Assignment 7: Time Series Analysis

Reed Leon-Hinton

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 Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

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.

```
#1

remove(list = ls()) # clearing the environment as always, it's still a pet peeve
getwd() # showing the correct working directory.
```

```
## [1] "C:/Users/shado/Documents/Graduate School Stuff/ENVIRON 872 - Environmental Data Analytics/Envir
```

```
library(tidyverse)
library(lubridate)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.0.4
```

```
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.0.4
```

```
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.0.4
```

```r
# bringing back the theme from the last assignment
super_cool_theme_mk2 <- theme_classic(base_size = 13) +
  theme(axis.text = element_text(color = "firebrick3"),
        legend.position = "top",
        axis.title = element_text(color = "black"),
        panel.background = element_rect(fill = "white", color = "black"),
        panel.grid.major = element_line(size = 0.5, color = "gray93"))

# setting the super_cool_theme_mk2 as the default
theme_set(super_cool_theme_mk2)

#2

# Importing all the data
ozone2010 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                      stringsAsFactors = TRUE)
ozone2011 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                      stringsAsFactors = TRUE)
ozone2012 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                      stringsAsFactors = TRUE)
ozone2013 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                      stringsAsFactors = TRUE)
ozone2014 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                      stringsAsFactors = TRUE)
ozone2015 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                      stringsAsFactors = TRUE)
ozone2016 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                      stringsAsFactors = TRUE)
ozone2017 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                      stringsAsFactors = TRUE)
ozone2018 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                      stringsAsFactors = TRUE)
ozone2019 <- read.csv(file = "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                      stringsAsFactors = TRUE)

# combining the ozone sets into a single dataframe
GaringerOzone <- rbind(ozone2010, ozone2011, ozone2012, ozone2013, ozone2014,
               ozone2015, ozone2016, ozone2017, ozone2018, ozone2019)
```

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$Date <- mdy(GaringerOzone$Date)

# 4
GaringerOzone <- select(.data = GaringerOzone,
                        Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
Days <- as.data.frame(seq.Date(ymd("2010-01-01"), ymd("2019-12-31"), "days"))
colnames(Days) <- c("Date")

