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

Assignment 4 - Due date 02/12/24

Samantha Pace

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

#install.packages("readxl")
#install.packages("ggplot2")
#install.packages("forecast")
#install.packages("tseries")
#install.packages("Kendall")
#install.packages("dplyr")
#install.packages("lubridate")

library(readxl)
library(ggplot2)
library(forecast)
```

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

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

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

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

```
getwd()
```

```
## [1] "C:/Users/saman/OneDrive/Desktop/Duke Spring 24/GITHUB/TSA_Sp24"
```

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 using read.xlsx
energy_data <-
  read_excel("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`
```

```
# Getting column names from row 11
read_col_names <-
  read_excel(path="./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`
```

```
energy_data <-
  read_excel(path="./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`
```

```
# inputting correct column names
colnames(energy_data) <- read_col_names
head(energy_data)
```

```
## # A tibble: 6 x 14
##   Month               `Wood Energy Production` `Biofuels Production`
##   <dttm>                                 <dbl> <chr>
## 1 1973-01-01 00:00:00                     130. Not Available
## 2 1973-02-01 00:00:00                     117. Not Available
## 3 1973-03-01 00:00:00                     130. Not Available
## 4 1973-04-01 00:00:00                     125. Not Available
```

```
## 5 1973-05-01 00:00:00                    130. Not Available
## 6 1973-06-01 00:00:00                    125. Not Available
## # i 11 more variables: 'Total Biomass Energy Production' <dbl>,
## #   'Total Renewable Energy Production' <dbl>,
## #   'Hydroelectric Power Consumption' <dbl>,
## #   'Geothermal Energy Consumption' <dbl>, 'Solar Energy Consumption' <chr>,
## #   'Wind Energy Consumption' <chr>, 'Wood Energy Consumption' <dbl>,
## #   'Waste Energy Consumption' <dbl>, 'Biofuels Consumption' <chr>,
## #   'Total Biomass Energy Consumption' <dbl>, ...
```

```r
# selecting the 3 columns needed
energy_data <- energy_data %>% select(1, 5)
help(select)
```

```
## starting httpd help server ... done
```

