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

Assignment 4 - Due date 02/12/24

Emma Kaufman

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("xlsx")
library(readxl)
library(ggplot2)
library(forecast)
```

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

```
library(tseries)
library(Kendall)
library(forecast)
library(tseries)
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
```

```
library(readxl)
library(ggplot2)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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
energy_data1 <- 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`
```

```
#Now let's extract the 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`
```

```
colnames(energy_data1) <- read_col_names

energy_interest <- select(energy_data1,
                          'Total Renewable Energy Production')
renewable_ts <- ts(energy_interest, start = c(1973,1), frequency = 12)
```

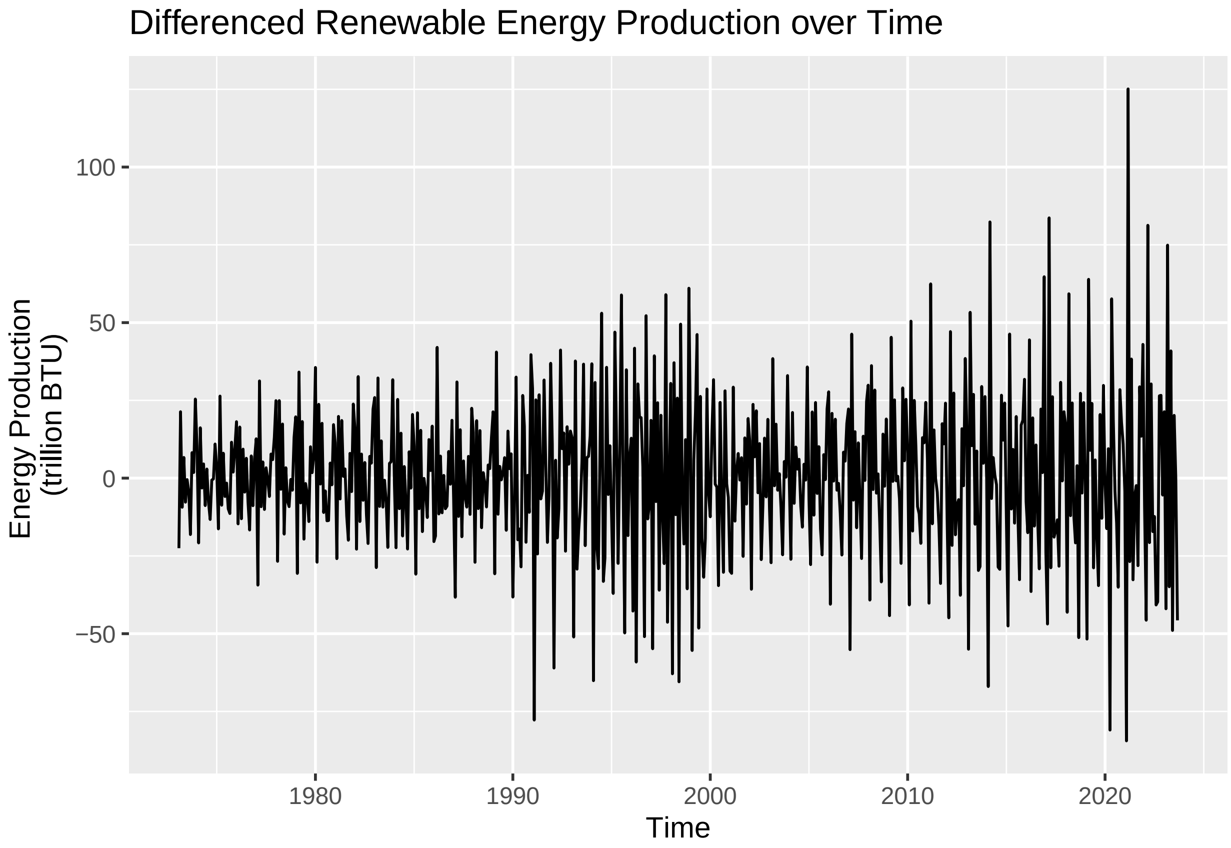
Stochastic Trend and Stationarity Tests

Q1

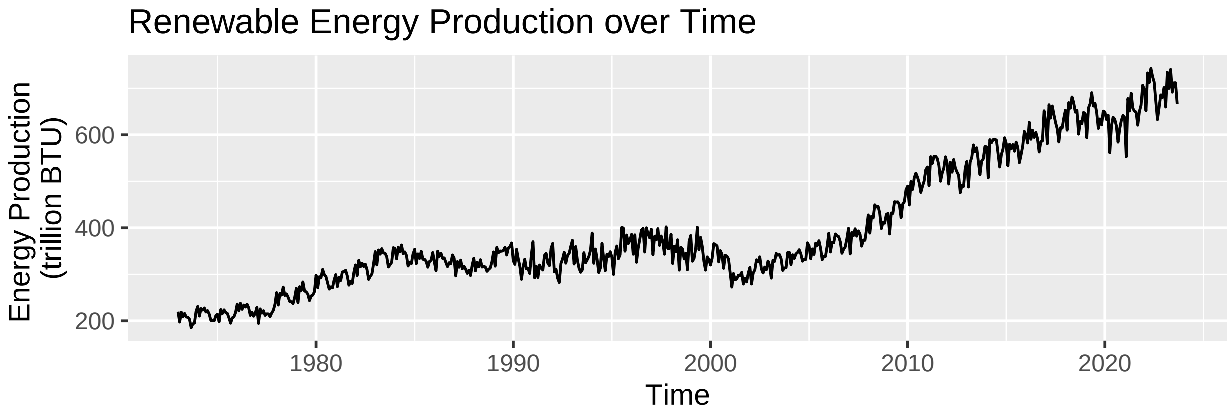
Difference the "Total Renewable Energy Production" series using function diff(). Function diff() is from package base and take three main arguments: * *x* vector containing values to be differenced; * *lag* integer indicating with lag to use; * *differences* integer indicating how many times series should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

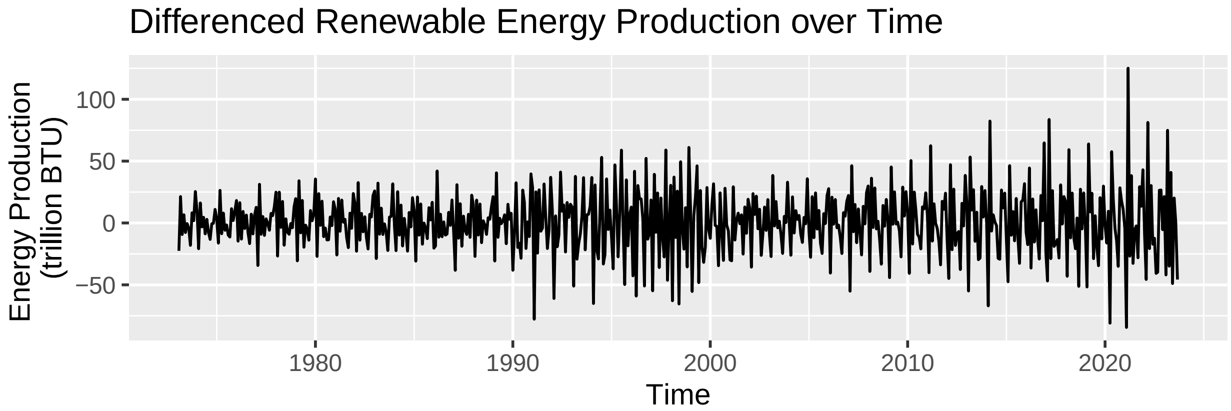
Try differencing at lag 1 only once, i.e., make `lag=1` and `differences=1`. Plot the differenced series Do the series still seem to have trend?

