ENV 790.30 - Time Series Analysis for Energy Data | Spring 2024

Assignment 4 - Due date 02/12/24

Jaimie Wargo

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.

```r
#Load/install required package here
library(readxl)
library(ggplot2)
library(forecast)
```

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

```r
library(tseries)
library(Kendall)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(here)
```

```
## here() starts at C:/Users/jaimi/OneDrive/Documents/Duke/Spring_2024/TSA_Sp24
```

```
library(cowplot)
```

```
##
## Attaching package: ’cowplot’
##
## The following object is masked from ’package:lubridate’:
##
##     stamp
```

Questions

Consider the same data you used for A3 from the spreadsheet “Table_10.1_Renewable_Energy_Production_and_Consumpti The data comes from the US Energy Information and Administration and corresponds to the January 2021 Monthly Energy Review. For this assignment you will work only with the column “Total Renewable Energy Production”.

```
#Importing data set - using readxl package
raw_data <- read_excel(path=here('Data',
          'Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx'),
          skip = 12, sheet="Monthly Data",col_names=FALSE)
```

```
## New names:
## * ‘‘ -> ‘...1‘
## * ‘‘ -> ‘...2‘
## * ‘‘ -> ‘...3‘
## * ‘‘ -> ‘...4‘
## * ‘‘ -> ‘...5‘
## * ‘‘ -> ‘...6‘
## * ‘‘ -> ‘...7‘
## * ‘‘ -> ‘...8‘
## * ‘‘ -> ‘...9‘
## * ‘‘ -> ‘...10‘
## * ‘‘ -> ‘...11‘
## * ‘‘ -> ‘...12‘
## * ‘‘ -> ‘...13‘
## * ‘‘ -> ‘...14‘
```

```
#Extract the column names from row 11
read_col_names <- read_excel(path=here('Data',
          'Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx'),
            skip = 10,n_max = 1, sheet="Monthly Data",col_names=FALSE)
```

```
## New names:
## * ‘‘ -> ‘...1‘
## * ‘‘ -> ‘...2‘
## * ‘‘ -> ‘...3‘
## * ‘‘ -> ‘...4‘
## * ‘‘ -> ‘...5‘
## * ‘‘ -> ‘...6‘
## * ‘‘ -> ‘...7‘
## * ‘‘ -> ‘...8‘
## * ‘‘ -> ‘...9‘
## * ‘‘ -> ‘...10‘
## * ‘‘ -> ‘...11‘
## * ‘‘ -> ‘...12‘
## * ‘‘ -> ‘...13‘
## * ‘‘ -> ‘...14‘
```

```
colnames(raw_data) <- read_col_names

#Data frame
energy_df <- data.frame(raw_data[,c('Month', 'Total Biomass Energy Consumption',
                         'Total Renewable Energy Consumption',
                         'Hydroelectric Power Consumption')])

nospace_colnames <- c('date', 'biomass', 'total_renew', 'hydro')
colnames(energy_df) <- nospace_colnames

#Time series structure
energy_ts <- ts(energy_df[,2:4], start=c(1973, 1), frequency = 12)
```

