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

Assignment 3 - Due date 02/10/23

Qiuying Liao

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 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_A02_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).

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. Submit this pdf using Sakai.

Questions

Consider the same data you used for A2 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. 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
library(forecast)
```

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

```
library(tseries)
library(Kendall)
library(readxl)
library(ggplot2)
```

##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: use par() function)

```
#data import
library(readxl)
energy_data <- read_excel(path="../Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sourc
```

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

```
#Extra column name
read_col_names <- read_excel(path="../Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_So
```

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

```
colnames(energy_data) <- read_col_names
head(energy_data)
```

```
## # A tibble: 6 x 14
##   Month               Wood Ene~1 Biofu~2 Total~3 Total~4 Hydro~5 Geoth~6 Solar~7
##   <dttm>                   <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl> <chr>
## 1 1973-01-01 00:00:00       130. Not Av~    130.    404.    273.    1.49 Not Av~
## 2 1973-02-01 00:00:00       117. Not Av~    117.    361.    242.    1.36 Not Av~
## 3 1973-03-01 00:00:00       130. Not Av~    130.    400.    269.    1.41 Not Av~
## 4 1973-04-01 00:00:00       125. Not Av~    126.    380.    253.    1.65 Not Av~
## 5 1973-05-01 00:00:00       130. Not Av~    130.    392.    261.    1.54 Not Av~
## 6 1973-06-01 00:00:00       125. Not Av~    126.    377.    250.    1.76 Not Av~
```

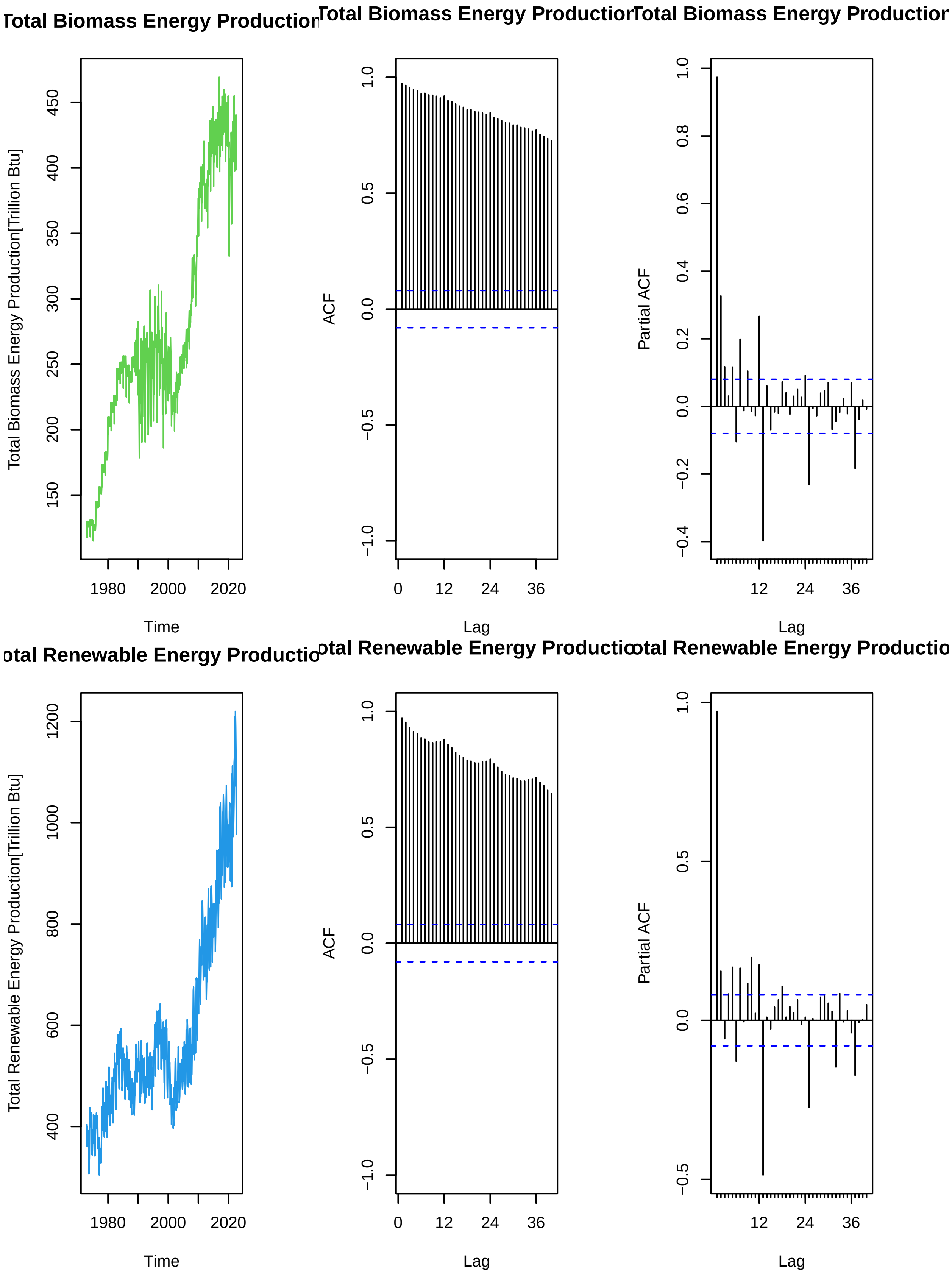
```
## # ... with 6 more variables: `Wind Energy Consumption` <chr>,
## #   `Wood Energy Consumption` <dbl>, `Waste Energy Consumption` <dbl>,
## #   `Biofuels Consumption` <chr>, `Total Biomass Energy Consumption` <dbl>,
## #   `Total Renewable Energy Consumption` <dbl>, and abbreviated variable names
## #   1: `Wood Energy Production`, 2: `Biofuels Production`,
## #   3: `Total Biomass Energy Production`,
## #   4: `Total Renewable Energy Production`, ...
```

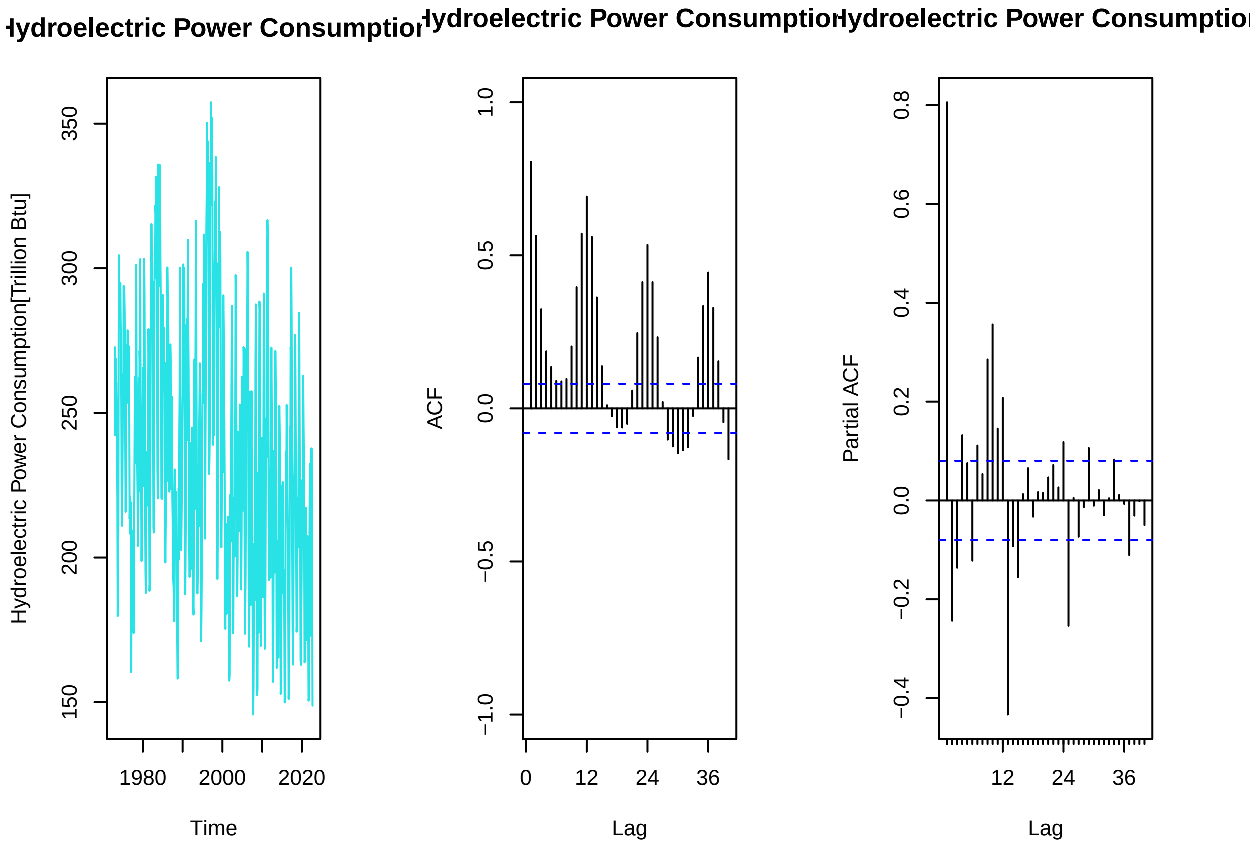
```r
#select columns for data frame
energy_data_df <- energy_data[,c("Month", "Total Biomass Energy Production","Total Renewable Energy Proc