```
#differencing parameters
lag=1
differences=1
renewable_diff_ts <- diff(renewable_ts, lag,differences)

Renew_ts_plot<- autoplot(renewable_ts) +
  labs(x = "Time",
       y = "Energy Production \n(trillion BTU)",
       title = "Renewable Energy Production over Time")

Renew_ts_dif_plot<- autoplot(renewable_diff_ts) +
  labs(x = "Time",
       y = "Energy Production \n(trillion BTU)",
       title = "Differenced Renewable Energy Production over Time")
Renew_ts_dif_plot
```



```
#plotting together
plot_grid(Renew_ts_plot,Renew_ts_dif_plot, nrow=2,align= 'h',
          rel_heights = c(2, 2))
```





> The series no longer appears to have an upwards trend. The differenced plot is centered around 0.

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.

```
#number of observations
nobs <- nrow(energy_interest)

#Create vector t
t<- 1:nobs

#from tibble to df
energy_interest<- as.data.frame(energy_interest)

#empty vectors
beta0 <- numeric(1)
beta1 <- numeric(1)

for (i in 1) {
  #define linear trend based on interested energy type
  linear_trend <- lm(energy_interest[, i] ~ t)
  print(summary(linear_trend))
  beta0[i] <- coef(linear_trend)[1]  # intercept
  beta1[i] <- coef(linear_trend)[2]  # slope
}
```

```
##
## Call:
## lm(formula = energy_interest[, i] ~ 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
```

