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

Assignment 3 - Due date 02/08/22

Emre Yurtbay

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_Consumption 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.\

```
#Load/install required package here
library(lubridate)
```

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

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

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

```
## Warning: package 'forecast' was built under R version 4.1.2
```

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

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

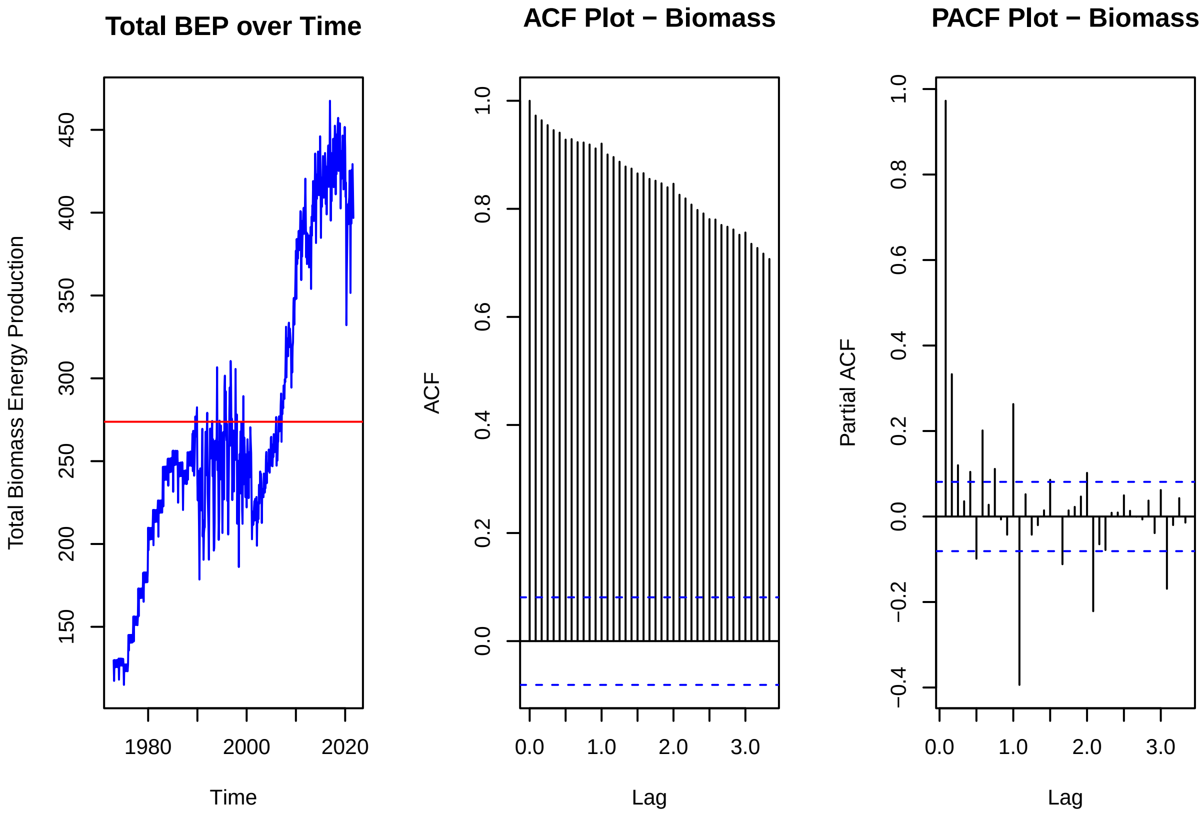
```
tab <- read_excel("/Users/emreyurtbay/Documents/Duke/env790/ENV790_TimeSeriesAnalysis_Sp2022/Data/Table_

# remove the first row
tab = tab[-1,]

# create data frame and ts object for the required series
df = tab[,c(1, 4:6)]
df$`Total Biomass Energy Production` = as.numeric(df$`Total Biomass Energy Production`)
df$`Total Renewable Energy Production` = as.numeric(df$`Total Renewable Energy Production`)
df$`Hydroelectric Power Consumption` = as.numeric(df$`Hydroelectric Power Consumption`)
ts_df <- ts(df[, 2:4], frequency = 12, start = c(1973, 1))
```

Trend Component

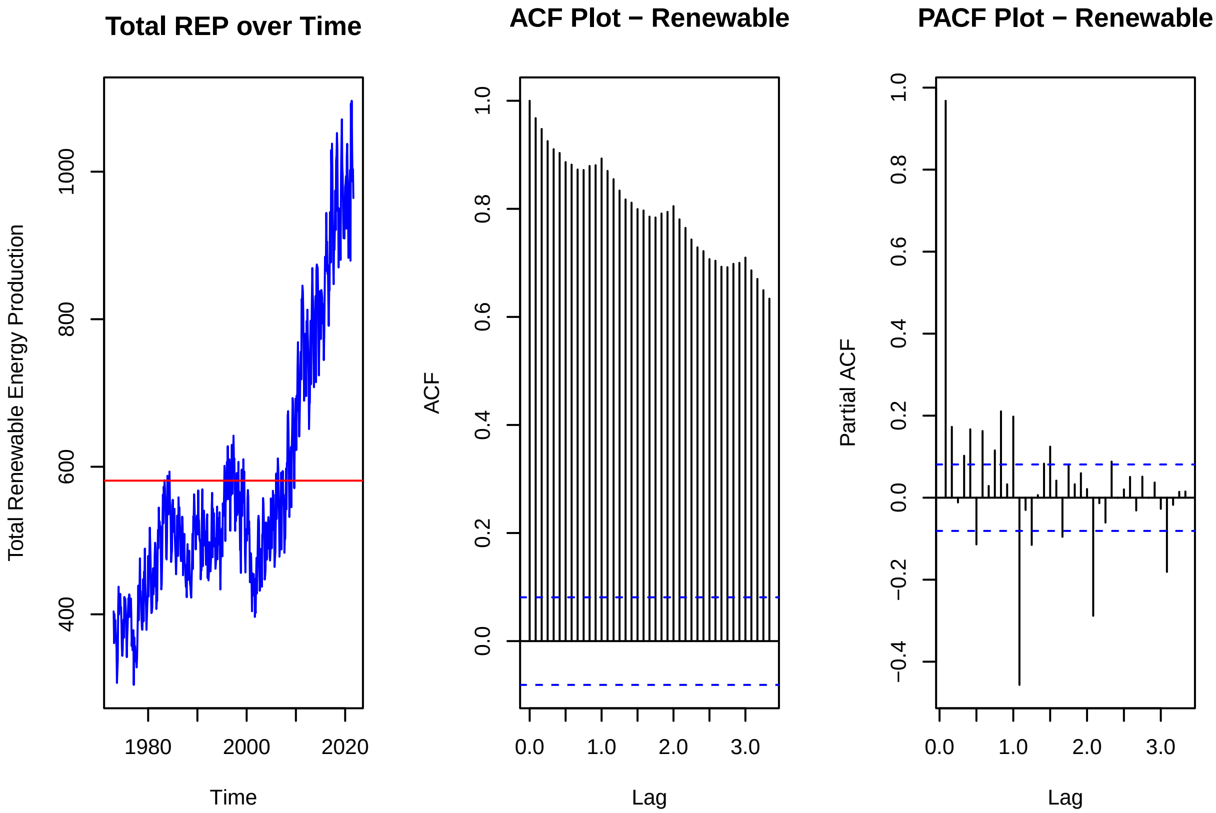
Q1

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)

