ENV 790.30 - Time Series Analysis for Energy Data | Spring 2023

Assignment 4 - Due date 02/17/23

Hugh Cipparone

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github. And to do so you will need to fork our repository and link it to your RStudio.

Once you have the file open on your local machine the first thing you will do is rename the file such that it includes your first and last name (e.g., "LuanaLima_TSA_A04_Sp23.Rmd"). Then change "Student Name" on line 4 with your name.

Then you will start working through the assignment by **creating code and output** that answer each question. Be sure to use this assignment document. Your report should contain the answer to each question and any plots/tables you obtained (when applicable).

When you have completed the assignment, **Knit** the text and code into a single PDF file. Submit this pdf using Sakai.

R packages needed for this assignment: "xlsx" or "readxl", "ggplot2", "forecast","tseries", and "Kendall". Install these packages, if you haven't done yet. Do not forget to load them before running your script, since they are NOT default packages.\

```
#Load/install required package here

library(readxl)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(forecast)
```

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

```
library(tseries)
library(Kendall)
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
```

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

Questions

Consider the same data you used for A3 from the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumption The data comes from the US Energy Information and Administration and corresponds to the December 2022 Monthly Energy Review. For this assignment you will work only with the column "Total Renewable Energy Production".

```
#Importing data set - using xlsx package

data<-read_excel("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx", skip=1
  slice(-1)%>%
  select("Month", "Total Renewable Energy Production")

data$'Total Renewable Energy Production'<-as.numeric(data$`Total Renewable Energy Production`)

date<-ymd(data$Month)

data.edit<-cbind(data,date)%>%
  select(-Month)
```

Stochastic Trend and Stationarity Tests

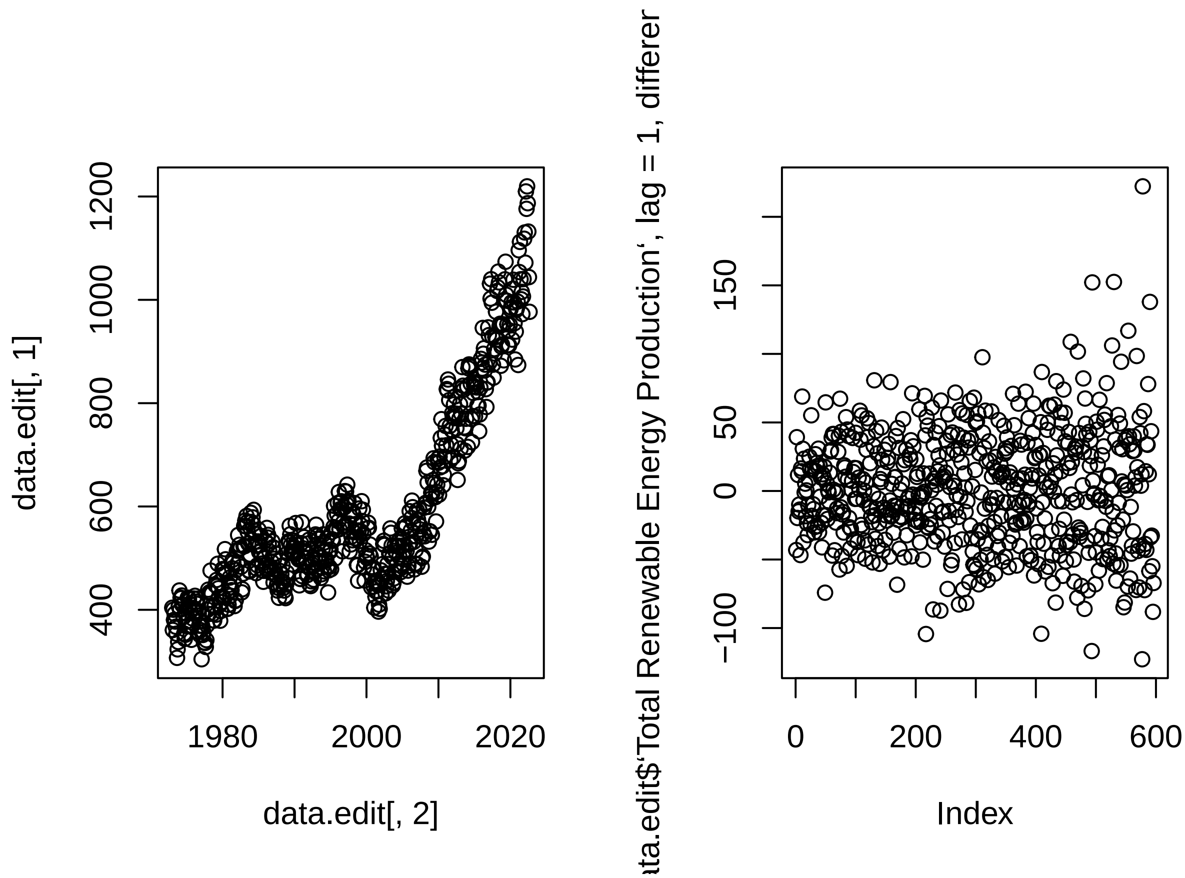
Q1

Difference the "Total Renewable Energy Production" series using function diff(). Function diff() is from package base and take three main arguments: * *x* vector containing values to be differenced; * *lag* integer indicating with lag to use; * *differences* integer indicating how many times series should be differenced.