head(energy_data_df)
```

```
## # A tibble: 6 x 4
##   Month               `Total Biomass Energy Production` Total Renewabl~1 Hydro~2
##   <dttm>                                          <dbl>            <dbl>   <dbl>
## 1 1973-01-01 00:00:00                              130.             404.    273.
## 2 1973-02-01 00:00:00                              117.             361.    242.
## 3 1973-03-01 00:00:00                              130.             400.    269.
## 4 1973-04-01 00:00:00                              126.             380.    253.
## 5 1973-05-01 00:00:00                              130.             392.    261.
## 6 1973-06-01 00:00:00                              126.             377.    250.
## # ... with abbreviated variable names 1: `Total Renewable Energy Production`,
## #   2: `Hydroelectric Power Consumption`
```

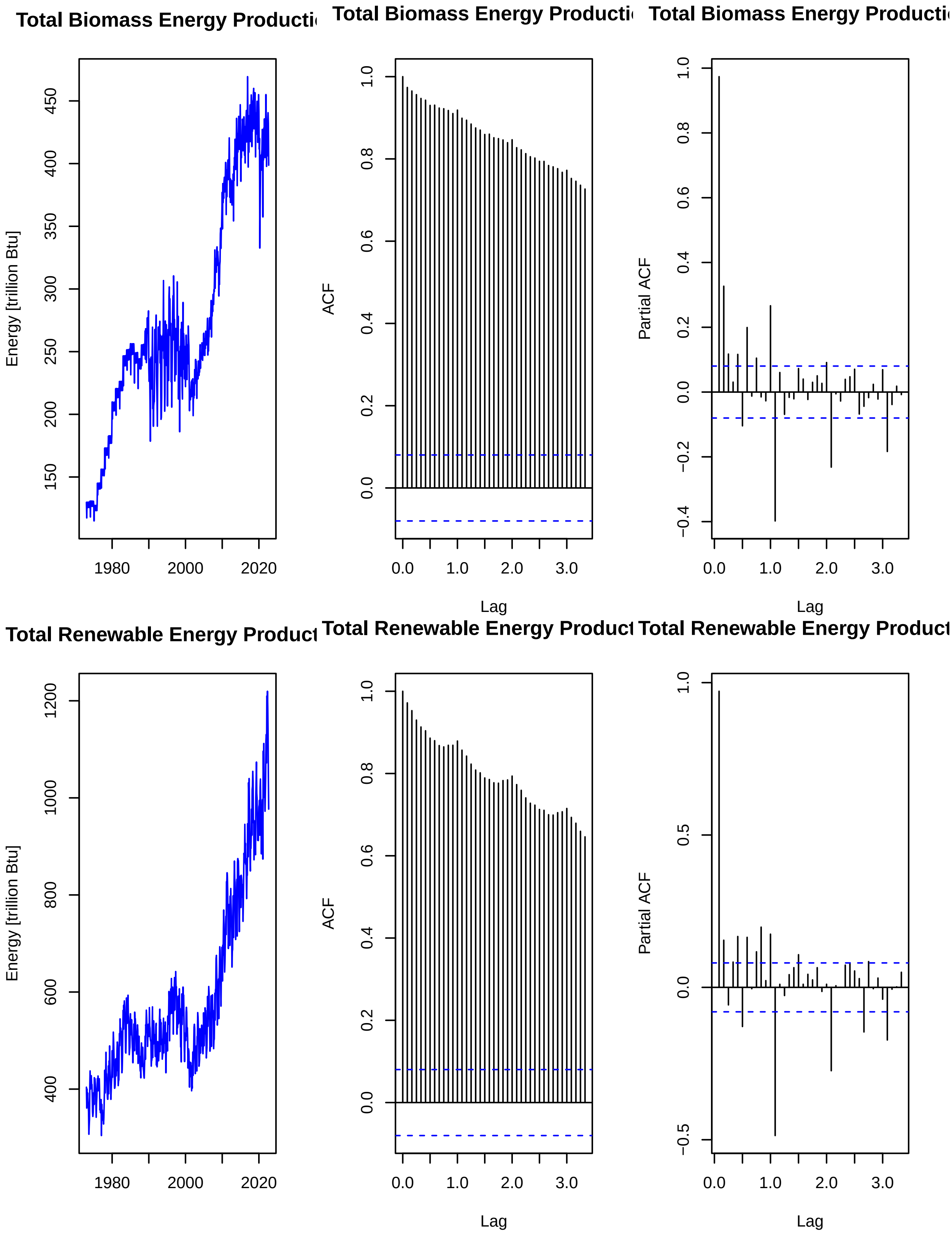
```r
#create time series
ts_energy_data_df <- ts(energy_data_df[,2:4], start = c(1973,1), frequency =12)
```

```r
#acf and pacf

col_names <- colnames(energy_data_df)
cc <- palette()

for(i in 1:3){
  par(mfrow = c(1,3), mar = c(4,4,4,4))
  plot(ts_energy_data_df[,i], type="l", col=cc[i+2], ylab = paste0(col_names[i+1], "[Trillion Btu]"), ma
  Acf(ts_energy_data_df[,i], lag.max = 40, main = paste0(col_names[i+1]), ylim=c(-1,1))
   Pacf(ts_energy_data_df[,i], lag.max = 40, main = paste0(col_names[i+1]))
}
```

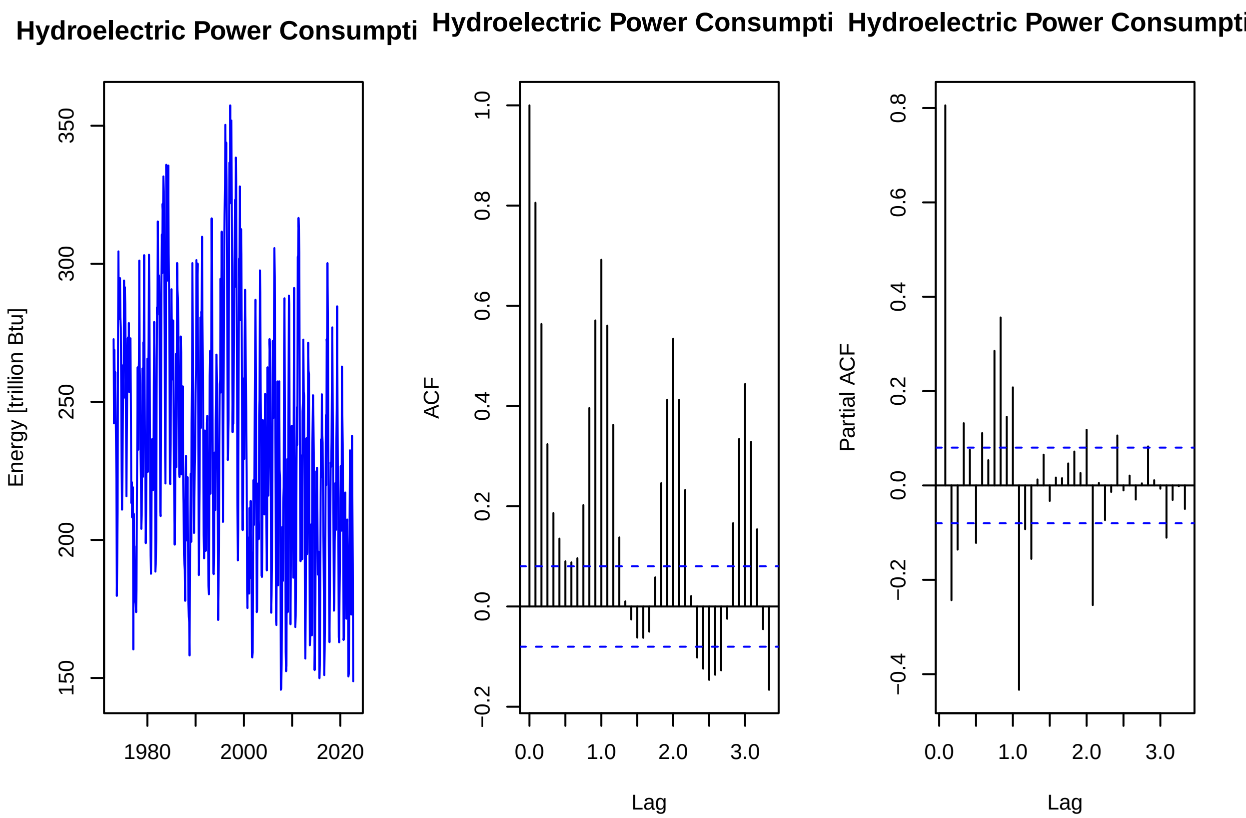

Total Biomass Energy Production
Total Biomass Energy Production
Total Biomass Energy Production
Total Biomass Energy Production[Trillion Btu]
Time
ACF
Lag
Partial ACF
Lag
1980
2000
2020
0
12
24
36
otal Renewable Energy Productio
otal Renewable Energy Productio
otal Renewable Energy Productio
Total Renewable Energy Production[Trillion Btu]
Time
ACF
Lag
Partial ACF
Lag




```r
#Luana's solution
par(mfrow=c(1,3))
for(i in 1:3){
  plot(energy_data_df$Month, ts_energy_data_df[,i], type="l", col= "blue", ylab="Energy [trillion Btu]"
  acf(ts_energy_data_df[,i], lag.max = 40, plot=TRUE, main=colnames(ts_energy_data_df)[i])
  pacf(ts_energy_data_df[,i], lag.max = 40, plot=TRUE, main=colnames(ts_energy_data_df)[i])
}
```



Total Biomass Energy Producti
Energy [trillion Btu]
450
400
350
300
250
200
150
1980
2000
2020
Total Biomass Energy Producti
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
1.0
2.0
3.0
Lag
Total Biomass Energy Producti
Partial ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
1.0
2.0
3.0
Lag
Total Renewable Energy Product
Energy [trillion Btu]
1200
1000
800
600
400
1980
2000
2020
Total Renewable Energy Product
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
1.0
2.0
3.0
Lag
Total Renewable Energy Product
Partial ACF
1.0
0.5
0.0
−0.5
0.0
1.0
2.0
3.0
Lag




