ENV 790.30 - Time Series Analysis for Energy Data | Spring 2022

Assignment 3 - Due date 02/08/22

Ben Joseph

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_A03_Sp22.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_Consumpti The data comes from the US Energy Information and Administration and corresponds to the January 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.\

```
library(forecast)
```

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

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

```r
df <- read_excel("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx"
   , sheet = 1, col_names = TRUE, skip =10, na="Not Available")
df2 <- df[2:586,4:6]
df2$`Total Biomass Energy Production`<- as.numeric(df2$`Total Biomass Energy Production`)
df2$`Total Renewable Energy Production`<- as.numeric(df2$`Total Renewable Energy Production`)
df2$`Hydroelectric Power Consumption`<-as.numeric(df2$`Hydroelectric Power Consumption`)
head(df2,10)
```

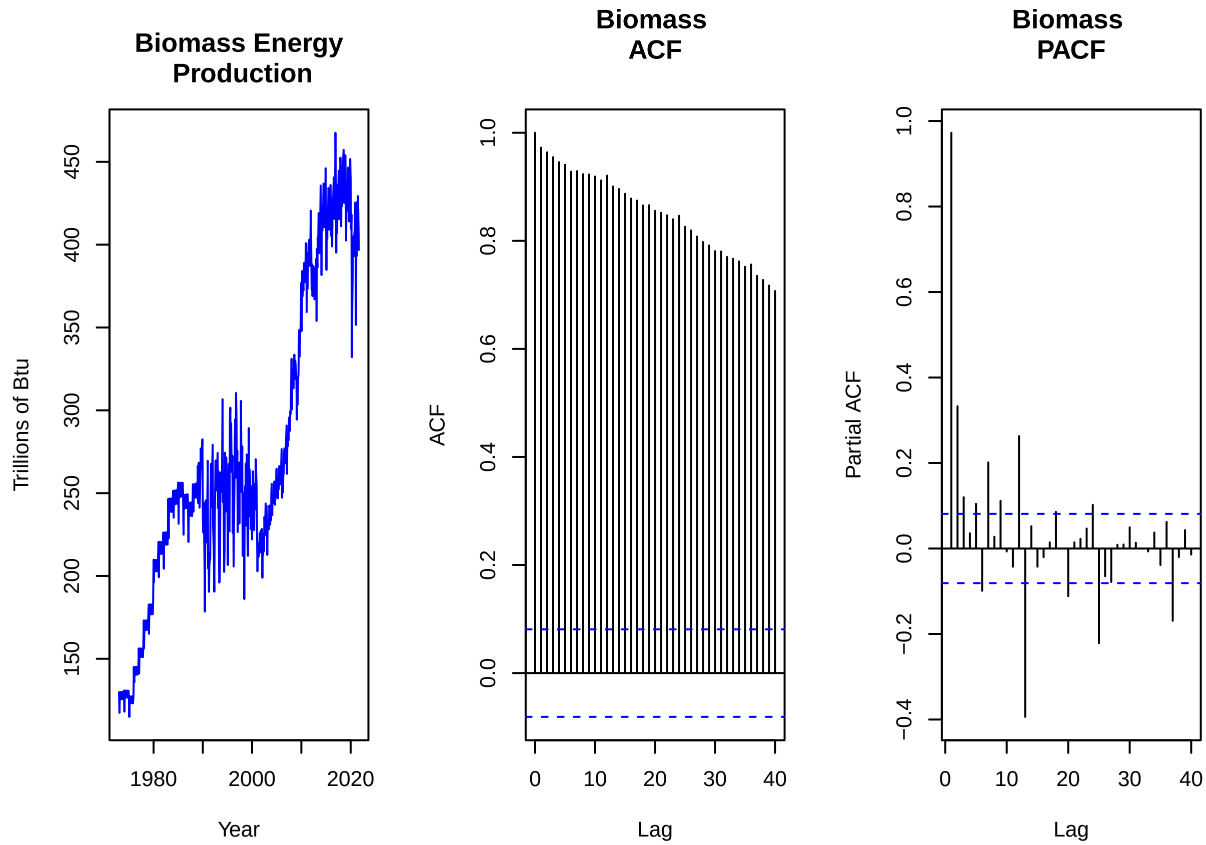
```
## # A tibble: 10 x 3
##    `Total Biomass Energy Production` `Total Renewable Ener~ `Hydroelectric Powe~
##                                <dbl>                  <dbl>                <dbl>
##  1                              130.                   404.                 273.
##  2                              117.                   361.                 242.
##  3                              130.                   400.                 269.
##  4                              126.                   380.                 253.
##  5                              130.                   392.                 261.
##  6                              126.                   377.                 250.
##  7                              130.                   367.                 236.
##  8                              130.                   354.                 222.
##  9                              126.                   307.                 180.
## 10                              130.                   323.                 192.
```

##Trend Component

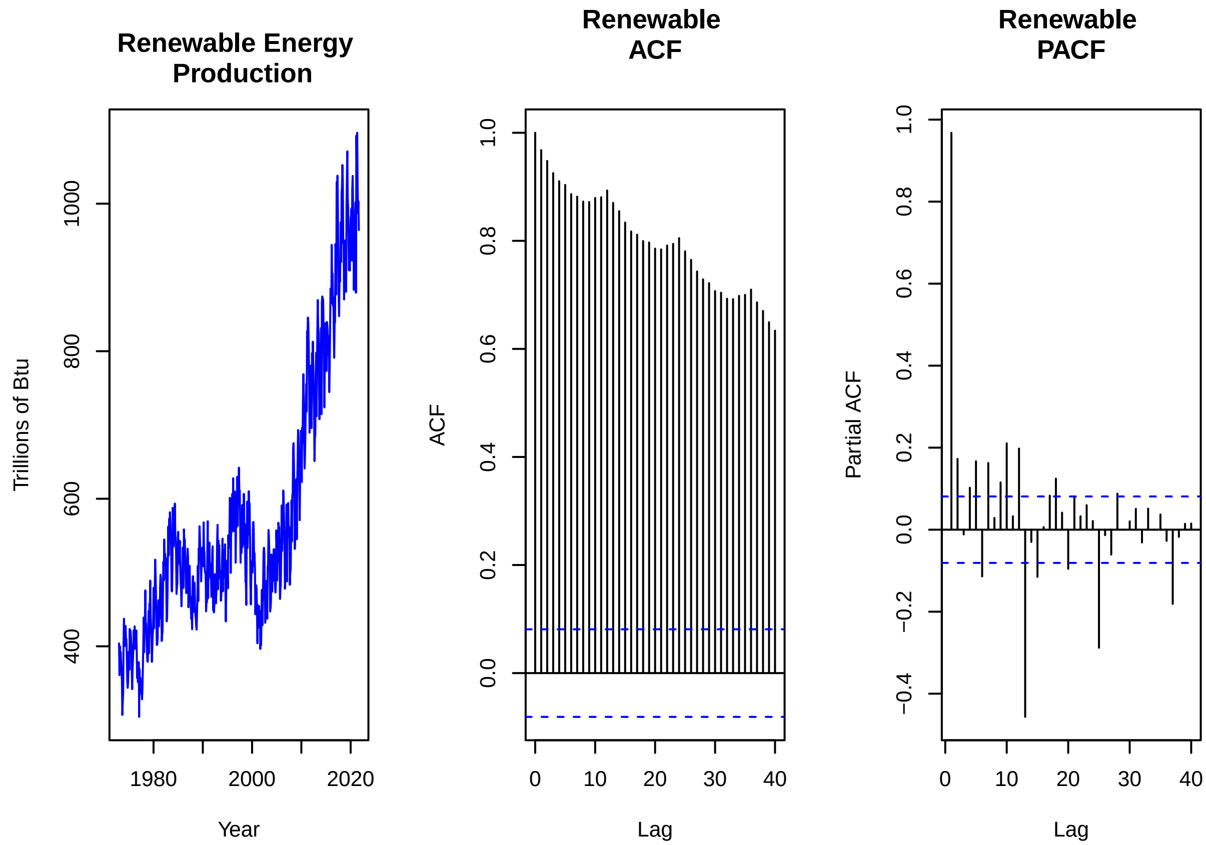
Q1

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)

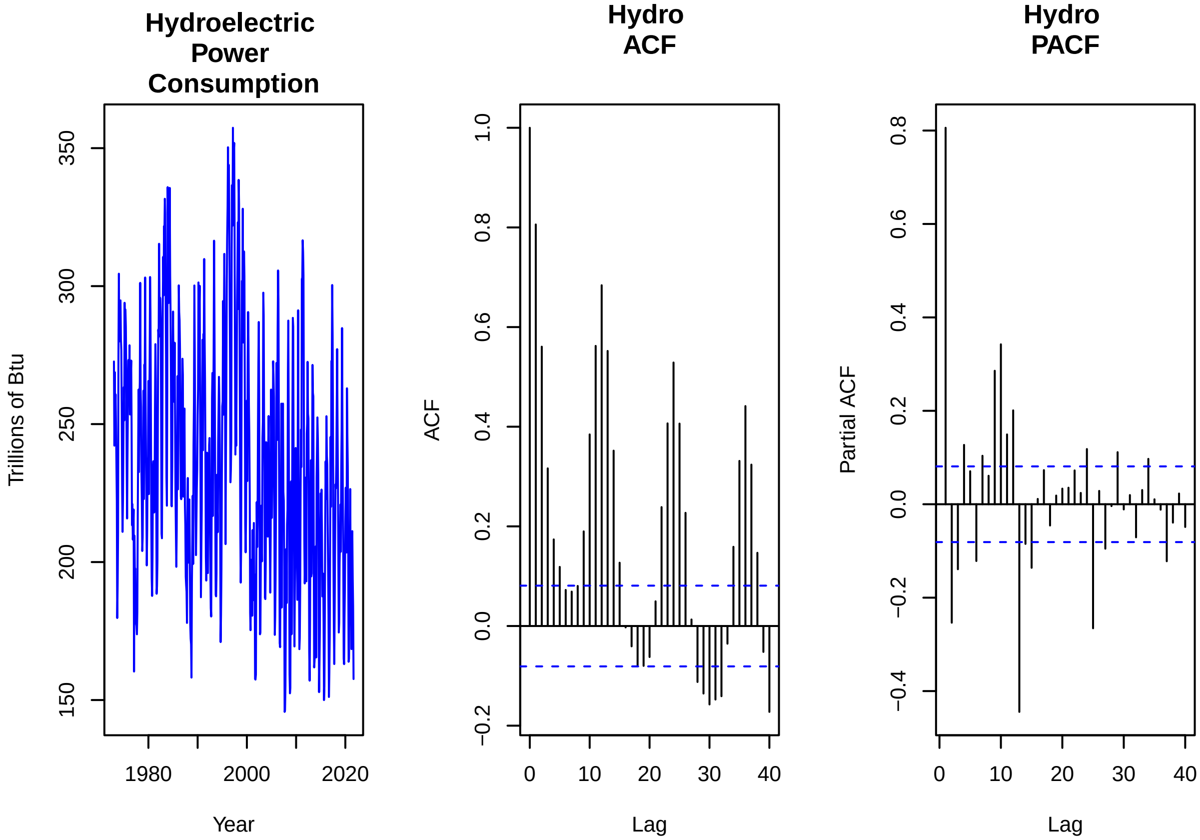
```r
ts <- ts(data = df2, start = 1973, frequency = 12)


#Biomass Plots
par(mfrow = c(1,3))
plot(ts[,1], col="blue", ylab="Trillions of Btu", main="Biomass Energy\n Production",
     xlab="Year")
tsBio <- ts(data = df2$`Total Biomass Energy Production`, start = 1973, frequency = 1)
acfBio <- acf(tsBio, lag.max = 40, type = "correlation",plot=TRUE, main="Biomass \nACF")
pacfBio <- pacf(tsBio, lag.max = 40, plot = T, main="Biomass \nPACF")
```




```
#Renewable Plots
par(mfrow = c(1,3))
plot(ts[,2], col="blue", ylab="Trillions of Btu", main="Renewable Energy \n Production",
     xlab="Year")
tsRE <- ts(data = df2$`Total Renewable Energy Production`, start = 1973, frequency = 1)
acfRE <- acf(tsRE, lag.max = 40, type = "correlation", plot=TRUE, main="Renewable \nACF")
pacfRE <- pacf(tsRE, lag.max = 40, plot = T, main="Renewable \nPACF")
```




```r
#Hydro Plot
par(mfrow = c(1,3))
plot(ts[,3], col="blue", ylab="Trillions of Btu", main="Hydroelectric \nPower \nConsumption",
     xlab="Year")
tsHydro <- ts(data = df2$`Hydroelectric Power Consumption`, start = 1973, frequency = 1)
acfHydro <- acf(tsHydro, lag.max = 40, type = "correlation", plot=TRUE, main="Hydro \nACF")
pacfHydro <- pacf(tsHydro, lag.max = 40, plot = T, main="Hydro \nPACF")
```



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?