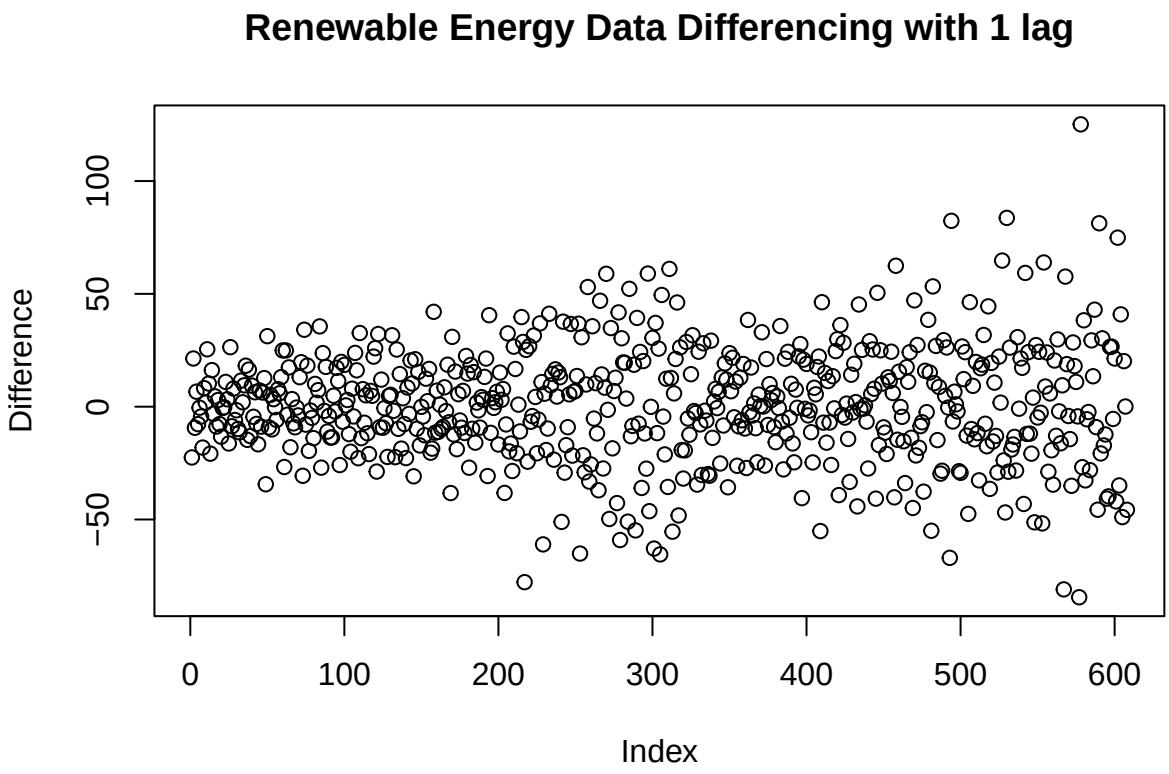
Stochastic Trend and Stationarity Tests

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?

```r
# differencing
REP.diff <- diff(energy_data$`Total Renewable Energy Production`,
     lag=1,
     differences = 1)

#plotting differenced data
plot(REP.diff,
     ylab = "Difference",
     main = "Renewable Energy Data Differencing with 1 lag")
```



Answer: The differenced series does not appear to have a discernible trend.

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.

```
# from A3:
# storing vectors
nobs <-nrow(energy_data)
t <- 1:nobs

# fitting linear trend to Renewable Energy Production Data
REP_linear_trend <- lm(energy_data$`Total Renewable Energy Production`~t)
summary(REP_linear_trend)
```

```
##
## Call:
## lm(formula = energy_data$`Total Renewable Energy Production` ~
##     t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -148.27  -35.63   11.58   41.51  144.27
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 180.98940    4.90151   36.92   <2e-16 ***
## t             0.70404    0.01392   50.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.41 on 607 degrees of freedom
## Multiple R-squared:  0.8081, Adjusted R-squared:  0.8078
## F-statistic:  2557 on 1 and 607 DF,  p-value: < 2.2e-16
```

