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

Assignment 3 - Due date 02/08/22

Aasha Reddy

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(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
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(tseries)
```

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

```
library(readxl)
library(lubridate)
```

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

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

```
#Importing data set
repc_raw <- read_excel("/Users/Aasha Reddy/Documents/Statistics - Duke University/2022 Spring/Time Seri
repc_raw <- repc_raw[-1,]
```

```
# Clean data
repc_raw <- repc_raw %>%
  select(`Total Biomass Energy Production`,
         `Total Renewable Energy Production`,
         `Hydroelectric Power Consumption`,
         Month)

# change variables to numeric
repc_raw <- repc_raw %>%
  mutate(`Total Biomass Energy Production` = as.numeric(`Total Biomass Energy Production`),
         `Total Renewable Energy Production` = as.numeric(`Total Renewable Energy Production`),
         `Hydroelectric Power Consumption` = as.numeric(`Hydroelectric Power Consumption`))

# Change Month column to date
repc_raw <- repc_raw %>%
  mutate(Month = ymd(Month))

# transform data into time series object
repc <- ts(data = repc_raw %>% select(-Month), start = c(1973, 1), frequency = 12)
```

##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)

Total Biomass Energy Production:

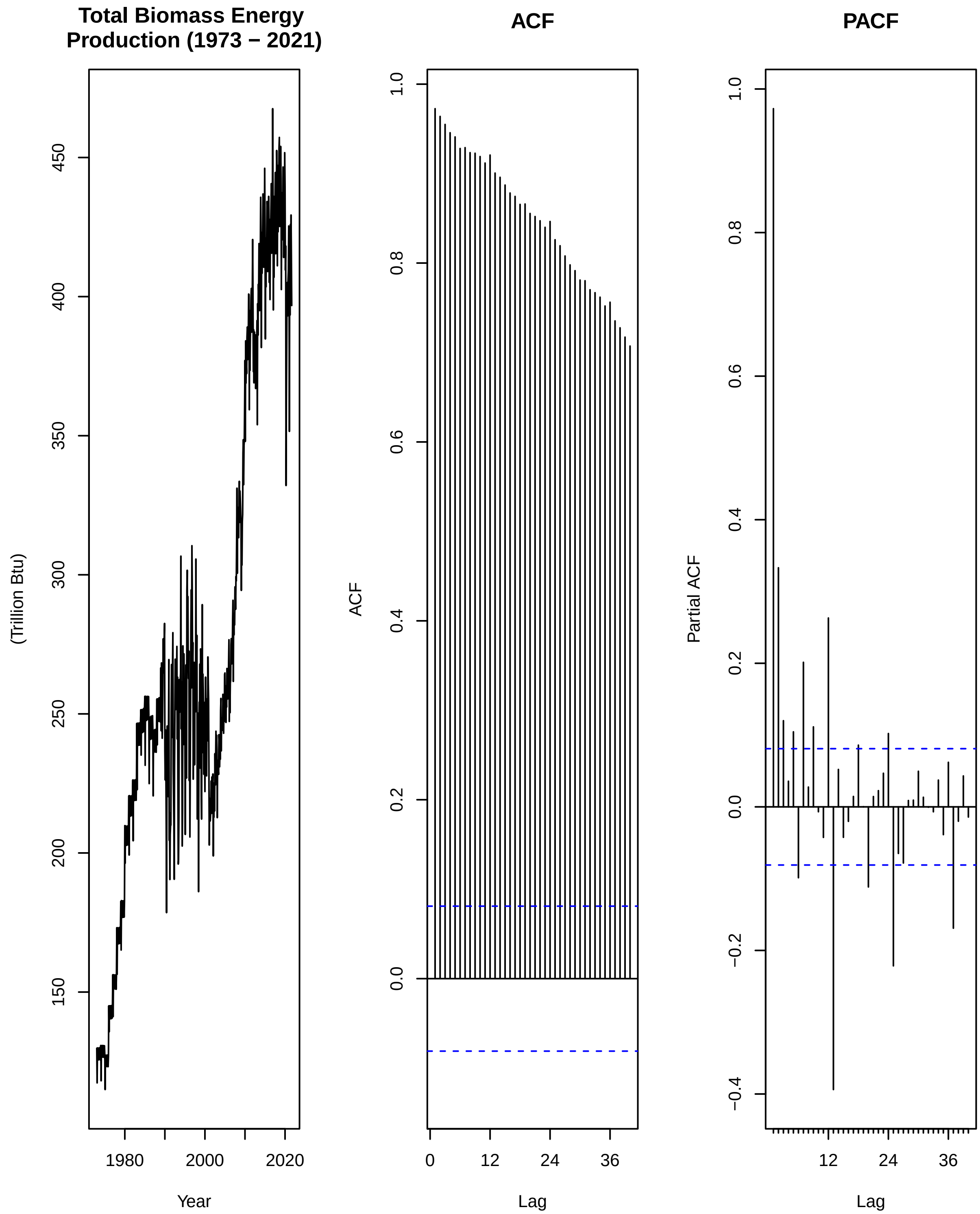
```
# Divide window into 1 row 3 columns
par(mfrow=c(1, 3))