There appears to be a positive stochastic trend in biomass energy production, a positive stochastic trend for renewable energy production, and a slight negative deterministic trend for hydroelectric power consumption.

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.

As discussed in class, I am going to ignore the intercept coefficient and focus solely on the slope coefficients.

The analysis shows that for every single month increase, biomass production increases by 0.4744 trillion Btu, renewable energy productioin increases by 0.8805 trillion Btu, and hydroelectric power consumption decreases by 0.0792 trillion Btu.

All three linear models have highly statistically significant trends as evidenced by the very small p-values associated with the slope coefficients shown in the row labeled "t" on each coefficients table.

```r
nobs = nrow(df2)
t<-c(1:nobs)
my_date <- df[,1]
my_date <- my_date[c(2:586),1]
df2 <- cbind(my_date,df2)
```

```r
lmBio = lm(df2$`Total Biomass Energy Production`~t)
b0Bio=as.numeric(lmBio$coefficients[1])
b1Bio=as.numeric(lmBio$coefficients[2])

lmRE = lm(df2$`Total Renewable Energy Production`~t)
b0RE=as.numeric(lmRE$coefficients[1])
b1RE=as.numeric(lmRE$coefficients[2])

lmHydro = lm(df2$`Hydroelectric Power Consumption`~t)
b0Hydro=as.numeric(lmHydro$coefficients[1])
b1Hydro=as.numeric(lmHydro$coefficients[2])

summary(lmBio)
```

```
##
## Call:
## lm(formula = df2$`Total Biomass Energy Production` ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -101.892  -24.306    4.932   33.103   82.292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.348e+02  3.282e+00   41.07   <2e-16 ***
## t           4.744e-01  9.705e-03   48.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.64 on 583 degrees of freedom
## Multiple R-squared:  0.8039, Adjusted R-squared:  0.8035
## F-statistic:  2389 on 1 and 583 DF,  p-value: < 2.2e-16
```

```r
summary(lmRE)
```

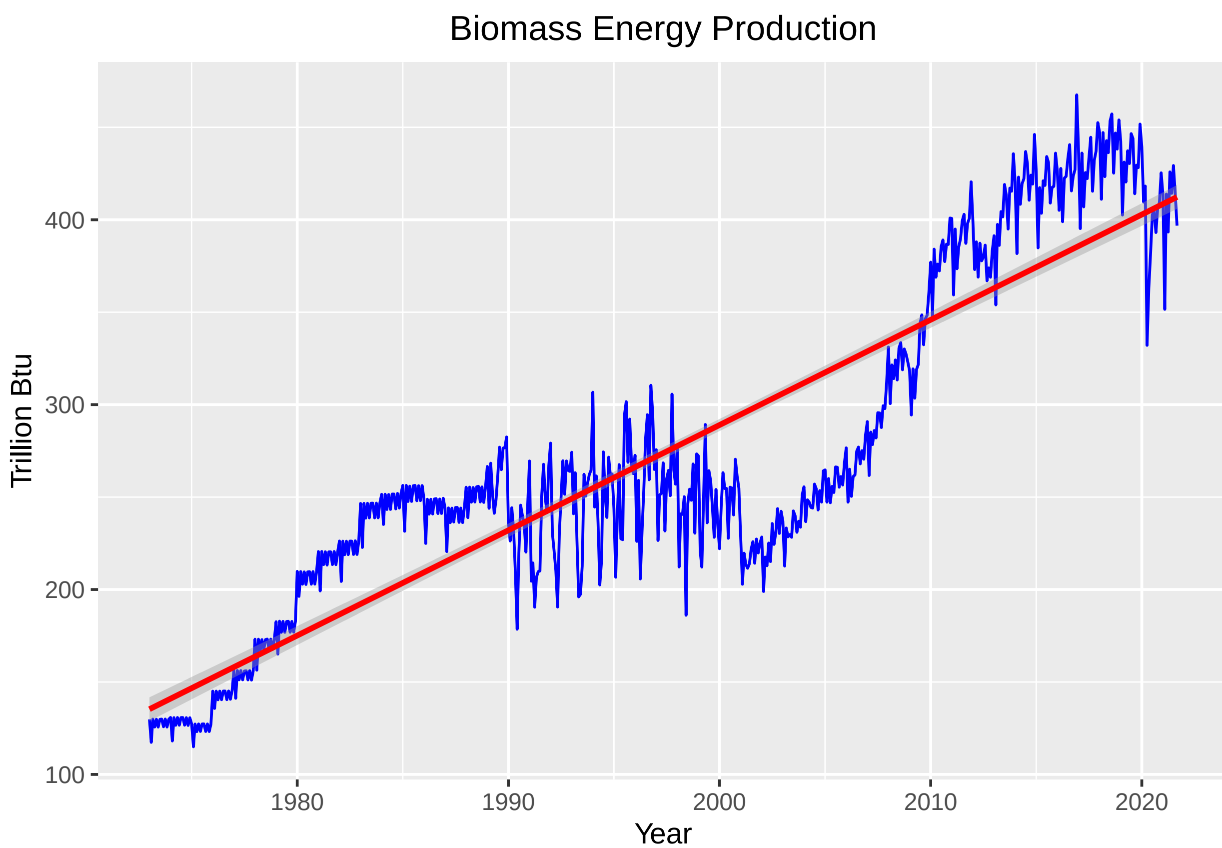
```
##
## Call:
## lm(formula = df2$`Total Renewable Energy Production` ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -230.488  -57.869    5.595   62.090  261.349
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 323.18243    8.02555   40.27   <2e-16 ***
## t             0.88051    0.02373   37.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96.93 on 583 degrees of freedom
## Multiple R-squared:  0.7025, Adjusted R-squared:  0.702
## F-statistic:  1377 on 1 and 583 DF,  p-value: < 2.2e-16
```

```r
summary(lmHydro)
```

```
##
## Call:
## lm(formula = df2$`Hydroelectric Power Consumption` ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -94.892 -31.300  -2.414  27.876 121.263
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 259.18303    3.47464  74.593  < 2e-16 ***
## t            -0.07924    0.01027  -7.712 5.36e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.97 on 583 degrees of freedom
## Multiple R-squared:  0.09258,    Adjusted R-squared:  0.09103
## F-statistic: 59.48 on 1 and 583 DF,  p-value: 5.364e-14
```

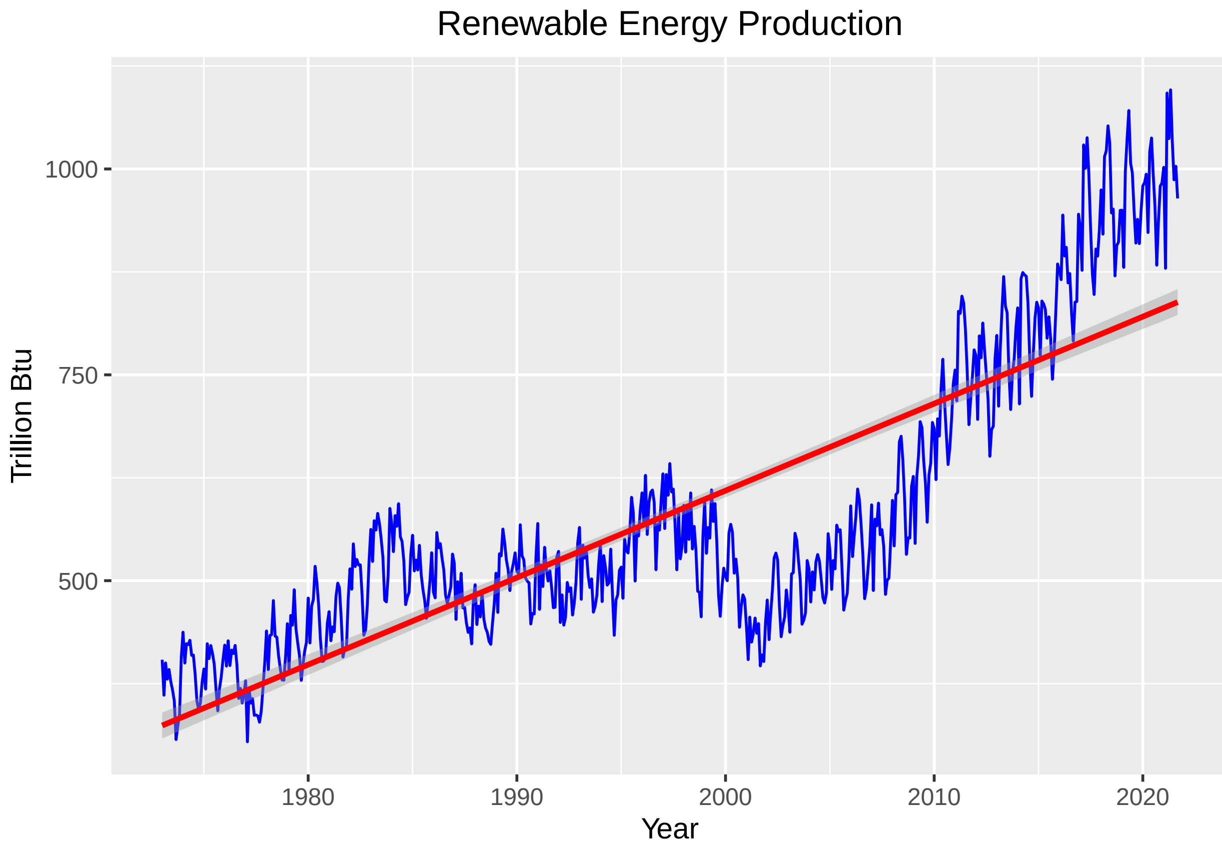
```r
ggplot(df2, aes(x=df2$Month, y=df2$`Total Biomass Energy Production`)) +
            ylab(paste0(colnames(df2))) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            ggtitle("Biomass Energy Production")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")
```

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



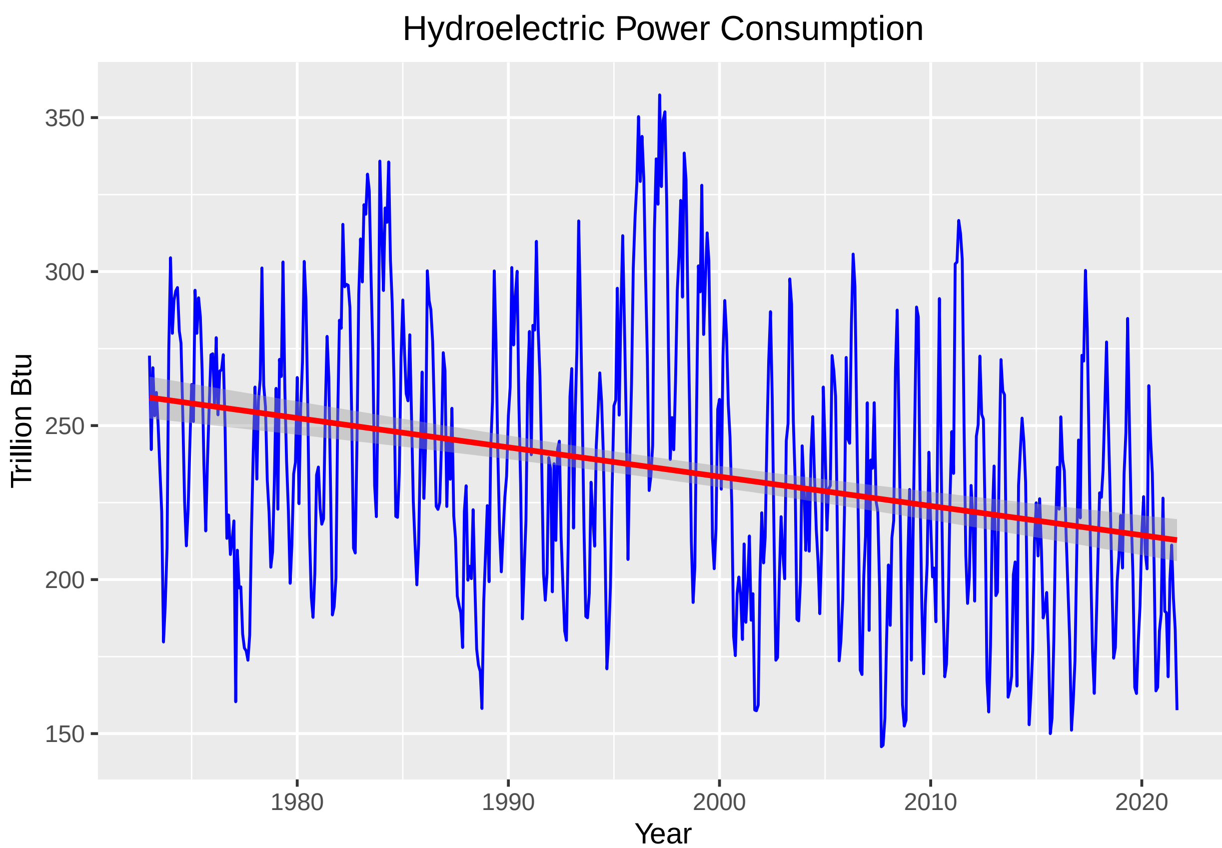

```r
ggplot(df2, aes(x=df2$Month, y=df2$`Total Renewable Energy Production`)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            ggtitle("Renewable Energy Production")+
            theme(plot.title = element_text(hjust = 0.5)) +  xlab("Year") + ylab("Trillion Btu")
```