```
#renewable: pacf have strike- indicate have seasonal component
#total renewable: have positive trend, there might be some seasonality from acf. Pacf telling us a litt
#Biomass: increasing trend, but can's say many thing about seasonality
```

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 total biomass energy production and total renewable energy production have a similar trend that both graph show a general upward trend. Both graph show that the values fluctuate before and increase sharply after around 2005. Hydroelectric power consumption appear to have a general downward trend and has high fluctuation. In terms of acf graph, total biomass energy production and total renewable energy production show nonstationarity and downward trend while hydroelectric power consumption shows stationarity in the time series. The autocorrelation is still strong at lag 36 for all three series since the value is above the blue line. In terms of pacf graph, there are positive and negative values in all series. However, most values are within the blue line as lag increases, showing a weak correlation and may not be significant.

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.

For biomass energy production, the slope is 0.48 and the intercept is 133.7; for renewable energy production, the slope is 0.94 and the intercept is 312.3; for hydroelectric power consumption, the slope is -0.083 and the intercept is 259.9.

Both biomass energy production and renewable energy production have a positive slope, showing that as time increases in 1 unit, it results in increases in biomass/renewable energy production for 0.48 unit and 0.94 unit, respectively. The expected mean value of Y when all X= 0 are 133.7 Trillion Btu and 312.3 Trillion Btu for biomass and renewable energy production, respectively. The p-value is less than 0.05, which shows that the values of slope and intercept are statistically significant. On the other hand, Hydroelectric consumption have a negative slop, showing that as time increase in 1 unit, the hydroelectric power consumption decreases for 0.083 unit. The expected mean value of Y when all X=0 is 259.9 Trillion Btu. The p-value is less than 0.05, showing that the values of slope and intercept are statistically significant.

```
#Fit a linear trend to TS
nobs <- nrow(energy_data_df)
t <- 1:nobs

#biomass
linear_trend <- lm(ts_energy_data_df[,1] ~ t)
summary(linear_trend)
```

```
##
## Call:
## lm(formula = ts_energy_data_df[, 1] ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.800  -23.994    5.667   32.265   82.192
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.337e+02  3.245e+00   41.22   <2e-16 ***
## t           4.800e-01  9.402e-03   51.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.59 on 595 degrees of freedom
## Multiple R-squared:  0.8142, Adjusted R-squared:  0.8138
## F-statistic:  2607 on 1 and 595 DF,  p-value: < 2.2e-16
```

```
#Renewable
linear_trend2 <- lm(ts_energy_data_df[,2] ~ t)
summary(linear_trend2)
```

```
##
## Call:
## lm(formula = ts_energy_data_df[, 2] ~ 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
```

```
#Hydroelectric
linear_trend3 <- lm(ts_energy_data_df[,3] ~ t)
summary(linear_trend3)
```

```
##
## Call:
## lm(formula = ts_energy_data_df[, 3] ~ t)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -95.42 -31.20  -2.56  27.32 121.61
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 259.898013   3.427300  75.832  < 2e-16 ***
## t            -0.082888   0.009931  -8.346 4.94e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.82 on 595 degrees of freedom
## Multiple R-squared:  0.1048, Adjusted R-squared:  0.1033
## F-statistic: 69.66 on 1 and 595 DF,  p-value: 4.937e-16
```

```
#solution:

#Renewable: have a trend
#hydroelectric: low completely difference from the coefficient (-0.082888) compared to renewable (0.936
```

Q4

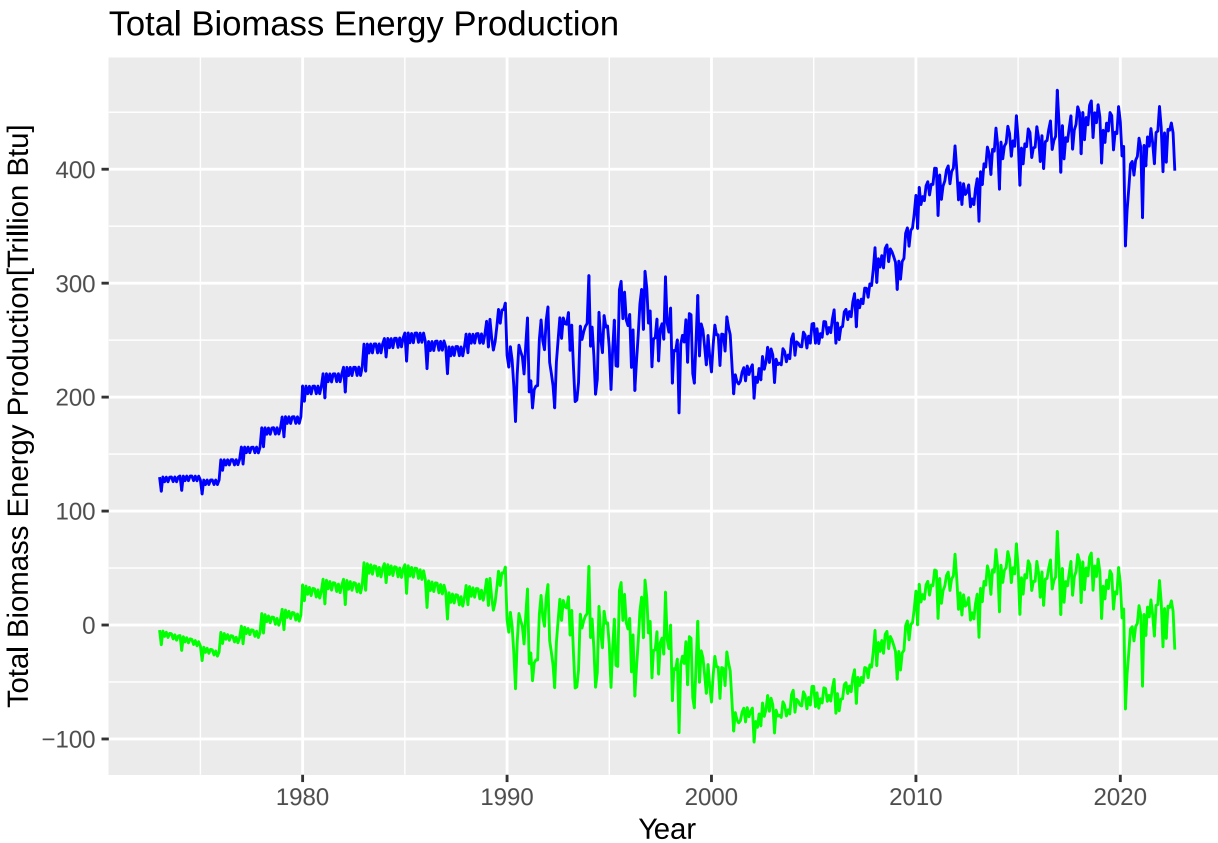
Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

The detrended series have values around 0 throughout the years but show almost same patterns as the original series. The biomass and renewable energy production have upward trend, but the detrended series have constant value through the years, which means there is no trend for the detrended series. The hydroelectric power consumption has slightly downward trend but the detrended series have constant value through the years.

```
#Biomass
beta0 <- linear_trend$coefficients[1]
beta1 <- linear_trend$coefficients[2]

detrend_energy_data_df <- ts_energy_data_df[,1]-(beta0+beta1*t)

ggplot(energy_data_df, aes(x=Month, y=ts_energy_data_df[,1]))+
  geom_line(color="blue")+
  geom_line(aes(y=detrend_energy_data_df),color="green")+
  ylab(paste0(colnames(energy_data_df)[(2)],sep="","[Trillion Btu]"))+
  xlab("Year")+
  ggtitle(paste0(colnames(energy_data_df)[(2)],sep=""))
```

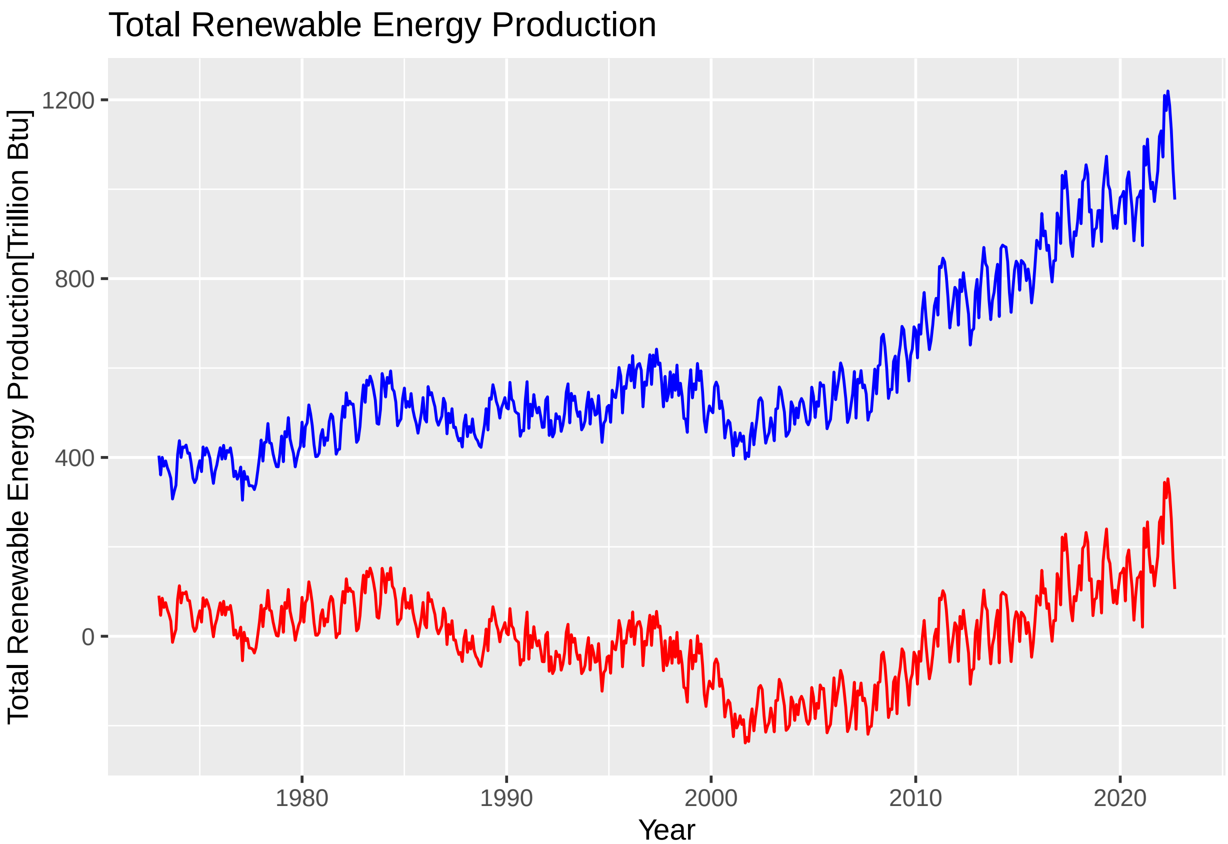
```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
```

