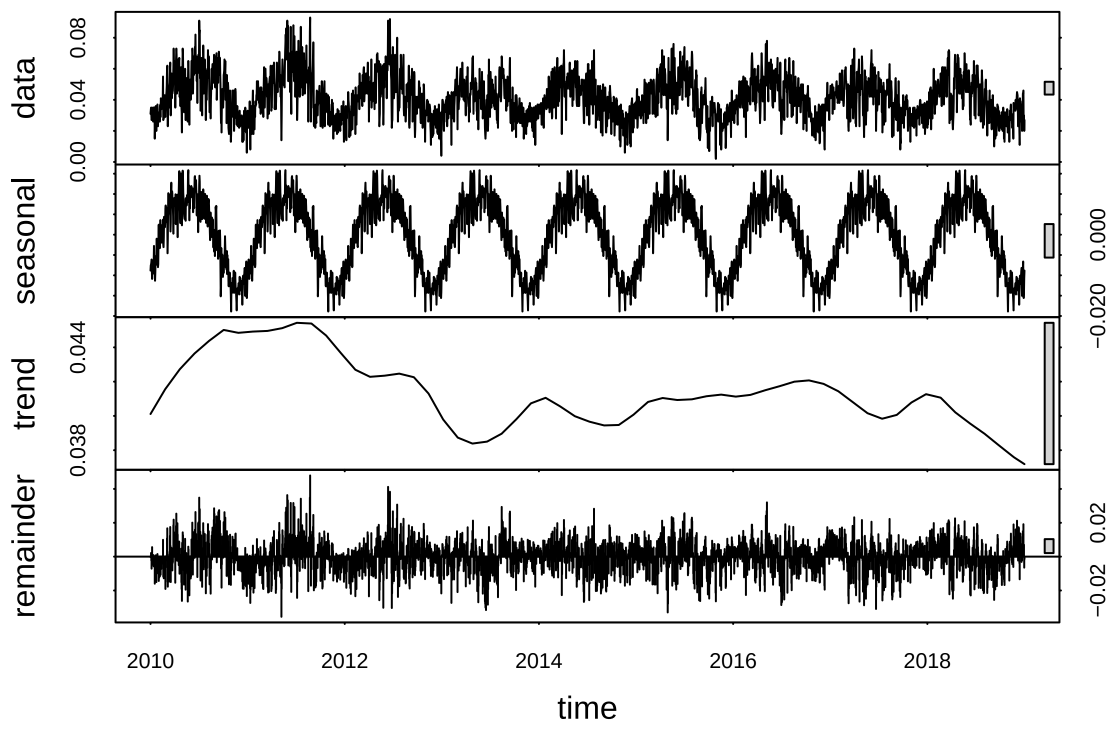
```
## `summarise()` has grouped output by 'Month_Year', 'Month'. You can override using the `.groups` argu
```

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.

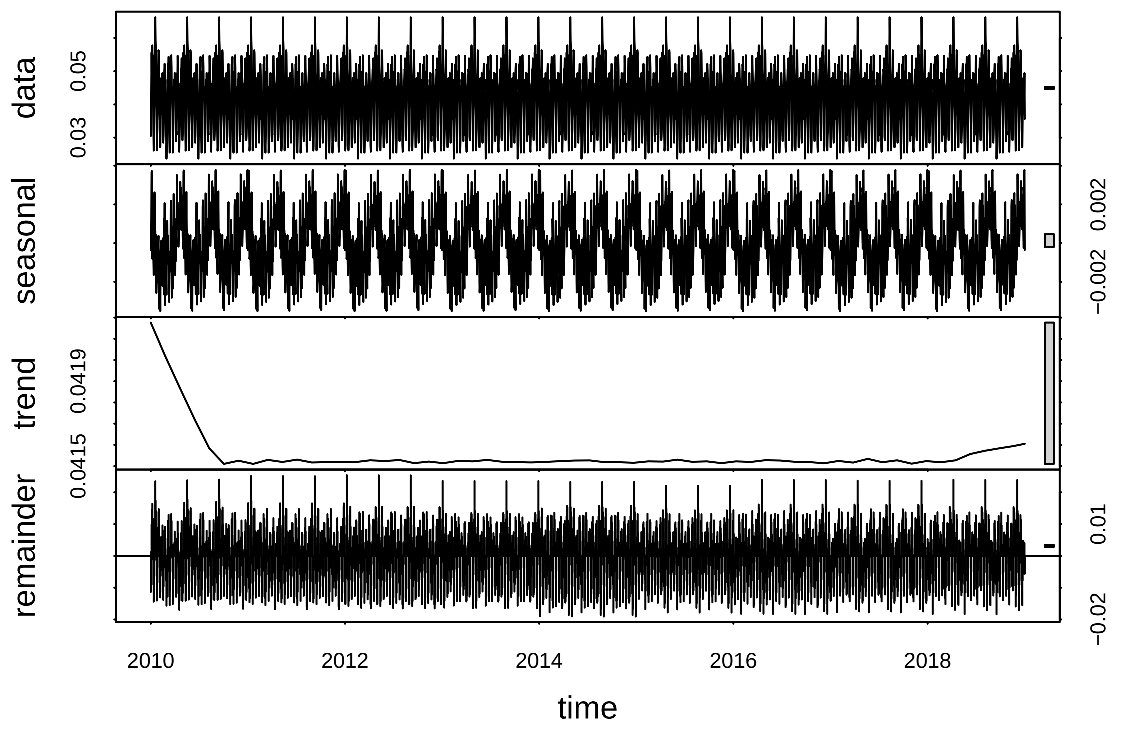
```
#10
GaringerOzone.daily.ts <- ts(GarOz.processed$Daily.Max.8.hour.Ozone.Concentration, start = c(2010, 1), 


GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanOzone, start = c(2010, 1), end = (2019), frequ
```