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




```r
ggplot(df2, aes(x=df2$Month, y=df2$`Hydroelectric Power Consumption`)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            ggtitle("Hydroelectric Power Consumption") +
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")
```

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



Q4

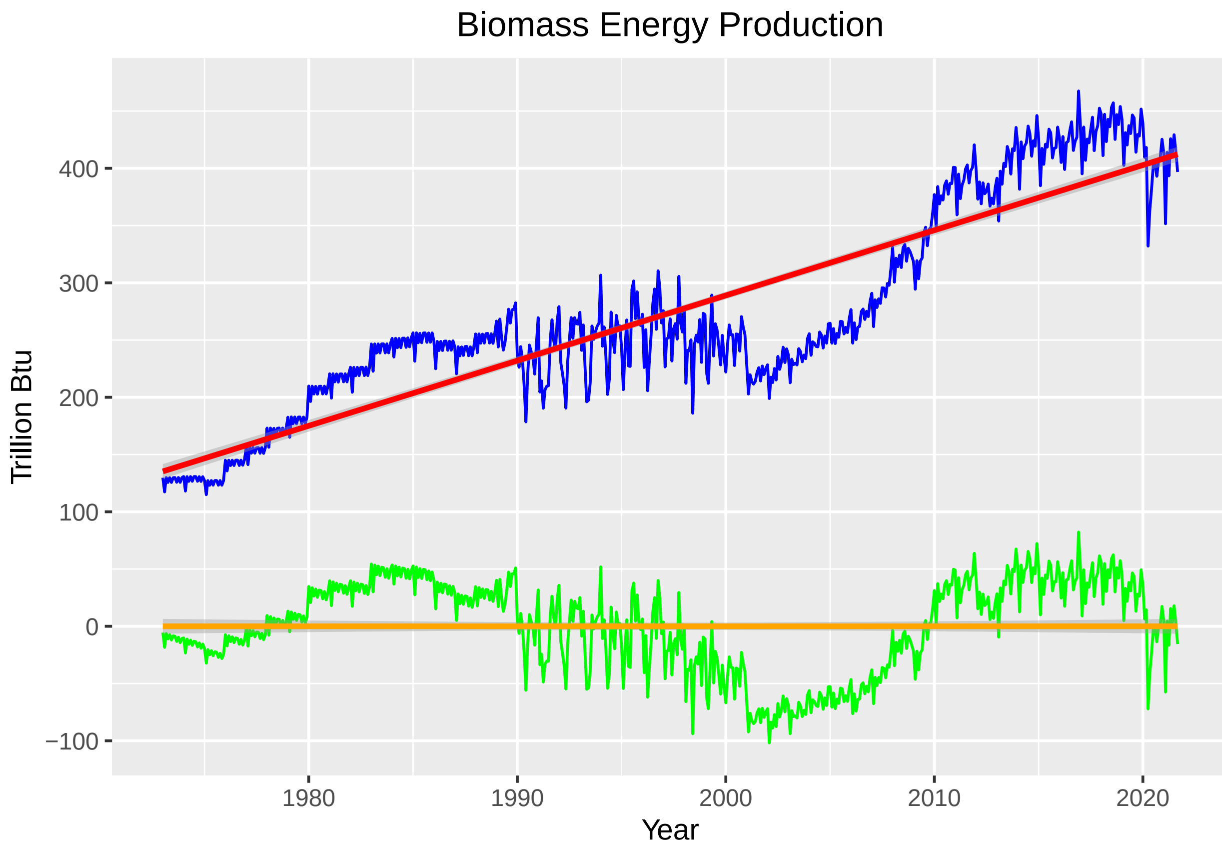
Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

I detrended the series and overlayed them with the original series, both with their trend lines. The data on each detrended series shifts down to be centered on zero. The trend lines on the detrended series are 0, showing the detrending was successful.

```
#Biomass
detrend_bio <- df2[,2]-(b0Bio+b1Bio*t)
detrend_bio <- as.data.frame(detrend_bio)

ggplot(df2, aes(x=df2$Month, y=df2$`Total Biomass Energy Production`)) +
            geom_line(color="blue") +
            ylab(paste0(colnames(df2))) +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=detrend_bio$detrend_bio), color="green") +
            geom_smooth(aes(y=detrend_bio$detrend_bio),color="orange",method="lm") +
            ggtitle("Biomass Energy Production")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")
```

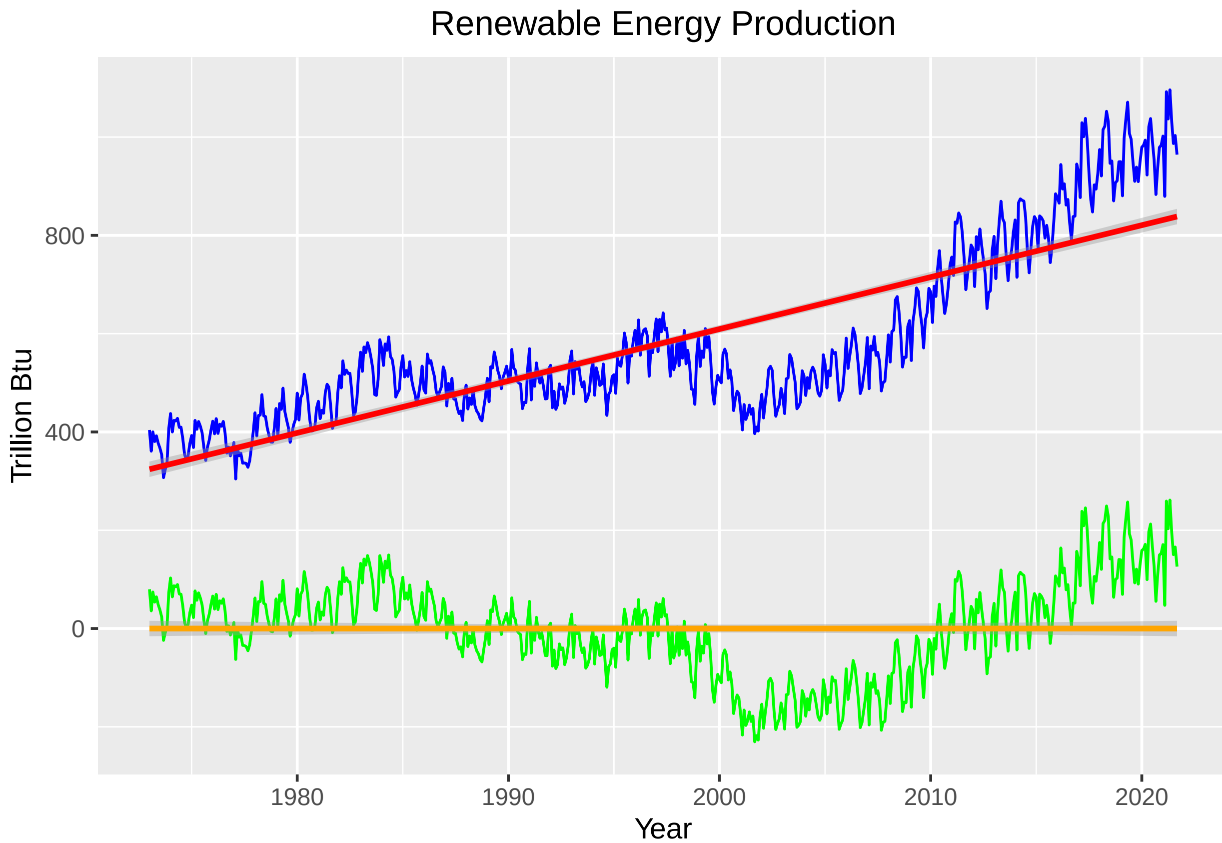
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



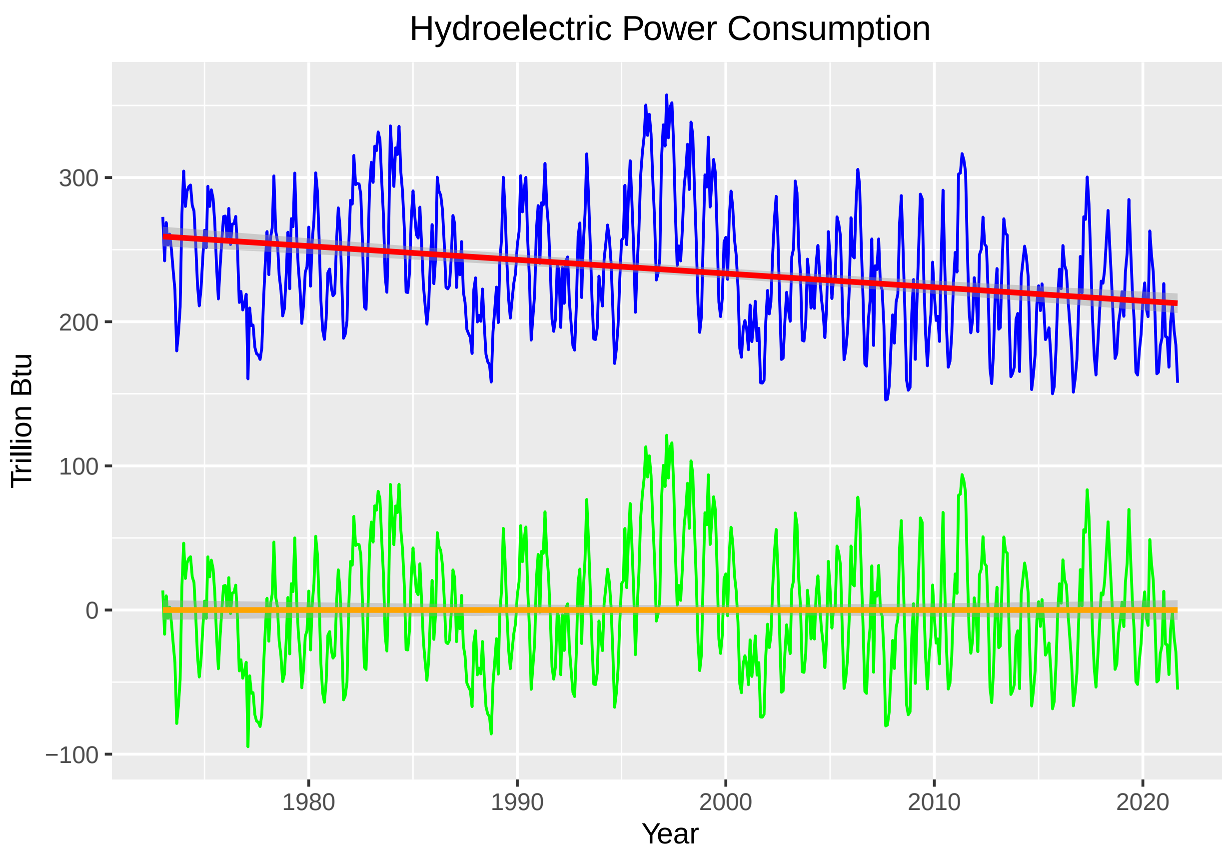

```
#Renewable
detrend_RE <- df2[,3]-(b0RE+b1RE*t)
detrend_RE <- as.data.frame(detrend_RE)

ggplot(df2, aes(x=df2$Month, y=df2$‘Total Renewable Energy Production‘)) +
            geom_line(color="blue") +
            ylab(paste0(colnames(df2))) +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=detrend_RE$detrend_RE), color="green") +
            geom_smooth(aes(y=detrend_RE$detrend_RE),color="orange",method="lm") +
            ggtitle("Renewable Energy Production")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
## ‘geom_smooth()‘ using formula ’y ~ x’
```




```
#Hydro
detrend_Hydro <- df2[,4]-(b0Hydro+b1Hydro*t)
detrend_Hydro <- as.data.frame(detrend_Hydro)

ggplot(df2, aes(x=df2$Month, y=df2$`Hydroelectric Power Consumption`, label="Hydro")) +
            geom_line(color="blue") +
            ylab(paste0(colnames(df2))) +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=detrend_Hydro$detrend_Hydro), color="green") +
            geom_smooth(aes(y=detrend_Hydro$detrend_Hydro),color="orange",method="lm") +
            ggtitle("Hydroelectric Power Consumption")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")
```