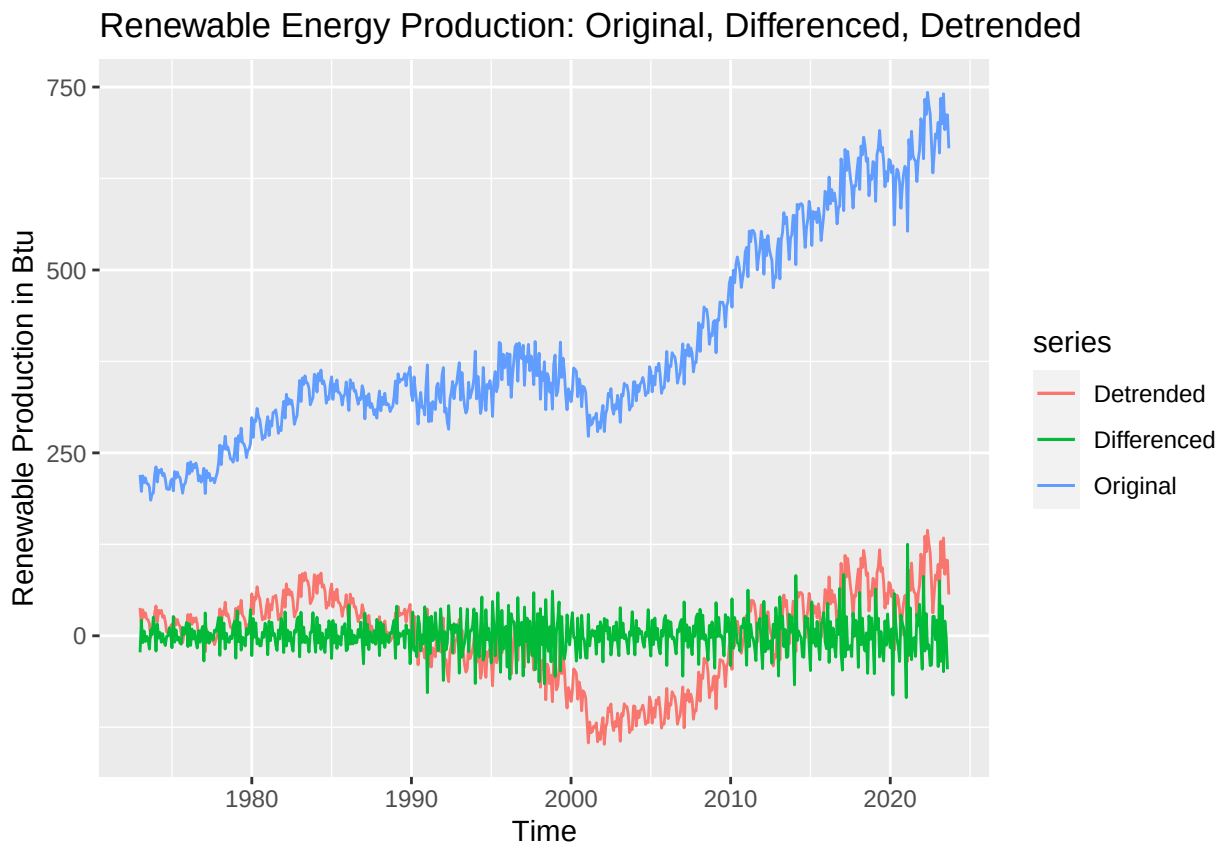
```r
# saving the regression coefficients for REP
REP_beta0 <- REP_linear_trend$coefficients[1]
REP_beta1 <- REP_linear_trend$coefficients[2]

# Creating REP detrended series
REP_detrend <- energy_data[,2] - (REP_beta0 + REP_beta1*t)

colnames(REP_detrend)[1]<-"detrend"

REP_detrend_df <- data.frame("Month" = energy_data$Month,
                             "Observed" = energy_data[,2],
                             "Detrend" = REP_detrend)

colnames(REP_detrend_df)[2] <- "observed"
```

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.

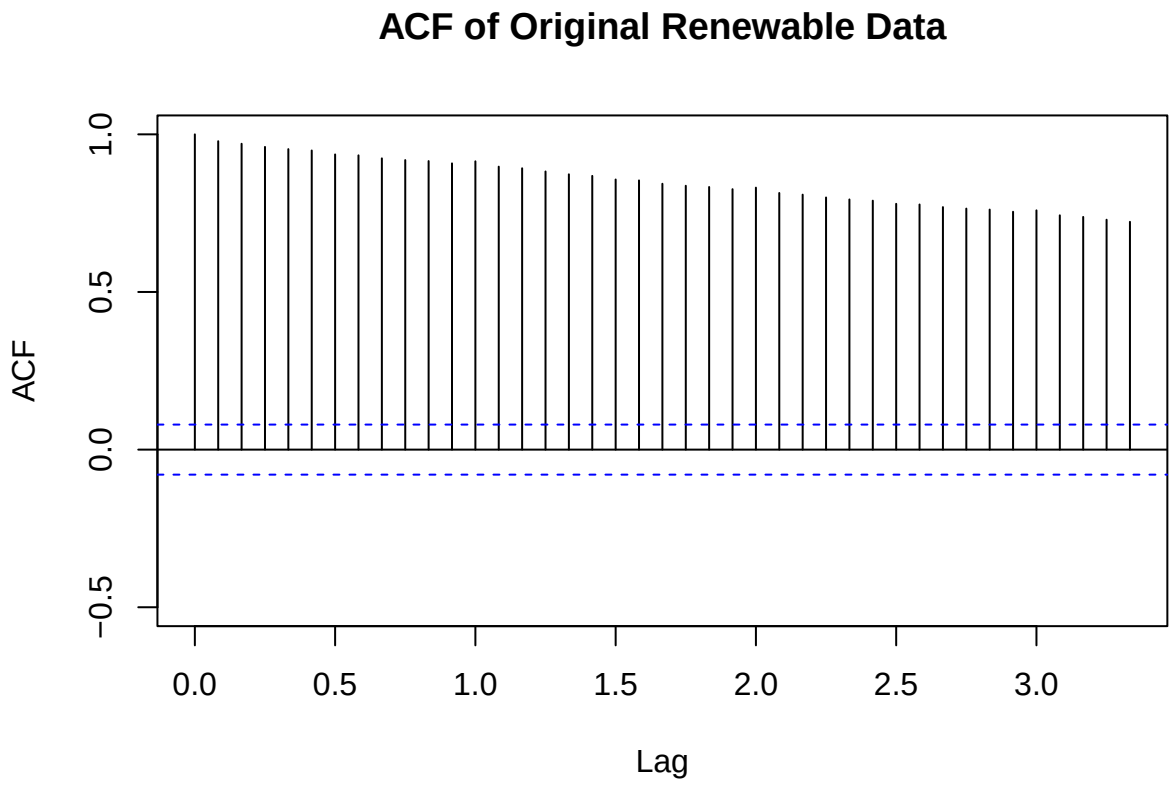
```r
# creating time series objects for each of the series
REP.original.ts <- ts(energy_data[,2], frequency = 12,
                      start = c(1973, 1))

REP.detrend.ts <- ts(REP_detrend,
                     frequency = 12,
                     start = c(1973, 1))

REP.differenced.ts <- ts(REP.diff, frequency = 12,
                         start = c(1973, 1))

# plotting original, detrended, and differenced
autoplot(REP.original.ts, series = "Original", ylab = "Renewable Production in Btu") +
  autolayer(REP.detrend.ts, series = "Detrended") +
  autolayer(REP.differenced.ts, series = "Differenced") +
  ggtitle("Renewable Energy Production: Original, Differenced, Detrended")
```