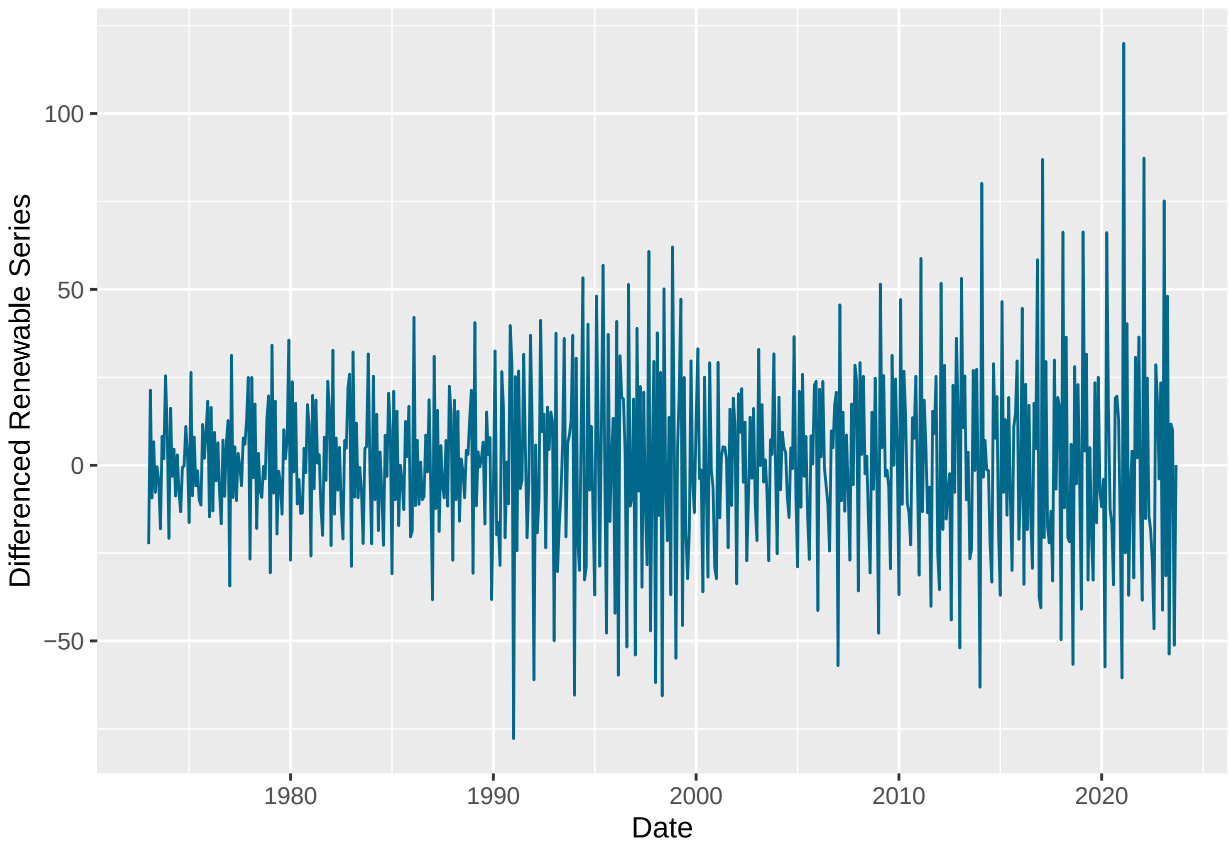
Stochastic Trend and Stationarity Tests

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 Does the series still seem to have trend?

```
energy_df$diff_renewable <- c(diff(energy_df$total_renew, lag=1, differences = 1),0)

ggplot(energy_df, aes(x=date, y=diff_renewable))+
  geom_line(color='deepskyblue4')+
  labs(x='Date', y='Differenced Renewable Series')
```



The series does not appear to have a trend anymore, as the observations are centered on 0 with no increasing or decreasing.

Q2

Copy and paste part of your code for A3 where you run the regression for Total Renewable Energy Production and subtract that from the original series. This should be the code for Q3 and Q4. make sure you use the same name for you time series object that you had in A3.

```
nobs <- nrow(energy_df)
t <- 1:nobs

renewable_trend <- lm(energy_df$total_renew~t)
summary(renewable_trend)
```

```
##
## Call:
## lm(formula = energy_df$total_renew ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -144.72  -33.24   11.30   39.21  135.99
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 184.40303    4.69129   39.31   <2e-16 ***
## t             0.68542    0.01333   51.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.81 on 607 degrees of freedom
## Multiple R-squared:  0.8134, Adjusted R-squared:  0.8131
## F-statistic:  2645 on 1 and 607 DF,  p-value: < 2.2e-16
```