```
## to continuous.
```



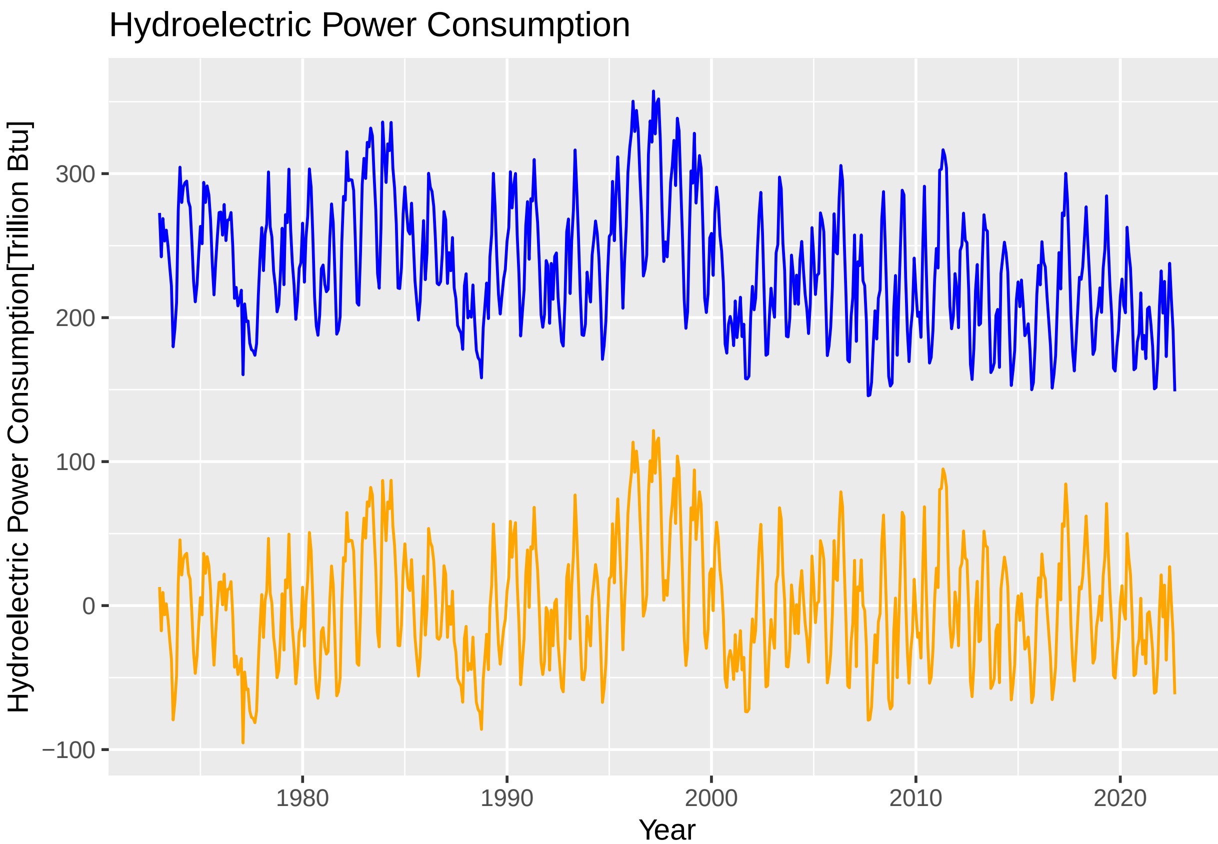

```
#Renewable
beta0_2 <- linear_trend2$coefficients[1]
beta1_2 <- linear_trend2$coefficients[2]

detrend_energy_data_df2 <- ts_energy_data_df[,2]-(beta0_2+beta1_2*t)

ggplot(energy_data_df, aes(x=Month, y=ts_energy_data_df[,2]))+
  geom_line(color="blue")+
  geom_line(aes(y=detrend_energy_data_df2),color="red")+
  ylab(paste0(colnames(energy_data_df)[(3)],sep="","[Trillion Btu]"))+
  xlab("Year")+
  ggtitle(paste0(colnames(energy_data_df)[(3)],sep=""))
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```




```r
#hydroelectric
beta0_3 <- linear_trend3$coefficients[1]
beta1_3 <- linear_trend3$coefficients[2]

detrend_energy_data_df3 <- ts_energy_data_df[,3]-(beta0_3+beta1_3*t)

ggplot(energy_data_df, aes(x=Month, y=ts_energy_data_df[,3]))+
  geom_line(color="blue")+
  geom_line(aes(y=detrend_energy_data_df3),color="orange")+
  ylab(paste0(colnames(energy_data_df)[(4)],sep="","[Trillion Btu]"))+
  xlab("Year")+
  ggtitle(paste0(colnames(energy_data_df)[(4)],sep=""))
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```




```
#Luana's solution
#biomass: the increasing trend is gone, but there is still trending at detrending. Maybe linear model i
#renewable: something similar happening again. we create a different trend when we use linear model
#hydroelectric : detrending is similar to the original one.

#The coefficient tell us a lot: if coefficient is equal to 1, there is stochastic trending yet.
```

Q5

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

The acf plot of biomass energy production changes that there is spike during 12, 24, and 36 lags. While it is not obvious, it towards to a stationary series that reflect seasonality pattern. The acf plot of renewable energy production changes that there is spike during 12, 24, and 36 lags, it shows stationary series that reflect seasonality pattern, which is different from plots from Q1. The acf of electric power consumption is still a stationary series, but the seasonality pattern is more obvious than the original plot. For example, the value reaches zero around 10 lags. The pacf for all three plots are similar than the original plot from Q1.

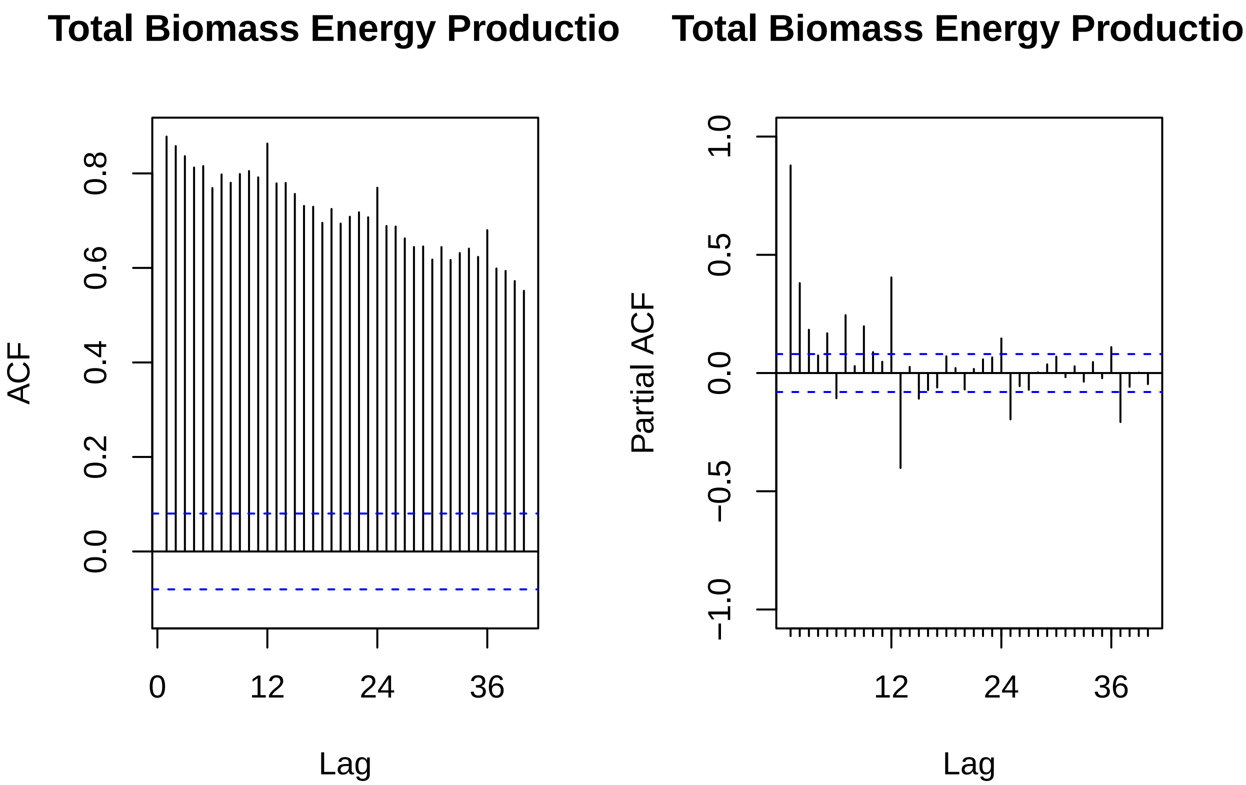
```
col_names <- colnames(energy_data_df)