# Total Biomass Energy Production
plot(repc[, "Total Biomass Energy Production"],
     main = "Total Biomass Energy \nProduction (1973 - 2021)",
```

```
     ylab = "(Trillion Btu)",
     xlab = "Year")

Acf(repc[,1],lag.max=40,
    main="ACF")

Pacf(repc[,1],lag.max=40,
     main="PACF")
```



Total Renewable Energy Production:

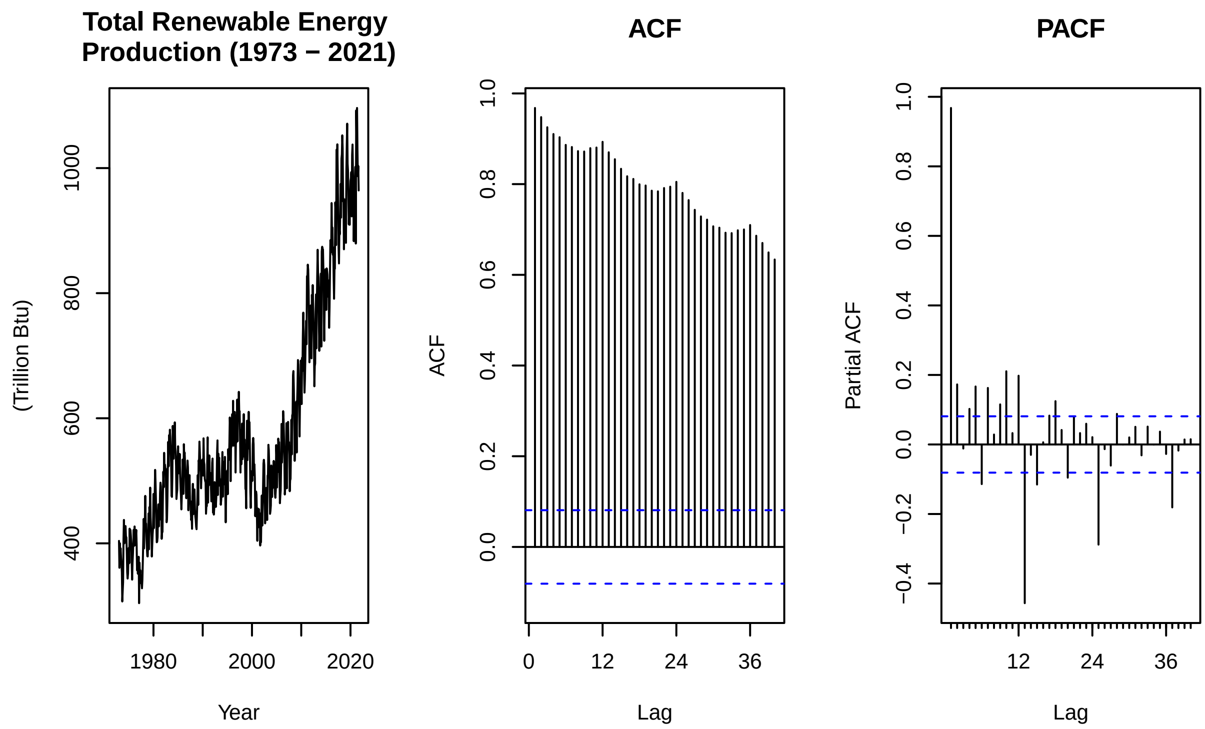
```
# Divide window into 1 row 3 columns
par(mfrow=c(1, 3))

# Total Renewable Energy Production
```

```
plot(repc[, "Total Renewable Energy Production"],
     main = "Total Renewable Energy \nProduction (1973 - 2021)",
     ylab = "(Trillion Btu)",
     xlab = "Year")