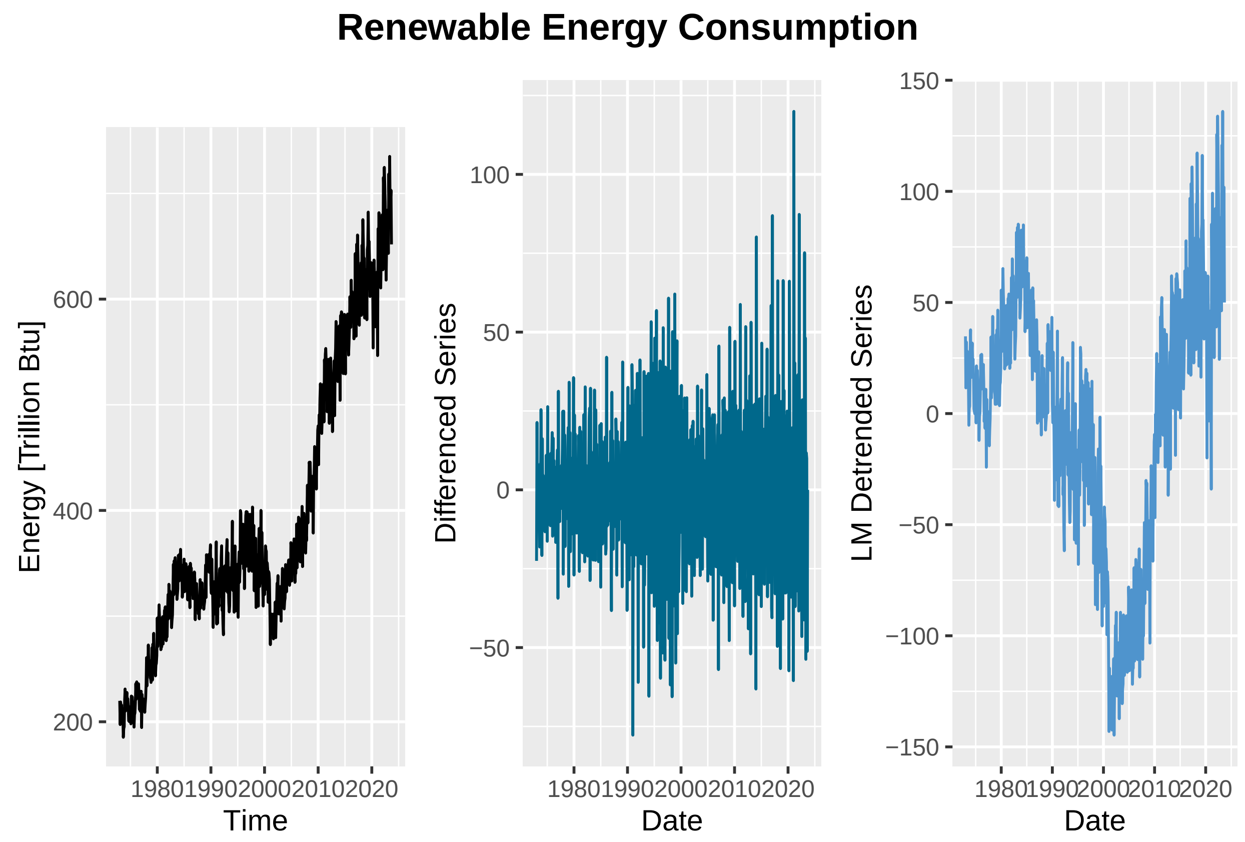
```
tre_beta0 <- renewable_trend$coefficients[1]
tre_beta1 <- renewable_trend$coefficients[2]

tre_detrend <- energy_df$total_renew - (tre_beta0+tre_beta1*t)

tre_df <- data.frame('date'=energy_df$date,
                         'observed'=energy_df[,3],
                         'detrend'=tre_detrend)
```

Q3

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 Q2 using linear regression.