#Biomass
par(mfrow=c(1,2))
Acf(detrend_energy_data_df,lag.max = 40,main=paste0(col_names[2]),ylaim=c(-1,1))
```

```
## Warning in plot.window(...): "ylaim" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "ylaim" is not a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in box(...): "ylaim" is not a graphical parameter

## Warning in title(...): "ylaim" is not a graphical parameter
```

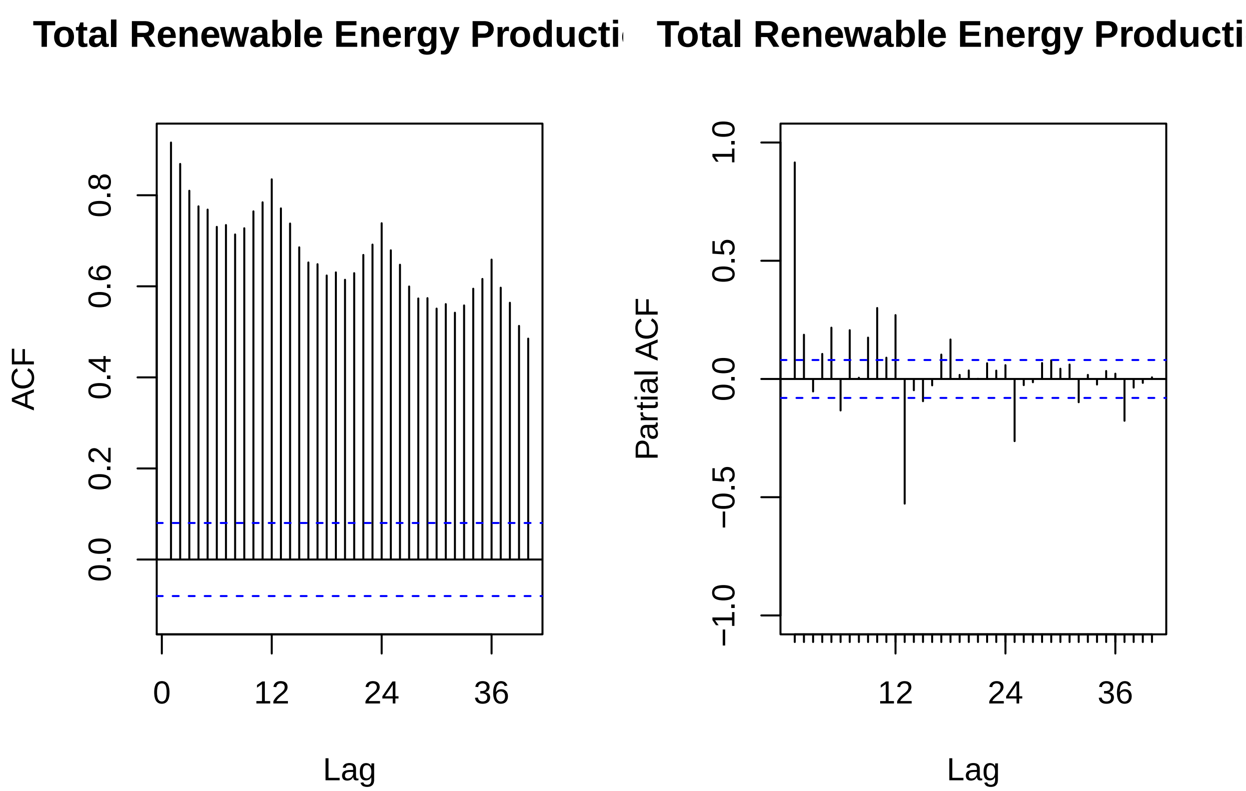
```
Pacf(detrend_energy_data_df,lag.max = 40,main=paste0(col_names[2]),ylim=c(-1,1))
```




```
#Renewable
par(mfrow=c(1,2))
Acf(detrend_energy_data_df2,lag.max = 40,main=paste0(col_names[3]),ylaim=c(-1,1))
```

```
## Warning in plot.window(...): "ylaim" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "ylaim" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in box(...): "ylaim" is not a graphical parameter

## Warning in title(...): "ylaim" is not a graphical parameter
```

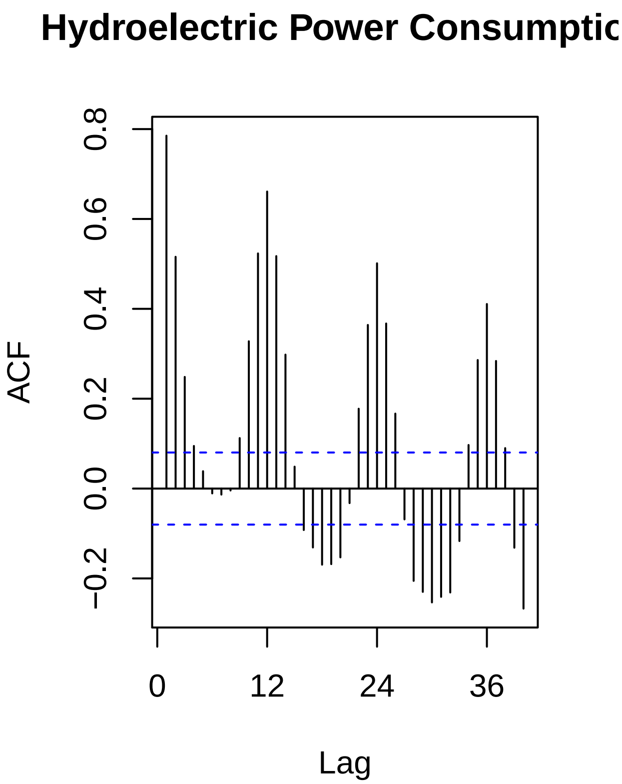
```
Pacf(detrend_energy_data_df2,lag.max = 40,main=paste0(col_names[3]),ylim=c(-1,1))
```

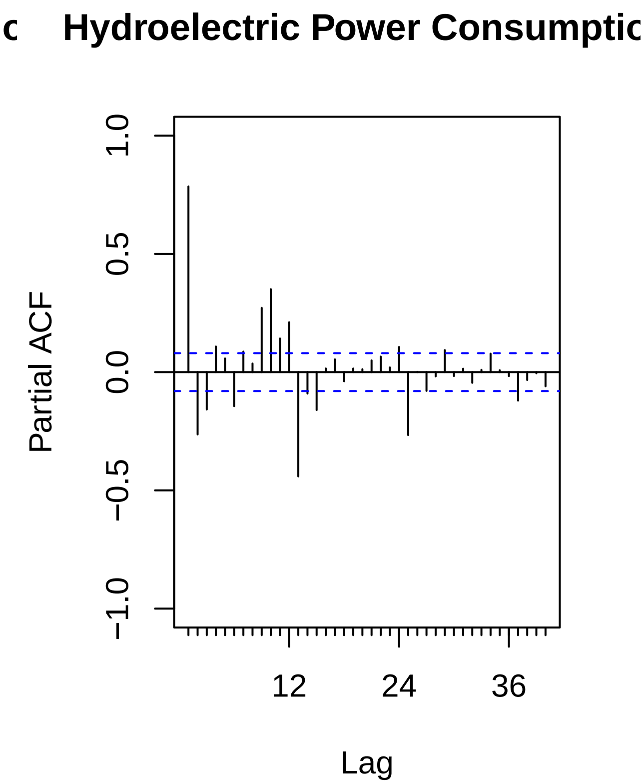



```
#Hydroelectric
par(mfrow=c(1,2))
Acf(detrend_energy_data_df3,lag.max = 40,main=paste0(col_names[4]),ylaim=c(-1,1))
```

```
## Warning in plot.window(...): "ylaim" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "ylaim" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in box(...): "ylaim" is not a graphical parameter

## Warning in title(...): "ylaim" is not a graphical parameter
```

```
Pacf(detrend_energy_data_df3,lag.max = 40,main=paste0(col_names[4]),ylim=c(-1,1))
```






```
#luana's solution
#biomass : the value of acf decrease when we detrend ; the detrending only change the pacf plot
#renewable: similar trend, eliminate some of the trend, but still have some correlations between the pr
#hydroelectric: no trend, we didn't change the behavior of independent of the detrend part
```

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 (i.e. using the seasonal dummies) to this/these time series. Ask R to print the summary of the regression. Interpret the regression output. Save the ri for further analysis.

hydroelectric power consumption series seem to have a seasonal trend (stationarity) while the other two don't (nonstationarity).

there is only 11 dummies because it drops one (December) to compare with. The intercept indicate the value of December. For biomass energy production and renewable energy production, all p-values are larger than 0.05, showing that the values are not statistically significant. However, we found seasonality (spikes) in acf for detrending biomass and renewable energy production. This shows that season might have some effects, but not constant. For electric power consumption, the p-value is less than 0.05, showing that values are statistically significant. Therefore, there is seasonal trend in hydroelectric power consumption series.

