ENV 790.30 - Time Series Analysis for Energy Data | Spring 2021

Assignment 3 - Due date 02/15/21

Xueying Feng

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github.

Once you have the project open the first thing you will do is change "Student Name" on line 3 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).

Please keep this R code chunk options for the report. It is easier for us to grade when we can see code and output together. And the tidy.opts will make sure that line breaks on your code chunks are automatically added for better visualization.

When you have completed the assignment, **Knit** the text and code into a single PDF file. Rename the pdf file such that it includes your first and last name (e.g., "LuanaLima_TSA_A01_Sp21.Rmd"). Submit this pdf using Sakai.

Questions

Consider the same data you used for A2 from the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumpt
The data comes from the US Energy Information and Administration and corresponds to the January 2021 Monthly Energy Review. Once again you will work only with the following columns: Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption. Create a data frame structure with these three time series only.

R packages needed for this assignment:"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")
library("readxl")
library(lubridate)
```

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

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

```r
library(ggplot2)
#install.packages("forecast")
library(forecast)
```

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

```r
#install.packages("tseries")
library(tseries)
library(Kendall)
```

##Trend Component

Q1

Create a plot window that has one row and three columns. And then for each object on your data frame, fill the plot window with time series plot, ACF and PACF. You may use the some code form A2, but I want all three plots on the same window this time. (Hint: watch videos for M4)

```r
#Importing data set
MonthlyData <- read_excel("../Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx

# number of obs
nobsv <- nrow(MonthlyData)


# Select columns: Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Pow
MonthlyData_subset<- MonthlyData[2:nobsv, c(1, 4, 5, 6)]

# Checking data
head(MonthlyData_subset)
```

```
## # A tibble: 6 x 4
##   Month               `Total Biomass Ene~ `Total Renewable E~ `Hydroelectric Po~
##   <dttm>              <chr>               <chr>               <chr>
## 1 1973-01-01 00:00:00 129.787             403.981             272.703
## 2 1973-02-01 00:00:00 117.338             360.9               242.199
## 3 1973-03-01 00:00:00 129.938             400.161             268.81
## 4 1973-04-01 00:00:00 125.636             380.47              253.185
## 5 1973-05-01 00:00:00 129.834             392.141             260.77
## 6 1973-06-01 00:00:00 125.611             377.232             249.859
```

```r
str(MonthlyData_subset)
```

```
## tibble [574 x 4] (S3: tbl_df/tbl/data.frame)
##  $ Month                            : POSIXct[1:574], format: "1973-01-01" "1973-02-01" ...
##  $ Total Biomass Energy Production  : chr [1:574] "129.787" "117.338" "129.938" "125.636" ...
##  $ Total Renewable Energy Production: chr [1:574] "403.981" "360.9" "400.161" "380.47" ...
##  $ Hydroelectric Power Consumption  : chr [1:574] "272.703" "242.199" "268.81" "253.185" ...
```

```r
# Change tbl_df to data frame
MonthlyData_subset = as.data.frame(MonthlyData_subset)
str(MonthlyData_subset)
```

```
## 'data.frame':    574 obs. of  4 variables:
##  $ Month                            : POSIXct, format: "1973-01-01" "1973-02-01" ...
##  $ Total Biomass Energy Production  : chr  "129.787" "117.338" "129.938" "125.636" ...
##  $ Total Renewable Energy Production: chr  "403.981" "360.9" "400.161" "380.47" ...
##  $ Hydroelectric Power Consumption  : chr  "272.703" "242.199" "268.81" "253.185" ...
```

```r
ncoln<- ncol(MonthlyData_subset)-1
```

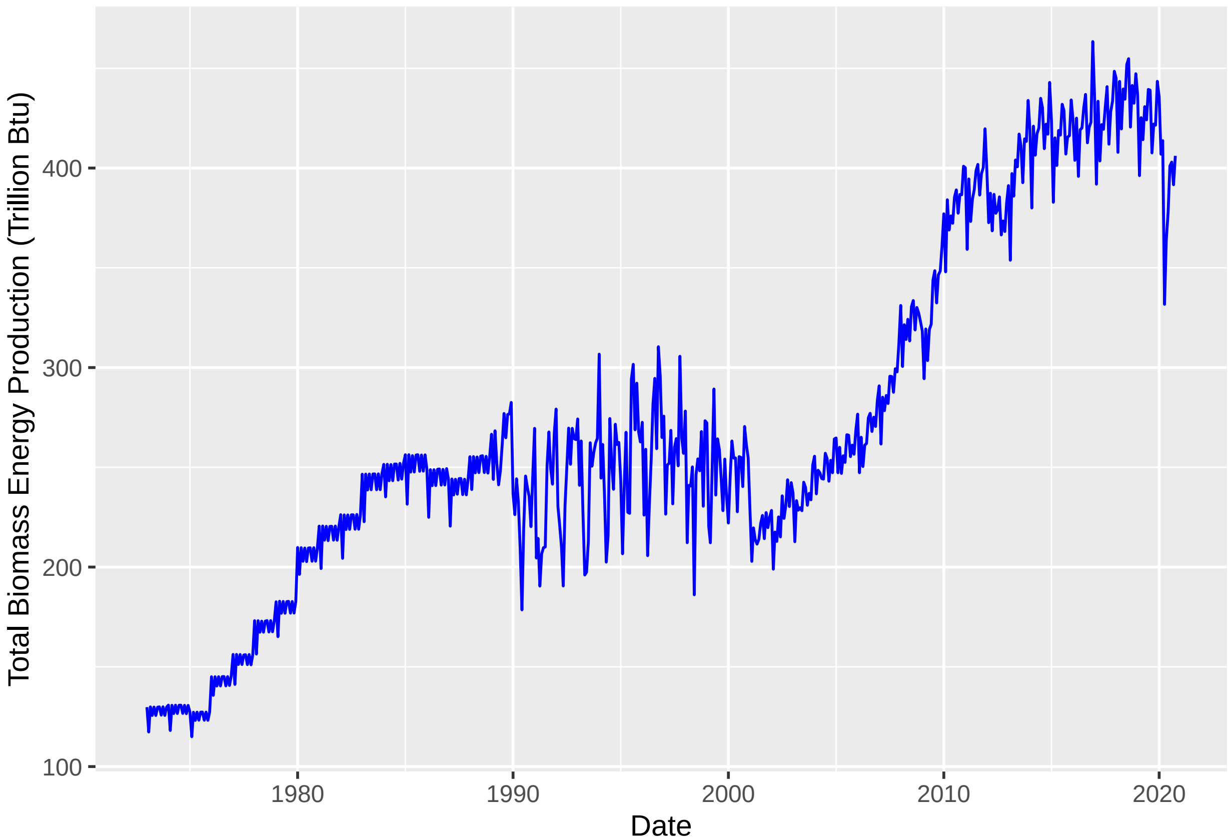
```r
# change character format to numeric format
MonthlyData_subset[,2:4] <- sapply(MonthlyData_subset[,2:4],as.numeric)
str(MonthlyData_subset)
```

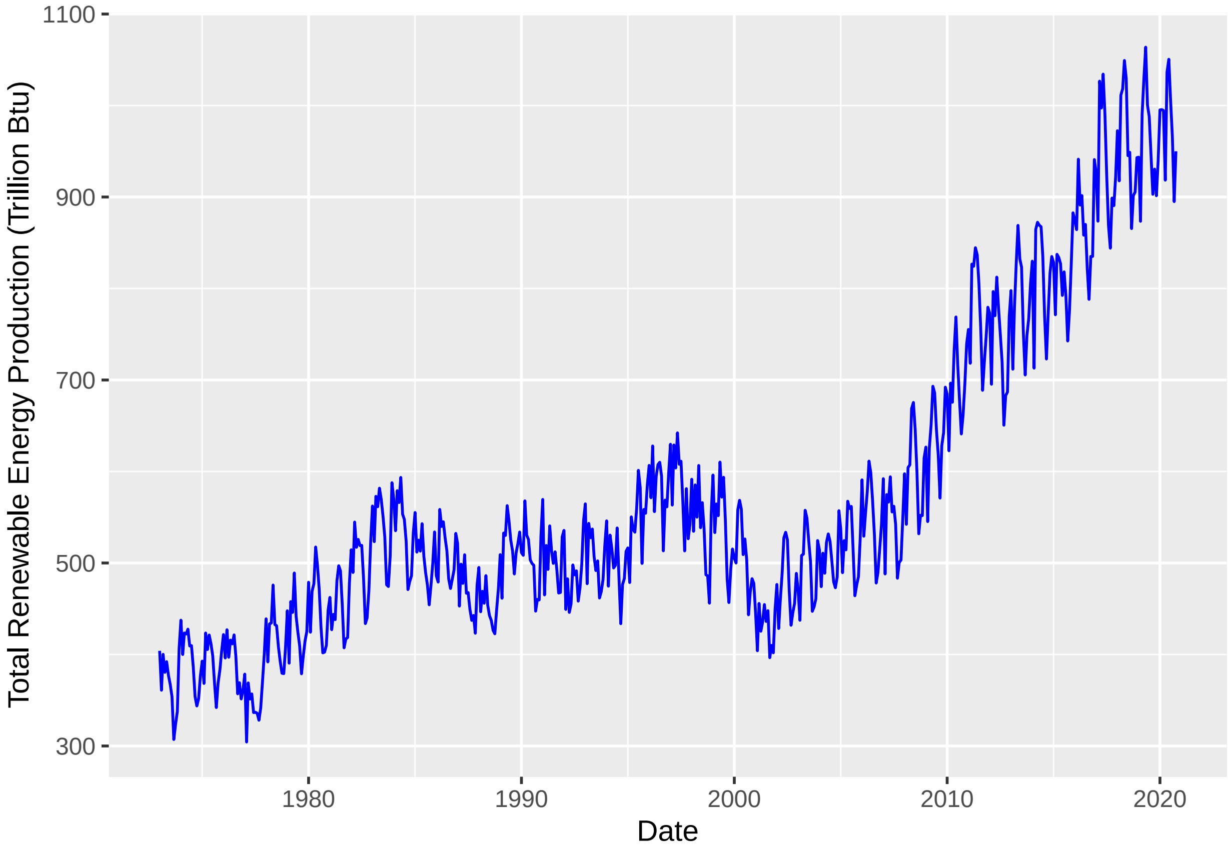
```
## 'data.frame':    574 obs. of  4 variables:
##  $ Month                            : POSIXct, format: "1973-01-01" "1973-02-01" ...
##  $ Total Biomass Energy Production  : num  130 117 130 126 130 ...
##  $ Total Renewable Energy Production: num  404 361 400 380 392 ...
##  $ Hydroelectric Power Consumption  : num  273 242 269 253 261 ...
```

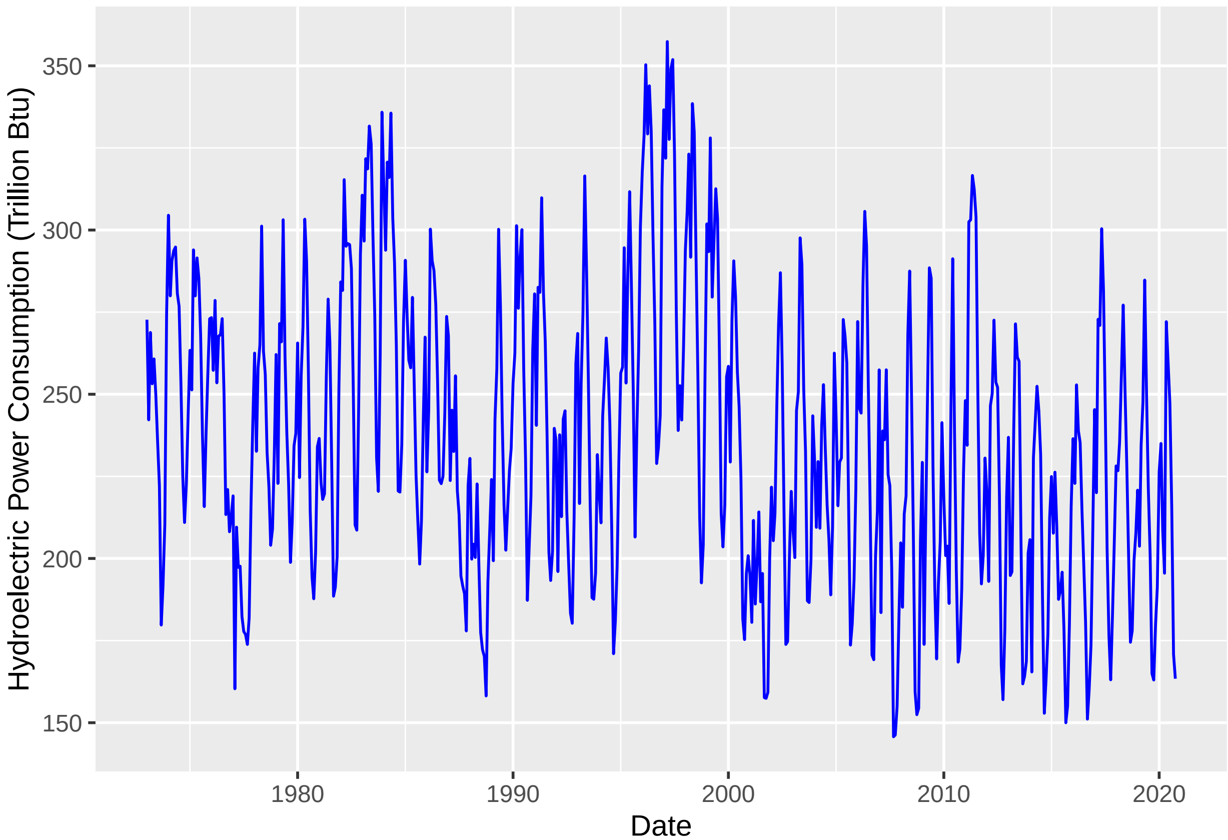
```r
# Change column names
colnames(MonthlyData_subset)[1] <- "Date"
#colnames(MonthlyData_subset)=c("Date","Biomass","Renewable","Hydroelectric")
str(MonthlyData_subset)
```