Using autoplot() + autolayer() create a plot that shows the three series together. Make sure your plot has a legend. The easiest way to do it is by adding the `series=` argument to each autoplot and autolayer function. Look at the key for A03 for an example.

```
#RenewableEnergyProduction
ts_plot<-autoplot(energy_ts[,2])+
  ylab("Energy [Trillion Btu]")+
  ggtitle("")

diff_plot<-ggplot(energy_df, aes(x=date, y=diff_renewable))+
  geom_line(color='deepskyblue4')+
  labs(x='Date', y='Differenced Series')

detrended_plot<-ggplot(tre_df, aes(x=date, y=detrend))+
  geom_line(color='steelblue3')+
  labs(x='Date', y='LM Detrended Series')

#Addingtitle
plot_row<-plot_grid(ts_plot,diff_plot,detrended_plot,nrow=1,ncol=3)

title<-ggdraw()+
  draw_label("Renewable Energy Consumption",fontface='bold')

plot_grid(title,plot_row,nrow=2,ncol=1,rel_heights=c(0.1,1))
```



Q4

Plot the ACF for the three series and compare the plots. Add the argument `ylim=c(-0.5,1)` to the autoplot() or Acf() function - whichever you are using to generate the plots - 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?

```
tre_acfplot <- autoplot(Acf(energy_ts[,2], lag.max=40, plot=F),
                        main="Renewable Energy ACF")
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: ‘main‘
```

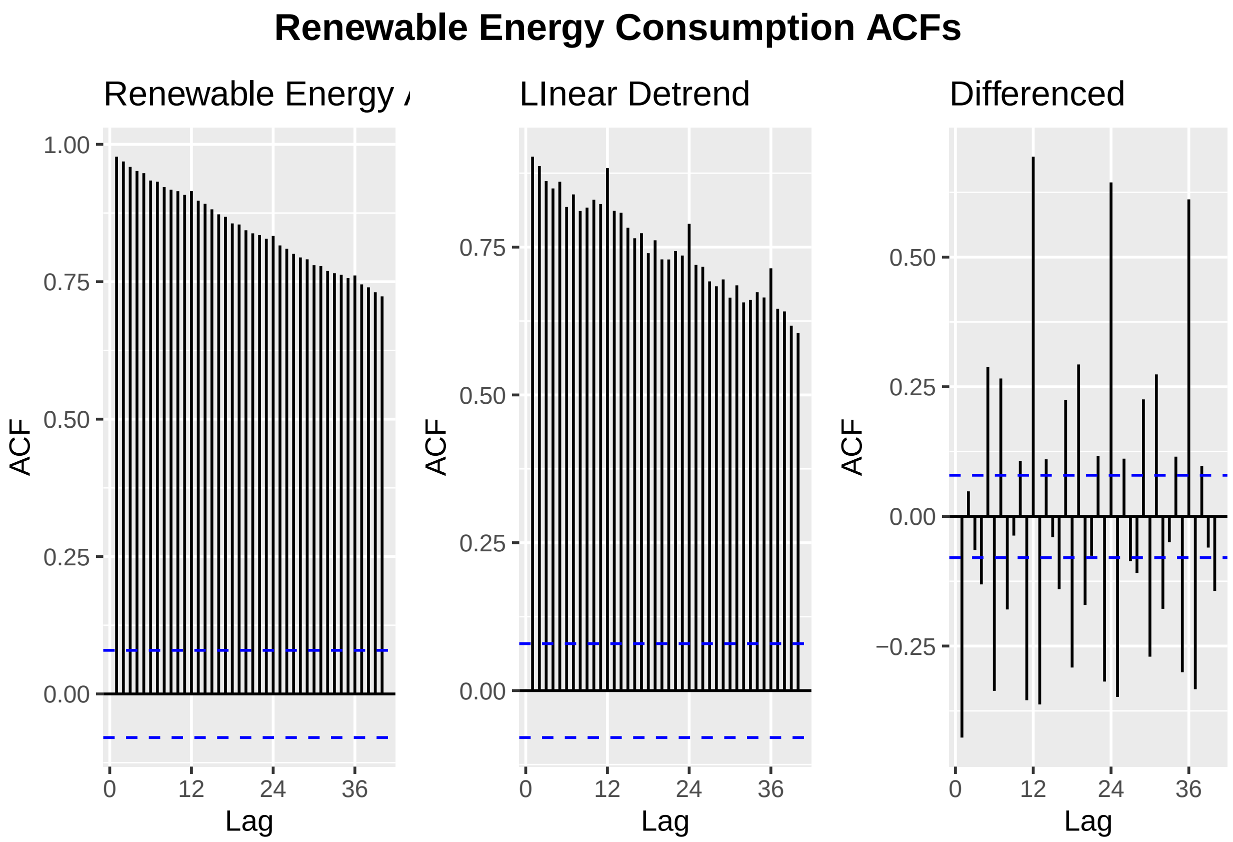
```
tre_ts_detrend <- ts(tre_df$detrend, frequency = 12, start(1973,1))

detrend_acf<- autoplot(Acf(tre_ts_detrend, lag.max=40, plot=F),
                       main="LInear Detrend")
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: ‘main‘
```

```
tre_diff_ts <- ts(energy_df$diff_renewable, frequency = 12, start(1973,1))

diff_acf <- autoplot(Acf(tre_diff_ts, lag.max=40,plot=F),
                     main="Differenced")
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: ‘main‘
```

```
plot_row<-plot_grid(tre_acfplot, detrend_acf, diff_acf,nrow=1,ncol=3)

title<-ggdraw()+
  draw_label("Renewable Energy Consumption ACFs",fontface='bold')

plot_grid(title,plot_row,nrow=2,ncol=1,rel_heights=c(0.1,1))
```