```
#Use seasonal means model

#Biomass
dummies <- seasonaldummy(ts_energy_data_df[,1])
```

```r
seas_means_model <- lm(ts_energy_data_df[,(1)]~dummies)
summary(seas_means_model)
```

```
##
## Call:
## lm(formula = ts_energy_data_df[, (1)] ~ dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -160.74  -53.67  -24.36   90.73  181.34
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  288.020     13.163  21.881   <2e-16 ***
## dummiesJan    -1.793     18.522  -0.097   0.9229
## dummiesFeb   -31.102     18.522  -1.679   0.0936 .
## dummiesMar    -9.104     18.522  -0.492   0.6232
## dummiesApr   -21.502     18.522  -1.161   0.2462
## dummiesMay   -14.238     18.522  -0.769   0.4424
## dummiesJun   -19.602     18.522  -1.058   0.2904
## dummiesJul    -3.674     18.522  -0.198   0.8428
## dummiesAug    -0.612     18.522  -0.033   0.9737
## dummiesSep   -13.335     18.522  -0.720   0.4718
## dummiesOct    -4.030     18.615  -0.216   0.8287
## dummiesNov    -9.849     18.615  -0.529   0.5970
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.14 on 585 degrees of freedom
## Multiple R-squared:  0.01018,    Adjusted R-squared:  -0.008437
## F-statistic: 0.5467 on 11 and 585 DF,  p-value: 0.8714
```

```r
#Renewable
dummies2 <- seasonaldummy(ts_energy_data_df[,2])
seas_means_model2 <- lm(ts_energy_data_df[,(2)]~dummies2)
summary(seas_means_model2)
```

```
##
## Call:
## lm(formula = ts_energy_data_df[, (2)] ~ dummies2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -284.92 -122.23  -68.42   91.22  585.68
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  601.022     27.260  22.048   <2e-16 ***
## dummies2Jan   11.468     38.358   0.299    0.765
## dummies2Feb  -41.456     38.358  -1.081    0.280
## dummies2Mar   23.130     38.358   0.603    0.547
## dummies2Apr    9.959     38.358   0.260    0.795
## dummies2May   38.853     38.358   1.013    0.312
## dummies2Jun   20.378     38.358   0.531    0.595
```

```
## dummies2Jul    8.298     38.358   0.216    0.829
## dummies2Aug  -19.450     38.358  -0.507    0.612
## dummies2Sep  -63.770     38.358  -1.662    0.097 .
## dummies2Oct  -52.612     38.551  -1.365    0.173
## dummies2Nov  -42.537     38.551  -1.103    0.270
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 190.8 on 585 degrees of freedom
## Multiple R-squared:  0.02844,    Adjusted R-squared:  0.01017
## F-statistic: 1.557 on 11 and 585 DF,  p-value: 0.1076
```

```
#Hydroelectric
dummies3 <- seasonaldummy(ts_energy_data_df[,3])
seas_means_model3 <- lm(ts_energy_data_df[,3]~dummies3)
summary(seas_means_model3)
```

```
##
## Call:
## lm(formula = ts_energy_data_df[, 3] ~ dummies3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -88.99 -23.47  -2.81  21.99 100.18
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  237.225      4.878  48.634  < 2e-16 ***
## dummies3Jan   13.594      6.864   1.981  0.04811 *
## dummies3Feb   -8.254      6.864  -1.203  0.22964
## dummies3Mar   19.980      6.864   2.911  0.00374 **
## dummies3Apr   15.649      6.864   2.280  0.02297 *
## dummies3May   39.210      6.864   5.713 1.77e-08 ***
## dummies3Jun   31.209      6.864   4.547 6.61e-06 ***
## dummies3Jul   10.436      6.864   1.520  0.12895
## dummies3Aug  -17.909      6.864  -2.609  0.00931 **
## dummies3Sep  -50.173      6.864  -7.310 8.82e-13 ***
## dummies3Oct  -48.262      6.898  -6.996 7.22e-12 ***
## dummies3Nov  -32.285      6.898  -4.680 3.56e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.14 on 585 degrees of freedom
## Multiple R-squared:  0.4132, Adjusted R-squared:  0.4022
## F-statistic: 37.45 on 11 and 585 DF,  p-value: < 2.2e-16
```

```
#luana's solution
#pvlaue is large: there is no constant seasonal component on your series. There might be a little bit;
```

Q7

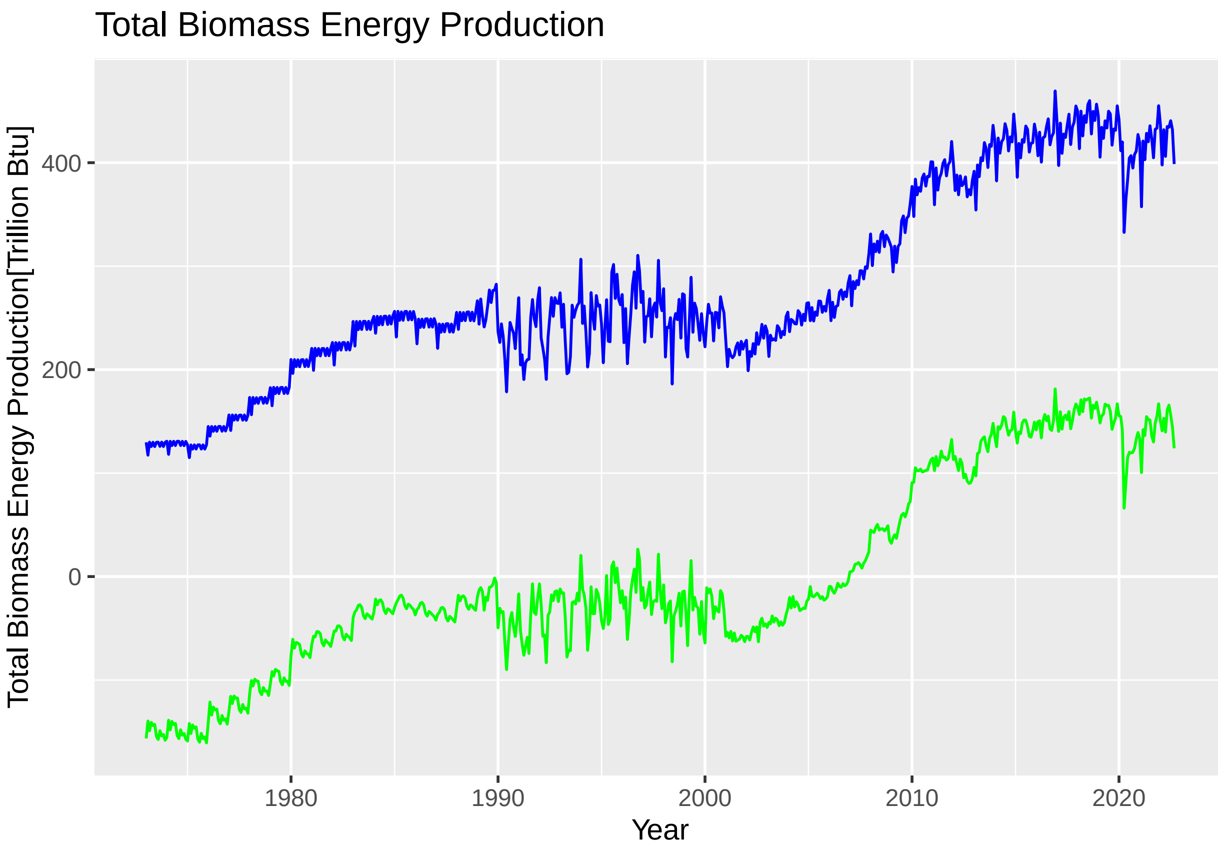
Use the regression coefficients from Q6 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

The deseason series have lower values (around 0 or less than 0) throughout the years but show almost same patterns and trends as the original series for biomass energy production and

renwable energy production. However, the deseason series have similar trend (downward) but different patterns than the original series for hydroelectric power consumption.

```
#Biomass
beta_int=seas_means_model$coefficients[1]
beta_coeff=seas_means_model$coefficients[2:12]

energy_seas_comp=array(0,nobs)
for(i in 1:nobs){
  energy_seas_comp[i]=(beta_int+beta_coeff%*%dummies[i,])
}