```
## 'data.frame':    574 obs. of  4 variables:
##  $ Date                             : POSIXct, format: "1973-01-01" "1973-02-01" ...
##  $ Total Biomass Energy Production  : num  130 117 130 126 130 ...
##  $ Total Renewable Energy Production: num  404 361 400 380 392 ...
##  $ Hydroelectric Power Consumption  : num  273 242 269 253 261 ...
```

```r
#using package ggplot2
#devtools::install_github('cran/ggplot2')


for(i in 1:ncoln){
  par(mfrow=c(1,3))
  print(ggplot(MonthlyData_subset, aes(x=Date, y=MonthlyData_subset[,(1+i)])) +
            geom_line(color="blue") +
            ylab(paste0(colnames(MonthlyData_subset)[(1+i)]," (Trillion Btu)",sep=""))
        )
}
```


Total Biomass Energy Production (Trillion Btu)
400
300
200
100
1980
1990
2000
2010
2020
Date


Total Renewable Energy Production (Trillion Btu)
1100
900
700
500
300
1980
1990
2000
2010
2020
Date




```r
# Create a data frame structure with these three time series
# From Jan 1973 to Oct 2020 as a time series object
MonthlyData_subset_ts <- ts(MonthlyData_subset[,2:4], frequency = 12, start = c(1973, 1, 1), end = c(202
str(MonthlyData_subset_ts)
```

```
##  Time-Series [1:574, 1:3] from 1973 to 2021: 130 117 130 126 130 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "Total Biomass Energy Production" "Total Renewable Energy Production" "Hydroelectri
```

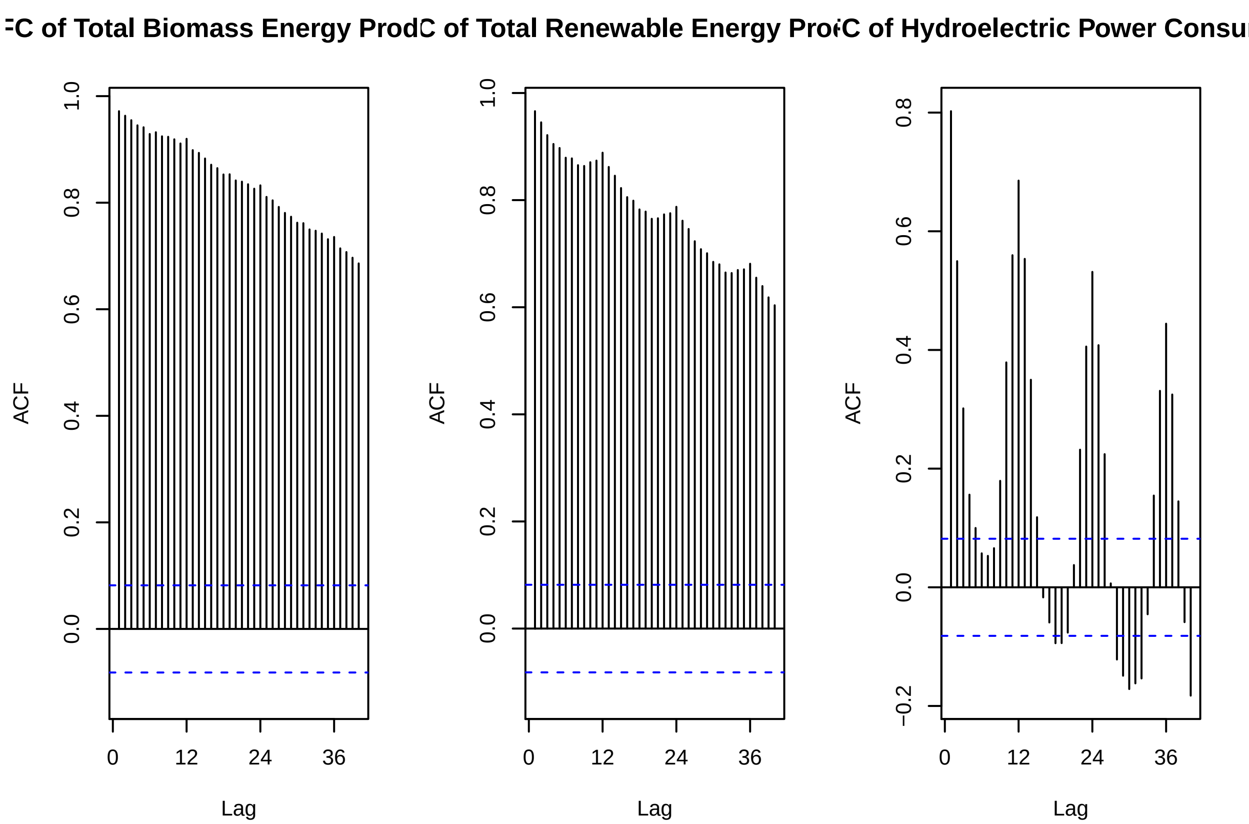
```r
head(MonthlyData_subset_ts)
```