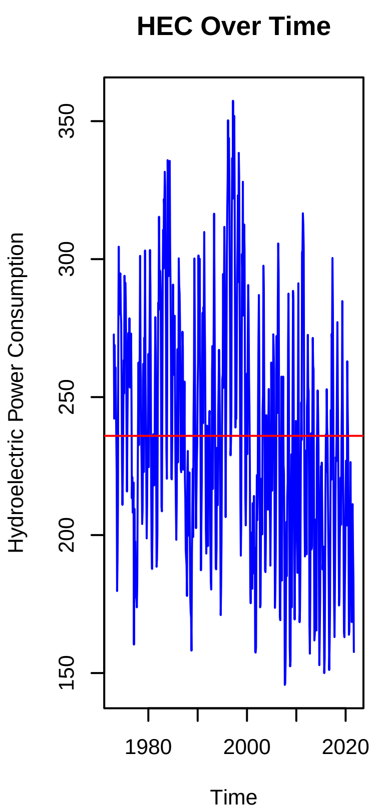
```
# Biomass
par(mfrow = c(1, 3))
plot(ts_df[,c("Total Biomass Energy Production")], type="l", col = "blue",
     main = "Total BEP over Time", xlab = "Time",
     ylab ="Total Biomass Energy Production")
abline(h = mean(ts_df[,c("Total Biomass Energy Production")]), col = "red")
ts_df[,c("Total Biomass Energy Production")] %>%
  acf(lag.max = 40,
      main = "ACF Plot - Biomass")
ts_df[,c("Total Biomass Energy Production")] %>%
  pacf(lag.max = 40,
      main = "PACF Plot - Biomass")
```

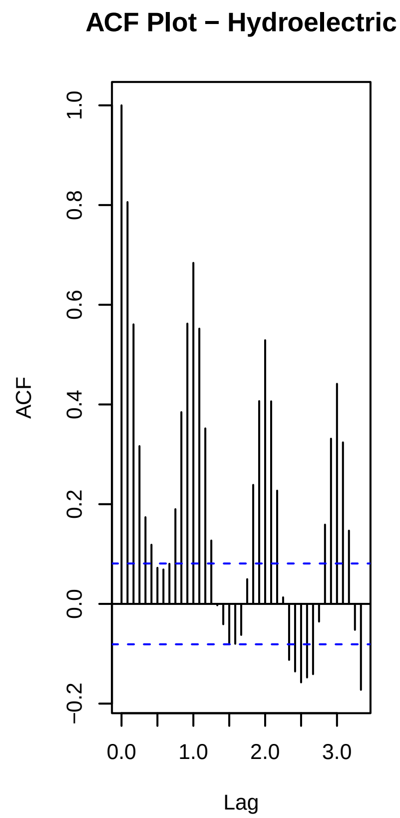


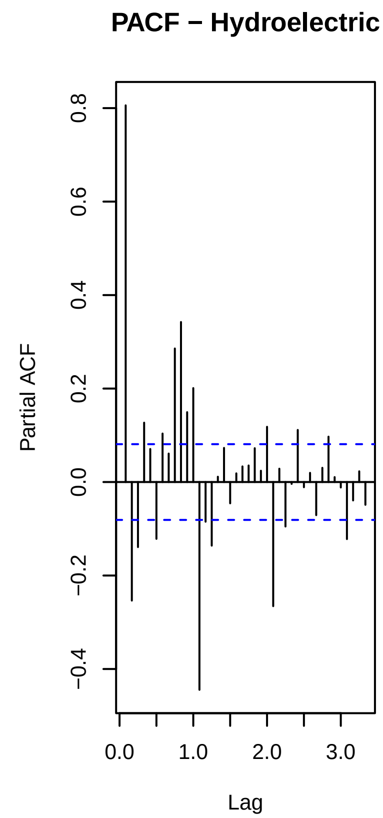

```r
par(mfrow = c(1, 3))
plot(ts_df[,c("Total Renewable Energy Production")], type="l", col = "blue",
     main = "Total REP over Time", xlab = "Time",
     ylab ="Total Renewable Energy Production")
abline(h = mean(ts_df[,c("Total Renewable Energy Production")]), col = "red")
ts_df[,c("Total Renewable Energy Production")] %>%
  acf(lag.max = 40,
      main = "ACF Plot - Renewable")
ts_df[,c("Total Renewable Energy Production")] %>%
  pacf(lag.max = 40,
      main = "PACF Plot - Renewable")
```




```r
par(mfrow = c(1, 3))
plot(ts_df[,c("Hydroelectric Power Consumption")], type="l", col = "blue",
     main = "HEC Over Time", xlab = "Time",
     ylab ="Hydroelectric Power Consumption")
abline(h = mean(ts_df[,c("Hydroelectric Power Consumption")]), col = "red")
ts_df[,c("Hydroelectric Power Consumption")] %>%
  acf(lag.max = 40,
      main = "ACF Plot - Hydroelectric")
ts_df[,c("Hydroelectric Power Consumption")] %>%
  pacf(lag.max = 40,
      main = "PACF - Hydroelectric")
```







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?

Total Biomass Energy Production seems to have a strong positive trend, though it's unclear to me whether that trend is linear or not, but my judgment tells me its probably linear. Similarly, Total Renewable Energy Production also seems to have a strong positive trend, with a linear-like trend. Hydroelectric Power Consumption doesn't seem to have a clear strong trend, but perhaps there is a very weak, approximately linear, negative 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.

```
#Create vector t
nobs <- nrow(df)
t <- c(1:nobs)

#Fit a linear trend to TS Biomass
reg_bio=lm(df$`Total Biomass Energy Production`~t)
summary(reg_bio)
```

```
##
## Call:
```

```
## lm(formula = df$`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
beta0_bio=as.numeric(reg_bio$coefficients[1])
beta1_bio=as.numeric(reg_bio$coefficients[2])
```

For biomass, the y intercept is 134.8 and the slope is 0.4744. This means that the predicted value of biomass energy production at the time we began data collection, January 1973, is 134.8. We expect that biomass energy production increases at a rate of 0.47 units per month.

```r
#Fit a linear trend to TS Biomass
reg_r=lm(df$`Total Renewable Energy Production`~t)
summary(reg_r)
```

```
##
## Call:
## lm(formula = df$`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
beta0_r=as.numeric(reg_r$coefficients[1])
beta1_r=as.numeric(reg_r$coefficients[2])
```

For renewable energy, the y intercept is 323.18 and the slope is 0.88. This means that the predicted value of renewable energy production at the time we began data collection, January 1973, is 323.18. We expect that renewable energy production increases at a rate of 0.88 units per month.

```r
#Fit a linear trend to TS Biomass
reg_he=lm(df$`Hydroelectric Power Consumption`~t)
summary(reg_he)
```

```
##
## Call:
## lm(formula = df$`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
```

```
beta0_he=as.numeric(reg_he$coefficients[1])
beta1_he=as.numeric(reg_he$coefficients[2])
```