```
#intercepts
beta0
```

```
## [1] 180.9894
```

```
#slopes
beta1
```

```
## [1] 0.7040391
```

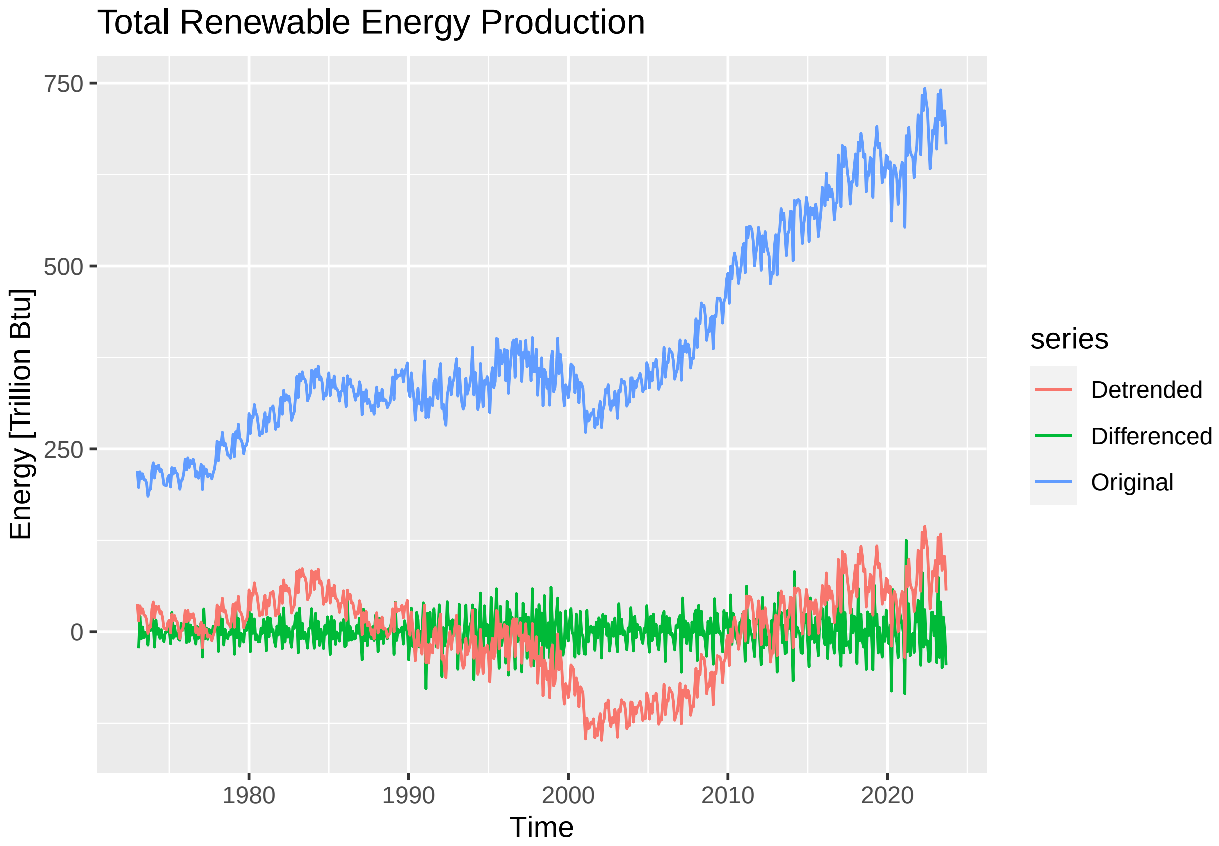
```
#Total Renewable
y_detrend_renew <- energy_interest - (beta0+beta1*t)
y_detrend_renew_ts <- ts(y_detrend_renew, start = c(1973,1), frequency = 12)
df_detrend_renew <- data.frame("Date"=energy_data1$Month,
                               Observed=energy_interest,
                               Detrend= y_detrend_renew)
colnames(df_detrend_renew) <- c("Date", "Observed", "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.

```
autoplot(renewable_ts,series="Original") +
  autolayer(renewable_diff_ts,series="Differenced") +
  autolayer(y_detrend_renew_ts, series= "Detrended") +
 ylab("Energy [Trillion Btu]") +
ggtitle("Total Renewable Energy Production")
```