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

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

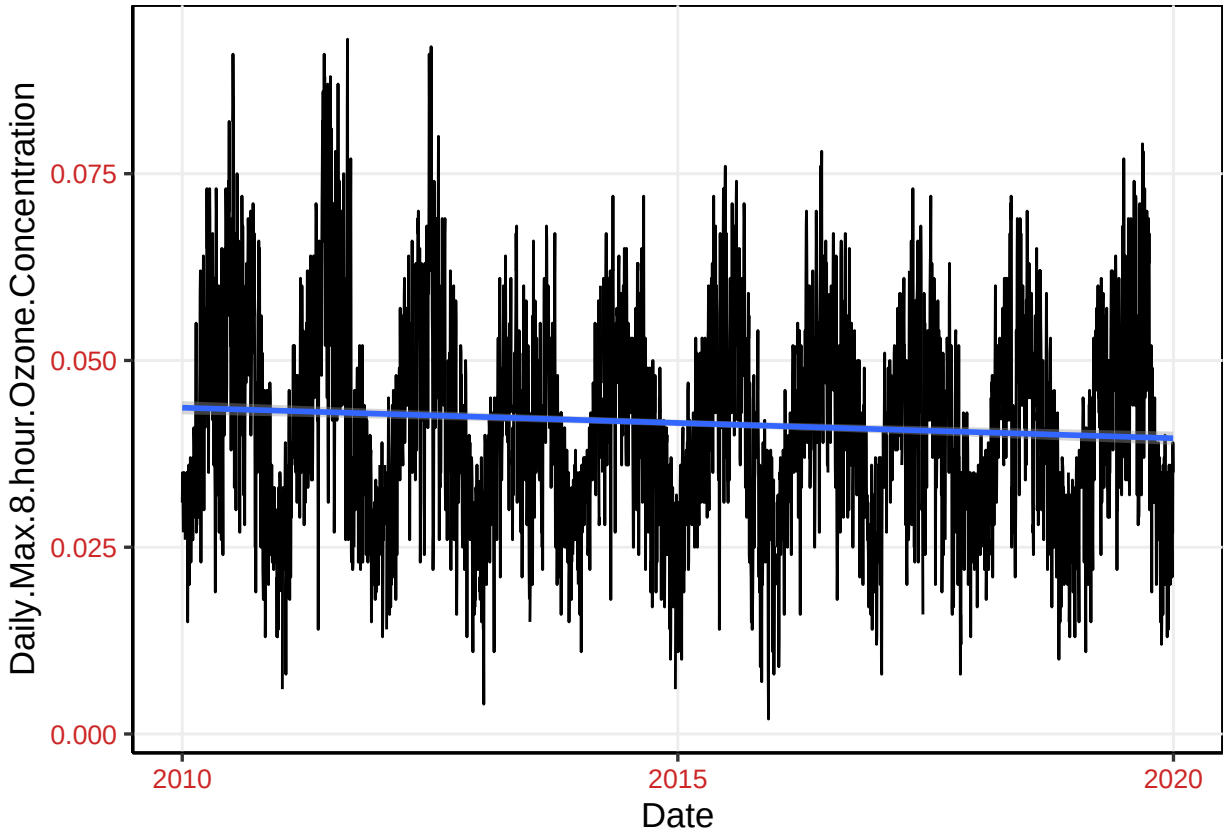
Visualize

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
timeplot <- ggplot(data = GaringerOzone) +
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_smooth(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration), method = "lm")
timeplot
```

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

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



Answer: Yes, it suggests a slow decrease in Ozone concentration over time between 2010 and 2020. You can clearly see a downward trend in the linear trend line created across the Ozone Concentration.

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
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- zoo::na.approx(GaringerOzone$Daily.Max.8.hour.Ozo
GaringerOzone$DAILY_AQI_VALUE <- zoo::na.approx(GaringerOzone$DAILY_AQI_VALUE)
```

Answer: With daily data, using the Piecewise Constant option would simply skew the trend as it would be equidistant from the measured data points on either side of the missing observation. Spline interpolation is also an inferior choice since it can also lead to greater variance than linear interpolation as the quadratic element changes the resulting filled spots more than a linear interpolation. A linear trend is also most accurate for this function as we know that the data points throughout are evenly spaced between each day's measurement point surrounding it.

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)