deseason_energy_data <- ts_energy_data_df[,(1)]-energy_seas_comp

ggplot(energy_data_df, aes(x=Month, y=ts_energy_data_df[,(1)])) +
            geom_line(color="blue") +
            ylab(paste0(colnames(energy_data_df)[(2)],sep="")) +
            geom_line(aes(y=deseason_energy_data), col="green")+
  ylab(paste0(colnames(energy_data_df)[(2)],sep="","[Trillion Btu]"))+
  xlab("Year")+
  ggtitle(paste0(colnames(energy_data_df)[(2)],sep=""))
```

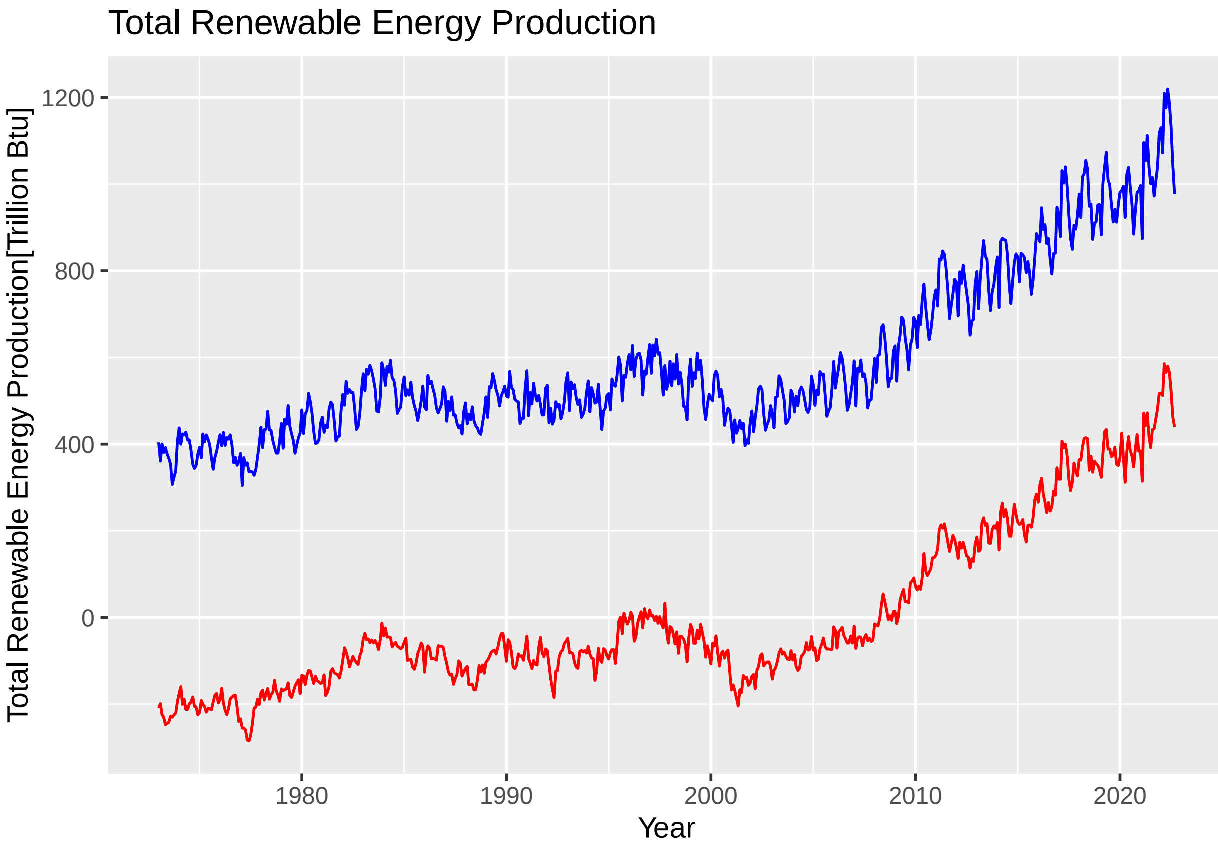
```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```




```
#Renewable
beta_int2=seas_means_model2$coefficients[1]
beta_coeff2=seas_means_model2$coefficients[2:12]
```

```r
energy_seas_comp2=array(0,nobs)
for(i in 1:nobs){
  energy_seas_comp2[i]=(beta_int2+beta_coeff2%*%dummies2[i,])
}

deseason_energy_data2 <- ts_energy_data_df[,(2)]-energy_seas_comp2

ggplot(energy_data_df, aes(x=Month, y=ts_energy_data_df[,(2)])) +
            geom_line(color="blue") +
            ylab(paste0(colnames(energy_data_df)[(3)],sep="")) +
            geom_line(aes(y=deseason_energy_data2), col="red")+
   ylab(paste0(colnames(energy_data_df)[(3)],sep="","[Trillion Btu]"))+
  xlab("Year")+
  ggtitle(paste0(colnames(energy_data_df)[(3)],sep=""))
```

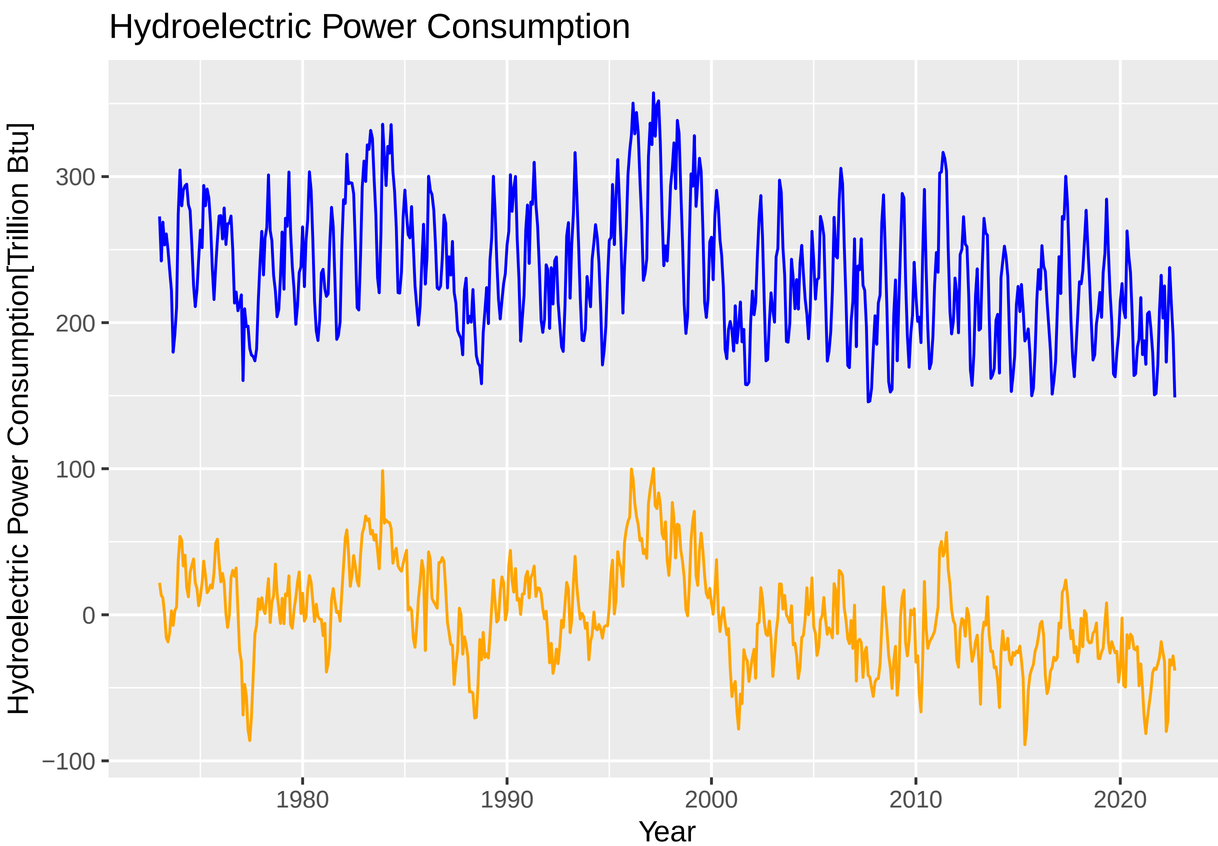
```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```




```r
#Hydroelectric
beta_int3=seas_means_model3$coefficients[1]
beta_coeff3=seas_means_model3$coefficients[2:12]

energy_seas_comp3=array(0,nobs)
for(i in 1:nobs){
  energy_seas_comp3[i]=(beta_int3+beta_coeff3%*%dummies3[i,])
}
```

```
deseason_energy_data3 <- ts_energy_data_df[,(3)]-energy_seas_comp3

ggplot(energy_data_df, aes(x=Month, y=ts_energy_data_df[,(3)])) +
            geom_line(color="blue") +
            ylab(paste0(colnames(energy_data_df)[(4)],sep="")) +
            geom_line(aes(y=deseason_energy_data3), col="orange")+
   ylab(paste0(colnames(energy_data_df)[(4)],sep="","[Trillion Btu]"))+
  xlab("Year")+
  ggtitle(paste0(colnames(energy_data_df)[(4)],sep=""))
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```



Q8

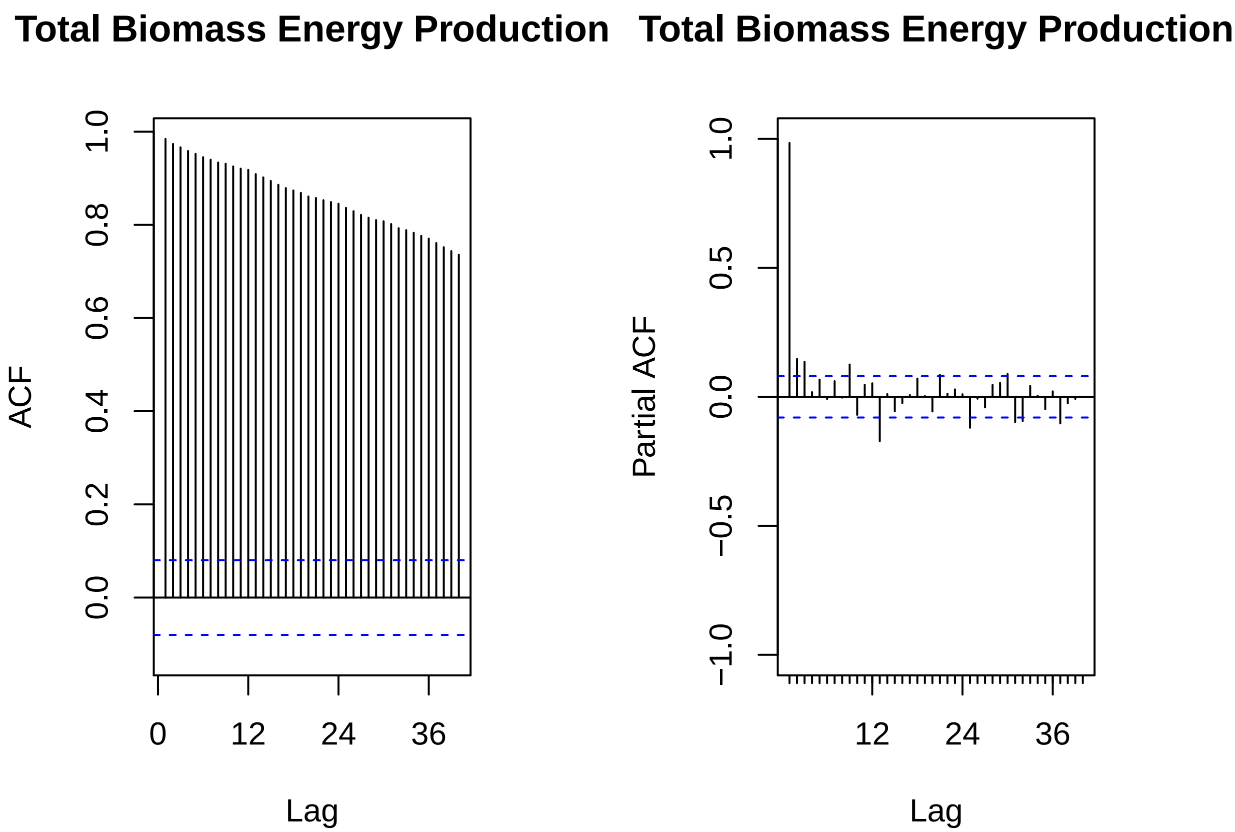
Plot ACF and PACF for the deseason series and compare with the plots from Q1. Did the plots change? How?