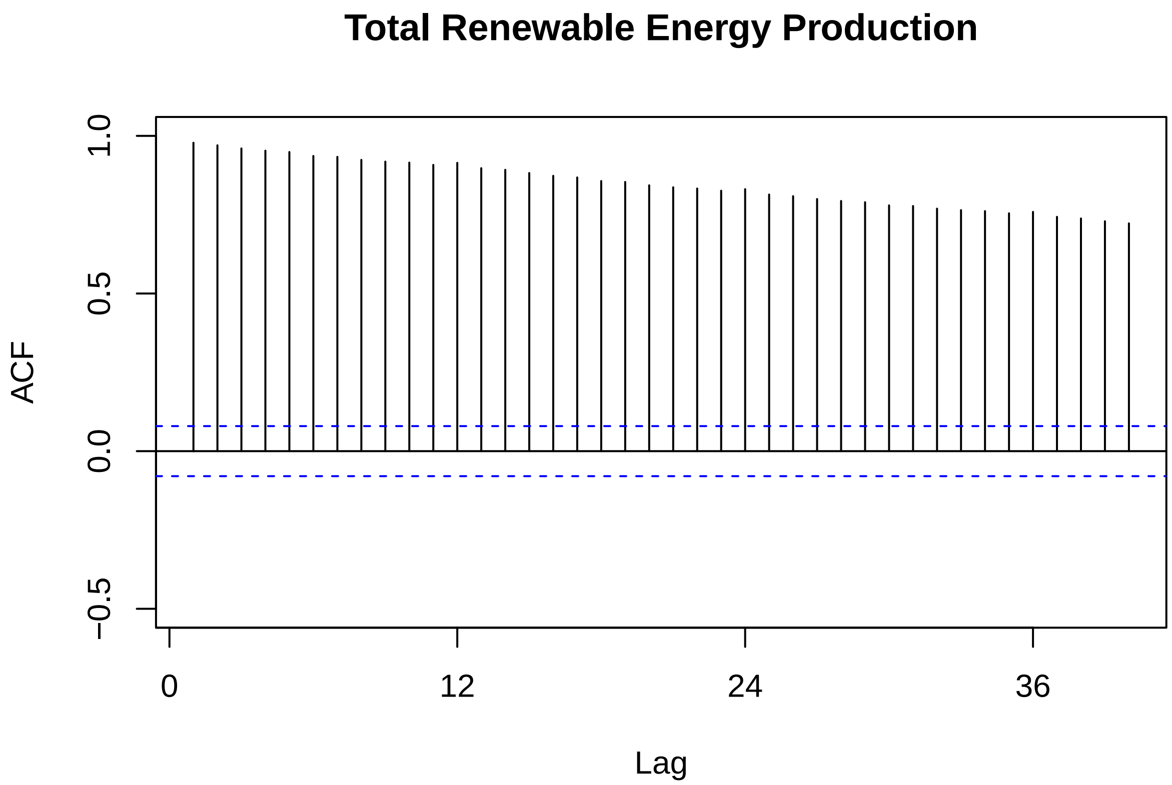


```
# #detrended plot
# Renew_detrend_plot<- autoplot(y_detrend_renew_ts) +
#   labs(x = "Time",
#        y = "Energy Production \n(trillion BTU)",
#        title = "Detrended Renewable Energy Production over Time")
# #original, detrended, and differenced plots
# plot_grid(Renew_ts_plot,Renew_ts_dif_plot, Renew_detrend_plot,
#           nrow=3 ,align= 'h',
#           rel_heights = c(2, 2, 2))
```