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

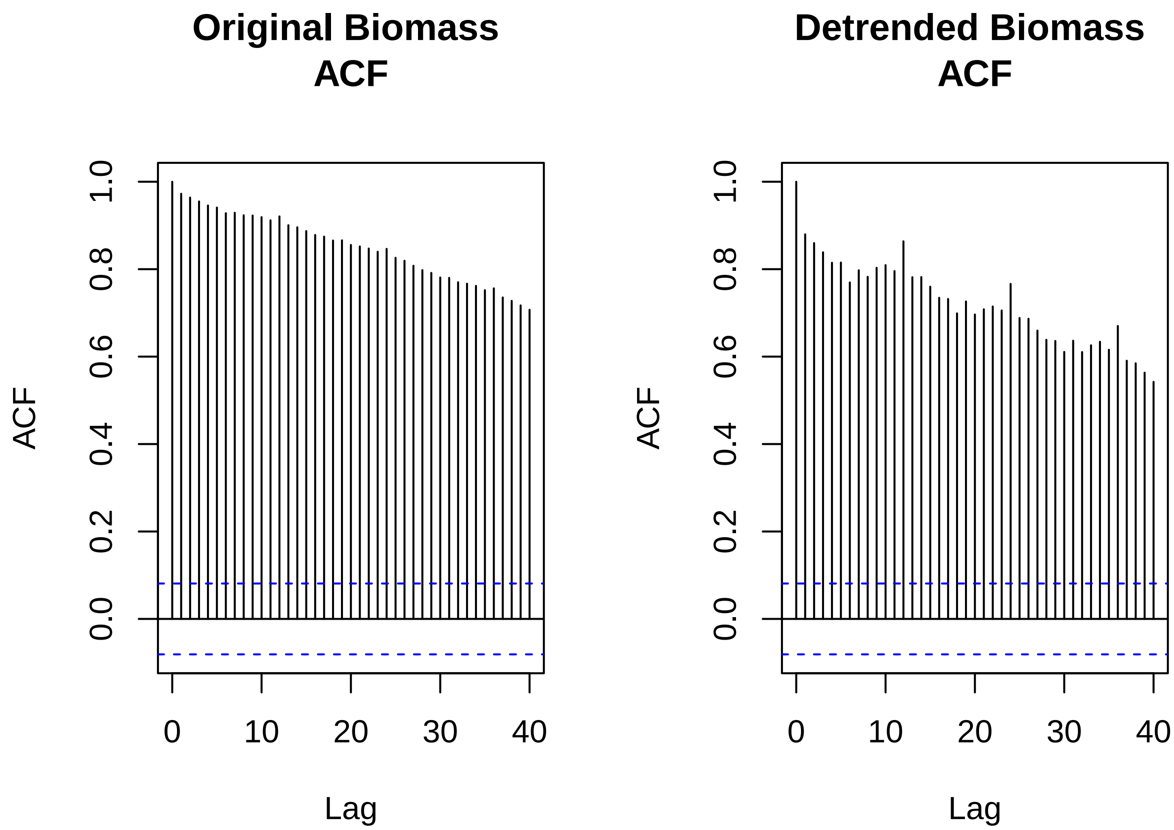


Q5

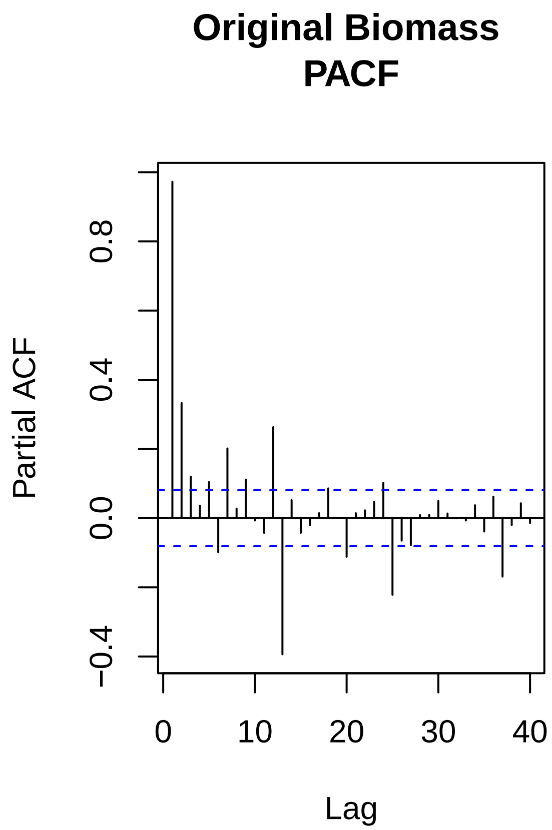
Plot ACF and PACF for the detrended series and compare with the plots from Q1. Did the plots change? How?

I plotted each ACF and PACF for each series next to their original ACF and PCF to show how the plots have chaanged. The ACF plots have only changed minorly. There is an excentuation of the seasonal effect on the ACF after deseasoning the graphs as the peaks and valleys are more differentiated in the deseasoned plots in comparison to the original plots. The PACF graphs look slightly changed. The Biomass and Renewable graphs look like the larger PACF bars actually grew after deseasoning while the Hydro PACF looks almost unchanged.

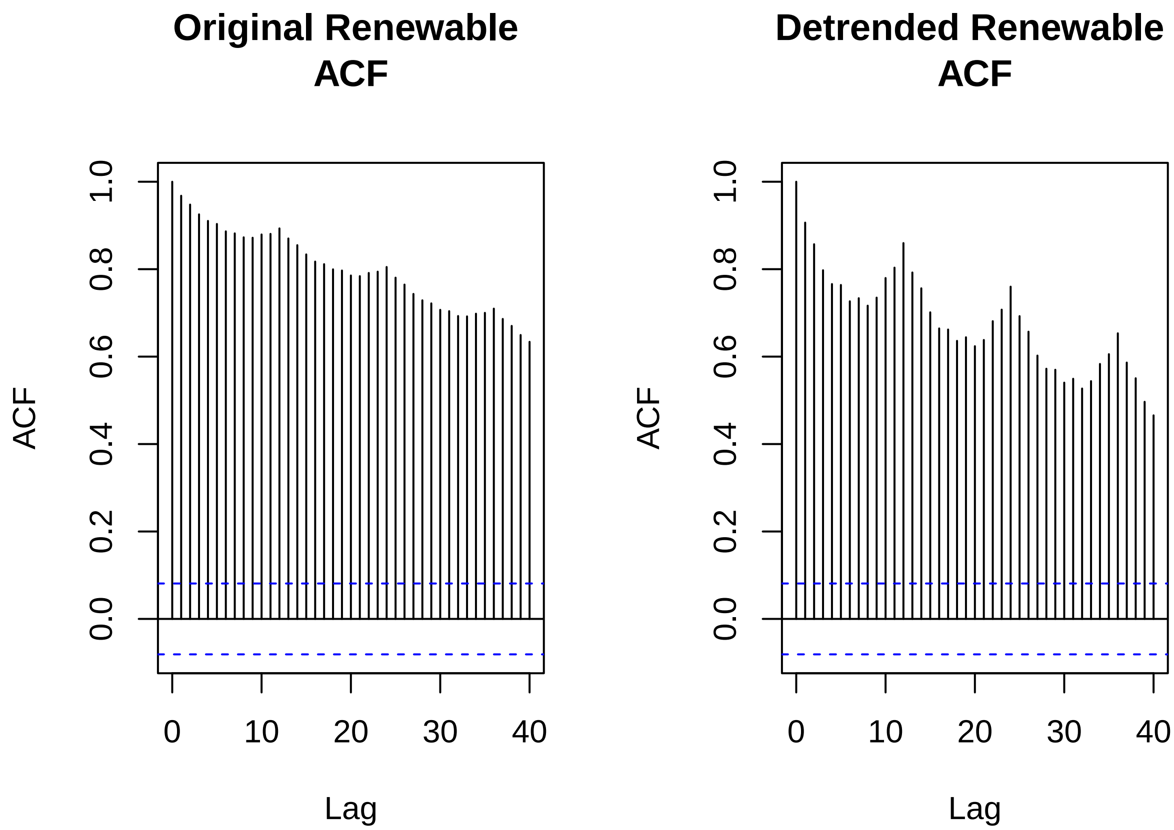
```
#Biomass
par(mfrow = c(1,2))
acf(tsBio, lag.max = 40, type = "correlation", plot=TRUE, main="Original Biomass \nACF")
acf(detrend_bio, lag.max = 40, type = "correlation", plot=TRUE, main="Detrended Biomass \nACF")
```



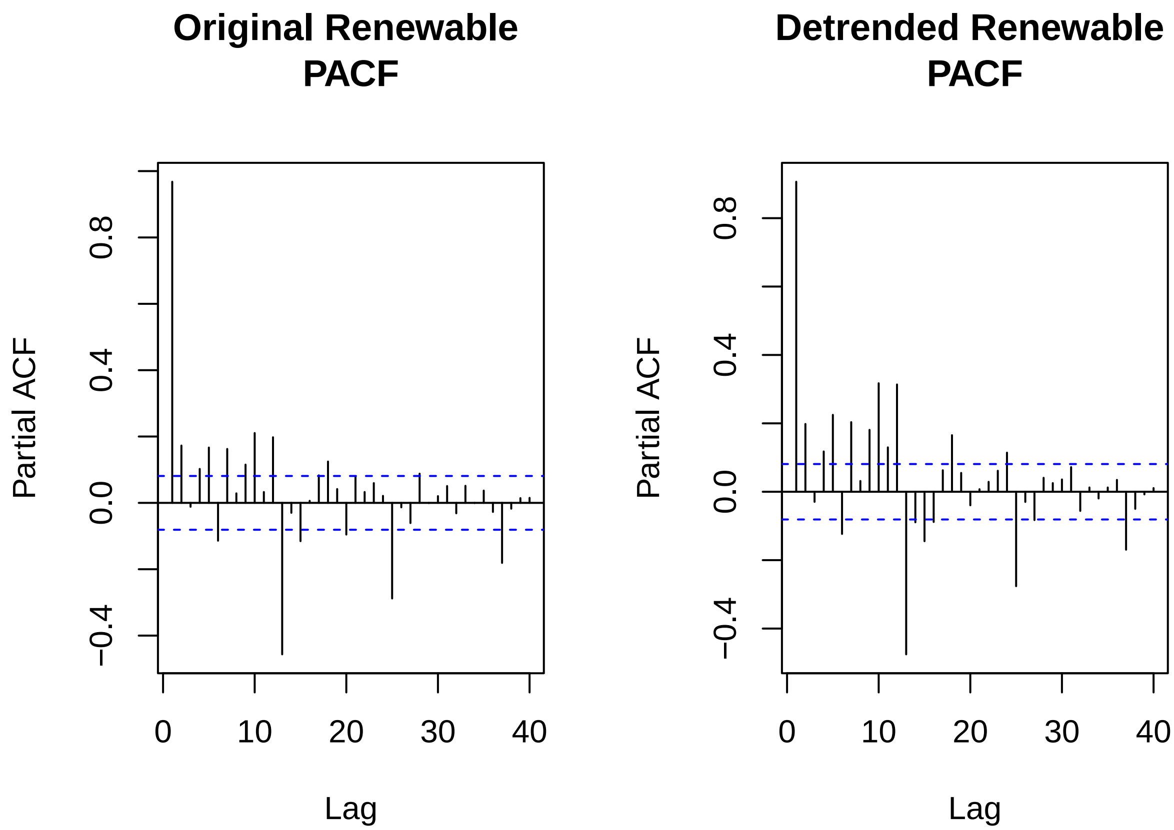

```
par(mfrow = c(1,2))
pacf(tsBio, lag.max = 40, plot = T, main="Original Biomass \nPACF")
pacf(detrend_bio, lag.max = 40, plot = T, main="Detrended Biomass \nPACF")
```