```
##          Total Biomass Energy Production Total Renewable Energy Production
## Jan 1973                         129.787                           403.981
## Feb 1973                         117.338                           360.900
## Mar 1973                         129.938                           400.161
## Apr 1973                         125.636                           380.470
## May 1973                         129.834                           392.141
## Jun 1973                         125.611                           377.232
##          Hydroelectric Power Consumption
## Jan 1973                         272.703
## Feb 1973                         242.199
## Mar 1973                         268.810
## Apr 1973                         253.185
## May 1973                         260.770
## Jun 1973                         249.859
```

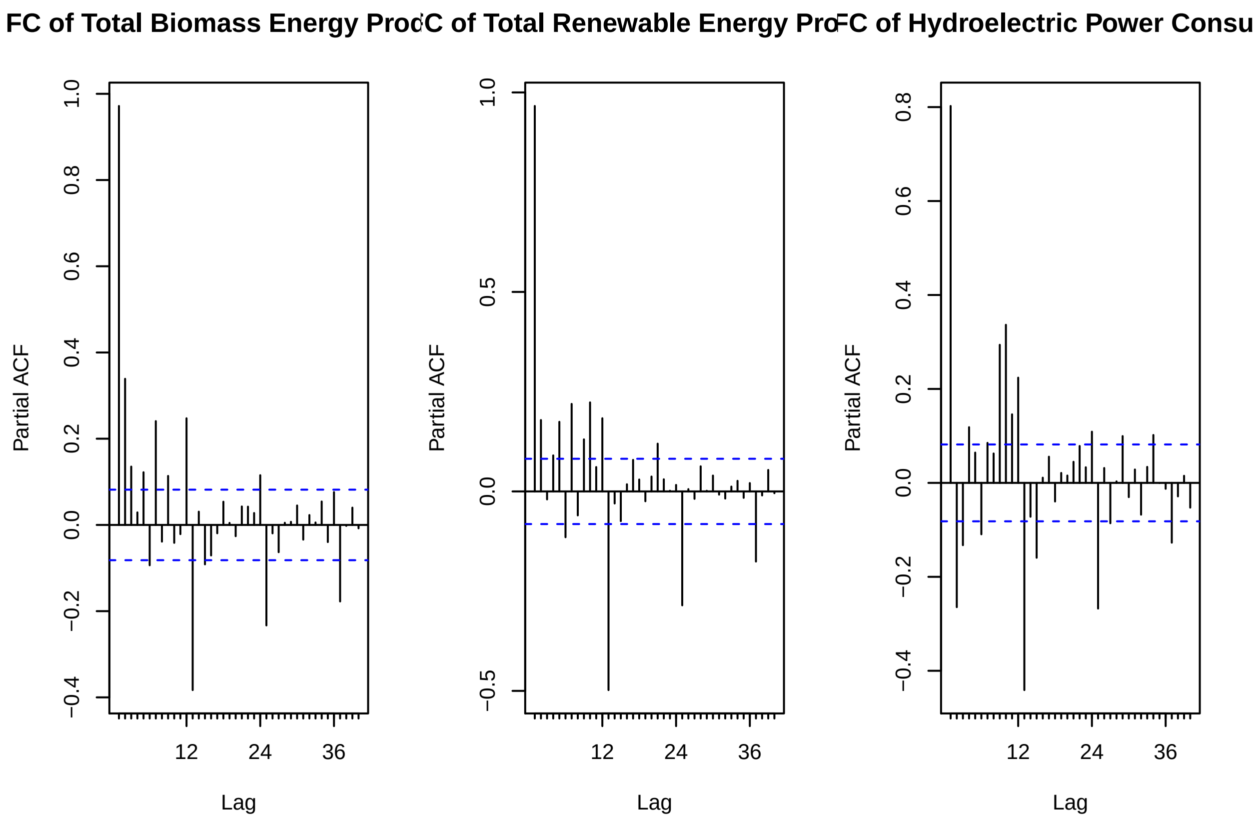
```
tail(MonthlyData_subset_ts)
```

```
##          Total Biomass Energy Production Total Renewable Energy Production
## May 2020                         363.894                          1036.515
## Jun 2020                         377.859                          1050.542
## Jul 2020                         401.014                          1006.388
## Aug 2020                         402.983                           965.785
## Sep 2020                         391.618                           894.957
## Oct 2020                         406.115                           949.990
##          Hydroelectric Power Consumption
## May 2020                         272.098
## Jun 2020                         259.445
## Jul 2020                         247.114
## Aug 2020                         215.725
## Sep 2020                         170.798
## Oct 2020                         163.392
```

```
# number of obs
#ncoln<- ncol(MonthlyData_subset)

#Acf and Pacf
par(mfrow=c(1,3))  #place plot side by side
for(i in 1:ncoln){
  # because I am not storing Acf() into any object, I don't need to specify plot=TRUE
  Acf(MonthlyData_subset_ts[,i],lag.max=40,main=paste("AFC of ",colnames(MonthlyData_subset)[(1+i)],sep=
}
```



```
par(mfrow=c(1,3))  #place plot side by side
for(i in 1:ncoln){
  Pacf(MonthlyData_subset_ts[,i],lag.max=40,main=paste("PAFC of ",colnames(MonthlyData_subset)[(1+i)],s
}
```



Q2

From the plot in Q1, do the series Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption appear to have a trend? If yes, what kind of trend?

> The series Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption appear to have a trend. Total Biomass Energy Production showsa clear upward trend overall, but 1990-2000 shows kind of random variation. >Total Renewable Energy Production: before 1980 and after 2002 show a clear upward trend, between 1985 and 2002, the cycles do not repeat at regular intervals and do not have the same shape. Hydroelectric Power Consumption shows cyclic movements, but in general is a decreasing trend.

Q3

Use the *lm()* function to fit a linear trend to the three time series. Ask R to print the summary of the regression. Interpret the regression output, i.e., slope and intercept. Save the regression coefficients for further analysis.

```
nobsv <-  nrow(MonthlyData_subset) #int
nobsv <- nobsv-1+1 #numeric
```

```r
#Create vector n
n <- c(1:nobsv)

par(mfrow=c(1,3))
for(EnergyType in 1:ncoln){
  #Fit a linear trend to TS of EnergyType
  LinearTrend_model <- lm(MonthlyData_subset[,EnergyType+1]~n)
  print(summary(LinearTrend_model))

  beta0=as.numeric(LinearTrend_model$coefficients[1])  #first coefficient is the intercept term or beta
  beta1=as.numeric(LinearTrend_model$coefficients[2])  #second coefficient is the slope or beta1
  print(beta0)
  print(beta1)

  #Let's plot the time series with its trend line  Command+Shift+C
  # print(ggplot(MonthlyData_subset, aes(x=Date, y=MonthlyData_subset[,(1+EnergyType)])) +
  #             geom_line(color="blue") +
  #             ylab(paste0(colnames(MonthlyData_subset)[(1+EnergyType)]," (Trillion Btu)",sep="")) +
  #             #geom_abline(intercept = beta0, slope = beta1, color="red")
  #             geom_smooth(color="red",method="lm") )
}
```