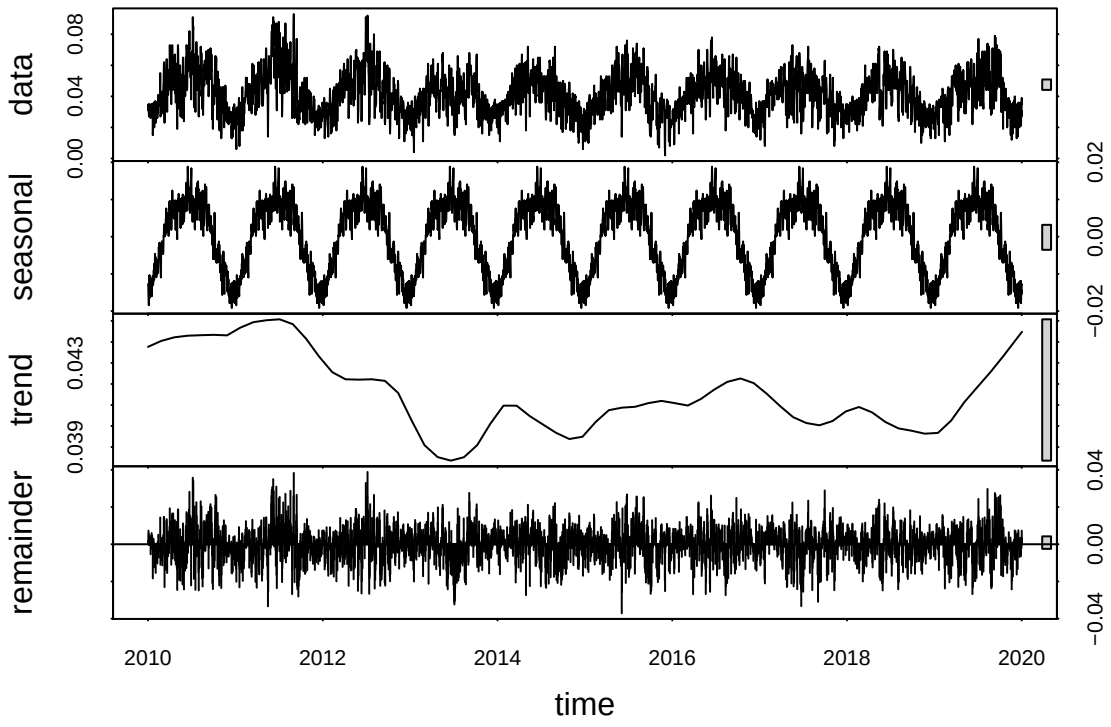
```r
#9
GaringerOzone_monthly <-
  GaringerOzone %>%
  mutate(Month = month(Date, label = TRUE, abbr = TRUE)) %>%
  mutate(Year = year(Date)) %>%
  group_by(Year, Month) %>%
  summarise_at(vars(Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE), list(name = mean)) %>%
  ungroup()

# Got to rename the Columns again
colnames(GaringerOzone_monthly) <- c("Year", "Month", "MonthlyAverage.Max.8.hour.Ozone.Concentration",
                                     "MonthlyAverage_AQI_VALUE")

# creating a date column
GaringerOzone_monthly$Day <- 1

# Merging the columns
GaringerOzone_monthly$Date <- as.Date(with(GaringerOzone_monthly, paste(Year, Month, Day, sep = "-")),"