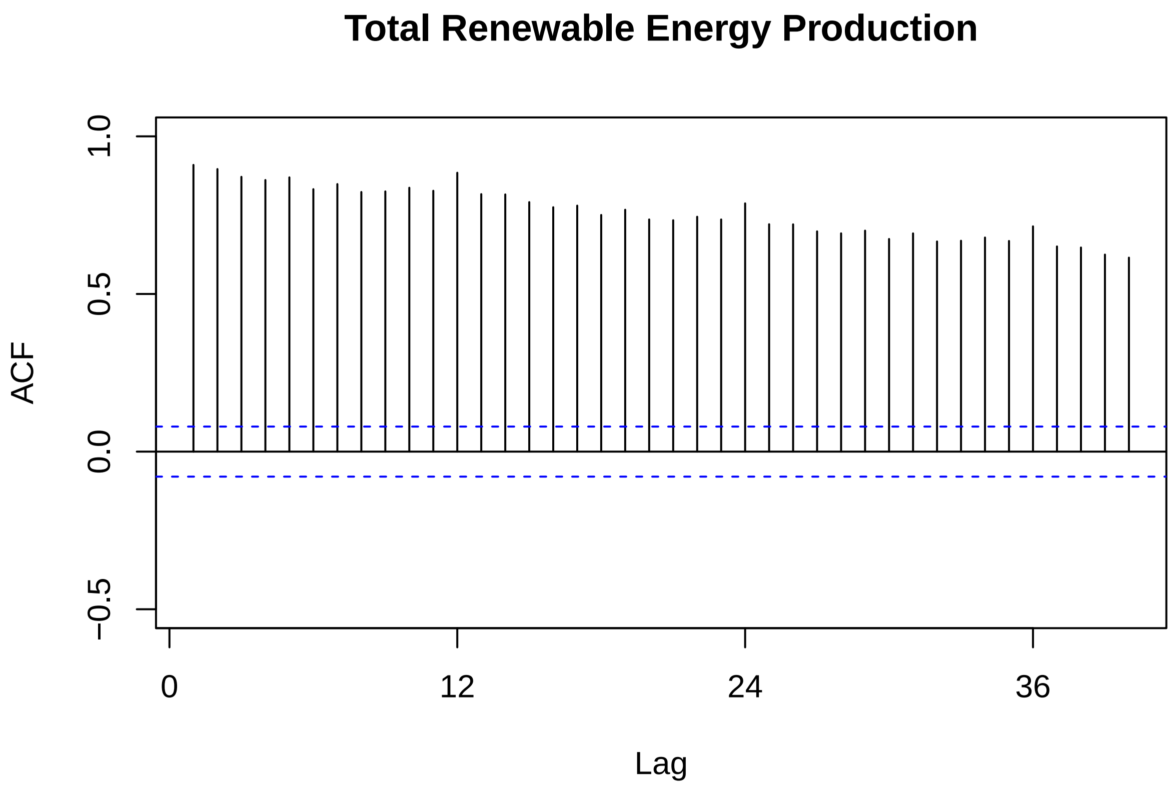
Q4

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?

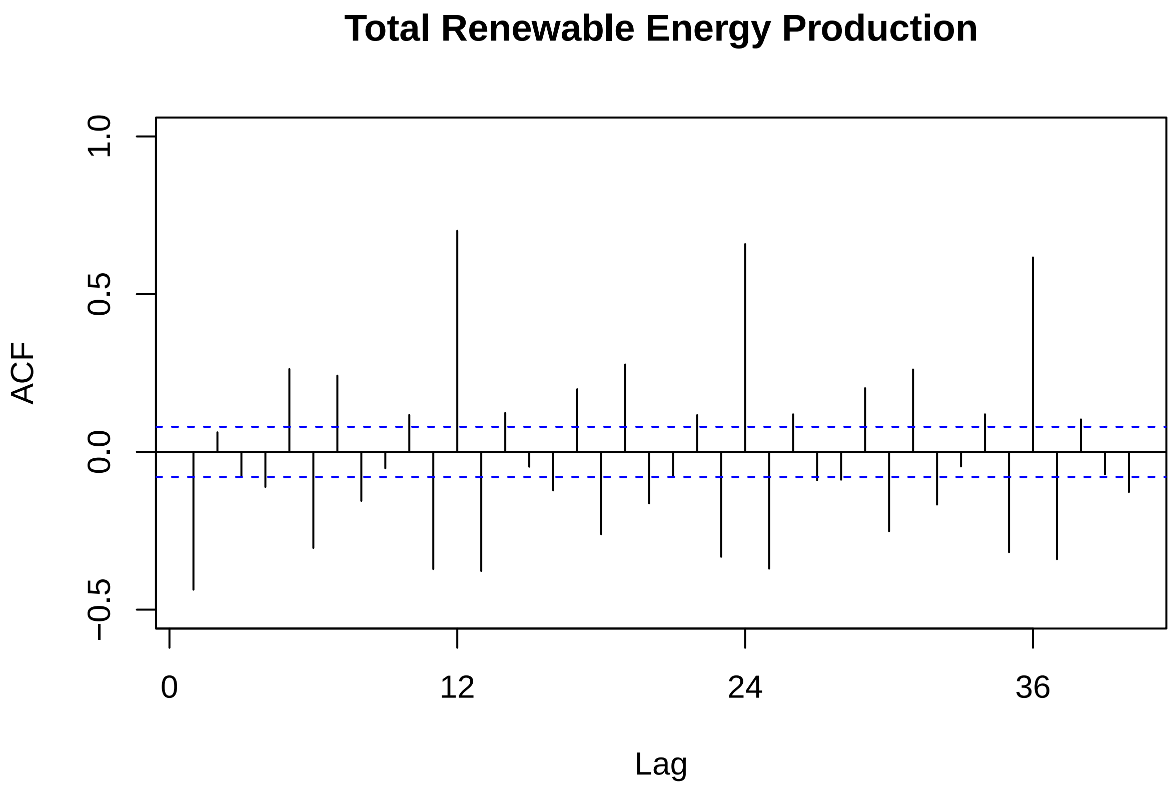
```
Renewable_acf= autoplot(Acf(renewable_ts,
                           lag.max=40,
                           ylim=c(-0.5,1)))
```