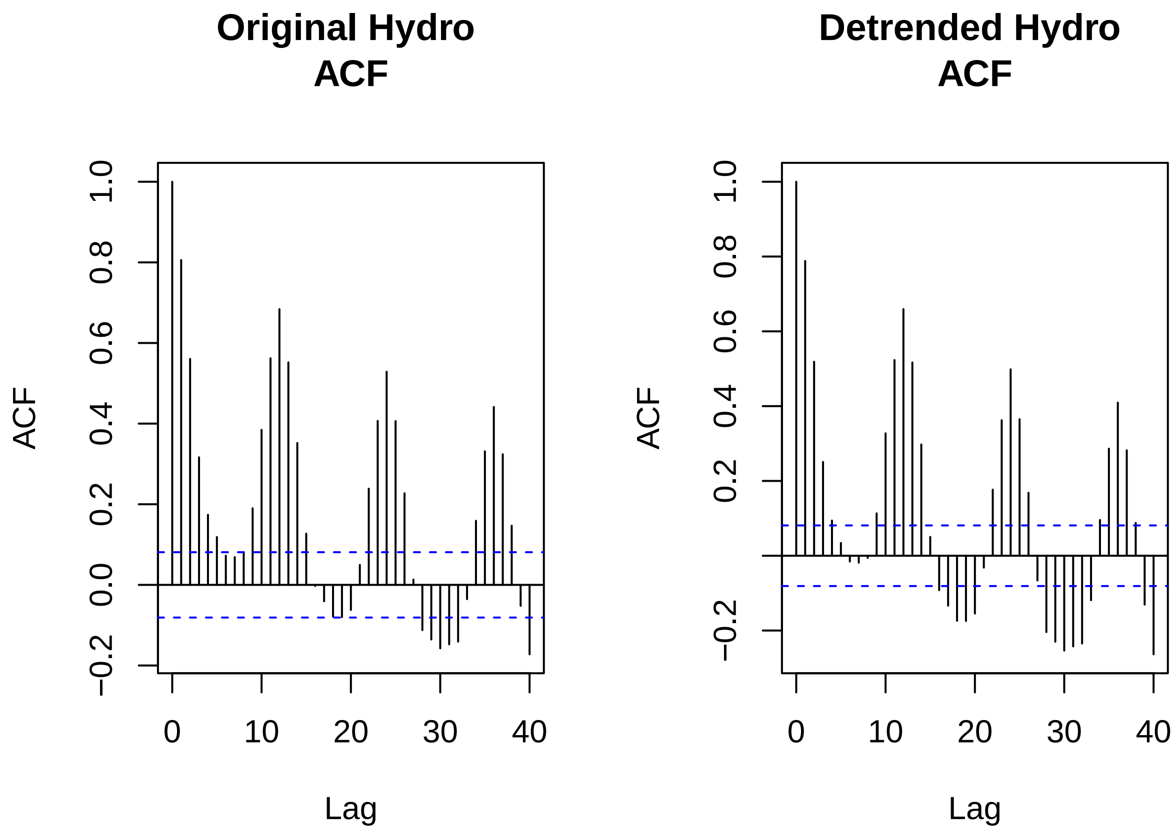

```
#Renewable
par(mfrow = c(1,2))
acf(tsRE, lag.max = 40, type = "correlation", plot=TRUE, main="Original Renewable \nACF")
acf(detrend_RE, lag.max = 40, type = "correlation", plot=TRUE, main="Detrended Renewable \nACF")
```



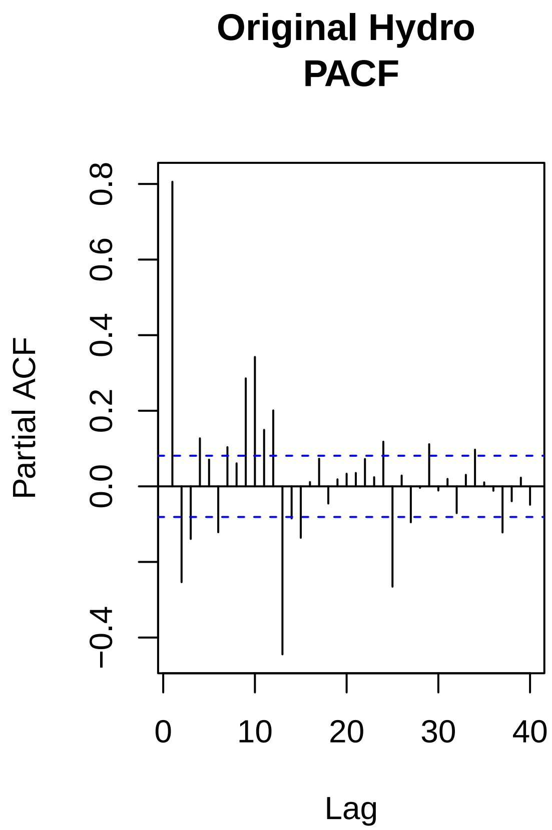

```
par(mfrow = c(1,2))
pacf(tsRE, lag.max = 40, plot = T, main="Original Renewable \nPACF")
pacf(detrend_RE, lag.max = 40, plot = T, main="Detrended Renewable \nPACF")
```




```
#Hydro
par(mfrow = c(1,2))
acf(tsHydro, lag.max = 40, type = "correlation", plot=TRUE, main="Original Hydro \nACF")
acf(detrend_Hydro, lag.max = 40, type = "correlation", plot=TRUE, main="Detrended Hydro \nACF")
```




```
par(mfrow = c(1,2))
pacf(tsHydro, lag.max = 40, plot = T, main="Original Hydro \nPACF")
pacf(detrend_Hydro, lag.max = 40, plot = T, main="Detrended Hydro \nPACF")
```



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 regression coefficients for further analysis.

I somewhat combined the code for questions 6 and 7 so the code is under number 7 but my answer for 6 is here.

When I fit a seasonal means model to these time series, it showed that the only time series with a statistically significant seasonal trend was hydro. I found this by investigating the summary of the seasonal means model. The righthand column of the coefficients table shows the probability of observing as extreme of a coefficient as our model produced if in fact the true value is zero. The fact that these probabilities were all above 0.05, our significance level, for biomass and renewables indicates there was no statistical significance to the seasonal variation in the data. For hydro, the p-values of the dummy coefficients were statistically significant for all dummy variables except for two, indicating a strong seasonal trend.

##$ These results make sense conseptually as there is nothing inherently seasonal about biomass energy production. For renewables, solar produces more during the summer, but wind produces more in the winter.

These could be cancelling each other our, effectively nullifying any seasonal trend. Hydro, however is highly affected by seasonal variations in windfall.

Q7

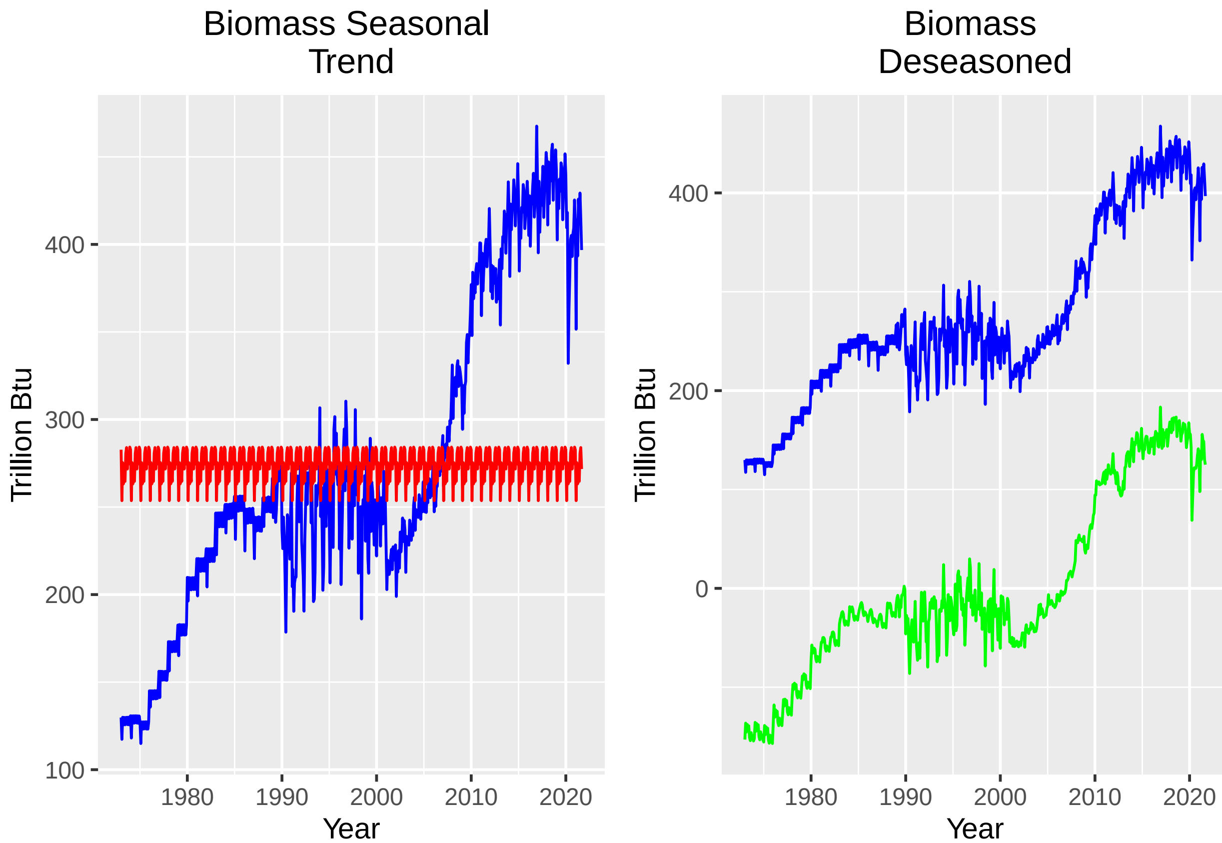
Use the regression coefficients from Q6 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

For each series, I displayed two plots side-by-side: one with the original data overlayed with the projected seasonal trend and one with the original data overlayed with the deseasoned data. This shifted the time series so that the mean was at zero. It also had the affect of smoothing out portions of the hydro production curve. The other two curves may have had some smoothing affect but it was not as obvious. This is unsurprising given the lack of the statistically significant seasonal trend as explained in the previous question.

```r
# install.packages("gridExtra")  #Installed to do par() equivalent on ggplot
library(gridExtra)

# Identifyinig the seasonal trend
dummiesBio <- seasonaldummy(ts[,1]) #creates dummies
seas_means_model_Bio=lm(ts[,1]~dummiesBio) #generate intercept and dummy coefficients for biomass data
summary(seas_means_model_Bio) # viewing dummy coefficients and intercept
```

```
##
## Call:
## lm(formula = ts[, 1] ~ dummiesBio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -156.96  -51.40  -22.15   60.65  183.31
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    284.241     12.962  21.928   <2e-16 ***
## dummiesBioJan   -1.498     18.238  -0.082   0.9346
## dummiesBioFeb  -30.582     18.238  -1.677   0.0941 .
## dummiesBioMar   -8.873     18.238  -0.486   0.6268
## dummiesBioApr  -21.009     18.238  -1.152   0.2498
## dummiesBioMay  -14.065     18.238  -0.771   0.4409
## dummiesBioJun  -19.601     18.238  -1.075   0.2829
## dummiesBioJul   -3.499     18.238  -0.192   0.8479
## dummiesBioAug   -0.252     18.238  -0.014   0.9890
## dummiesBioSep  -12.518     18.238  -0.686   0.4928
## dummiesBioOct   -3.629     18.331  -0.198   0.8432
## dummiesBioNov   -9.592     18.331  -0.523   0.6010
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 89.81 on 573 degrees of freedom
## Multiple R-squared:  0.01056,    Adjusted R-squared:  -0.008439
## F-statistic: 0.5557 on 11 and 573 DF,  p-value: 0.8647
```

```
beta_int_Bio=seas_means_model_Bio$coefficients[1] #saving intercept in beta_int_Bio
beta_coeff_Bio=seas_means_model_Bio$coefficients[2:12] #saving dummy coefficients in array called beta_
Bio_seas_comp=array(0,nobs) #creating array to hold seasonal trend
for(i in 1:nobs){
  Bio_seas_comp[i]=(beta_int_Bio+beta_coeff_Bio%*%dummiesBio[i,])
} #populating Bio_seas_comp using for loop with beta coefficients from the lm() function

#Removing seasonal component

bioSeasonPlot1 <- ggplot(df2, aes(x=df2$Month, y=df2$`Total Biomass Energy Production`)) + #plotting or
            geom_line(color="blue") +
            ylab(paste0(colnames(df2))) +
            geom_line(aes(y=Bio_seas_comp), color="red") +
            ggtitle("Biomass Seasonal \nTrend")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")

deseason_Bio <- ts[,1]-Bio_seas_comp # subtracting seasonal component in deseason_Bio

bioSeasonPlot2 <- ggplot(df2, aes(x=df2$Month, y=df2$`Total Biomass Energy Production`)) + #plotting or
            geom_line(color="blue") +
            ylab(paste0("Inflow ",colnames(ts)[1],sep="")) +
            geom_line(aes(y=deseason_Bio), col="green") +
            ggtitle("Biomass \nDeseasoned")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")

grid.arrange(bioSeasonPlot1, bioSeasonPlot2, ncol=2) #plotting side by side
```