Try differencing at lag 1 only once, i.e., make `lag=1` and `differences=1`. Plot the differenced series. Do the series still seem to have trend?

Answer: There doesn't appear to be a trend after the differencing

```
par(mfrow=c(1,2))
plot(data.edit[,2],data.edit[,1])
plot(diff(data.edit$`Total Renewable Energy Production`,lag=1,differences=1))
```




```
diff.1<-diff(data.edit$`Total Renewable Energy Production`,lag=1,differences=1)
```

Q2

Now let's compare the differenced series with the detrended series you calculated on A3. In other words, for the "Total Renewable Energy Production" compare the differenced series from Q1 with the series you detrended in A3 using linear regression. (Hint: Just copy and paste part of your code for A3)

Copy and paste part of your code for A3 where you compute regression for Total Energy Production and the detrended Total Energy Production

```
t<-c(1:nrow(data.edit))

linear_trend_model=lm(data.edit[,1]~t)
summary(linear_trend_model)
```

```
##
## Call:
## lm(formula = data.edit[, 1] ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -238.75  -61.85    8.59   64.48  352.27
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 312.2475     8.4902   36.78   <2e-16 ***
```

```
## t              0.9362     0.0246   38.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 103.6 on 595 degrees of freedom
## Multiple R-squared:  0.7088, Adjusted R-squared:  0.7083
## F-statistic:  1448 on 1 and 595 DF,  p-value: < 2.2e-16
```

```
beta0=as.numeric(linear_trend_model$coefficients[1])
beta1=as.numeric(linear_trend_model$coefficients[2])

#Renewables
detrend_data <- data.edit[,1]-(beta0+beta1*t)
```

Q3

Create a data frame with 4 columns: month, original series, detrended by Regression Series and differenced series. Make sure you properly name all columns. Also note that the differenced series will have only 584 rows because you loose the first observation when differencing. Therefore, you need to remove the first observations for the original series and the detrended by regression series to build the new data frame.

```
#Data frame - remember to not include January 1973

data.edit.1<-data.edit %>%
  slice(-1)

detrend_data.1<-detrend_data[2:597]

data.df<-data.frame(data.edit.1$date, data.edit.1$`Total Renewable Energy Production`, diff.1, detrend_d
  rename(Date=data.edit.1.date, "Normal Time Series"=data.edit.1..Total.Renewable.Energy.Production., "
```