# Eliminating the undesired columns and reordering
GaringerOzone_monthly <- select(.data = GaringerOzone_monthly,
                                Date, MonthlyAverage.Max.8.hour.Ozone.Concentration, MonthlyAverage_AQI_
```

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.

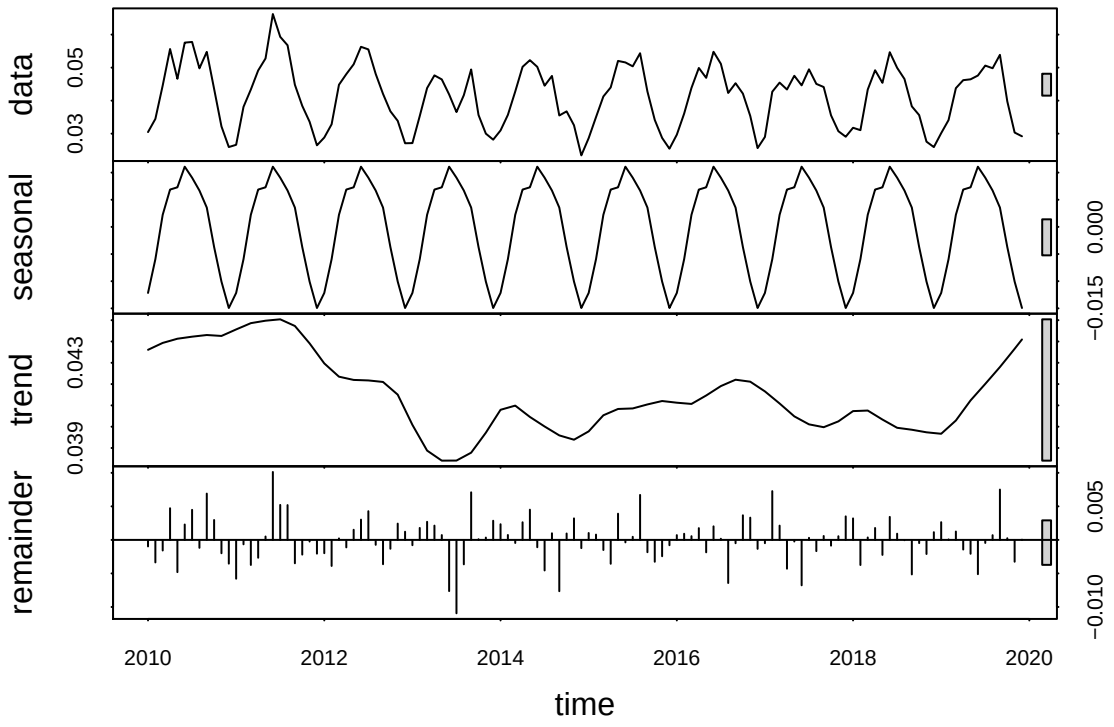
```r
#10
monthf <- month(first(GaringerOzone$Date))
yearf <- year(first(GaringerOzone$Date))

GaringerOzone_daily_ts <- ts(data = GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(yearf, monthf), frequency = 365)
GaringerOzone_monthly_ts <- ts(data = GaringerOzone_monthly$MonthlyAverage.Max.8.hour.Ozone.Concentrati
                               start = c(yearf, monthf), frequency = 12)
```