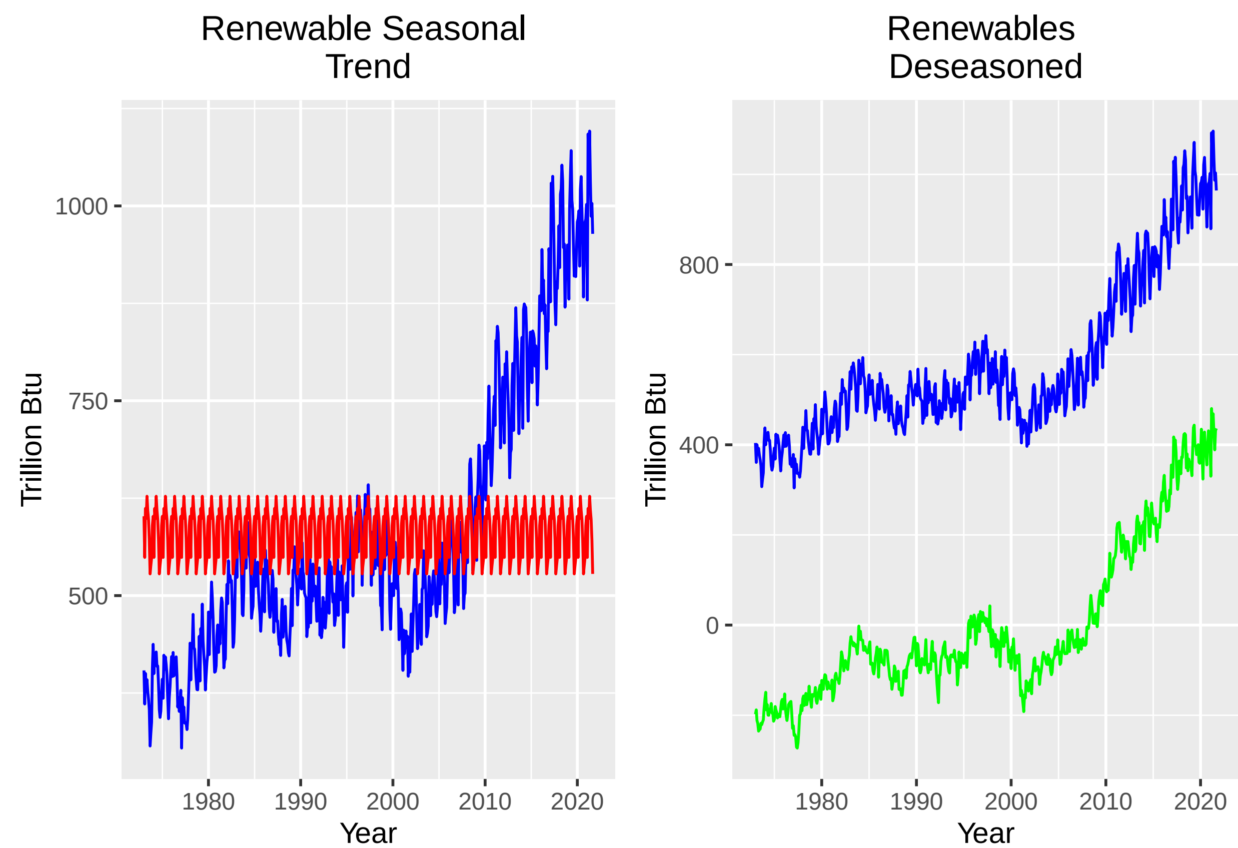



```r
# Identifyinig the seasonal trend
dummiesRE <- seasonaldummy(ts[,2]) #creates dummies
seas_means_model_RE=lm(ts[,2]~dummiesRE) #generate intercept and dummy coefficients for renewable data
summary(seas_means_model_RE) # viewing dummy coefficients and intercept
```

```
##
## Call:
## lm(formula = ts[, 2] ~ dummiesRE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -272.95 -111.55  -59.35   65.68  480.41
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   589.971     25.464  23.169   <2e-16 ***
## dummiesREJan   11.793     35.828   0.329   0.7422
## dummiesREFeb  -40.992     35.828  -1.144   0.2530
## dummiesREMar   21.892     35.828   0.611   0.5414
## dummiesREApr    8.908     35.828   0.249   0.8037
## dummiesREMay   37.500     35.828   1.047   0.2957
## dummiesREJun   19.465     35.828   0.543   0.5871
## dummiesREJul    8.115     35.828   0.227   0.8209
## dummiesREAug  -18.359     35.828  -0.512   0.6086
## dummiesRESep  -62.115     35.828  -1.734   0.0835 .
## dummiesREOct  -51.377     36.012  -1.427   0.1542
```

```
## dummiesRENov  -41.789     36.012  -1.160   0.2464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 176.4 on 573 degrees of freedom
## Multiple R-squared:  0.03139,    Adjusted R-squared:  0.0128
## F-statistic: 1.688 on 11 and 573 DF,  p-value: 0.07235
```

```
beta_int_RE=seas_means_model_RE$coefficients[1] #saving intercept in beta_int_RE
beta_coeff_RE=seas_means_model_RE$coefficients[2:12] #saving dummy coefficients in array called beta_co
RE_seas_comp=array(0,nobs) #creating array to hold seasonal trend
for(i in 1:nobs){
  RE_seas_comp[i]=(beta_int_RE+beta_coeff_RE%*%dummiesRE[i,])
} #populating RE_seas_comp using for loop with beta coefficients from the lm() function

#Removing seasonal component

reSeasonPlot1 <- ggplot(df2, aes(x=df2$Month, y=df2$`Total Renewable Energy Production`)) + #plotting o
            geom_line(color="blue") +
            ylab(paste0(colnames(df2))) +
            geom_line(aes(y=RE_seas_comp), color="red") +
            ggtitle("Renewable Seasonal \nTrend")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")

deseason_RE <- ts[,2]-RE_seas_comp # subtracting seasonal component in deseason_RE

reSeasonPlot2 <- ggplot(df2, aes(x=df2$Month, y=df2$`Total Renewable Energy Production`)) + #plotting o
            geom_line(color="blue") +
            ylab(paste0("Inflow ",colnames(ts)[2],sep="")) +
            geom_line(aes(y=deseason_RE), col="green") +
            ggtitle("Renewables \nDeseasoned")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")

grid.arrange(reSeasonPlot1, reSeasonPlot2, ncol=2) #plotting side by side
```

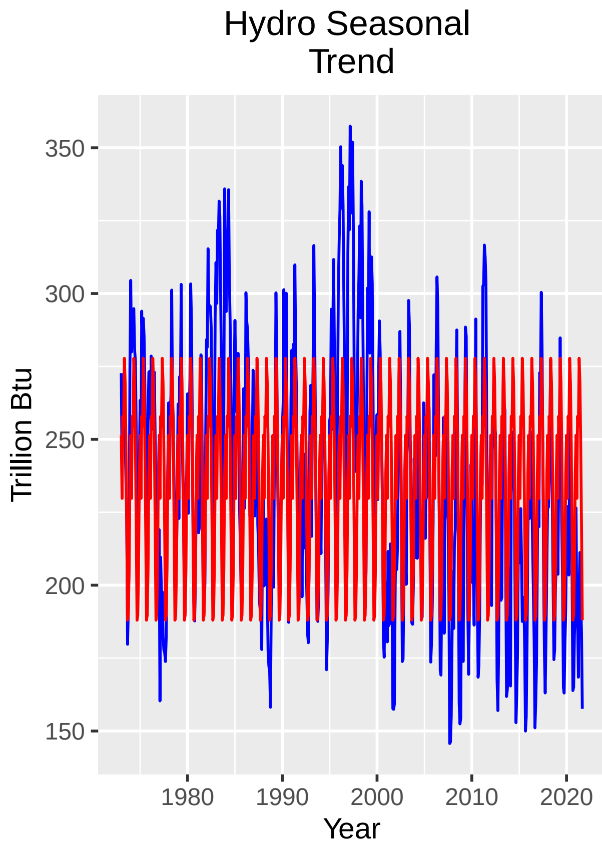


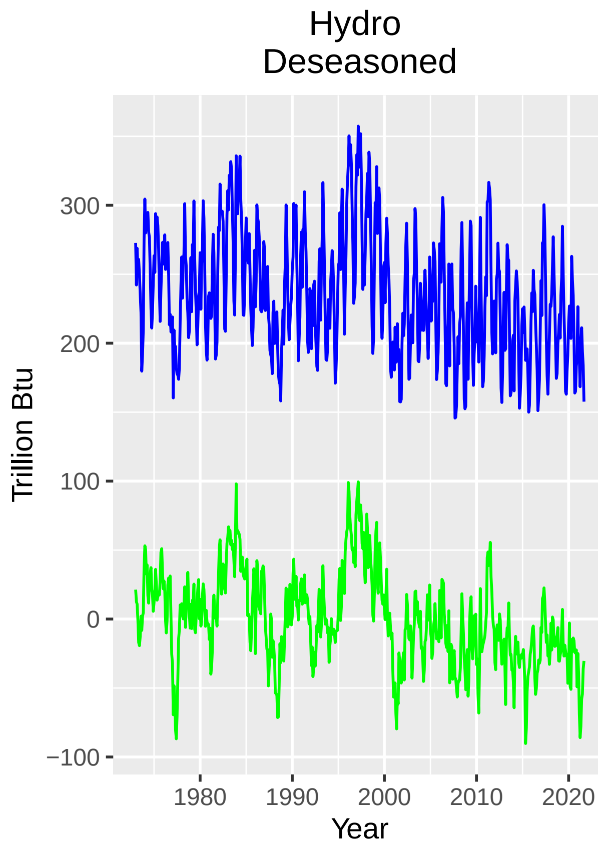

```
# Identifyinig the seasonal trend
dummiesHydro <- seasonaldummy(ts[,3]) #creates dummies
seas_means_model_Hydro=lm(ts[,3]~dummiesHydro) #generate intercept and dummy coefficients for renewable
summary(seas_means_model_Hydro) # viewing dummy coefficients and intercept
```

```
##
## Call:
## lm(formula = ts[, 3] ~ dummiesHydro)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.253 -23.017  -3.042  21.487  99.478
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       237.841      4.892  48.616  < 2e-16 ***
## dummiesHydroJan    13.558      6.883   1.970  0.04936 *
## dummiesHydroFeb    -8.090      6.883  -1.175  0.24037
## dummiesHydroMar    20.067      6.883   2.915  0.00369 **
## dummiesHydroApr    16.619      6.883   2.414  0.01607 *
## dummiesHydroMay    39.961      6.883   5.805 1.06e-08 ***
## dummiesHydroJun    31.315      6.883   4.549 6.57e-06 ***
## dummiesHydroJul    10.511      6.883   1.527  0.12732
## dummiesHydroAug   -17.853      6.883  -2.594  0.00974 **
## dummiesHydroSep   -49.852      6.883  -7.242 1.43e-12 ***
## dummiesHydroOct   -48.086      6.919  -6.950 9.96e-12 ***
```

```
## dummiesHydroNov  -32.187      6.919  -4.652 4.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.89 on 573 degrees of freedom
## Multiple R-squared:  0.4182, Adjusted R-squared:  0.4071
## F-statistic: 37.45 on 11 and 573 DF,  p-value: < 2.2e-16
```

```r
beta_int_Hydro=seas_means_model_Hydro$coefficients[1] #saving intercept in beta_int_Hydro
beta_coeff_Hydro=seas_means_model_Hydro$coefficients[2:12] #saving dummy coefficients in array called b
Hydro_seas_comp=array(0,nobs) #creating array to hold seasonal trend
for(i in 1:nobs){
  Hydro_seas_comp[i]=(beta_int_Hydro+beta_coeff_Hydro%*%dummiesHydro[i,])
} #populating Hydro_seas_comp using for loop with beta coefficients from the lm() function

#Removing seasonal component

HydroSeasonPlot1 <- ggplot(df2, aes(x=df2$Month, y=df2$`Hydroelectric Power Consumption`)) + #plotting
            geom_line(color="blue") +
            ylab(paste0(colnames(df2))) +
            geom_line(aes(y=Hydro_seas_comp), color="red") +
            ggtitle("Hydro Seasonal \nTrend")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")

deseason_Hydro <- ts[,3]-Hydro_seas_comp # subtracting seasonal component in deseason_Hydro

HydroSeasonPlot2 <- ggplot(df2, aes(x=df2$Month, y=df2$`Hydroelectric Power Consumption`)) + #plotting
            geom_line(color="blue") +
            ylab(paste0("Inflow ",colnames(ts)[3],sep="")) +
            geom_line(aes(y=deseason_Hydro), col="green") +
            ggtitle("Hydro \nDeseasoned")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")

grid.arrange(HydroSeasonPlot1, HydroSeasonPlot2, ncol=2) #plotting side by side
```