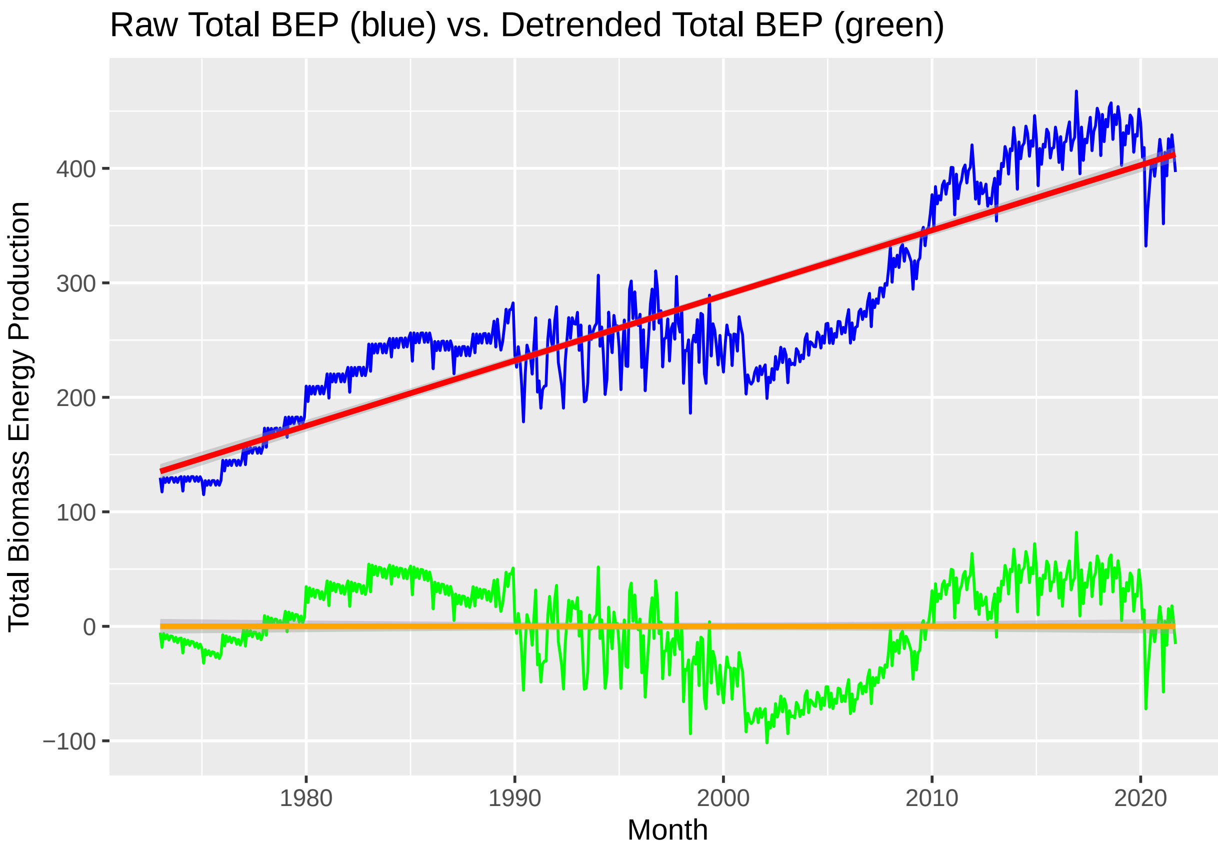
For hydroelectric power consumption, the y intercept is 259.18 and the slope is 0.88. This means that the predicted value of hydroelectric power consumption at the time we began data collection, January 1973, is 259.18. We expect that hydroelectric power consumption decreases at a rate of 0.079 units per month.

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?

```
#remove the trend from series
detrend_bio <- df$`Total Biomass Energy Production`-(beta0_bio+beta1_bio*t)

#Understanding what we did
ggplot(df, aes(x=Month, y=`Total Biomass Energy Production`)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=detrend_bio), col="green")+
            geom_smooth(aes(y=detrend_bio),color="orange",method="lm")+
  ggtitle("Raw Total BEP (blue) vs. Detrended Total BEP (green)")
```

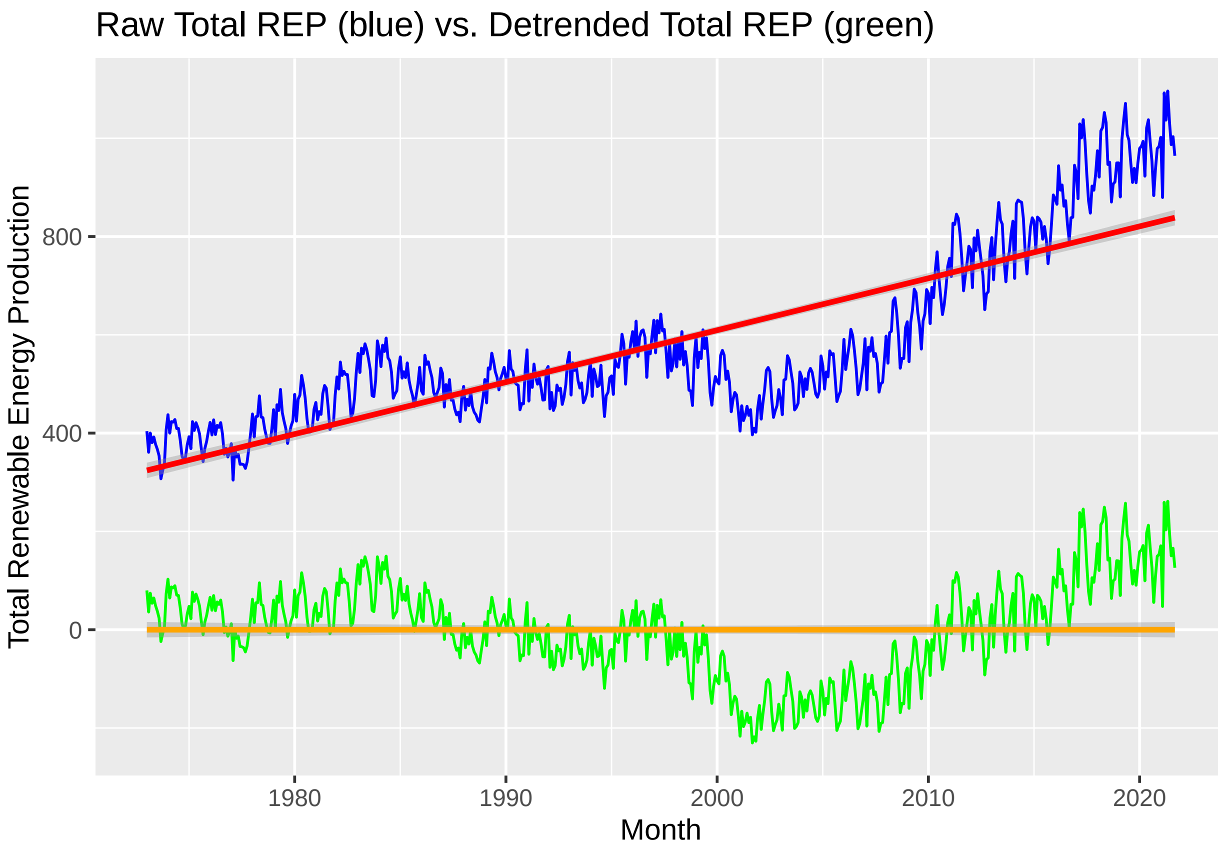
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



For the biomass data, we see a difference between the raw series and the detrended series. The raw series has a strong increasing trend, while the detrended series no longer does. However, the overall "structure" (local peaks and troughs) of the raw series and the detrended series do indeed look very similar.

```
#remove the trend from series
detrend_r <- df$`Total Renewable Energy Production`-(beta0_r+beta1_r*t)

#Understanding what we did
ggplot(df, aes(x=Month, y=`Total Renewable Energy Production`)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=detrend_r), col="green")+
            geom_smooth(aes(y=detrend_r),color="orange",method="lm")+
  ggtitle("Raw Total REP (blue) vs. Detrended Total REP (green)")
```