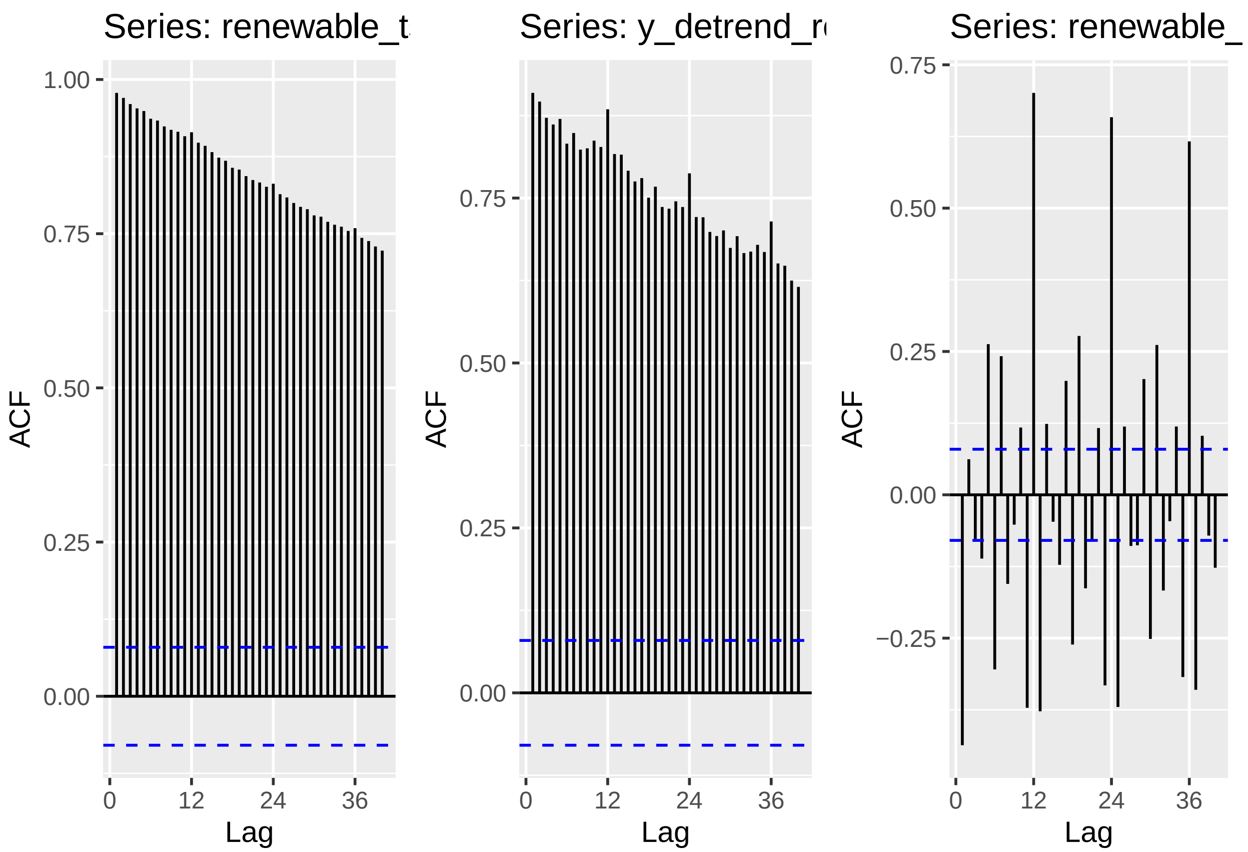
```
Renew_acf_detrend= autoplot(Acf(y_detrend_renew_ts,
                               lag.max=40,
                               ylim=c(-0.5,1)))
```



```
Renew_acf_differenced= autoplot(Acf(renewable_diff_ts,
                                    lag.max=40,
                                    ylim=c(-0.5,1)))
```



```
plot_grid(Renewable_acf,Renew_acf_detrend, Renew_acf_differenced, ncol=3 ,align= 'h',
          rel_heights = c(2, 2, 2))
```



> The differencing seems more effective in eliminating the trend than the linear regression. We see overall much lower significance in the autocorrelation for the differenced plot and no more decreasing trend. The peaks for the ACF plot we do see repeat every 12 months, so they are likely from seasonality.

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

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

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

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

```r
#ADF test
print("Results for ADF test:")
```

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

```r
print(adf.test(renewable_ts,alternative = "stationary"))
```

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

```r
#question what if s-value tells you you have an increasing trend but you are rejecting the null?
# what do we say in relation to whether or not this matches above?
```