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

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



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



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

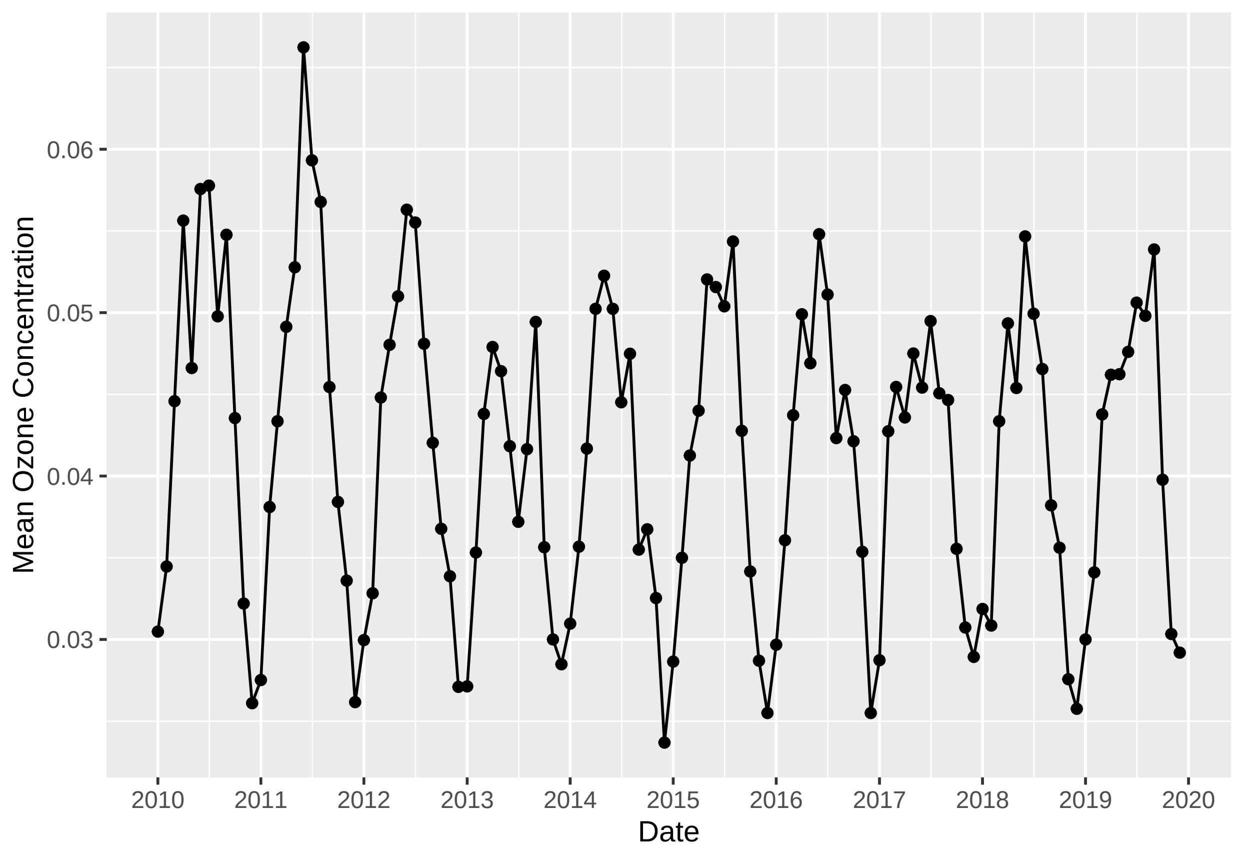
```
#12
GarOz.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
GarOz.monthly.trend
```

```
## tau = 0.0118, 2-sided pvalue =0.39787
```

Answer: Given that the monthly Ozone series is largely seasonal, the seasonal M-K analysis makes the most sense.

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
ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = meanOzone)) +
  geom_line() +
  geom_point() +
  scale_x_date(date_labels="%Y",date_breaks  ="1 year") +
  xlab("Date") + ylab("Mean Ozone Concentration")
```



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: Mean ozone concentration appears to be largely seasonal, with standard variations in seasons throughout the years. Ozone peaks in the mid-year summer months and decreases substantially towards the end of each year. Aside from a large peak in mid-2011 and some lower peaks in mid-2013 and latw 2017, ozone levels and their variations are largely similar year-to-year.

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
GaringerOzone.monthly.ts_Components <-
  as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])

GaringerOzone.monthly.ts_Components <- mutate(GaringerOzone.monthly.ts_Components,
                                              Observed = GaringerOzone.monthly.ts_Components$meanOzone,
                                              Date = GaringerOzone.monthly.ts_Components$Date)

#16
GaringerOzone.nonseason.ts <- ts(GaringerOzone.monthly.ts_Components)


GarOz.monthly.trend.nonseason <- Kendall::SeasonalMannKendall(GaringerOzone.nonseason.ts)
GarOz.monthly.trend.nonseason
```

```
## tau = 0.0207, 2-sided pvalue =0.0020885
```

> Answer: This nonseasonal analysis has a extremely high p-value- much larger than that of the analysis which was run with the seasonal component included (listed in output for #12). This suggests that the seasonal component was very strong, and contributed significantly to the observed trends in monthly Ozone concentration.