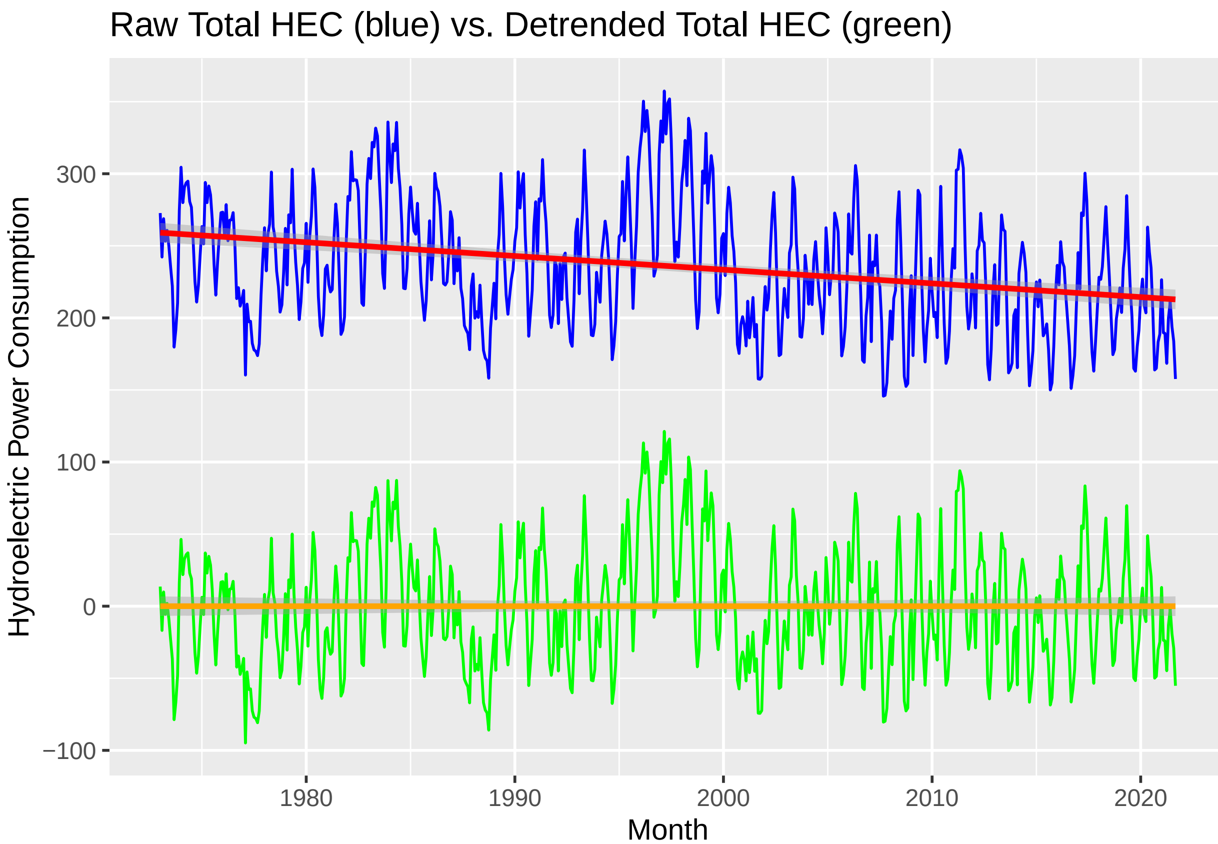
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



For the renewable energy data, we see a difference between the raw series and the detrended series. The raw series has a strong increasing trend, while the detrended series no longer does. However, the overall "structure" (local peaks and troughs) of the raw series and the detrended series do indeed look very similar.

```
#remove the trend from series
detrend_he <- df$`Hydroelectric Power Consumption`-(beta0_he+beta1_he*t)

#Understanding what we did
ggplot(df, aes(x=Month, y=`Hydroelectric Power Consumption`)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=detrend_he), col="green")+
            geom_smooth(aes(y=detrend_he),color="orange",method="lm")+
  ggtitle("Raw Total HEC (blue) vs. Detrended Total HEC (green)")
```

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



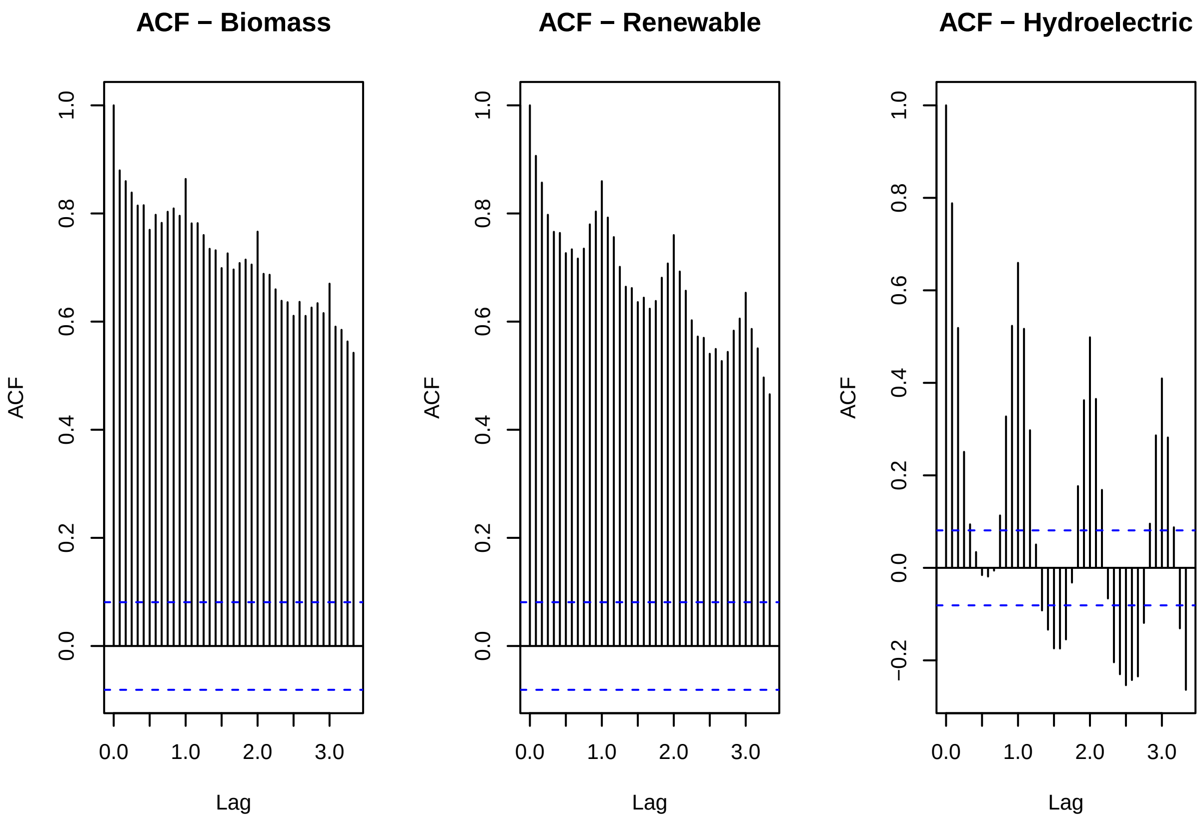
The detrended HEC looks very similar to the raw HEC, but the detrended HEC is shifted downward such that the mean is approximately 0. The green series does not show the weak negative trend present in the blue series. The overall "structure", as it were, of the raw series and the detrended series do indeed look very similar.