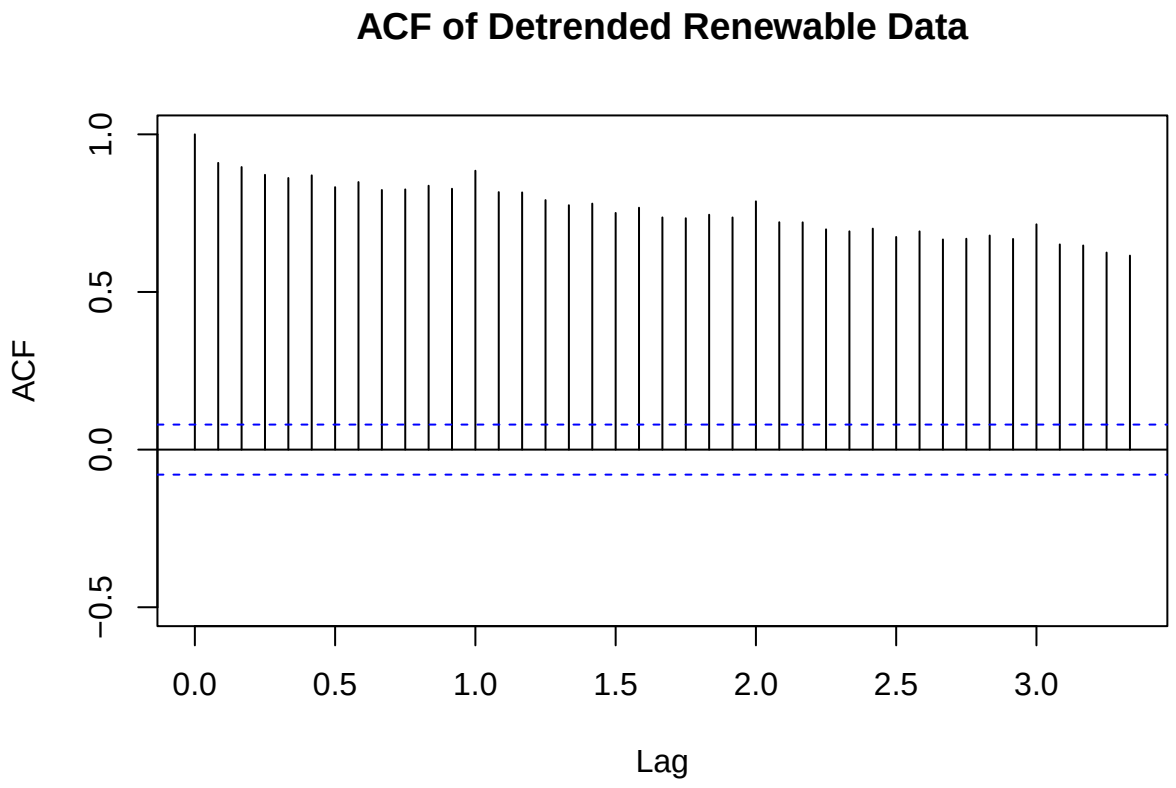
Q4

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?

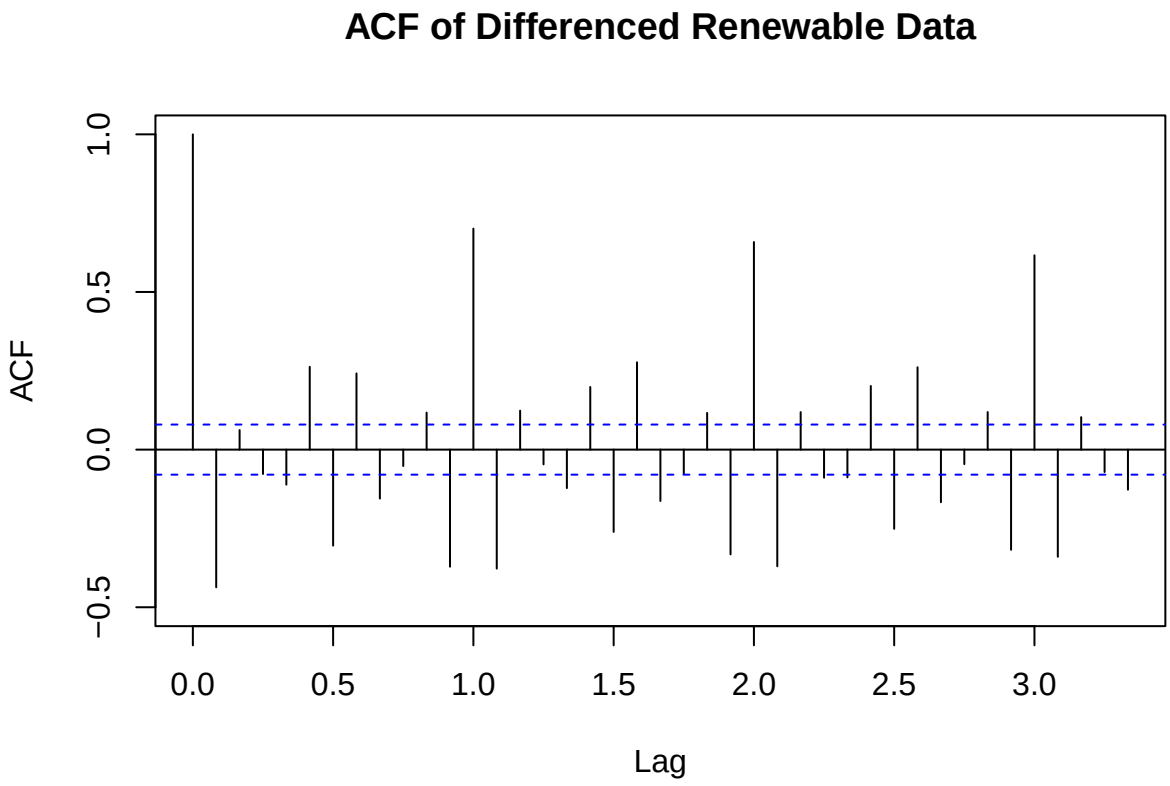
```
# creating autoplots for comparison

REP_original_ACF <- autoplot(acf(REP.original.ts,
                                 lag.max = 40,
                                 ylab = "ACF",
                                 main = "ACF of Original Renewable Data",
                                 ylim = c(-0.5, 1)))
```




```
REP_detrend_ACF <- autoplot(acf(REP.detrend.ts,
                                lag.max = 40,
                                ylab = "ACF",
                                main = "ACF of Detrended Renewable Data",
                                ylim = c(-0.5, 1)))
```




```
REP_diff_ACF <- autoplot(acf(REP.differenced.ts,
                             lag.max = 40,
                             ylab = "ACF",
                             main = "ACF of Differenced Renewable Data",
                             ylim = c(-0.5, 1)))
```