```
##
## Call:
## lm(formula = MonthlyData_subset[, EnergyType + 1] ~ n)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -101.149  -25.456    4.985   33.353   79.634
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.355e+02  3.296e+00   41.11   <2e-16 ***
## n           4.702e-01  9.934e-03   47.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.44 on 572 degrees of freedom
## Multiple R-squared:  0.7966, Adjusted R-squared:  0.7962
## F-statistic:  2240 on 1 and 572 DF,  p-value: < 2.2e-16
##
## [1] 135.525
## [1] 0.4701605
##
## Call:
## lm(formula = MonthlyData_subset[, EnergyType + 1] ~ n)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -224.735  -55.673    5.418   60.453  263.849
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 330.37156    7.86270   42.02   <2e-16 ***
## n              0.84299    0.02369   35.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 94.07 on 572 degrees of freedom
## Multiple R-squared:  0.6887, Adjusted R-squared:  0.6882
## F-statistic:  1266 on 1 and 572 DF,  p-value: < 2.2e-16
##
## [1] 330.3716
## [1] 0.8429932
##
## Call:
## lm(formula = MonthlyData_subset[, EnergyType + 1] ~ n)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -94.06 -31.57  -1.63  27.73 120.69
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 258.05622    3.52899  73.125  < 2e-16 ***
## n            -0.07341    0.01063  -6.903 1.36e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.22 on 572 degrees of freedom
## Multiple R-squared:  0.07689,    Adjusted R-squared:  0.07528
## F-statistic: 47.64 on 1 and 572 DF,  p-value: 1.361e-11
##
## [1] 258.0562
## [1] -0.07340757
```

> Formula Call:the output is the formula R used to fit the data. The formula just needs the predictor (vector n) and the target/response variable (production), together with the data being used (MonthlyData_subset). The Residuals section of the model output breaks it down into 5 summary points. When assessing how well the model fit the data, you should look for a symmetrical distribution across these points on the mean value zero (0). In simple linear regression, the coefficients are two unknown constants that represent the intercept and slope terms in the linear model. The coefficient Standard Error measures the average amount that the coefficient estimates vary from the actual average value of our response variable. A small p-value for the intercept and the slope indicates that we can reject the null hypothesis which allows us to conclude that there is a relationship between predictor and response variable. The R-squared statistic provides a measure of how well the model is fitting the actual data.

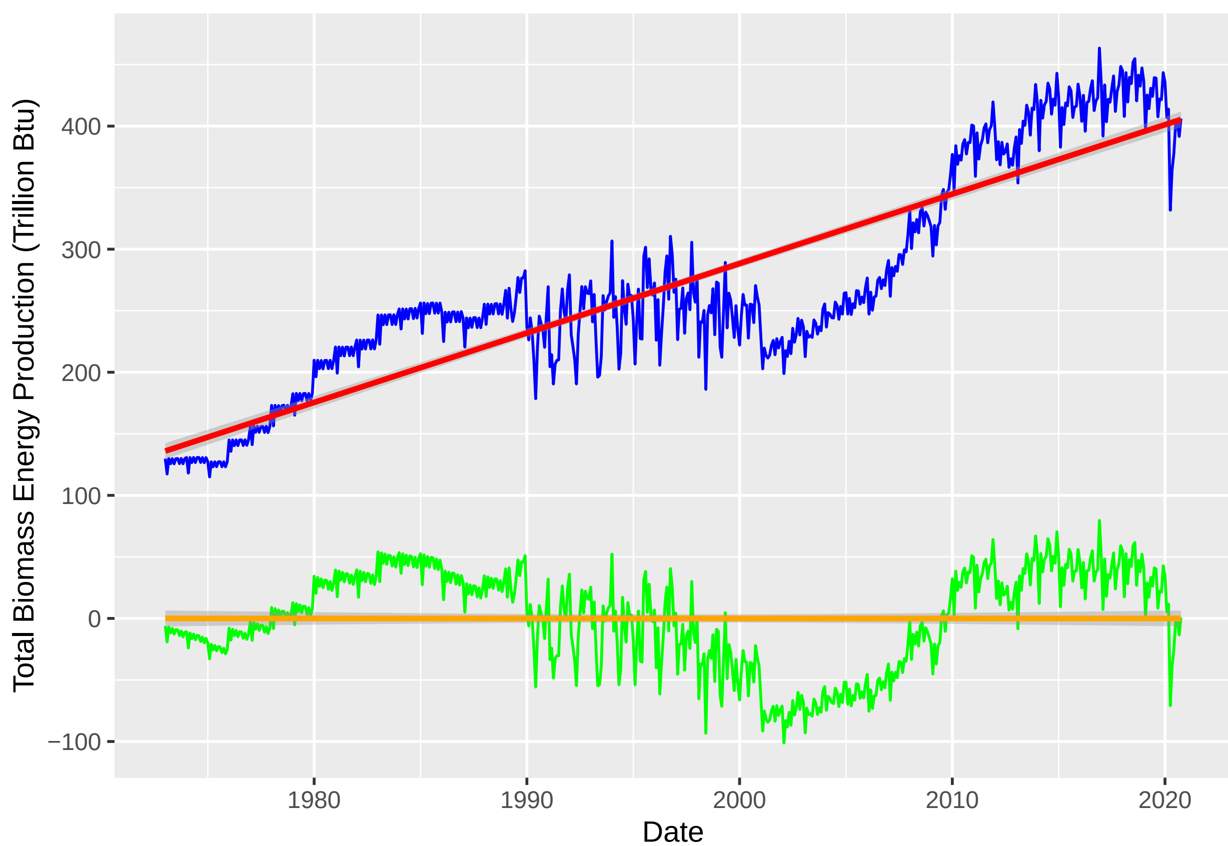
Q4

Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

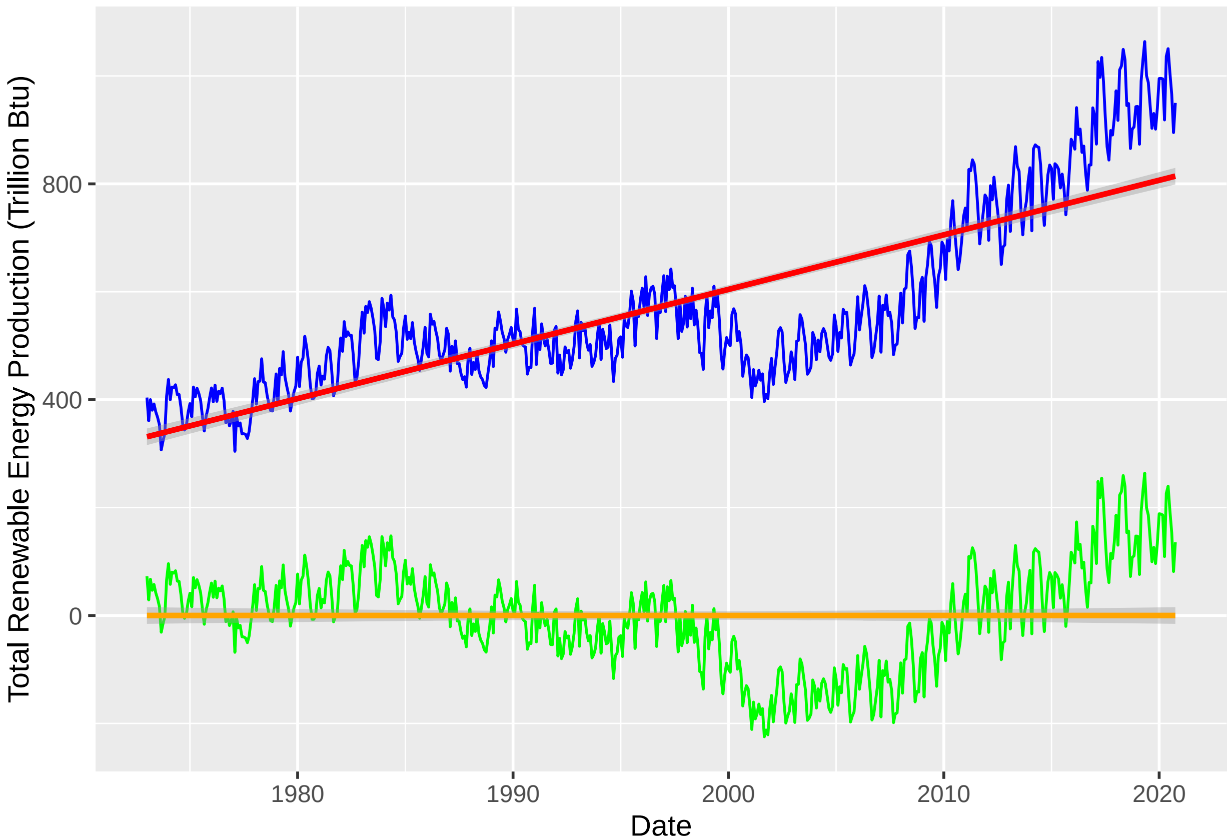
```r
#remove the trend from series
par(mfrow=c(1,3))
```

```r
for(EnergyType in 1:ncoln){
  #Fit a linear trend to TS of EnergyType
  LinearTrend_model <- lm(MonthlyData_subset[,EnergyType+1]~n)
  summary(LinearTrend_model)

  beta0=as.numeric(LinearTrend_model$coefficients[1])  #first coefficient is the intercept term or beta
  beta1=as.numeric(LinearTrend_model$coefficients[2])  #second coefficient is the slope or beta1

  #Remove the trend
  DetrendInflow_data <- MonthlyData_subset[,(EnergyType+1)]-(beta0+beta1*n)

  print(ggplot(MonthlyData_subset, aes(x=Date, y=MonthlyData_subset[,(1+EnergyType)])) +
              geom_line(color="blue") +
              ylab(paste0(colnames(MonthlyData_subset)[(1+EnergyType)]," (Trillion Btu)",sep="")) +
              #geom_abline(intercept = beta0, slope = beta1, color="red")
              geom_smooth(color="red",method="lm") +
              geom_line(aes(y=DetrendInflow_data), col="green")+
              geom_smooth(aes(y=DetrendInflow_data),color="orange",method="lm"))
}
```

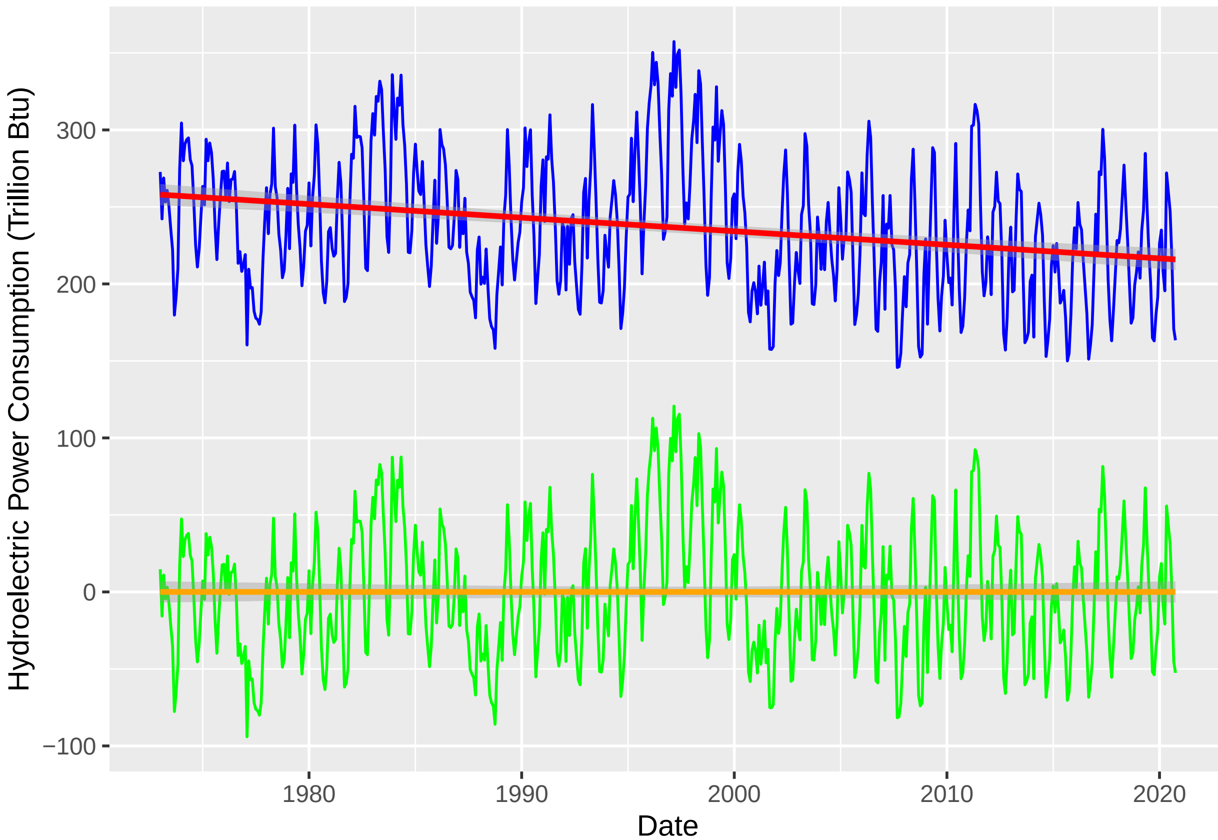
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```




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




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



>First, Detrend plot makes a time series more smoothed and stationary, which does not have obvious increasing or decreasing trend. Second, the value changes near the 0.

Q5

Plot ACF and PACF for the detrended series and compare with the plots from Q1. Did the plots change? How?