Acf(repc[,2],lag.max=40,
    main="ACF")

Pacf(repc[,2],lag.max=40,
     main="PACF")
```



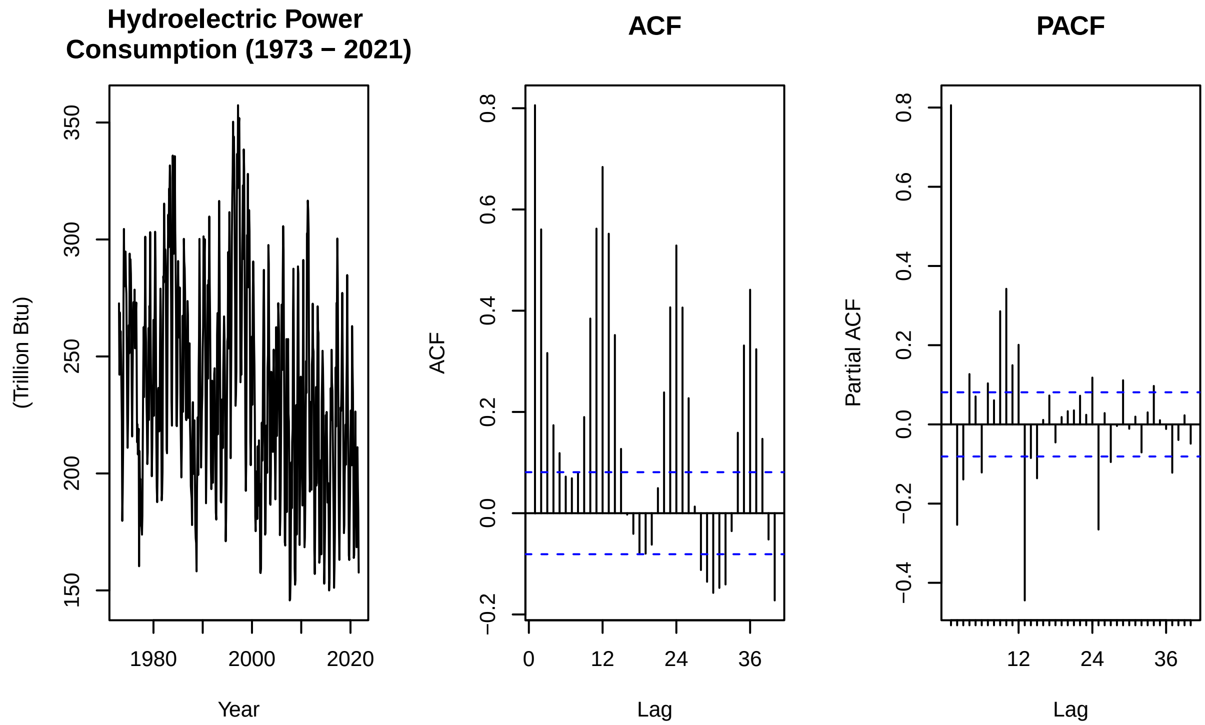
Hydroelectric Power Consumption:

```
# Divide window into 1 row 3 columns
par(mfrow=c(1, 3))

# Hydroelectric Power Consumption
plot(repc[, "Hydroelectric Power Consumption"],
     main = "Hydroelectric Power \nConsumption (1973 - 2021)",
     ylab = "(Trillion Btu)",
     xlab = "Year")

Acf(repc[,3],lag.max=40,
    main="ACF")

Pacf(repc[,3],lag.max=40,
     main="PACF")
```



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?

For Total Biomass Energy Production, the trend appears to be a linear increase. For Total Renewable Energy Production, the trend also appears to be a linear increase. For Hydroelectric Power Consumption, the trend is more difficult to assess in the plot, but it looks to be a linear downward 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.

Total Biomass Energy:

For the below results, we see the coefficient on t is significant. This can be interpreted as: as the time period increases by one unit, we expect the total biomass energy to increase on average by ~0.47. We can interpret the intercept as, at time period 0, we expect the biomass energy to be 134.8.

```r
#Create vector t
t <- c(1:nrow(repc))

be_linear <- lm(repc[,1]~t)
summary(be_linear)
```

```
##
## Call:
## lm(formula = repc[, 1] ~ 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
```

```
# save coefficients
be_linear_coefs = coef(be_linear)
```

Total Renewable Energy:

For the below results, we see the coefficient on t is significant. This can be interpreted as: as the time period increases by one unit, we expect the total renewable energy to increase on average by ~0.88. We can interpret the intercept as, at time period 0, we expect the biomass energy to be 323.18.