Q8

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

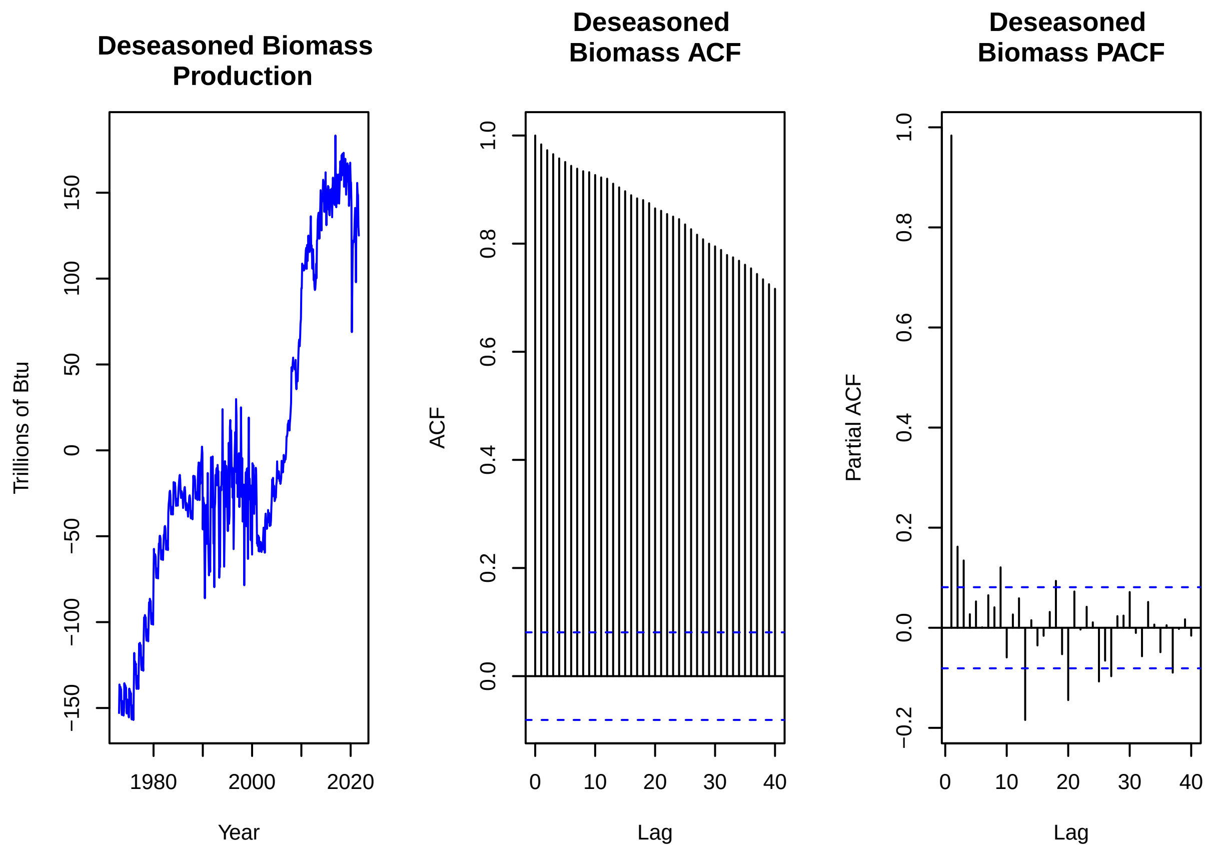
I plotted the time series, ACF, and PACF in the same plot window like I did in Q1 and also displayed the same plots as in Q1 for the sake of comparison.

Deseasoning seems to have taken away seasonal variation in the ACFs and significantly shrunk the PACFs for both Biomass and Renewable Energy time series. This indicates that it was successful in removing a seasonal trend which is surprising given the lack of statistical significance in the summary of the regression analysis.

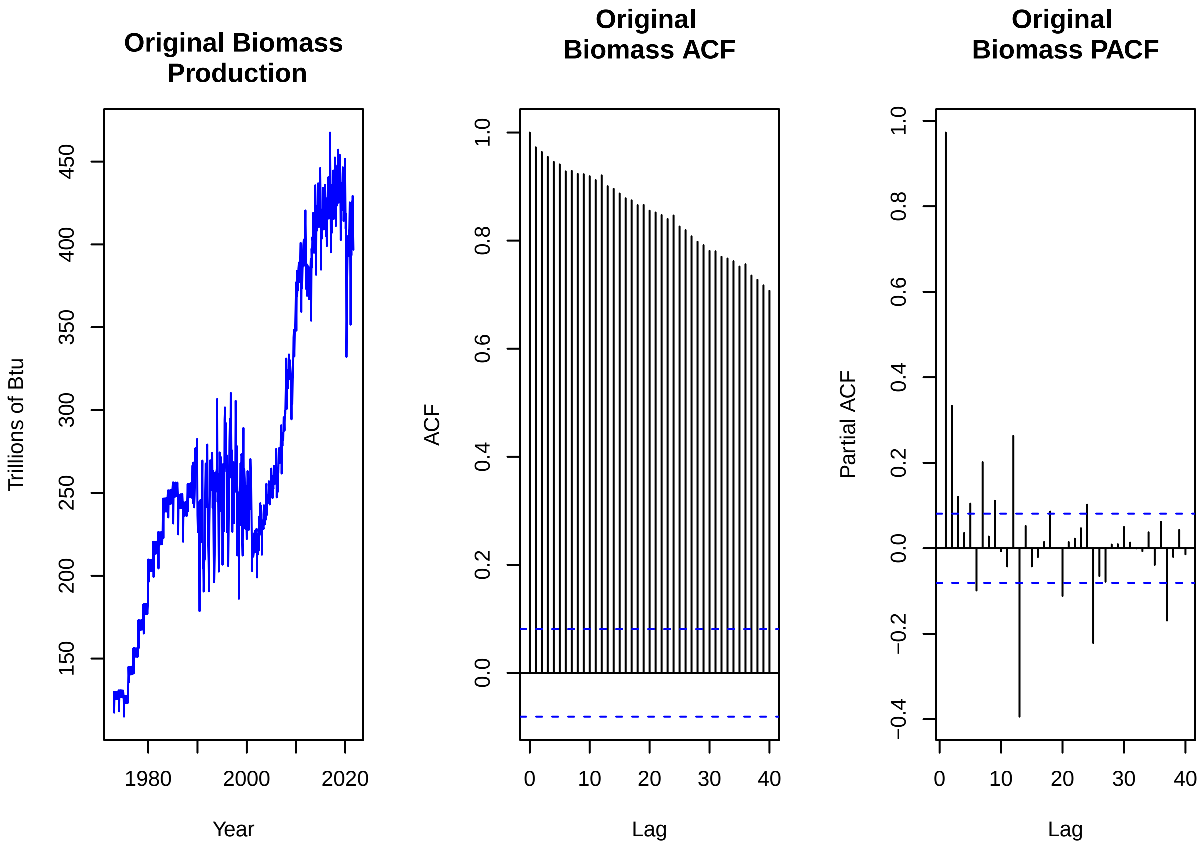
Unsurprisingly, the largest change is in the hydro time series. Where the ACF was originally shaped like a sine wave, it is now mostly steadily declining. Where the PACF originally showed seasonality, all the data past lag 2 is mostly negligable.

```
##plot deseasoned data, acf, pcf

par(mfrow = c(1,3))
plot(deseason_Bio, col="blue", ylab="Trillions of Btu", main="Deseasoned Biomass \n Production",
     xlab="Year")
tsDeseasBio <- ts(data = deseason_Bio, start = 1973, frequency = 1)
```

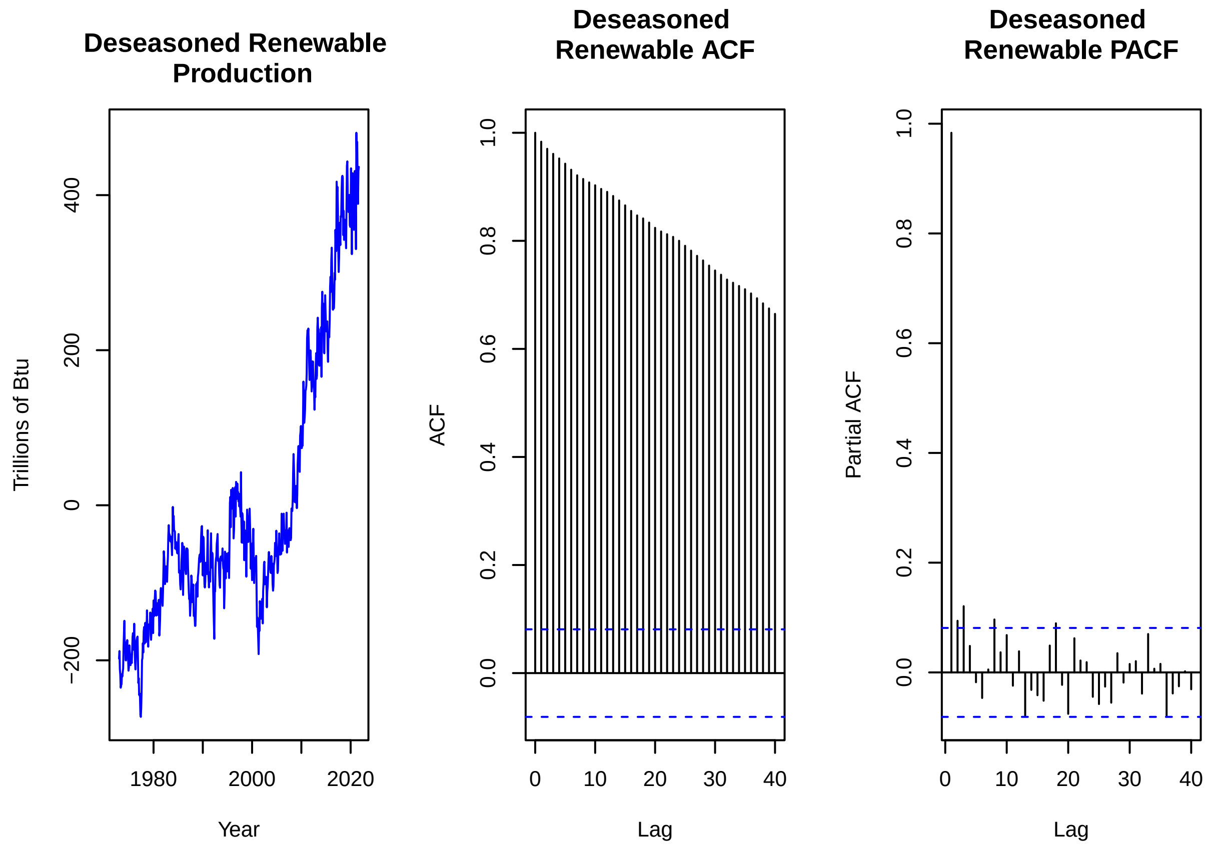
```
acfDeseasBio <- acf(tsDeseasBio, lag.max = 40, type = "correlation",plot=TRUE, main="Deseasoned \nBiomas
pacf(tsDeseasBio, lag.max = 40, plot = T, main="Deseasoned \nBiomass PACF")
```