```
Data_Detrend <- MonthlyData_subset

# Add detrend data column in data frame
for(i in 1:ncoln) {
  # Create new column
  detrend <- rep(i, nrow(Data_Detrend))
  # Append new column
  Data_Detrend[ , ncol(Data_Detrend) + 1] <- MonthlyData_subset[,(i+1)]-(beta0+beta1*n)
  # Rename column name
  colnames(Data_Detrend)[ncol(Data_Detrend)] <- paste0(colnames(Data_Detrend)[(1+i)]," Detrend",sep="")
}

# Check data
str(Data_Detrend)
```

```
## 'data.frame':    574 obs. of  7 variables:
##  $ Date                                   : POSIXct, format: "1973-01-01" "1973-02-01" ...
##  $ Total Biomass Energy Production        : num  130 117 130 126 130 ...
##  $ Total Renewable Energy Production      : num  404 361 400 380 392 ...
```

```
##  $ Hydroelectric Power Consumption         : num  273 242 269 253 261 ...
##  $ Total Biomass Energy Production Detrend  : num  -128 -141 -128 -132 -128 ...
##  $ Total Renewable Energy Production Detrend: num  146 103 142 123 134 ...
##  $ Hydroelectric Power Consumption Detrend  : num  14.72 -15.71 10.97 -4.58 3.08 ...
```

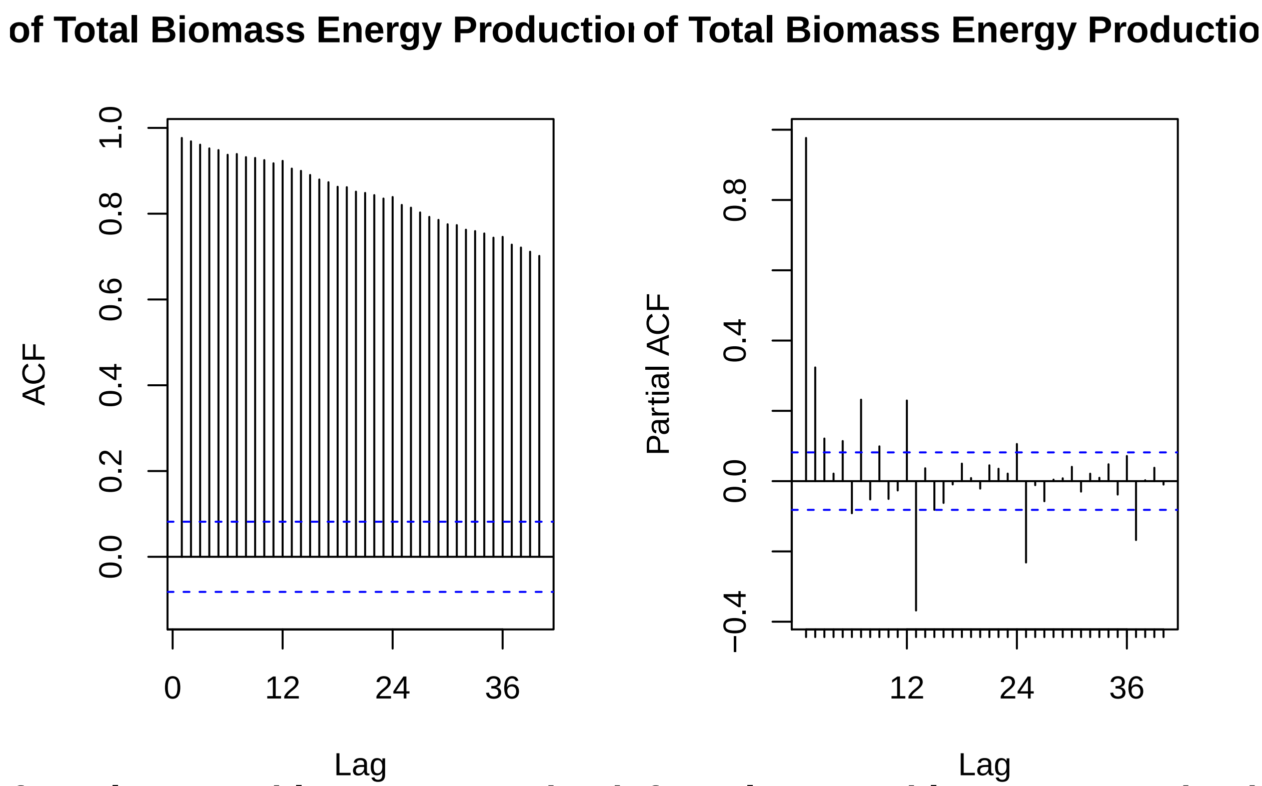
```
head(Data_Detrend)
```

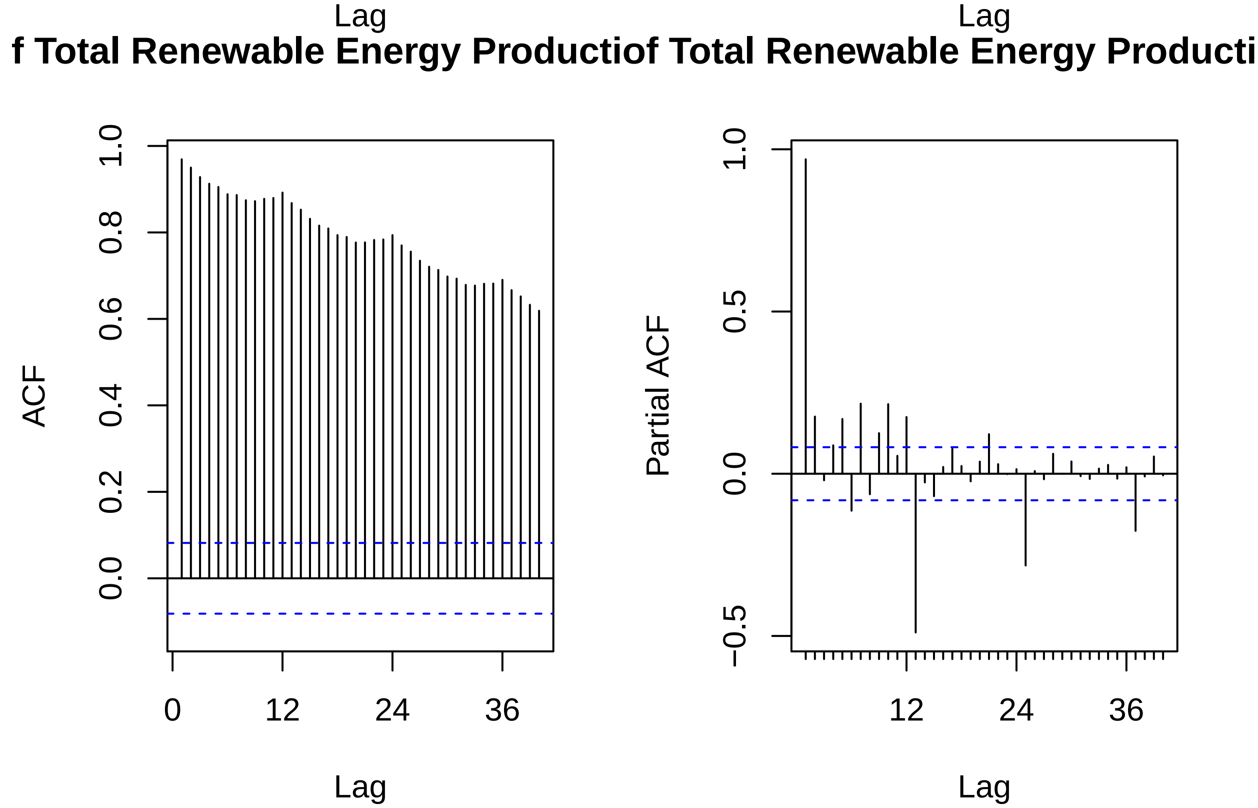
```
##         Date Total Biomass Energy Production Total Renewable Energy Production
## 1 1973-01-01                         129.787                           403.981
## 2 1973-02-01                         117.338                           360.900
## 3 1973-03-01                         129.938                           400.161
## 4 1973-04-01                         125.636                           380.470
## 5 1973-05-01                         129.834                           392.141
## 6 1973-06-01                         125.611                           377.232
##   Hydroelectric Power Consumption Total Biomass Energy Production Detrend
## 1                         272.703                               -128.1958
## 2                         242.199                               -140.5714
## 3                         268.810                               -127.8980
## 4                         253.185                               -132.1266
## 5                         260.770                               -127.8552
## 6                         249.859                               -132.0048
##   Total Renewable Energy Production Detrend
## 1                                  145.9982
## 2                                  102.9906
## 3                                  142.3250
## 4                                  122.7074
## 5                                  134.4518
## 6                                  119.6162
##   Hydroelectric Power Consumption Detrend
## 1                               14.720189
## 2                              -15.710403
## 3                               10.974005
## 4                               -4.577588
## 5                                3.080820
## 6                               -7.756773
```

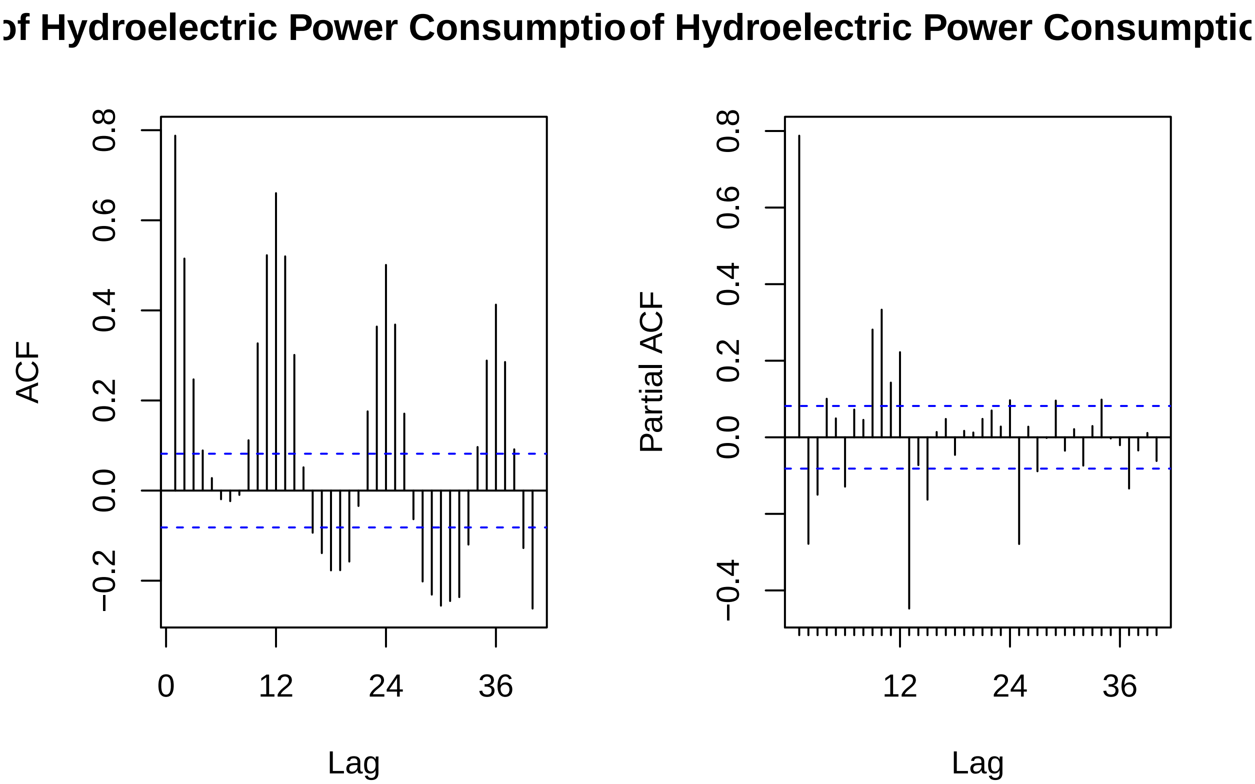
```
#change Detrend_data to TS
Data_Detrend_ts <- ts(Data_Detrend[,2:7],frequency=12)
str(Data_Detrend_ts)
```