Q5

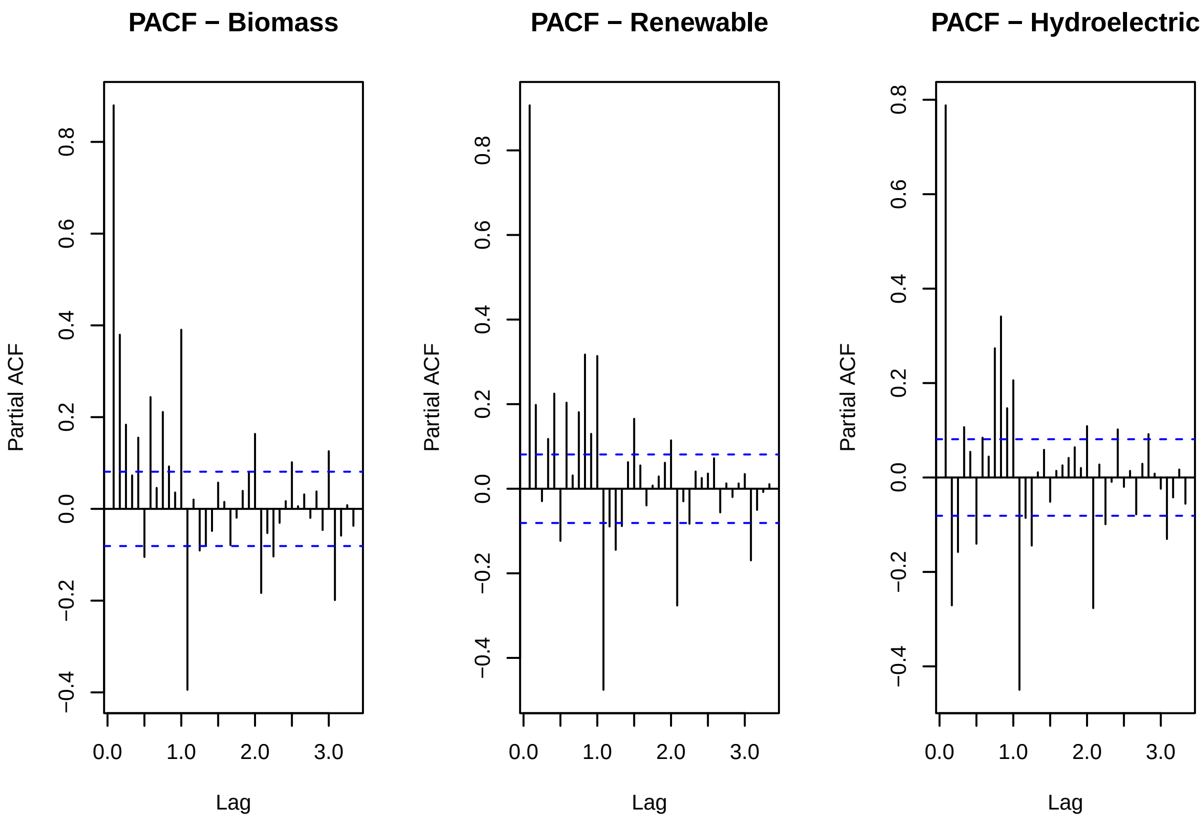
Plot ACF and PACF for the detrended series and compare with the plots from Q1. Did the plots change? How?

```
ts_df_detrend <- ts(cbind(detrend_bio, detrend_r, detrend_he),
            frequency = 12, start = c(1973, 1))
```

```
par(mfrow = c(1, 3))
ts_df_detrend[,c("detrend_bio")] %>%
  acf(lag.max = 40,
      main = "ACF - Biomass")
ts_df_detrend[,c("detrend_r")] %>%
  acf(lag.max = 40,
      main = "ACF - Renewable")
ts_df_detrend[,c("detrend_he")] %>%
  acf(lag.max = 40,
      main = "ACF - Hydroelectric")
```




```
par(mfrow = c(1, 3))
ts_df_detrend[,c("detrend_bio")] %>%
  pacf(lag.max = 40,
      main = "PACF - Biomass")
ts_df_detrend[,c("detrend_r")] %>%
  pacf(lag.max = 40,
      main = "PACF - Renewable")
ts_df_detrend[,c("detrend_he")] %>%
  pacf(lag.max = 40,
      main = "PACF - Hydroelectric")
```



The ACF plots aren't exactly the same as before - for instance, the scalloped shape we see in the renewable energy plot is more pronounced in the detrended series, and the negative values in the ACF for the hydroelectric series are more pronounced. However, the overall shapes of the detrended plots are pretty similar to their "raw" counterparts. The PACF plots don't look to different, though its hard to detect differences just looking at the plots alone/

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.

```
nobs <- nrow(df)

dummies <- seasonaldummy(ts_df[,c("Total Biomass Energy Production")])
#Then fit a linear model to the seasonal dummies
seas_means_model_b=lm(ts_df[,c("Total Biomass Energy Production")]~dummies)
summary(seas_means_model_b)
```

```
##
## Call:
## lm(formula = ts_df[, c("Total Biomass Energy Production")] ~
##     dummies)
```

```
##
## 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 ***
## dummiesJan    -1.498     18.238  -0.082   0.9346
## dummiesFeb   -30.582     18.238  -1.677   0.0941 .
## dummiesMar    -8.873     18.238  -0.486   0.6268
## dummiesApr   -21.009     18.238  -1.152   0.2498
## dummiesMay   -14.065     18.238  -0.771   0.4409
## dummiesJun   -19.601     18.238  -1.075   0.2829
## dummiesJul    -3.499     18.238  -0.192   0.8479
## dummiesAug    -0.252     18.238  -0.014   0.9890
## dummiesSep   -12.518     18.238  -0.686   0.4928
## dummiesOct    -3.629     18.331  -0.198   0.8432
## dummiesNov    -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
```

```r
#Look at the regression coefficient. These will be the values of Beta
#Store regression coefficients
beta_int_b=seas_means_model_b$coefficients[1]
beta_coeff_b=seas_means_model_b$coefficients[2:12]
```

The intercept represents the baseline level of the time series in the month of December, our baseline month. The coefficient for January represents the difference between the level of the series in December and January. So, since the coefficient for January is -1.48, the level of the series is 1.48 units lower in January than in December. This interpretations holds for all subsequent months. Notice the p-values for all regression coefficients except the intercept are non-significant; this may indicate we do not see significant seasonality in the Biomass data.

```r
nobs <- nrow(df)

dummies <- seasonaldummy(ts_df[,c("Total Renewable Energy Production")])
#Then fit a linear model to the seasonal dummies
seas_means_model_r=lm(ts_df[,c("Total Renewable Energy Production")]~dummies)
summary(seas_means_model_r)
```

```
##
## Call:
## lm(formula = ts_df[, c("Total Renewable Energy Production")] ~
##     dummies)
##
## 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 ***
## dummiesJan     11.793     35.828   0.329   0.7422
## dummiesFeb    -40.992     35.828  -1.144   0.2530
## dummiesMar     21.892     35.828   0.611   0.5414
## dummiesApr      8.908     35.828   0.249   0.8037
## dummiesMay     37.500     35.828   1.047   0.2957
## dummiesJun     19.465     35.828   0.543   0.5871
## dummiesJul      8.115     35.828   0.227   0.8209
## dummiesAug    -18.359     35.828  -0.512   0.6086
## dummiesSep    -62.115     35.828  -1.734   0.0835 .
## dummiesOct    -51.377     36.012  -1.427   0.1542
## dummiesNov    -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
```

```r
#Look at the regression coefficient. These will be the values of Beta
#Store regression coefficients
beta_int_r=seas_means_model_r$coefficients[1]
beta_coeff_r=seas_means_model_r$coefficients[2:12]
```