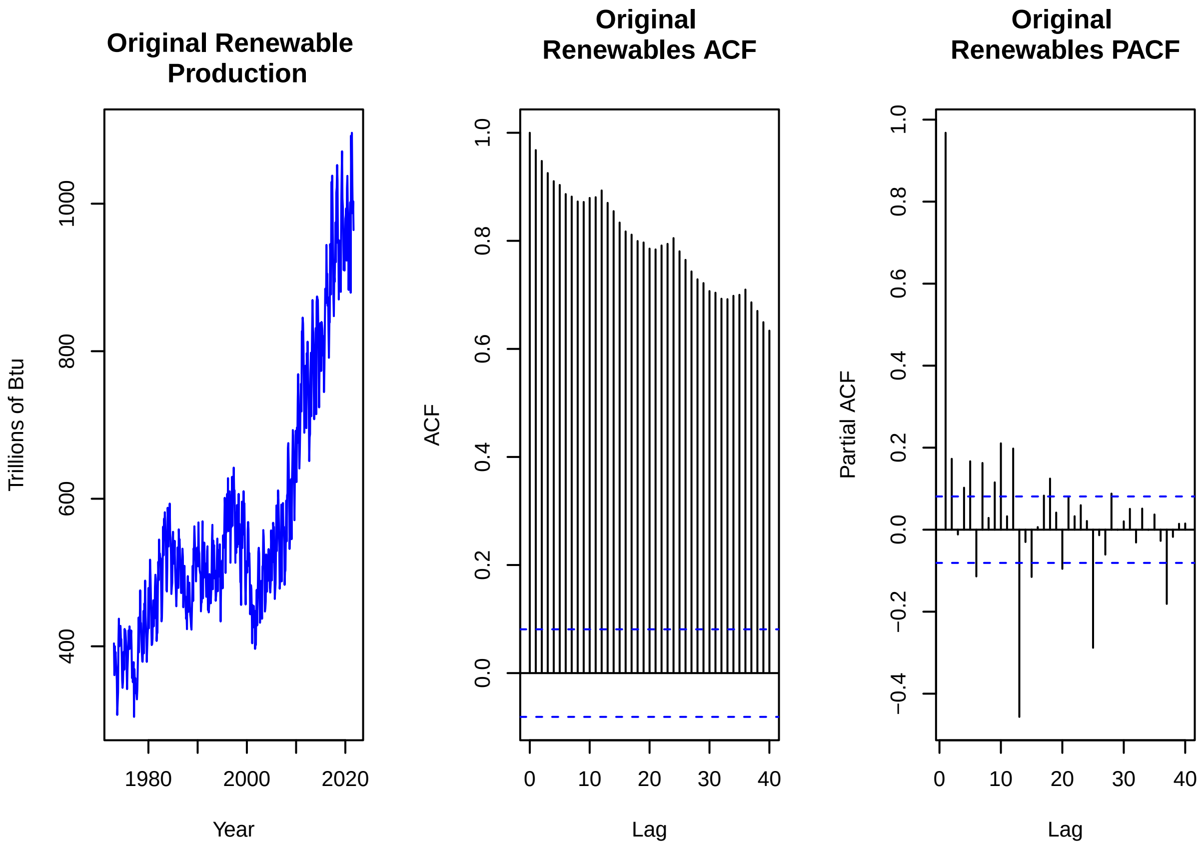

```
##include original plots for comparison

par(mfrow = c(1,3))
plot(ts[,1], col="blue", ylab="Trillions of Btu", main="Original Biomass\n Production",
     xlab="Year")
plot(acfBio, main = "Original \nBiomass ACF")
plot(pacfBio, main = "Original \nBiomass PACF")
```



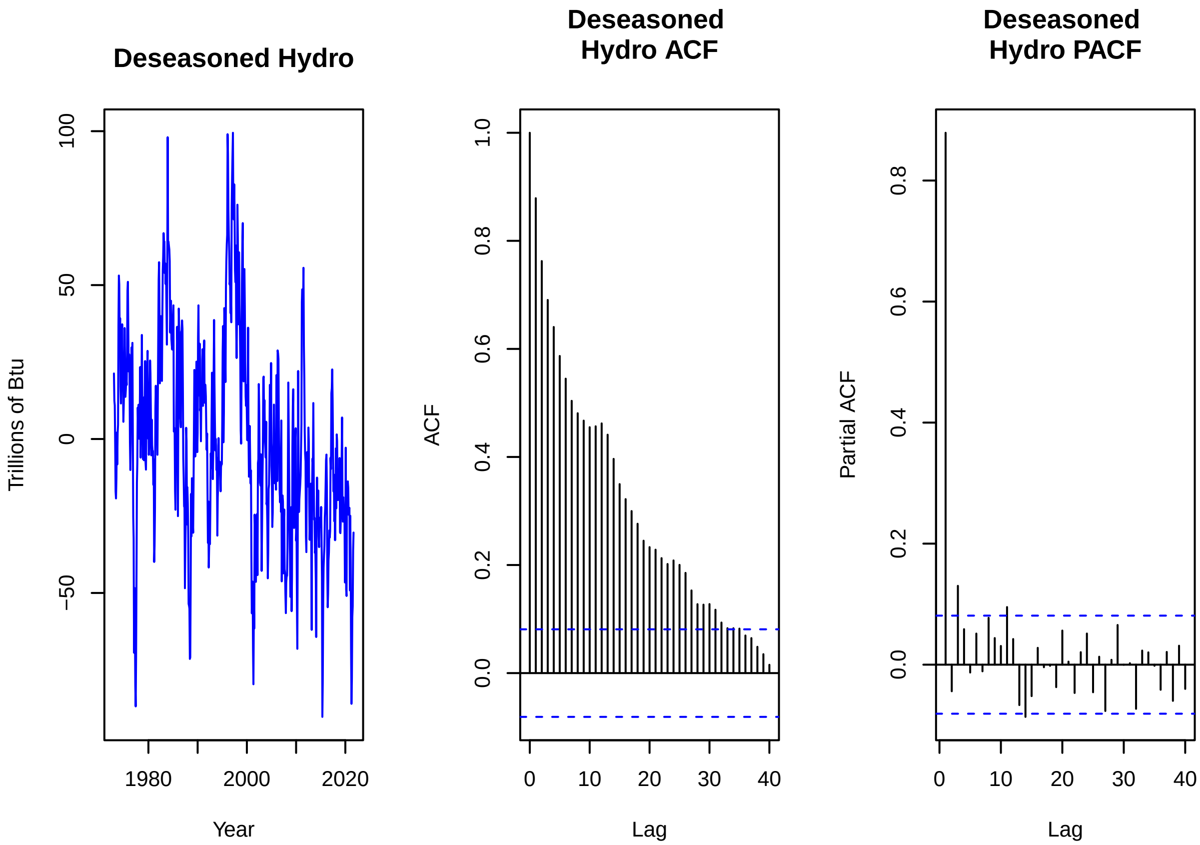

```r
##plot deseasoned data, acf, pacf

par(mfrow = c(1,3))
plot(deseason_RE, col="blue", ylab="Trillions of Btu", main="Deseasoned Renewable \n Production",
     xlab="Year")
tsDeseasRE <- ts(data = deseason_RE, start = 1973, frequency = 1)
acfDeseasRE <- acf(tsDeseasRE, lag.max = 40, type = "correlation",plot=TRUE, main="Deseasoned \nRenewabl
pacf(tsDeseasRE, lag.max = 40, plot = T, main="Deseasoned \nRenewable PACF")
```



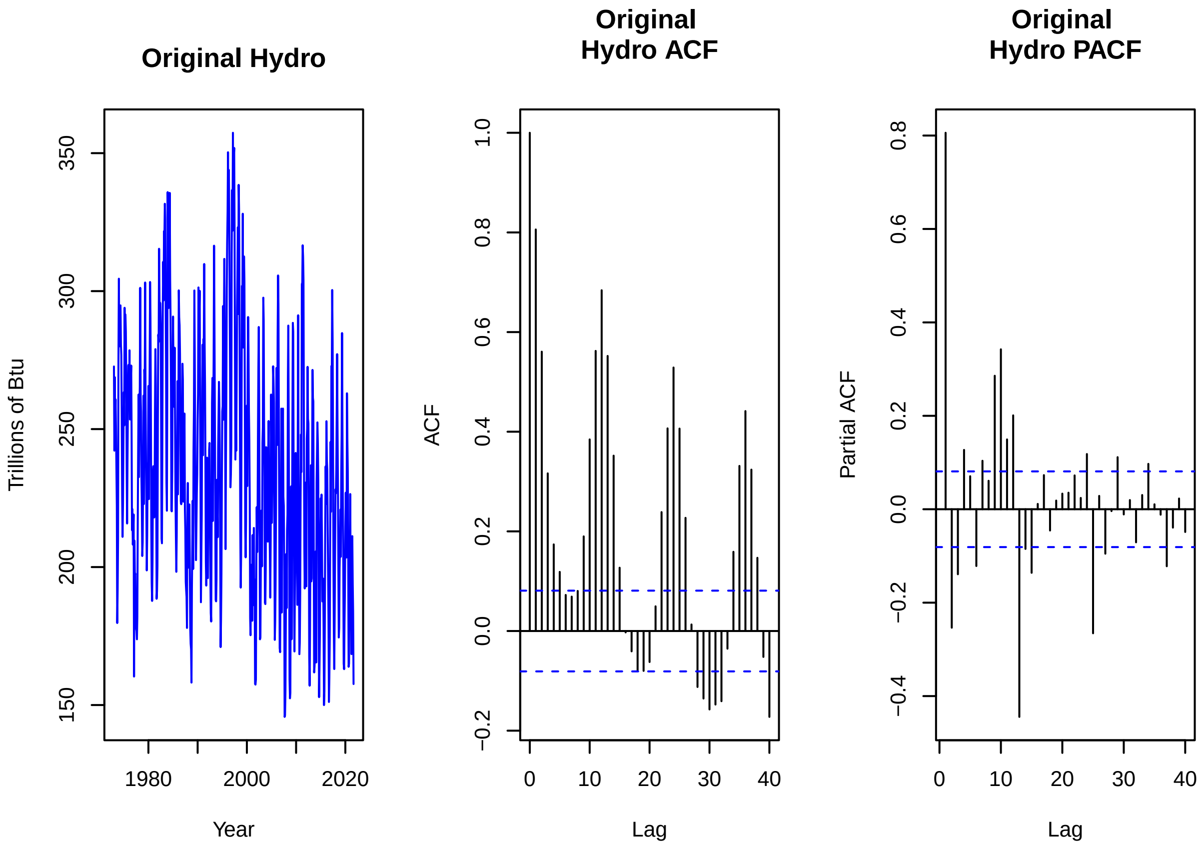

```r
##include original plots for comparison

par(mfrow = c(1,3))
plot(ts[,2], col="blue", ylab="Trillions of Btu", main="Original Renewable \n Production",
     xlab="Year")
plot(acfRE, main = "Original \nRenewables ACF")
plot(pacfRE, main = "Original \nRenewables PACF")
```




```r
##plot deseasoned data, acf, pacf

par(mfrow = c(1,3))
plot(deseason_Hydro, col="blue", ylab="Trillions of Btu", main="Deseasoned Hydro",
     xlab="Year")
tsDeseasHydro <- ts(data = deseason_Hydro, start = 1973, frequency = 1)
acfDeseasHydro <- acf(tsDeseasHydro, lag.max = 40, type = "correlation",plot=TRUE, main="Deseasoned \nHy
pacf(tsDeseasHydro, lag.max = 40, plot = T, main="Deseasoned \nHydro PACF")
```




```r
##include original plots for comparison

par(mfrow = c(1,3))
plot(ts[,3], col="blue", ylab="Trillions of Btu", main="Original Hydro",
     xlab="Year")
plot(acfHydro, main = "Original \nHydro ACF")
plot(pacfHydro, main = "Original \nHydro PACF")
```


Original Hydro
Trillions of Btu
Year
Original
Hydro ACF
ACF
Lag
Original
Hydro PACF
Partial ACF
Lag