```
##  Time-Series [1:574, 1:6] from 1 to 48.8: 130 117 130 126 130 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:6] "Total Biomass Energy Production" "Total Renewable Energy Production" "Hydroelectri
```

```
#Acf and Pacf plots
for(i in 1:ncoln){
  par(mfrow=c(1,2))  #place plot side by side
  Acf(Data_Detrend_ts[,i+3],lag.max=40,main=paste("AFC of ",colnames(Data_Detrend_ts)[(3+i)],sep=""))
  # because I am not storing Acf() into any object, I don't need to specify plot=TRUE
  Pacf(Data_Detrend_ts[,i+3],lag.max=40,main=paste("PAFC of ",colnames(Data_Detrend_ts)[(3+i)],sep=""))
}
```


of Total Biomass Energy Productioı of Total Biomass Energy Productio
ACF
Partial ACF
Lag
Lag
0
12
24
36
1.0
0.8
0.6
0.4
0.2
0.0
−0.4


f Total Renewable Energy Productiof Total Renewable Energy Producti
ACF
Partial ACF
Lag
Lag
0
12
24
36
1.0
0.5
0.0
−0.5



>The plot change a little bit, but not too much for plot 1 and 2. >The third ACF at lag 5 6 7 8 are changed to nagetive and less lines fall into two dotted lines.

Seasonal Component

Set aside the detrended series and consider the original series again from Q1 to answer Q6 to Q8.

Q6

Do the series seem to have a seasonal trend? Which serie/series? Use function *lm()* to fit a seasonal means model to this/these time series. Ask R to print the summary of the regression. Interpret the regression output. Save the regression coefficients for further analysis.

```
for(EnergyType in 1:ncoln){
  #Use seasonal means model
  #First create the seasonal dummies
  Energy_dummies <- seasonaldummy(MonthlyData_subset_ts[,EnergyType])
  #this function only accepts ts object, no need to add one here because date
  #object is not a column

  #Then fit a linear model to the seasonal dummies
  seasonal_model=lm(MonthlyData_subset[,(EnergyType+1)]~Energy_dummies)
  print(summary(seasonal_model))

  #Look at the regression coefficient. These will be the values of Beta

  #Store regression coefficients
  beta_int=seasonal_model$coefficients[1]
```

```r
  beta_coeff=seasonal_model$coefficients[2:12]

  #compute seasonal component
  seasonal_comp=array(0,nobsv)
  for(n in 1:nobsv){
    seasonal_comp[n]=(beta_int+beta_coeff%*%Energy_dummies[n,])
  }

  #Understanding what we did
  # print(ggplot(MonthlyData_subset, aes(x=Date, y=MonthlyData_subset[,(1+EnergyType)])) +
  #             geom_line(color="blue") +
  #             ylab(paste0(colnames(MonthlyData_subset)[(1+EnergyType)]," (Trillion Btu)",sep="")) +
  #             geom_line(aes(y=seasonal_comp), col="red"))
}
```