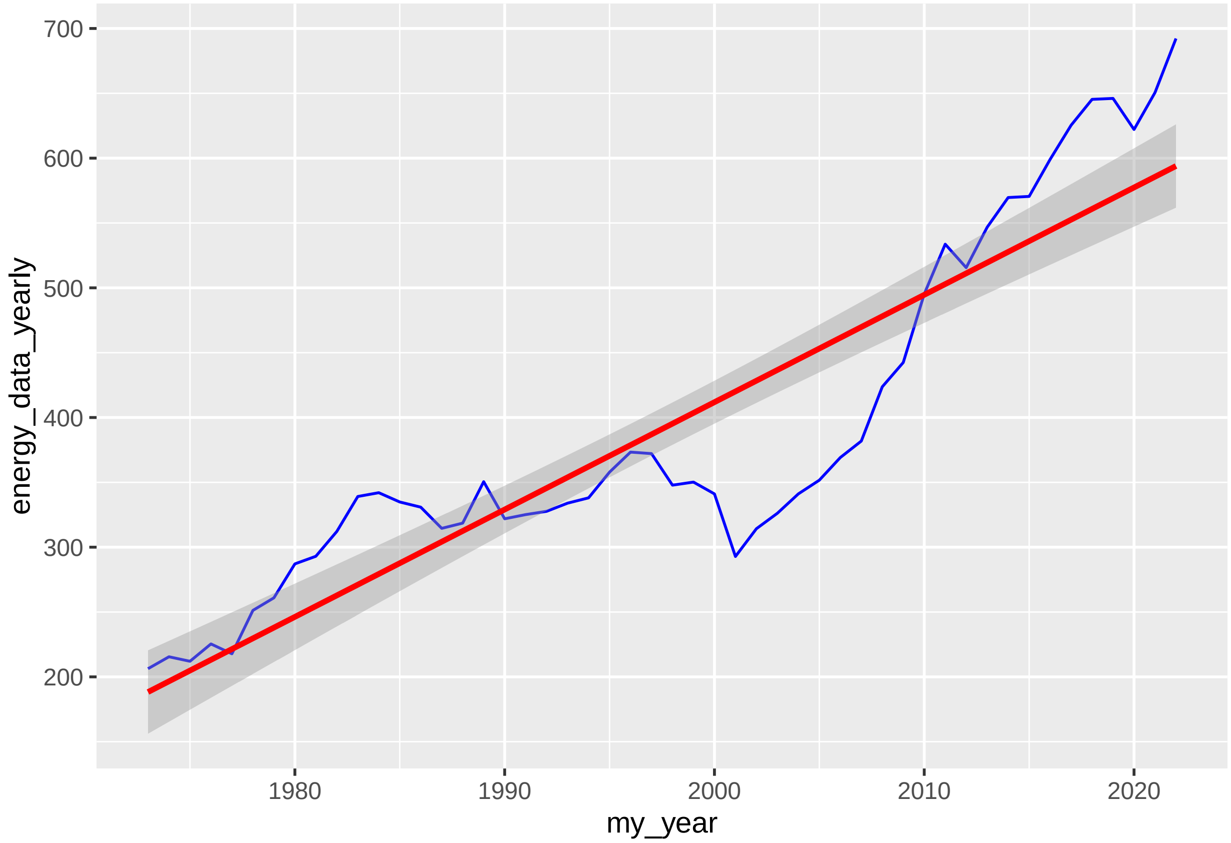
> For the seasonal mann kendall test our null hypothesis is that the data are stationary. Because our p-value is less than 0.05, that means we can reject the null hypothesis. So we can send our data follow a trend, and because we have a positive S-value that means there is an increasing trend. For the ADF test, our null hypothesis is that our model has a unit root. We have a p-value of 0.9 so we cannot reject the null hypothesis, and we can say that our data have a stochastic trend. This matches what we observed above, we saw an increasing trend that was not deterministic (it wasn't removed by detrending the data).