```
#Create vector t
re_linear <- lm(repc[,2]~t)
summary(re_linear)
```

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

```
# save coefficients
re_linear_coefs = coef(re_linear)
```

Hydroelectric Power:

For the below results, we see the coefficient on t is significant. This can be interpreted as: as the time period increases by one unit, we expect hydroelectric power to decrease on average by ~0.08. We can interpret the intercept as, at time period 0, we expect the biomass energy to be 259.18.

```
#Create vector t
hp_linear <- lm(repc[,3]~t)
summary(hp_linear)
```

```
##
## Call:
## lm(formula = repc[, 3] ~ 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
```

```
# save coefficints
hp_linear_coefs <- coef(hp_linear)
```

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?

Biomass Energy:

We can see from the below plot after detrending the series is no longer linear increasing. However, the overall pattern and structure of the series is similar.

```
beta0=as.numeric(be_linear_coefs[1])
beta1=as.numeric(be_linear_coefs[2])

# combine the detrended data into
be_detrend <- repc_raw$`Total Biomass Energy Production`-(beta0+beta1*t)


#Understanding what we did
ggplot(repc_raw, aes(x = Month, y = `Total Biomass Energy Production`)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=be_detrend), col="green") +
            geom_smooth(aes(y=be_detrend),color="orange",method="lm")
```

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

Total Renewable Energy:

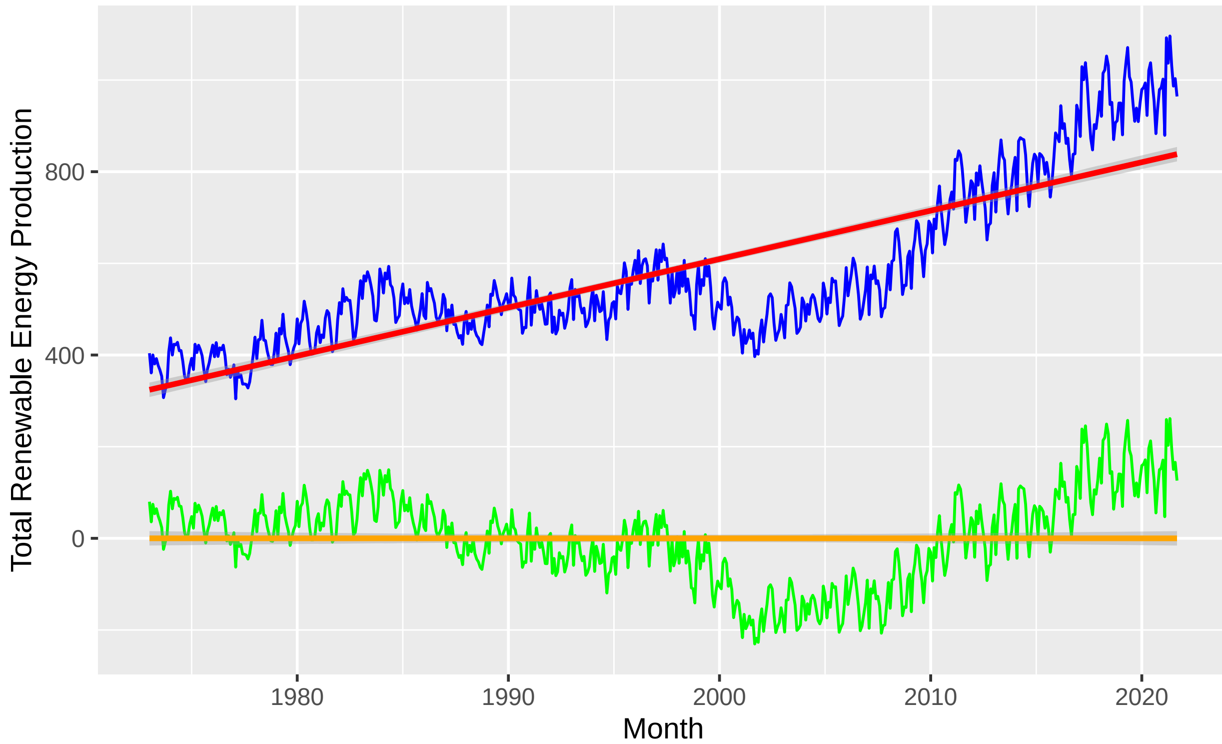
We can see from the below plot after detrending the series is no longer linear increasing. However, the overall pattern and structure of the series is similar.