The intercept represents the baseline level of the time series in the month of December, our baseline month. The coefficient for January represents the difference between the level of the series in December and January. So, since the coefficient for January is 11.793, the level of the series is 11.793 units higher in January than in December. This interpretations holds for all subsequent months. Notice the p-values for all regression coefficients except the intercept are non-significant; this may indicate we do not see significant seasonality in the renewable energy data.

```r
nobs <- nrow(df)

dummies <- seasonaldummy(ts_df[,c("Hydroelectric Power Consumption")])
#Then fit a linear model to the seasonal dummies
seas_means_model_h=lm(ts_df[,c("Hydroelectric Power Consumption")]~dummies)
summary(seas_means_model_h)
```

```
##
## Call:
## lm(formula = ts_df[, c("Hydroelectric Power Consumption")] ~
##     dummies)
##
## 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 ***
## dummiesJan     13.558      6.883   1.970  0.04936 *
## dummiesFeb     -8.090      6.883  -1.175  0.24037
## dummiesMar     20.067      6.883   2.915  0.00369 **
## dummiesApr     16.619      6.883   2.414  0.01607 *
```

```
## dummiesMay    39.961      6.883   5.805 1.06e-08 ***
## dummiesJun    31.315      6.883   4.549 6.57e-06 ***
## dummiesJul    10.511      6.883   1.527  0.12732    
## dummiesAug   -17.853      6.883  -2.594  0.00974 ** 
## dummiesSep   -49.852      6.883  -7.242 1.43e-12 ***
## dummiesOct   -48.086      6.919  -6.950 9.96e-12 ***
## dummiesNov   -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
```

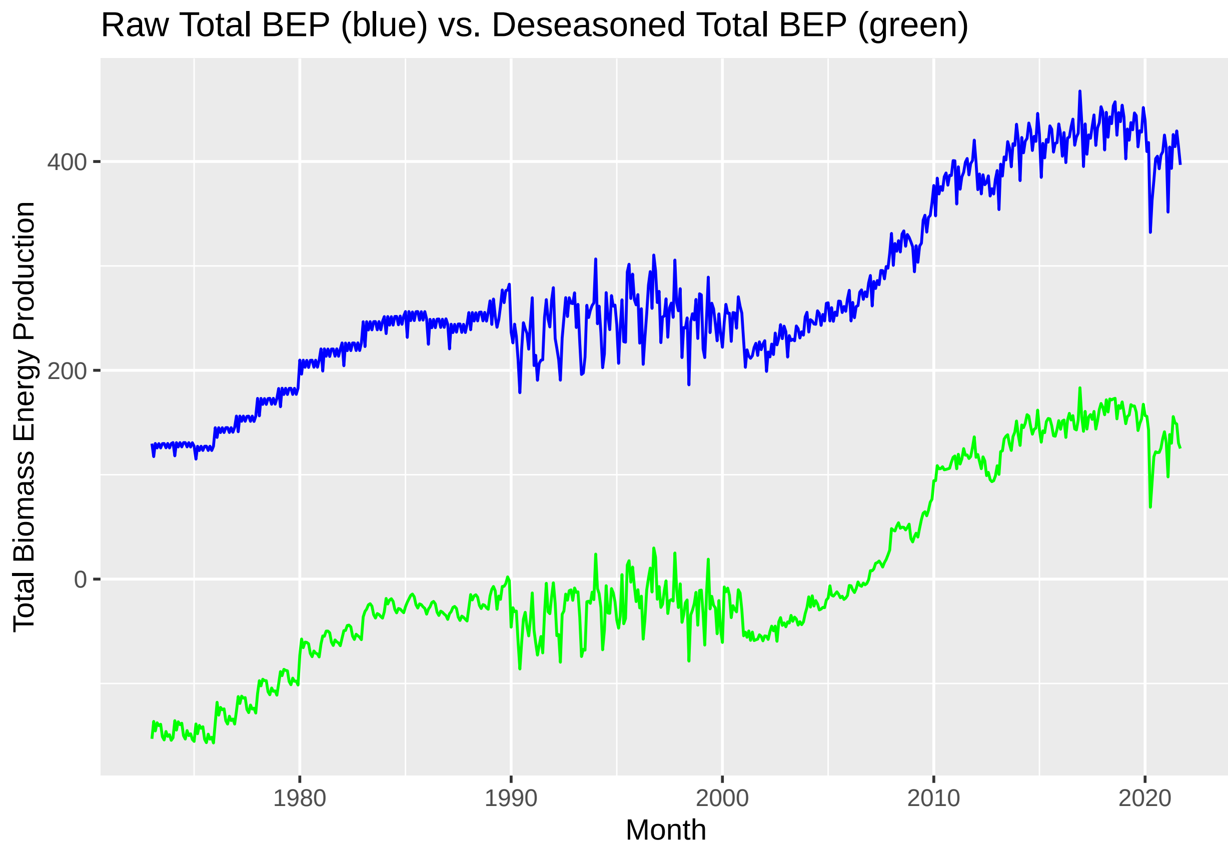
```
#Look at the regression coefficient. These will be the values of Beta
#Store regression coefficients
beta_int_h=seas_means_model_h$coefficients[1]
beta_coeff_h=seas_means_model_h$coefficients[2:12]
```

The intercept represents the baseline level of the time series in the month of December, our baseline month. The coefficient for January represents the difference between the level of the series in December and January. So, since the coefficient for January is 13.558, the level of the series is 13.558 units higher in January than in December. This interpretations holds for all subsequent months. Notice the p-values for many regression coefficients except the intercept are now significant; this may indicate we do see some significant seasonality in the hydroelectric energy data.

Q7

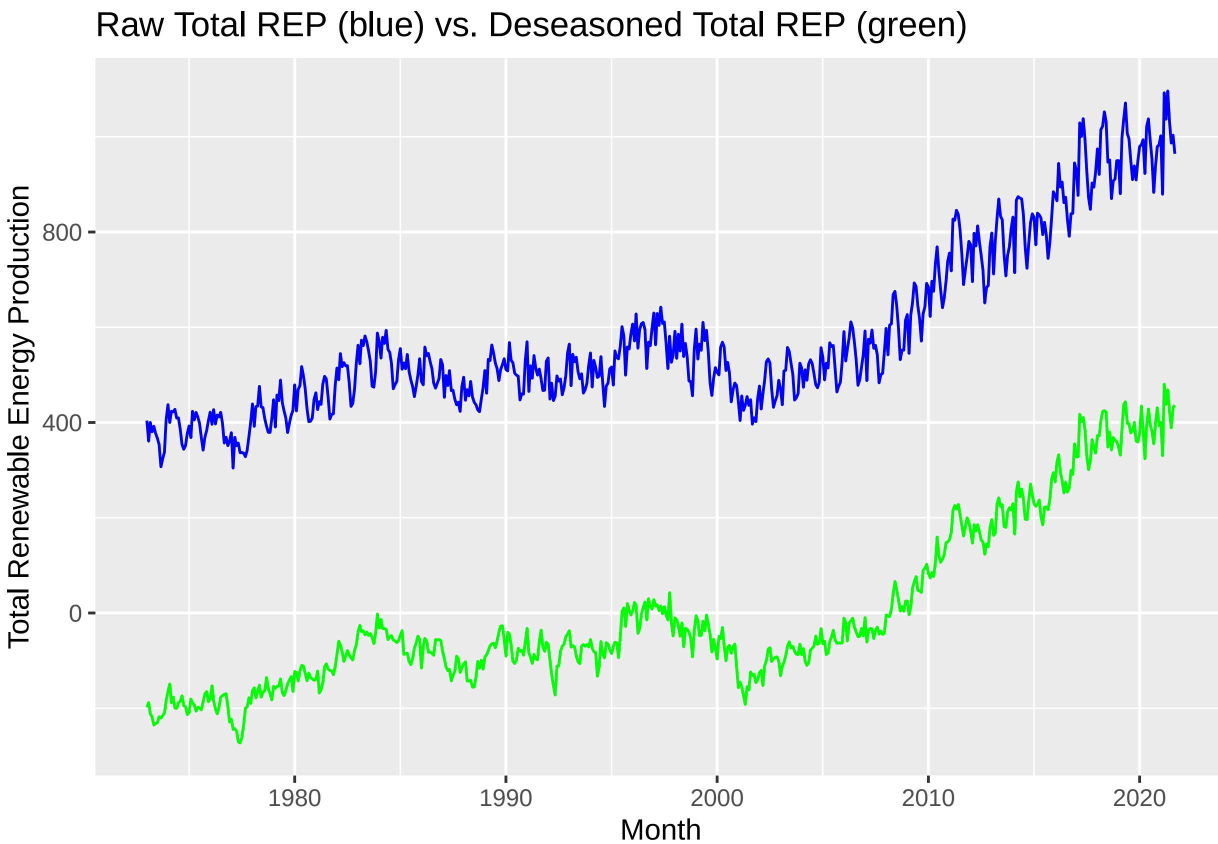
Use the regression coefficients from Q6 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

```
b_comp=array(0,nobs)
for(i in 1:nobs){
  b_comp[i]=(beta_int_b+beta_coeff_b%*%dummies[i,])
}

deseason_b <- ts_df[,c("Total Biomass Energy Production")]-b_comp

#Understanding what we did
ggplot(df, aes(x=Month, y=`Total Biomass Energy Production`)) +
            geom_line(color="blue") +
            geom_line(aes(y=deseason_b), col="green") +
  ggtitle("Raw Total BEP (blue) vs. Deseasoned Total BEP (green)")
```