The differenced series ACF shows that this method was the most efficient in differencing the trend. The values now do not exceed ~0.7. It does seem like there may be some seasonality remaining with strong lags at 12, 24, and 36 months, but overall it succeeded at removing the trend.

Q5

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. What is 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.

```
tre_mk <- SeasonalMannKendall(energy_ts[,2])

print("Results for Seasonal Mann Kendall")
```

```
## [1] "Results for Seasonal Mann Kendall"
```

```
print(summary(tre_mk))
```

```
## Score =  11903 , Var(Score) = 179299
## denominator =  15149.5
## tau = 0.786, 2-sided pvalue =< 2.22e-16
## NULL
```

```
tre_adf <- adf.test(energy_ts[,2], alternative = c('stationary'))

print("Results for ADF Test")
```

```
## [1] "Results for ADF Test"
```

```
print(tre_adf)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  energy_ts[, 2]
## Dickey-Fuller = -1.2501, Lag order = 8, p-value = 0.8957
## alternative hypothesis: stationary
```

The ADF test gave a p-value of 0.8957, meaning that we reject the alternate hypothesis of stationarity. This indicates that the renewable energy series has a unit root, meaning the series needs to be detrended with differencing rather than the function. The Mann Kendall test returned a p-value less than 0.05, signaling that there is a trend present in this series.