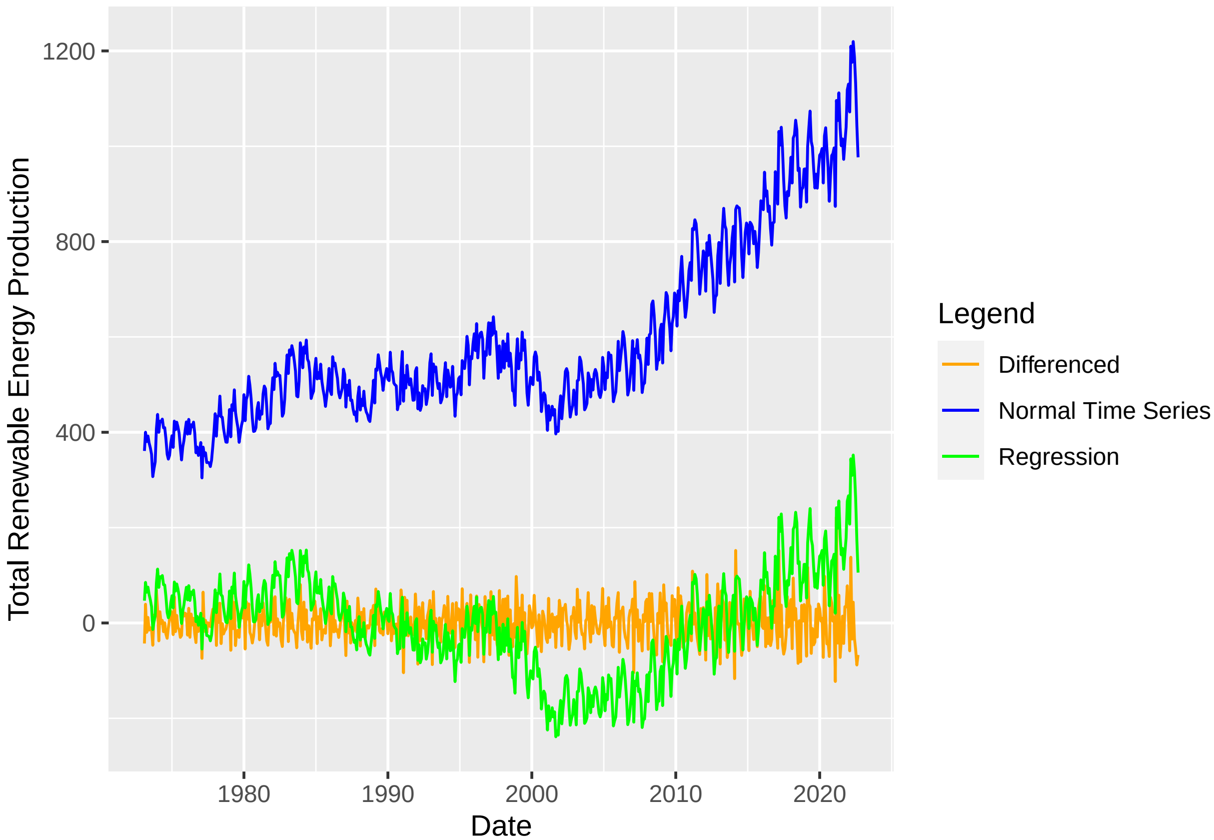
Q4

Using ggplot() create a line plot that shows the three series together. Make sure you add a legend to the plot.

```
#Use ggplot

colors<-c("Normal Time Series"="blue", "Differenced"="orange", "Regression"="green")

ggplot(data.df, aes(x=Date))+
  geom_line(aes(y=data.df[,2], color="Normal Time Series"))+
  geom_line(aes(y=data.df[,3], color="Differenced"))+
  geom_line(aes(y=data.df[,4], color="Regression"))+
  labs(
    x="Date",
    y="Total Renewable Energy Production",
    color="Legend")+
  scale_color_manual(values=colors)
```



Q5

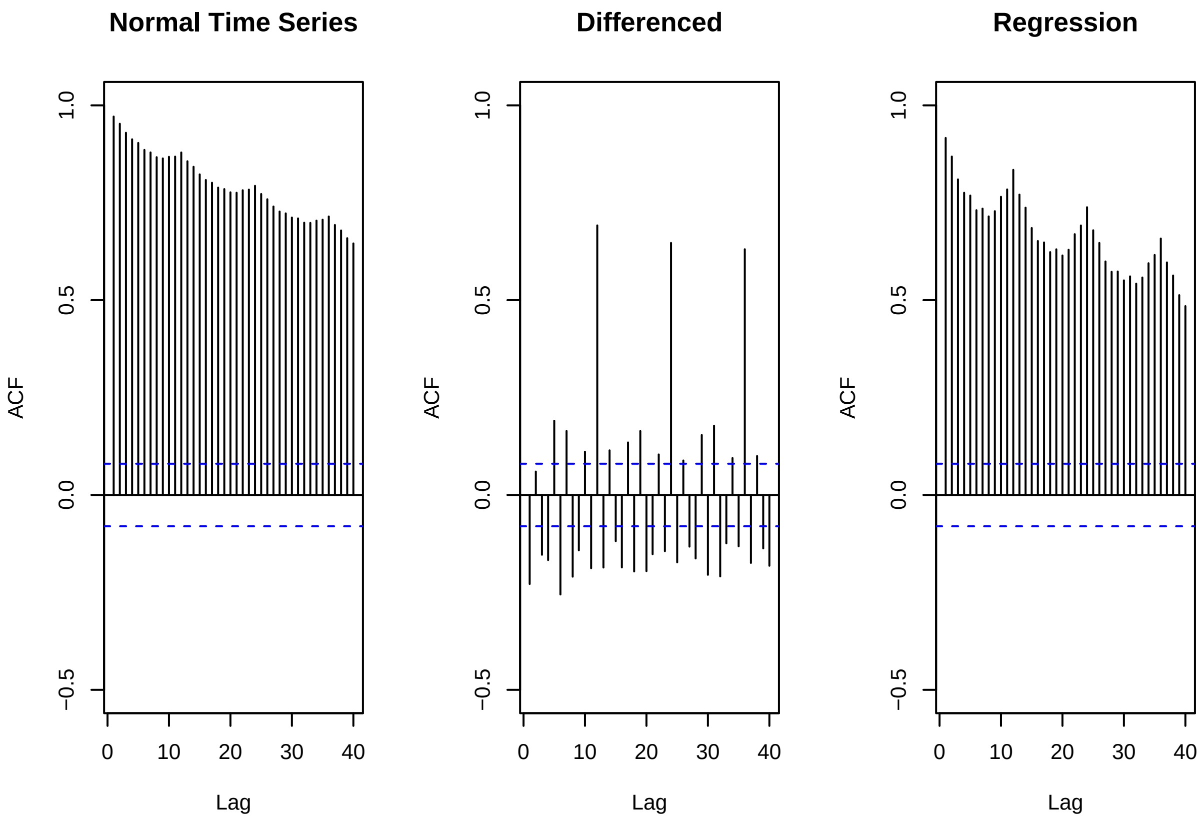
Plot the ACF for the three series and compare the plots. Add the argument `ylim=c(-0.5,1)` to the Acf() function to make sure all three y axis have the same limits. Which method do you think was more efficient in eliminating the trend? The linear regression or differencing?