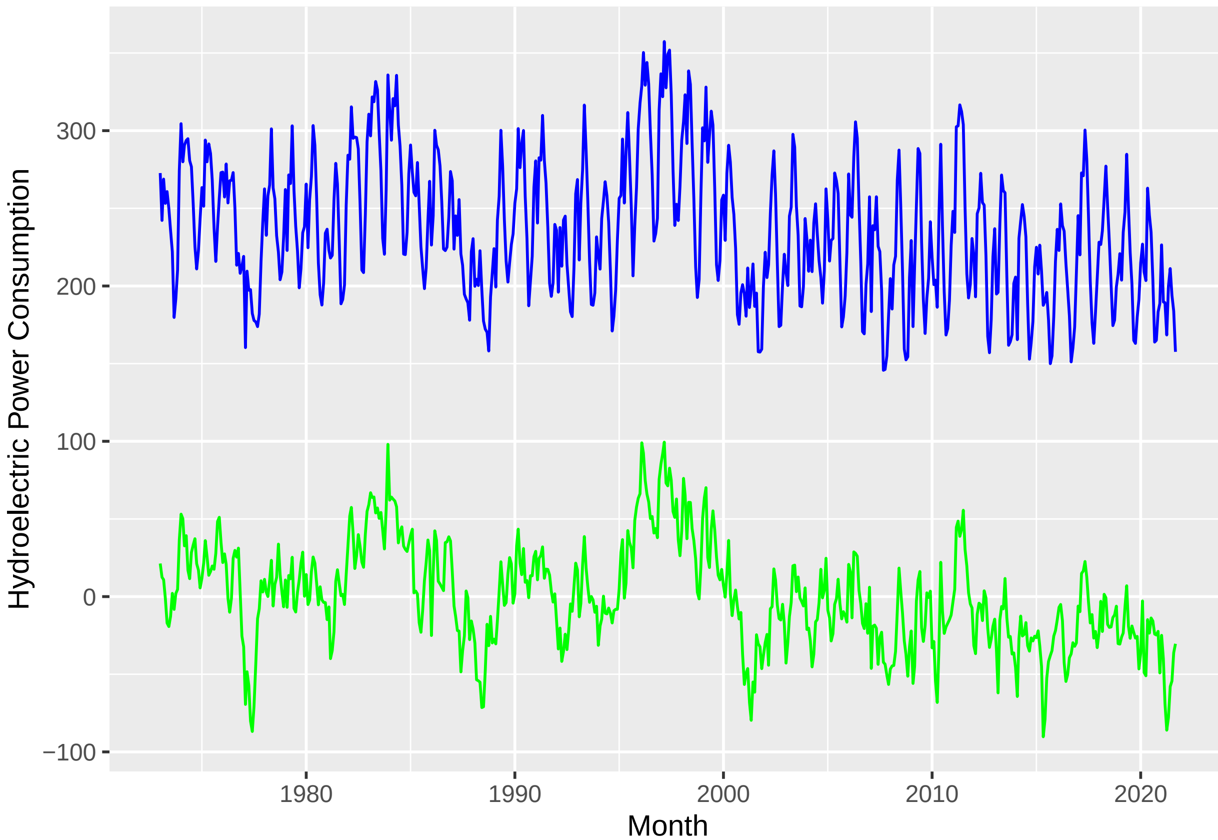
In the case of the biomass energy, the deseasoned series looks pretty similar to the raw series, except that the deseasoned series is shifted so its mean is near zero. The local structures (that is, peaks and troughs) of the 2 series are pretty similar, and this is to be expected since there was not much monthly seasonal behavior in the original series.

```
r_comp=array(0,nobs)
for(i in 1:nobs){
  r_comp[i]=(beta_int_r+beta_coeff_r%*%dummies[i,])
}

deseason_r <- ts_df[,c("Total Renewable Energy Production")]-r_comp

#Understanding what we did
ggplot(df, aes(x=Month, y=`Total Renewable Energy Production`)) +
            geom_line(color="blue") +
            geom_line(aes(y=deseason_r), col="green")+
  ggtitle("Raw Total REP (blue) vs. Deseasoned Total REP (green)")
```



Again, In the case of renewable energy, the deseasoned series looks pretty similar to the raw series, except that the deseasoned series is shifted so its mean is near zero. The local structures (that is, peaks and troughs) of the 2 series are more different than in the biomass example. The seasonality of the renewable energy series is slightly stronger than that of the biomass series.

```
h_comp=array(0,nobs)
for(i in 1:nobs){
  h_comp[i]=(beta_int_h+beta_coeff_h%*%dummies[i,])
}

deseason_h <- ts_df[,c("Hydroelectric Power Consumption")]-h_comp

#Understanding what we did
ggplot(df, aes(x=Month, y=`Hydroelectric Power Consumption`)) +
            geom_line(color="blue") +
            geom_line(aes(y=deseason_h), col="green")
```




```
ggtitle("Raw Total HPC (blue) vs. Deseasoned Total HPC (green)")
```

```
## $title
## [1] "Raw Total HPC (blue) vs. Deseasoned Total HPC (green)"
##
## attr(,"class")
## [1] "labels"
```

Again, the deseasoned series is shifted so its mean is near zero. The local structures (that is, peaks and troughs) of the 2 series are pretty different, especially compared to the previous 2 examples The seasonality of the hydroelectric power series is stronger than that of the previous two series.