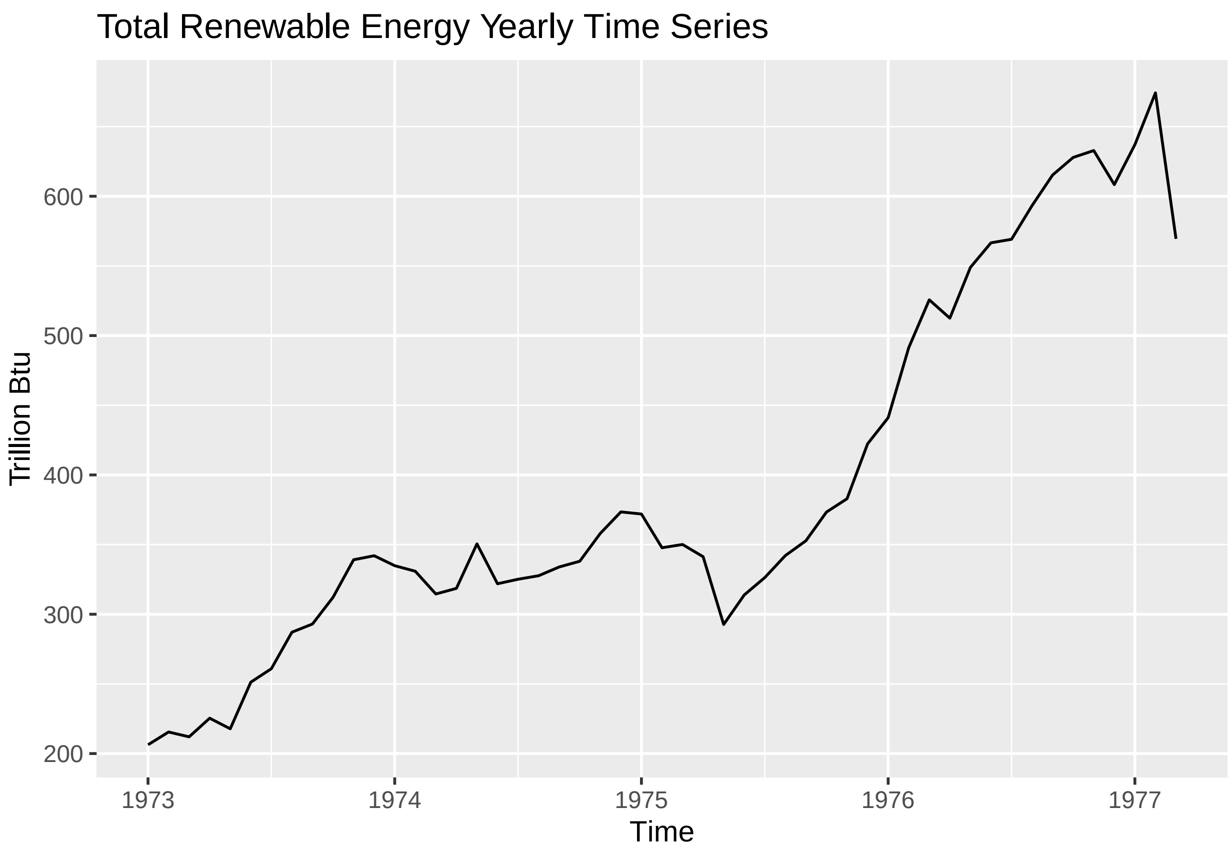
Q6

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. Convert the accumulates yearly series into a time series object and plot the series using autoplot().

```
energy_data_matrix <- matrix(energy_ts[,2],byrow=FALSE,nrow=12)
```

```
## Warning in matrix(energy_ts[, 2], byrow = FALSE, nrow = 12): data length [609]
## is not a sub-multiple or multiple of the number of rows [12]
```

```
energy_data_yearly <- colMeans(energy_data_matrix)
my_year <- c(year(first(energy_df$date)):year(last(energy_df$date)))

energy_data_yearly <- data.frame(my_year, energy_data_yearly)

ts_yearly <- ts(energy_data_yearly[,2], start=c(1973,1), frequency = 12)

autoplot(ts_yearly)+
  ggtitle("Total Renewable Energy Yearly Time Series")+
  ylab("Trillion Btu")
```



Q7

Apply the Mann Kendall, Spearman correlation rank test and ADF. Are the results from the test in agreement with the test results for the monthly series, i.e., results for Q6?

```
tre_mk <- SeasonalMannKendall(ts_yearly)

print("Results for Seasonal Mann Kendall")
```

```
## [1] "Results for Seasonal Mann Kendall"
```

```
print(summary(tre_mk))
```

```
## Score =  80 , Var(Score) = 128
## denominator =  84
## tau = 0.952, 2-sided pvalue =1.5374e-12
## NULL
```

```
tre_adf <- adf.test(ts_yearly, alternative = c('stationary'))

print("Results for ADF Test")
```

```
## [1] "Results for ADF Test"
```

```
print(tre_adf)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_yearly
## Dickey-Fuller = -2.1483, Lag order = 3, p-value = 0.5147
## alternative hypothesis: stationary
```

```
sp_tre <- cor.test(ts_yearly,my_year,method="spearman")

print("Results for Spearman Test")
```

```
## [1] "Results for Spearman Test"
```

```
print(sp_tre)
```

```
##
##  Spearman's rank correlation rho
##
## data:  ts_yearly and my_year
## S = 1850, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9162896
```

The results are in agreement with the procedure for Q5. While the ADF p-value did decrease from 0.8957 to 0.5147, this value is still not statistically significant, meaning this is still a stochastic series. The Mann Kendall test provided the same results as well, indicating a trend. The Spearman test also indicated a strong trend, with a rho of 0.92 and p-value below 0.05.