Answer: I think the differencing was more effective at eliminating the trend because the differencing operation reduced the magnitude of the ACF considerably across the lags and showed that there is a still a periodic higher magnitude ACF which may be indicative of a seasonal component. The linear regression model marginally lowered the magnitude of the ACFs of the lags, so it didn't do much to eliminate 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.

```
# Seasonal Mann-Kendall Test
SMKtest <- SeasonalMannKendall(REP.original.ts)
print("Results of Seasonal Mann Kendall on Original data")
```

```
## [1] "Results of Seasonal Mann Kendall on Original data"
```

```
print(summary(SMKtest))
```

```
## Score =  11865 , Var(Score) = 179299
## denominator =  15149.5
## tau = 0.783, 2-sided pvalue =< 2.22e-16
## NULL
```

```r
# ADF test: unit root/stochastic trend
print("Results for ADF test")
```

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

```r
print(adf.test(REP.original.ts,alternative = "stationary"))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  REP.original.ts
## Dickey-Fuller = -1.24, Lag order = 8, p-value = 0.9
## alternative hypothesis: stationary
```

Answer: The results of the Seasonal Mann-Kendall test show that the p value is less than 0.05, which means we can reject the null hypothesis which was that the data is stationary. This means that the data has a trend. The s value is positive so there is an increasing trend.

The p value of the ADF test is 0.9, which is greater than 0.05, so we fail to reject the null hypothesis. This means there may be a stochastic trend in the data and the series is non-stationary. The results of the Seasonal Mann-Kendall Test match the results I observed in Q3 - it appeared that the original data series has a trend. The results of the ADF test show there may be a stochastic trend in the data, which also does match what was observed previously. The data don't appear to be uniformly trend stationary, it looks like there may be a stochastic trend.

Q6

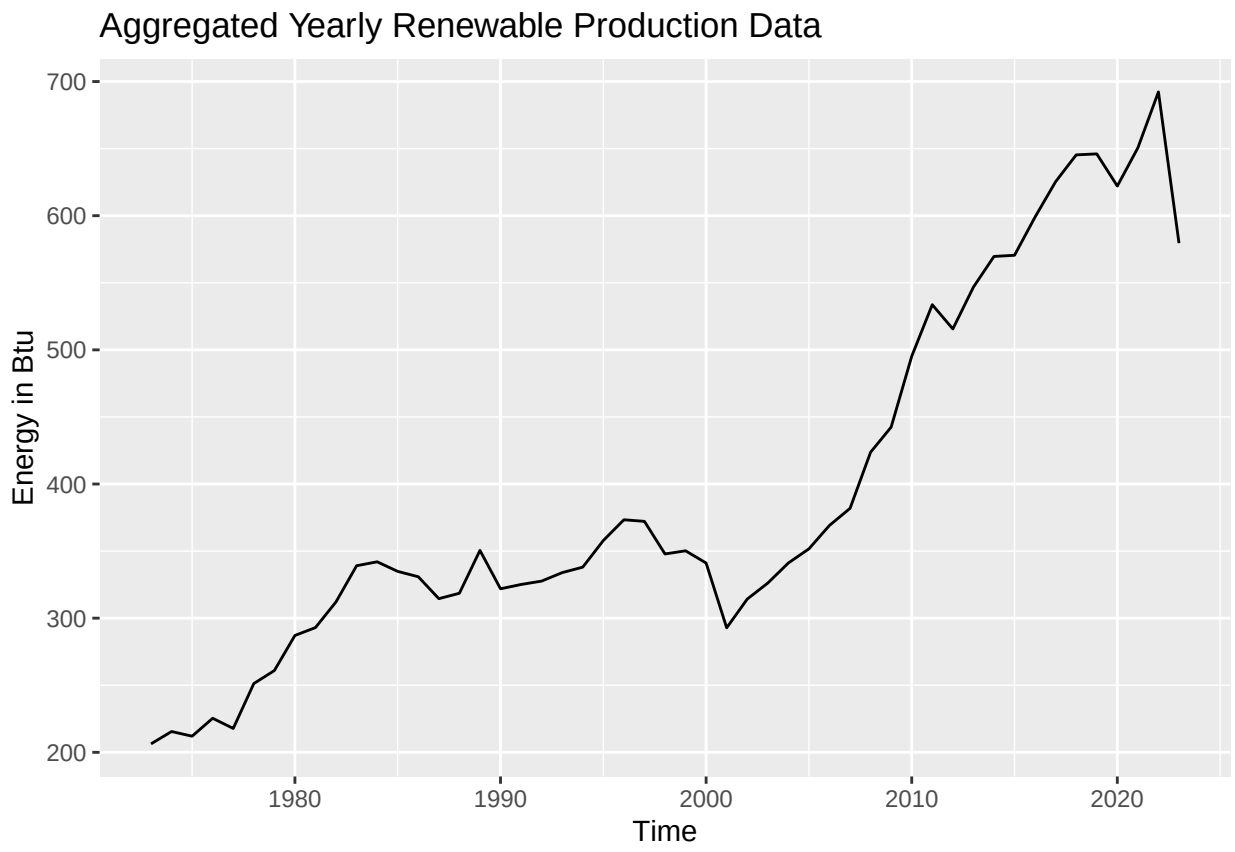
Aggregate the original "Total Renewable Energy Production" series by year. You can use the same procedure we used in class. Store series in a matrix where rows represent months and columns represent years. And then take the columns mean using function colMeans(). Recall the goal is the remove the seasonal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

```r
# aggregating the original data by year
REP_data_matrix <- matrix(REP.original.ts,
                          byrow = FALSE,
                          nrow = 12)
```