```
##
## Call:
## lm(formula = MonthlyData_subset[, (EnergyType + 1)] ~ Energy_dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -153.47  -50.56  -20.25   52.13  182.84
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       280.5693    12.7954  21.927   <2e-16 ***
## Energy_dummiesJan  -1.0039    18.0009  -0.056    0.956
## Energy_dummiesFeb -29.3891    18.0009  -1.633    0.103
## Energy_dummiesMar  -8.6090    18.0009  -0.478    0.633
## Energy_dummiesApr -20.5046    18.0009  -1.139    0.255
## Energy_dummiesMay -14.0960    18.0009  -0.783    0.434
## Energy_dummiesJun -19.5548    18.0009  -1.086    0.278
## Energy_dummiesJul  -3.4306    18.0009  -0.191    0.849
## Energy_dummiesAug   0.2220    18.0009   0.012    0.990
## Energy_dummiesSep -11.9821    18.0009  -0.666    0.506
## Energy_dummiesOct  -0.5379    18.0009  -0.030    0.976
## Energy_dummiesNov  -9.3753    18.0954  -0.518    0.605
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87.72 on 562 degrees of freedom
## Multiple R-squared:  0.01116,    Adjusted R-squared:  -0.008199
## F-statistic: 0.5764 on 11 and 562 DF,  p-value: 0.8486
##
##
## Call:
## lm(formula = MonthlyData_subset[, (EnergyType + 1)] ~ Energy_dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -263.99 -102.98  -52.33   36.68  453.58
##
## Coefficients:
```

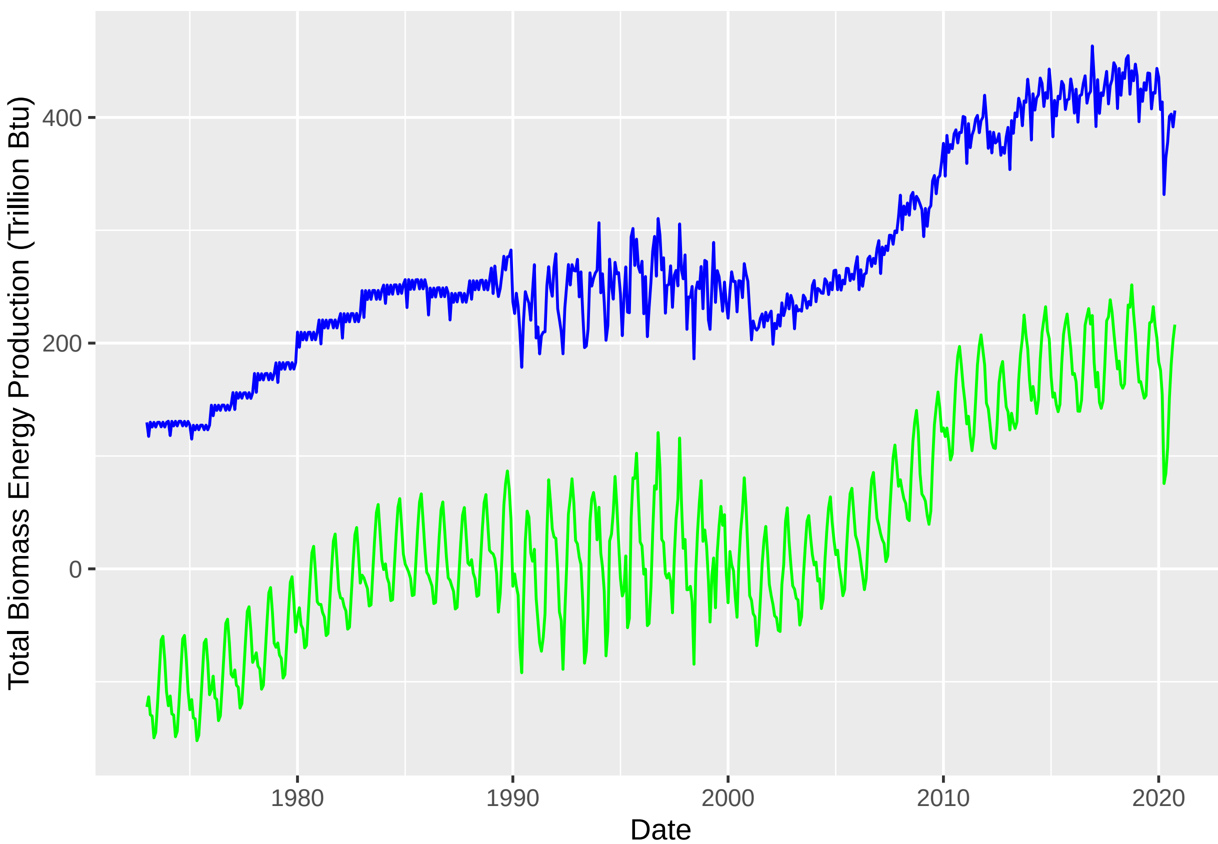
```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        580.912     24.406  23.802   <2e-16 ***
## Energy_dummiesJan   12.451     34.335   0.363   0.7170
## Energy_dummiesFeb  -38.964     34.335  -1.135   0.2569
## Energy_dummiesMar   20.515     34.335   0.597   0.5504
## Energy_dummiesApr    8.294     34.335   0.242   0.8092
## Energy_dummiesMay   36.628     34.335   1.067   0.2865
## Energy_dummiesJun   19.560     34.335   0.570   0.5691
## Energy_dummiesJul    8.863     34.335   0.258   0.7964
## Energy_dummiesAug  -18.480     34.335  -0.538   0.5906
## Energy_dummiesSep  -62.410     34.335  -1.818   0.0696 .
## Energy_dummiesOct  -42.649     34.335  -1.242   0.2147
## Energy_dummiesNov  -42.516     34.515  -1.232   0.2185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 167.3 on 562 degrees of freedom
## Multiple R-squared:  0.03244,    Adjusted R-squared:  0.01351
## F-statistic: 1.713 on 11 and 562 DF,  p-value: 0.06702
##
##
## Call:
## lm(formula = MonthlyData_subset[, (EnergyType + 1)] ~ Energy_dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -92.064 -22.897  -2.654  20.642  98.058
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        238.887      4.863  49.125  < 2e-16 ***
## Energy_dummiesJan   13.270      6.841   1.940  0.05291 .
## Energy_dummiesFeb   -8.133      6.841  -1.189  0.23499
## Energy_dummiesMar   20.442      6.841   2.988  0.00293 **
## Energy_dummiesApr   17.199      6.841   2.514  0.01221 *
## Energy_dummiesMay   40.726      6.841   5.953 4.64e-09 ***
## Energy_dummiesJun   31.764      6.841   4.643 4.28e-06 ***
## Energy_dummiesJul   10.858      6.841   1.587  0.11306
## Energy_dummiesAug  -17.907      6.841  -2.618  0.00909 **
## Energy_dummiesSep  -50.121      6.841  -7.326 8.26e-13 ***
## Energy_dummiesOct  -49.165      6.841  -7.187 2.12e-12 ***
## Energy_dummiesNov  -32.757      6.877  -4.763 2.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.34 on 562 degrees of freedom
## Multiple R-squared:  0.4345, Adjusted R-squared:  0.4234
## F-statistic: 39.25 on 11 and 562 DF,  p-value: < 2.2e-16
```

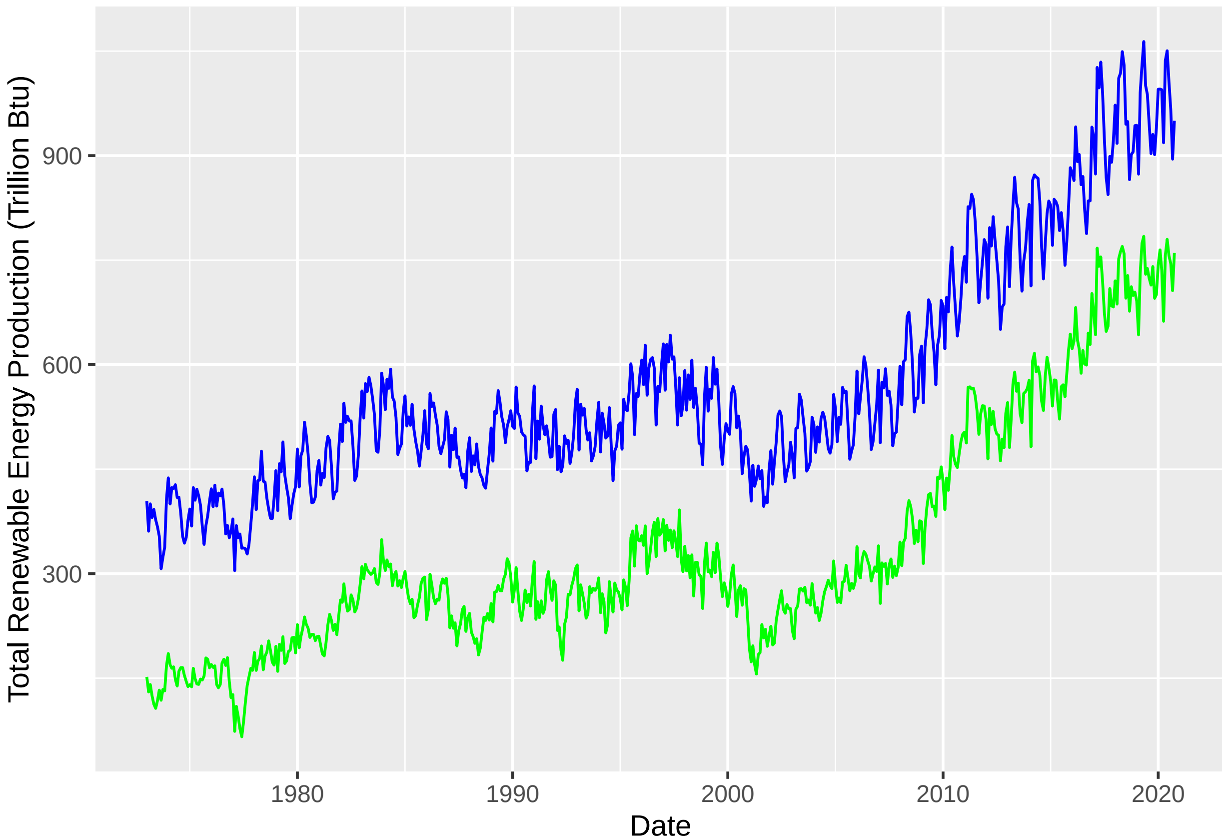
There series all seem to have a seasonal trend.

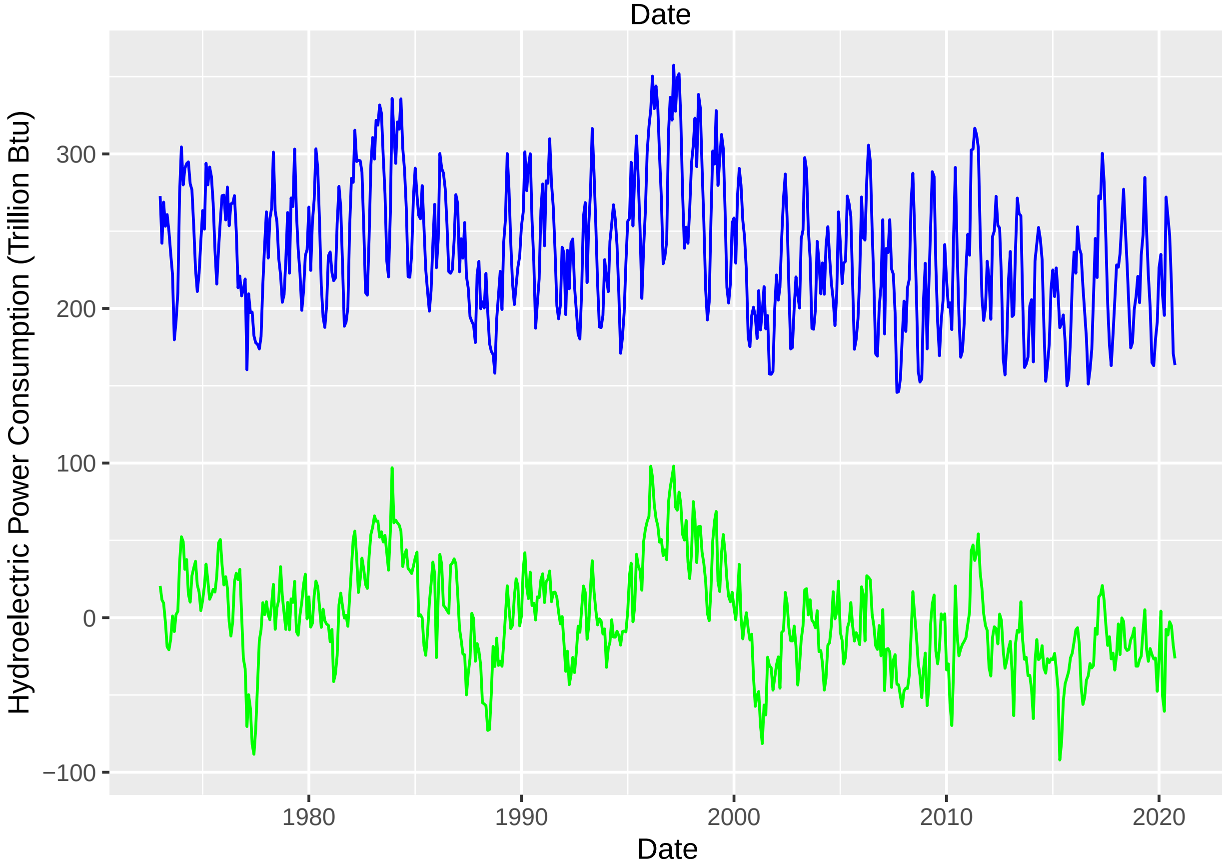
Q7

Use the regression coefficients from Q6 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

```
for(EnergyType in 1:ncoln){

  #Removing seasonal component
  Deseason_data <- MonthlyData_subset[,(1+EnergyType)]-seasonal_comp