Answer: The differenced was the most efficient in eliminating the trend, because the level of autocorrelation between lags is much less significant than the regression.

```
#Compare ACFs

column.name=colnames(data.df)

par(mfrow=c(1,3))
for(i in 2:4){
  Acf(data.df[,i],lag.max=40, main=column.name[i], ylim=c(-0.5,1))
}
```



Q6

Compute the Seasonal Mann-Kendall and ADF Test for the original "Total Renewable Energy Production" series. Ask R to print the results. Interpret the results for both test. Whats the conclusion from the Seasonal Mann Kendall test? What's the conclusion for the ADF test? Do they match what you observed in Q2? Recall that having a unit root means the series has a stochastic trend. And when a series has stochastic trend we need to use a different procedure to remove the trend.

Interpretation: The p-value for our Seasonal Mann-Kendall is far less than 0.05 - meaning that we reject the null hypothesis that the series is stationary. The p-value for our ADF test is far greater than 0.05, meaning we accept the null hypothesis that the series contains a unit root (aka is stochastic). This matches the finding in Q5 (which is where we stated whether to use the regression versus the differencing process to remove the trend) because the regression is not going to work well with a stochastic process and - low and behold - we observed that the differenced works better than the regression because - as we now know - this is a stochastic process.

```
renewable.ts<-ts(data.edit[,1], frequency = 12, start=c(1973,1))

smk<-SeasonalMannKendall(renewable.ts)
print(smk)
```

```
## tau = 0.727, 2-sided pvalue =< 2.22e-16
```

```
summary(smk)
```

```
## Score =  10577 , Var(Score) = 169001
## denominator =  14553
## tau = 0.727, 2-sided pvalue =< 2.22e-16
```

```r
adf<-adf.test(renewable.ts, alternative=c("stationary"))
print(adf)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  renewable.ts
## Dickey-Fuller = -1.2055, Lag order = 8, p-value = 0.9056
## alternative hypothesis: stationary
```

Q7

Aggregate the original "Total Renewable Energy Production" series by year. You can use the same procedure we used in class. Store series in a matrix where rows represent months and columns represent years. And then take the columns mean using function colMeans(). Recall the goal is the remove the seasonal variation from the series to check for trend.

```r
#Delete 2022 because only 9 observations instead of full 12
renewable.ts.yearly<-renewable.ts[1:588]
date.yearly<-date[1:588]

iHP=1
#Group data in yearly steps instances
renewable_data_matrix <- matrix(renewable.ts.yearly, byrow=FALSE,nrow=12)
renewable_data_yearly <- colMeans(renewable_data_matrix)

library(dplyr)  #move this to package chunk later
my_year <- c(year(first(date.yearly)):year(last(date.yearly)))

renewable_data_new_yearly <- data.frame(my_year, renewable_data_yearly)
```

Q8

Apply the Mann Kendal, Spearman correlation rank test and ADF. Are the results from the test in agreement with the test results for the non-aggregated series, i.e., results for Q6?

Answer:They are in agreement! Mann Kendall and Spearman reject their null (that there is not trend) and the ADF accepts the null (that there is a stochastic process)

```r
renewable.ts.yearly<-ts(renewable_data_new_yearly[,2], frequency = 12, start=c(1973,1))

mk<-MannKendall(renewable.ts.yearly)
print(mk)
```

```
## tau = 0.735, 2-sided pvalue =< 2.22e-16
```

```r
cor.test(my_year, renewable.ts.yearly, method=c("spearman"))
```

```
##
##  Spearman's rank correlation rho
##
## data:  my_year and renewable.ts.yearly
## S = 2548, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##  rho
## 0.87
```

```r
adf<-adf.test(renewable.ts.yearly, alternative=c("stationary"))
print(adf)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  renewable.ts.yearly
## Dickey-Fuller = -0.68508, Lag order = 3, p-value = 0.9654
## alternative hypothesis: stationary
```