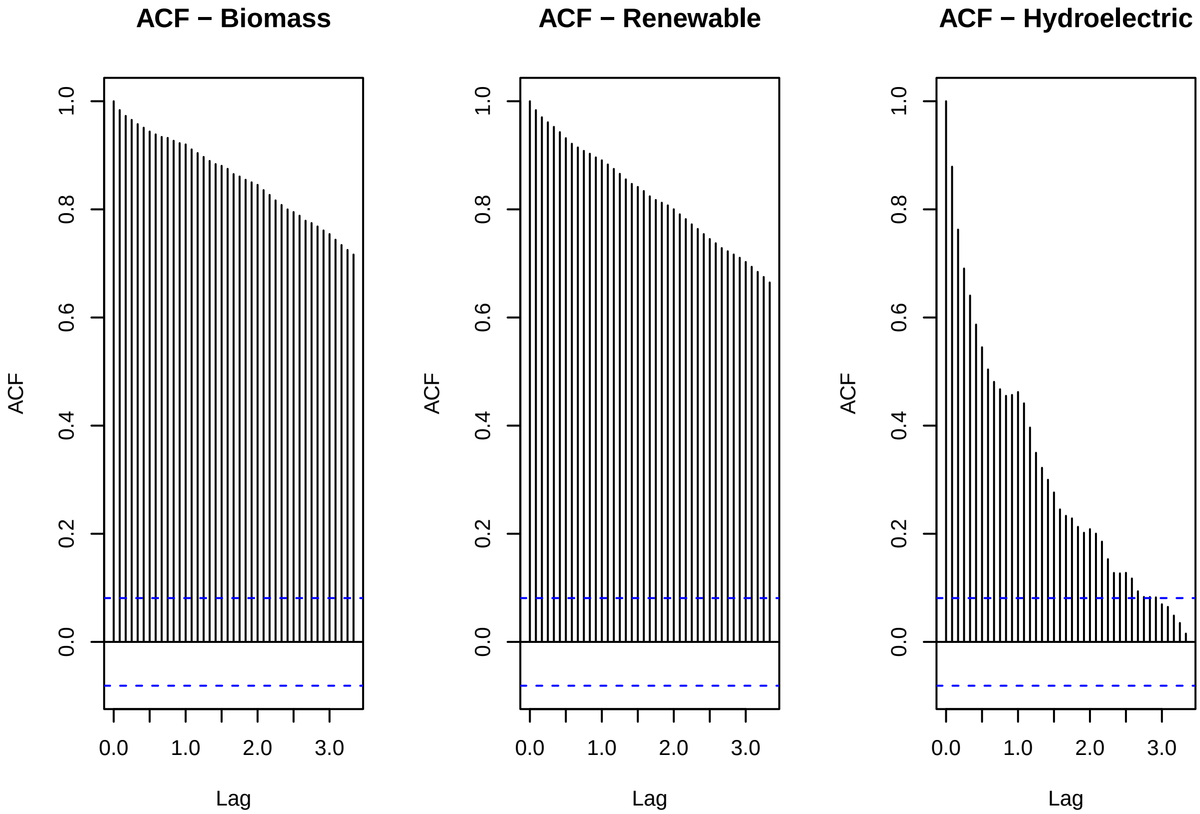
Q8

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

```
ts_df_deseason <- ts(cbind(deseason_b, deseason_r, deseason_h),
            frequency = 12, start = c(1973, 1))
```

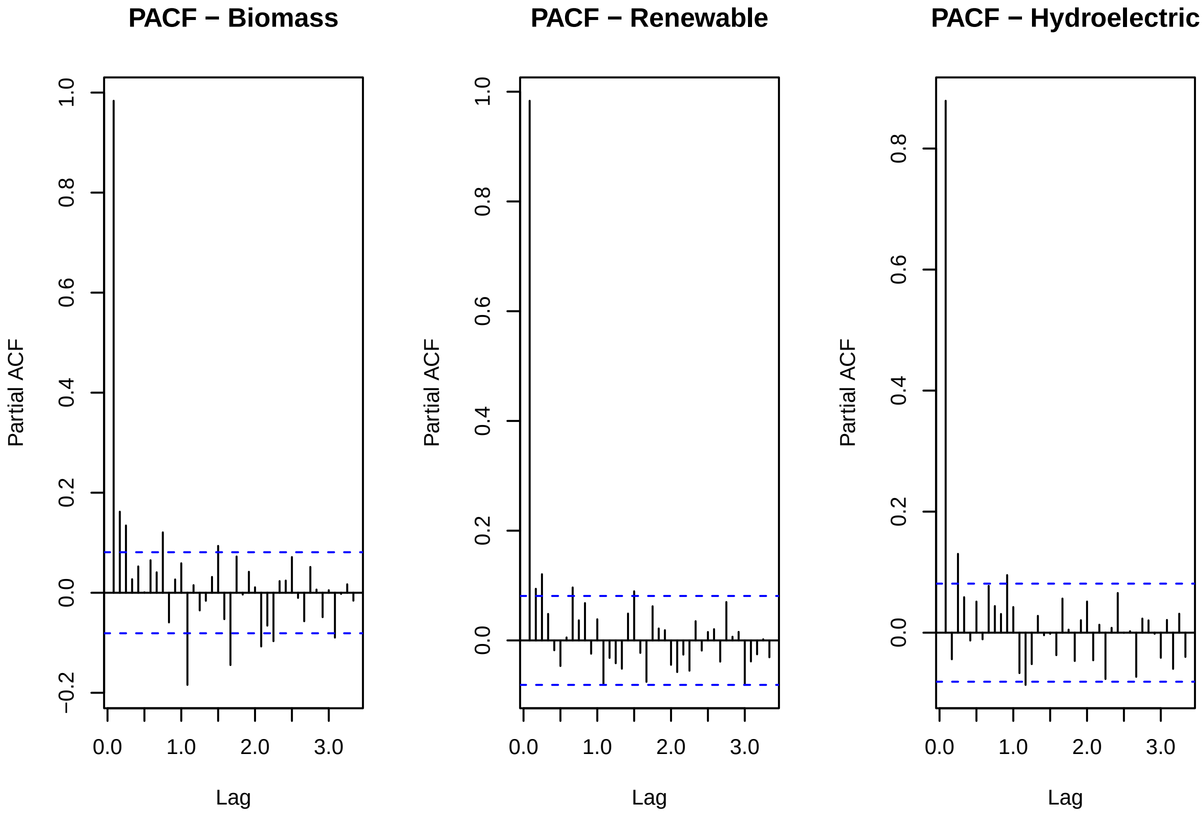
```
par(mfrow = c(1, 3))
ts_df_deseason[,c("deseason_b")] %>%
  acf(lag.max = 40,
      main = "ACF - Biomass")
ts_df_deseason[,c("deseason_r")] %>%
  acf(lag.max = 40,
      main = "ACF - Renewable")
ts_df_deseason[,c("deseason_h")] %>%
```

```r
  acf(lag.max = 40,
      main = "ACF - Hydroelectric")
```



The ACF for biomass looks very similar. The ACF for renewable has changed somewhat, as the scalloped shape in the plot is now gone. The ACF for hydroelectric looks extremely different - the negative values have disappeared. This makes sense - we have seseasoned the data, and the hydroelectric data showed significant seasonality.

```r
par(mfrow = c(1, 3))
ts_df_deseason[,c("deseason_b")] %>%
  pacf(lag.max = 40,
      main = "PACF - Biomass")
ts_df_deseason[,c("deseason_r")] %>%
  pacf(lag.max = 40,
      main = "PACF - Renewable")
ts_df_deseason[,c("deseason_h")] %>%
  pacf(lag.max = 40,
      main = "PACF - Hydroelectric")
```



The PACF plots don't look drastically different, though perhaps more of the values fall within the dotted lines for the deseasoned data compared to that of the raw data.