  #Understanding what we did
  print(ggplot(MonthlyData_subset, aes(x=Date, y=MonthlyData_subset[,(1+EnergyType)])) +
            geom_line(color="blue") +
            ylab(paste0(colnames(MonthlyData_subset)[(1+EnergyType)]," (Trillion Btu)",sep="")) +
            geom_line(aes(y=Deseason_data), col="green"))
}
```







>First plot: the magnitude of the change increases compared to original one, which means has more obvious

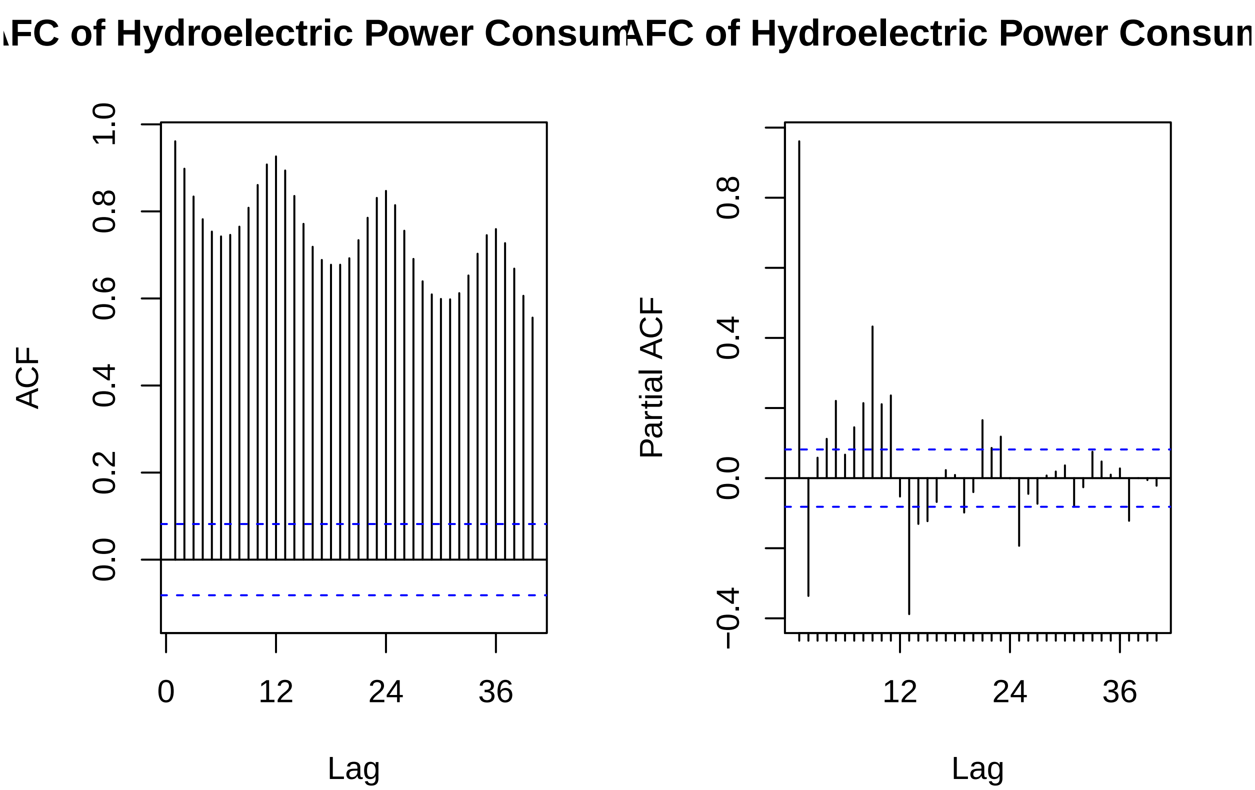
oscillations. The other two plots has less obvious oscillations.Moreover, the values are smaller than original ones.

Q8

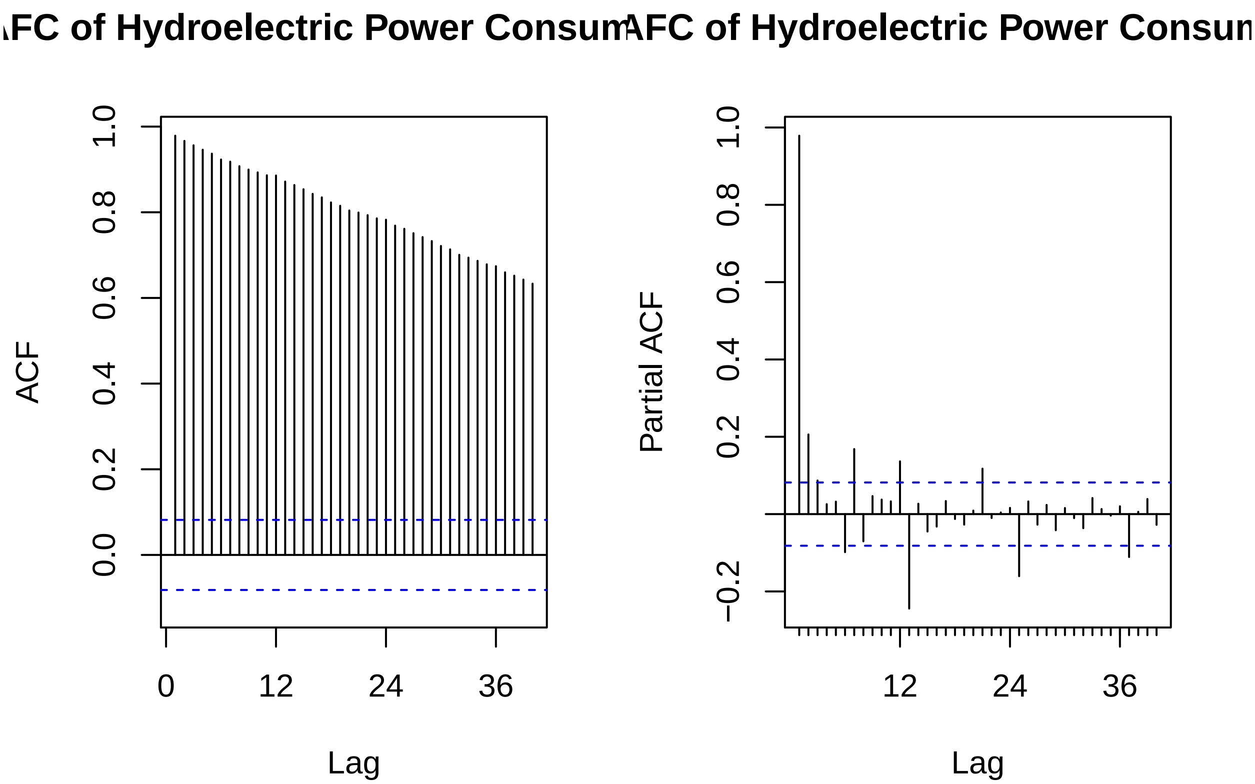
Plot ACF and PACF for the deseason series and compare with the plots from Q1. Did the plots change? How?

```
for(EnergyType in 1:ncoln){
  Deseason_data <- MonthlyData_subset[,(1+EnergyType)]-seasonal_comp
  str(Deseason_data)

  #change Deseason_data to TS
  Deseason_data_ts <- ts(Deseason_data, frequency = 12)

  #plot ACF and PACF for Detrend_data_ts
  par(mfrow=c(1,2))
  Acf(Deseason_data_ts,lag.max=40,main=paste("AFC of ",colnames(MonthlyData_subset)[(1+i)],sep=""))
  Pacf(Deseason_data_ts,lag.max=40,main=paste("PAFC of ",colnames(MonthlyData_subset)[(1+i)],sep=""))
}
```

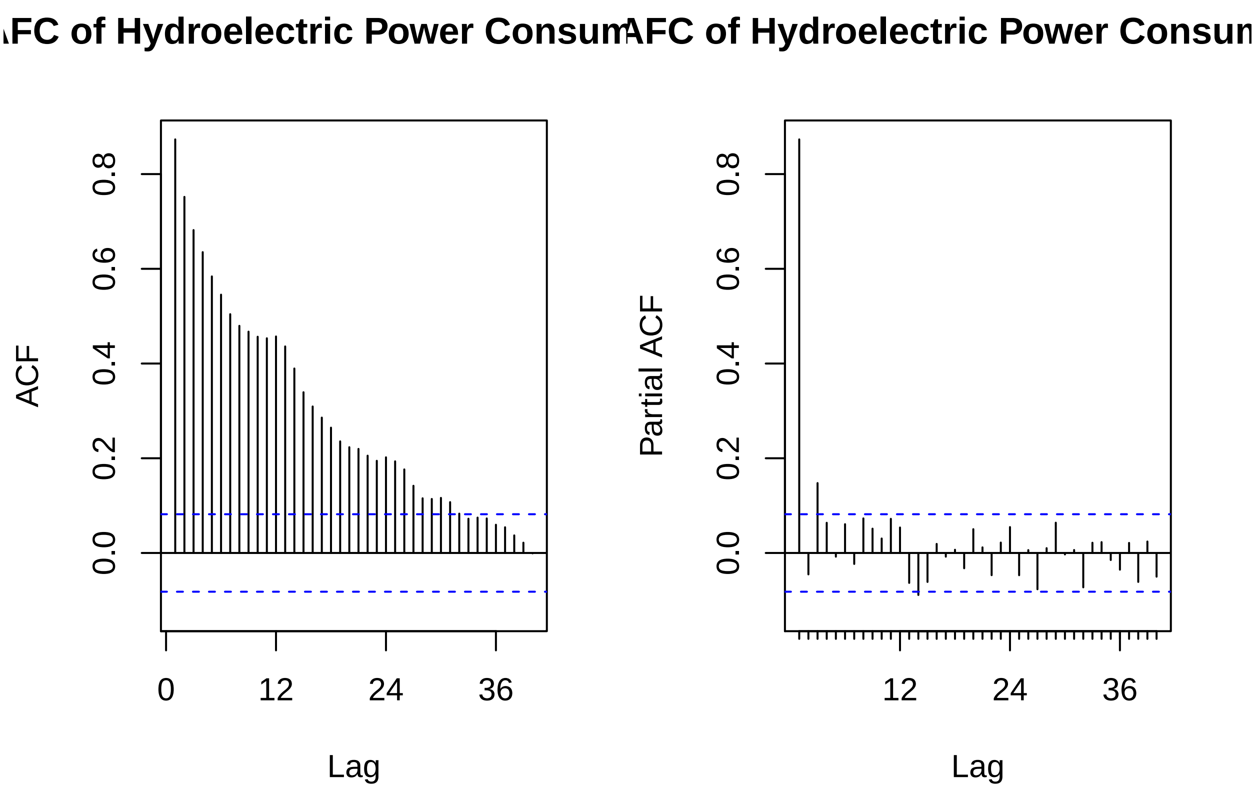
```
##  num [1:574(1d)] -122 -113 -129 -130 -150 ...
```




```
##  num [1:574(1d)] 152 130 141 124 113 ...
```




```
##  num [1:574(1d)] 20.55 11.45 9.48 -2.9 -18.84 ...
```



> The spike values of ACF in plot 1 and 2 are different from that of plots from Q1. Plot 1 has more obvious changes, but plot 2 is less obvious. > For Thrid plot, the value of ACF are all positive, which is total different from Q1.