For biomass and renewable energy production, the acf for the deseason series also show nonstationarity and have decreased trend. There is a small upward trends in 12, 24, 36 lags in original plot of renewable energy production, but the trend decreases at all in acf of deseason series. All values are positive (above 0) in acf. For hydroelectric power consumption, it changes a lot that the acf of deseason series shows nonstationarity and have decreased trend, which is very different from the original plot that is stationary. This reflects that season has impact on hydroelectric power consumption. The pacf also changes that most values are within the blue line for the deseason series for all three components, showing a weak correlation and may not be significant.

```
#Biomass
par(mfrow=c(1,2),mar = c(4,4,4,4))
Acf(deseason_energy_data,lag.max = 40,main=paste0(col_names[2]),ylaim=c(-1,1))
```

```
## Warning in plot.window(...): "ylaim" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "ylaim" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in box(...): "ylaim" is not a graphical parameter

## Warning in title(...): "ylaim" is not a graphical parameter
```

```
Pacf(deseason_energy_data,lag.max = 40,main=paste0(col_names[2]),ylim=c(-1,1))
```

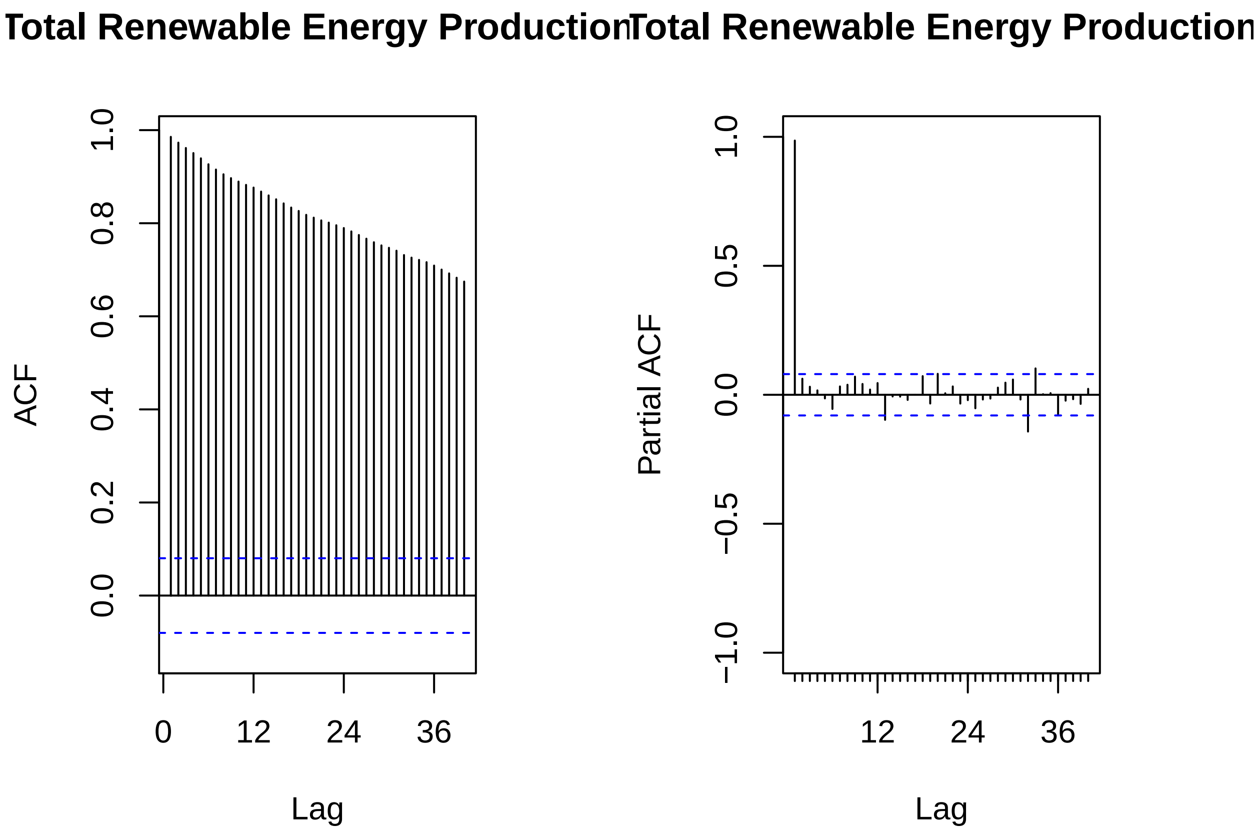



```
#Renewable
par(mfrow=c(1,2))
Acf(deseason_energy_data2,lag.max = 40,main=paste0(col_names[3]),ylaim=c(-1,1))
```

```
## Warning in plot.window(...): "ylaim" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "ylaim" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
```

```
## graphical parameter

## Warning in box(...): "ylaim" is not a graphical parameter

## Warning in title(...): "ylaim" is not a graphical parameter
```

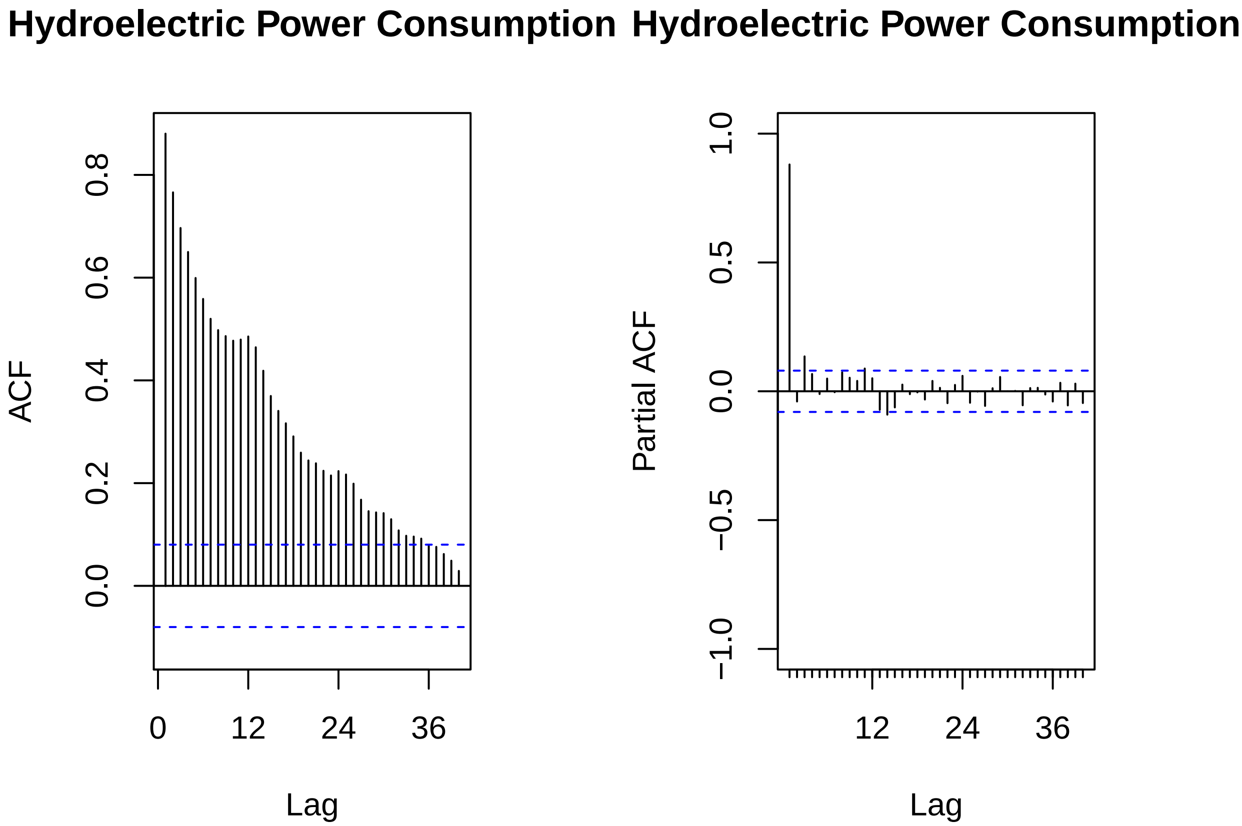
```
Pacf(deseason_energy_data2,lag.max = 40,main=paste0(col_names[3]),ylim=c(-1,1))
```




```
#Hydroelectric
par(mfrow=c(1,2))
Acf(deseason_energy_data3,lag.max = 40,main=paste0(col_names[4]),ylaim=c(-1,1))
```

```
## Warning in plot.window(...): "ylaim" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "ylaim" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylaim" is not a
## graphical parameter

## Warning in box(...): "ylaim" is not a graphical parameter

## Warning in title(...): "ylaim" is not a graphical parameter
```

```
Pacf(deseason_energy_data3,lag.max = 40,main=paste0(col_names[4]),ylim=c(-1,1))
```




```
#Luana 's solution
#hydroelectric: pacf: after removing season, there is only 1 spike (deseason) which is what we want
```