Q6

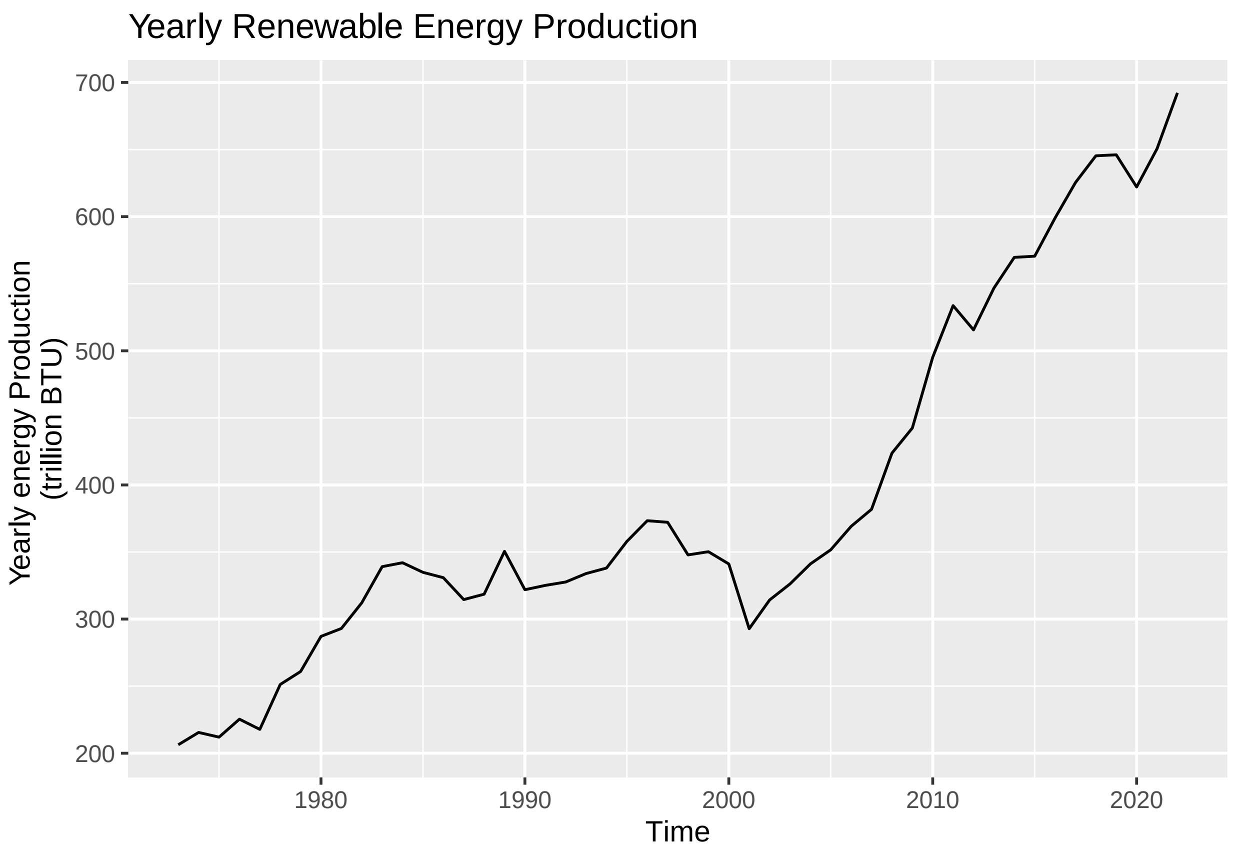
Aggregate the original "Total Renewable Energy Production" series by year. You can use the same procedure we used in class. Store series in a matrix where rows represent months and columns represent years. And then take the columns mean using function colMeans(). Recall the goal is the remove the seasonal 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
#remove the last 9 months because we only want whole years of data
renewable_ts <- as.ts(renewable_ts[1:600,])
my_date <- energy_data1$Month[1:600]

#create new df
renewable_data_new <- cbind(my_date, renewable_ts)


#Group data in yearly steps instances
energy_data_matrix <- matrix(renewable_ts,byrow=FALSE,nrow=12) #populate matrix by column
energy_data_yearly <- colMeans(energy_data_matrix) #mean for each column

my_year <- c(year(first(my_date)):year(last(my_date)))

energy_data_new_yearly <- data.frame(my_year, energy_data_yearly)

ggplot(energy_data_new_yearly, aes(x=my_year, y=energy_data_yearly)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm")
```

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



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

autoplot(energy_yearly_ts) +
  labs(x = "Time",
       y = "Yearly energy Production \n(trillion BTU)",
       title = "Yearly Renewable Energy Production")
```



```
#why as timeseries and not as a dataframe?
```

Q7

Apply the Mann Kendal, 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
MKtest_year <- MannKendall(energy_yearly_ts)
print("Results for Seasonal Mann Kendall: ")
```

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

```
print(summary(MKtest_year))
```

```
## Score =  983 , Var(Score) = 14291.67
## denominator =  1225
## tau = 0.802, 2-sided pvalue =< 2.22e-16
## NULL
```

```
#Spearman
#Deterministic trend with Spearman Correlation Test
print("Results from Spearman Correlation")
```

```
## [1] "Results from Spearman Correlation"
```

```
sp_rho=cor(energy_yearly_ts,my_year,method="spearman")
print(sp_rho)
```

```
## [1] 0.9110684
```

```
#with cor.test you can get test statistics
sp_rho=cor.test(energy_yearly_ts,my_year,method="spearman")
print(sp_rho)
```

```
##
##  Spearman's rank correlation rho
##
## data:  energy_yearly_ts and my_year
## S = 1852, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9110684
```

```
#ADF
print("Results for ADF test:")
```

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

```
print(adf.test(energy_yearly_ts,alternative = "stationary"))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  energy_yearly_ts
## Dickey-Fuller = -1.0881, Lag order = 3, p-value = 0.9156
## alternative hypothesis: stationary
```

> For the Mann Kendall the p-value is less than 0.05 so we can reject the null hypothesis that the data are stationary. The S-value of 983 tells us that we have a positive trend. For the Spearman correlation test we get a coefficient of 0.911, which means there is a trend, and our p-value is much less than 0.05 so we can reject the null hypothesis that the data are stationary. For ADF, our p-value is 0.9, so we cannot reject the null. This means that the data are stochastic and follow a trend. These results are in agreement with the stationarity tests on the monthly data. They all say there is a trend in the data and that it is stochastic.