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

```r
#11
Ozone_daily_decomp <- stl(GaringerOzone_daily_ts,s.window = "periodic")
plot(Ozone_daily_decomp)
```




```
Ozone_monthly_decomp <- stl(GaringerOzone_monthly_ts,s.window = "periodic")
plot(Ozone_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
Ozone_daily_smk <- Kendall::SeasonalMannKendall(GaringerOzone_daily_ts)
Ozone_daily_smk
```

```
## tau = -0.0456, 2-sided pvalue =0.00051075
```

```
summary(Ozone_daily_smk)
```

```
## Score =  -739 , Var(Score) = 45223.67
## denominator =  16213.86
## tau = -0.0456, 2-sided pvalue =0.00051075
```

```
Ozone_monthly_smk <- Kendall::SeasonalMannKendall(GaringerOzone_monthly_ts)
Ozone_monthly_smk
```

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

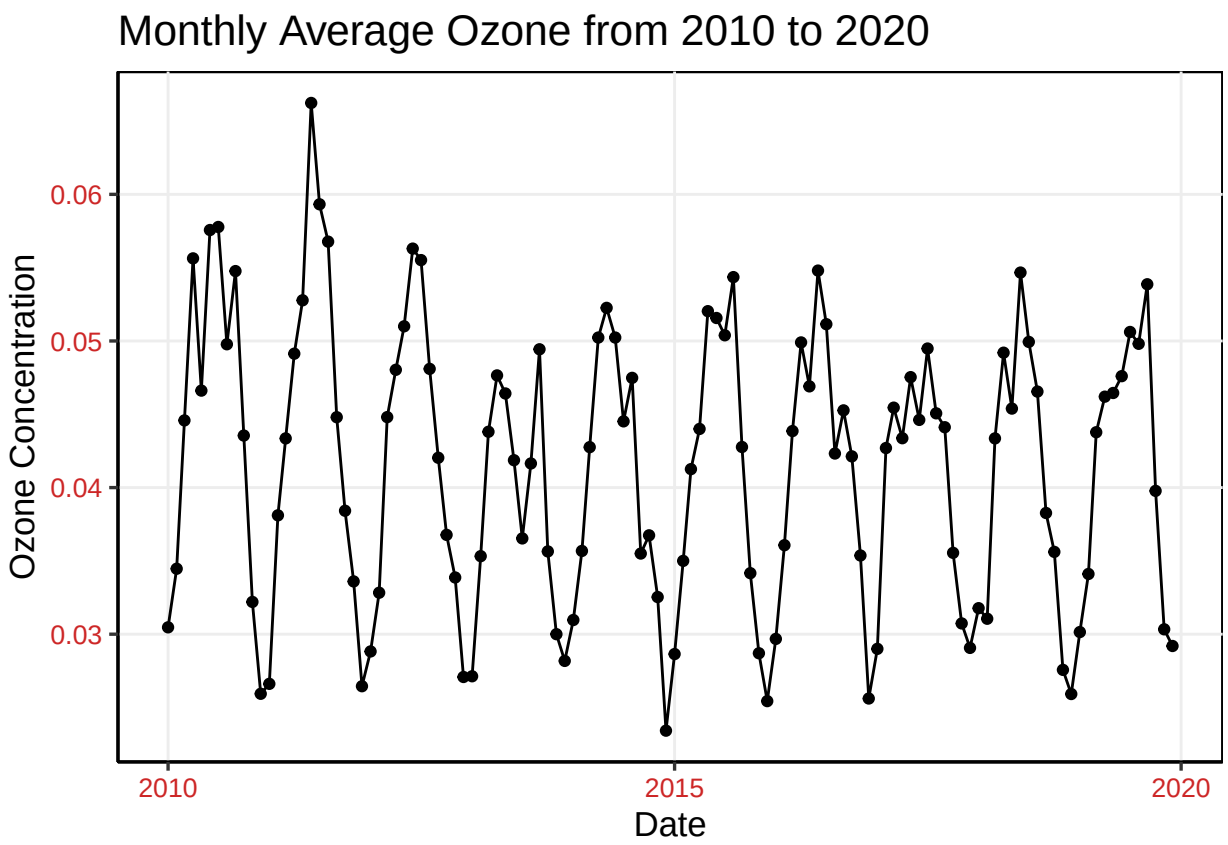
```
summary(Ozone_monthly_smk)
```

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

Answer: There is a clear seasonality to the Ozone data, both daily and monthly, with the observations in ozone concentration. The usage of the seasonal MK test allows us to account for the seasonality and works for non-parametric data.

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

summary_graph <- ggplot() +
  geom_point(data = GaringerOzone_monthly, aes(x = Date, y = MonthlyAverage.Max.8.hour.Ozone.Concentrat
  geom_line(data = GaringerOzone_monthly, aes(x = Date, y = MonthlyAverage.Max.8.hour.Ozone.Concentrati
  labs(y = "Ozone Concentration", x = "Date", title = "Monthly Average Ozone from 2010 to 2020")
summary_graph
```



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: The seasonal Mann Kendall test returned results with p-values that were less than 0.05, indicating that the null hypothesis would be rejected and that the alternative hypothesis is most likely correct, therefore should be accepted. Meaning that the data follows a trend and is non-stationary. (Daily SMK test = 0.00051, Monthly SMK test = 0.04672).

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_comp <- as.data.frame(Ozone_monthly_decomp$time.series[,1:3])
```

```r
# creating another column  of the data without the seasonal component
GaringerOzone_monthly$NonSeasonal <- GaringerOzone_monthly$MonthlyAverage.Max.8.hour.Ozone.Concentratio
  GaringerOzone_monthly_comp$seasonal

# making that into a time series
GraingerOzone_NS_TS <- ts(data = GaringerOzone_monthly$NonSeasonal,
                          start = c(yearf, monthf), frequency = 12)

#16
Ozone_monthly_mk <- MannKendall(GraingerOzone_NS_TS)
Ozone_monthly_mk
```

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

```r
summary(Ozone_monthly_mk)
```

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

Answer: The results of the Mann Kendall on the deseasonalized dataset demonstrate that the p-value for this analysis is also less than 0.05. Therefore, even when utilizing the deseasonalize then test process, it is proven that the null hypothesis is rejected and the alternative hypothesis should be accepted. This indicates that there is a trend in the dataset. (p-value = 0.00754)