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

```r
REP_data_yearly <- colMeans(REP_data_matrix)

# creating date object
energy_data$Month  <- as.Date(energy_data$Month)
my_year <- c(year(first(energy_data$Month)): year(last(energy_data$Month)))

# creating data frame
REP_yearly <- data.frame(my_year, REP_data_yearly)

# converting to a ts
```

```
REP_yearly_ts <- ts(REP_yearly[,2],
                    frequency = 1,
                    start = c(1973))

# plotting ts
autoplot(REP_yearly_ts,
         ylab = "Energy in Btu") +
  ggtitle("Aggregated Yearly Renewable Production Data")
```



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?

```
# Mann-Kendall Test
yearly_MKtest <- MannKendall(REP_yearly_ts)
print("Results for Mann Kendall on yearly data")
```

```
## [1] "Results for Mann Kendall on yearly data"
```

```
print(summary(yearly_MKtest))
```

```
## Score =  1019 , Var(Score) = 15158.33
```

```
## denominator =  1275
## tau = 0.799, 2-sided pvalue =< 2.22e-16
## NULL
```

```
# Spearman correlation
sp_corr <- cor.test(REP_data_yearly, my_year, method = "spearman")
print(sp_corr)
```

```
##
##  Spearman's rank correlation rho
##
## data:  REP_data_yearly and my_year
## S = 1908, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9136652
```

```
# ADF
print(adf.test(REP_data_yearly, alternative = "stationary"))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  REP_data_yearly
## Dickey-Fuller = -2.0953, Lag order = 3, p-value = 0.5361
## alternative hypothesis: stationary
```

Answer: The results of the yearly Mann Kendall test are in agreement with the monthly series, only that the s value is lower due to there being less observations because the data is aggregated. This aggregated data test also shows that the p value is less than 0.05 and we can reject the null. There is a trend and it is positive.

For the Spearman correlation, the p value is less than 0.05, so we can reject the null hypothesis and we conclude that the data is not stationary. It follows a trend, which is in agreement with the monthly data.

The ADF for the yearly aggregated data has a p value of 0.54, so we fail to reject the null hypothesis, which is aligned with the results of the monthly ADF findings.