```
beta0=as.numeric(re_linear_coefs[1])
beta1=as.numeric(re_linear_coefs[2])

# detrend
re_detrend <- repc_raw$`Total Renewable Energy Production`-(beta0+beta1*t)


#Understanding what we did
ggplot(repc_raw, aes(x = Month, y = `Total Renewable Energy Production`)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=re_detrend), col="green") +
            geom_smooth(aes(y=re_detrend),color="orange",method="lm")
```

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



Hydroelectric Power:

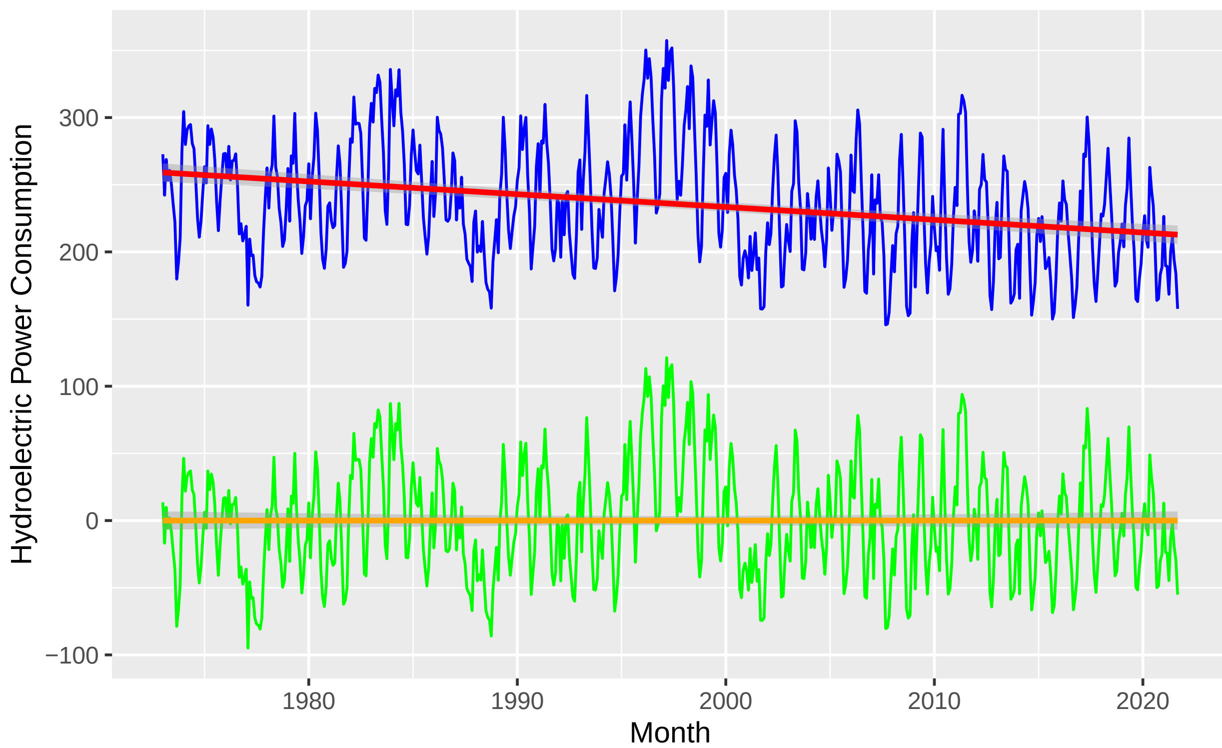
We can see from the below plot after detrending the series is no longer linear decreasing. However, the overall pattern and structure of the series is similar.

```
beta0=as.numeric(hp_linear_coefs[1])
beta1=as.numeric(hp_linear_coefs[2])

# detrend
hp_detrend <- repc_raw$`Hydroelectric Power Consumption`-(beta0+beta1*t)


#Understanding what we did
ggplot(repc_raw, aes(x = Month, y = `Hydroelectric Power Consumption`)) +
            geom_line(color="blue") +
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=hp_detrend), col="green") +
            geom_smooth(aes(y=hp_detrend),color="orange",method="lm")
```

```
## `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?

Total Biomass Energy Production:

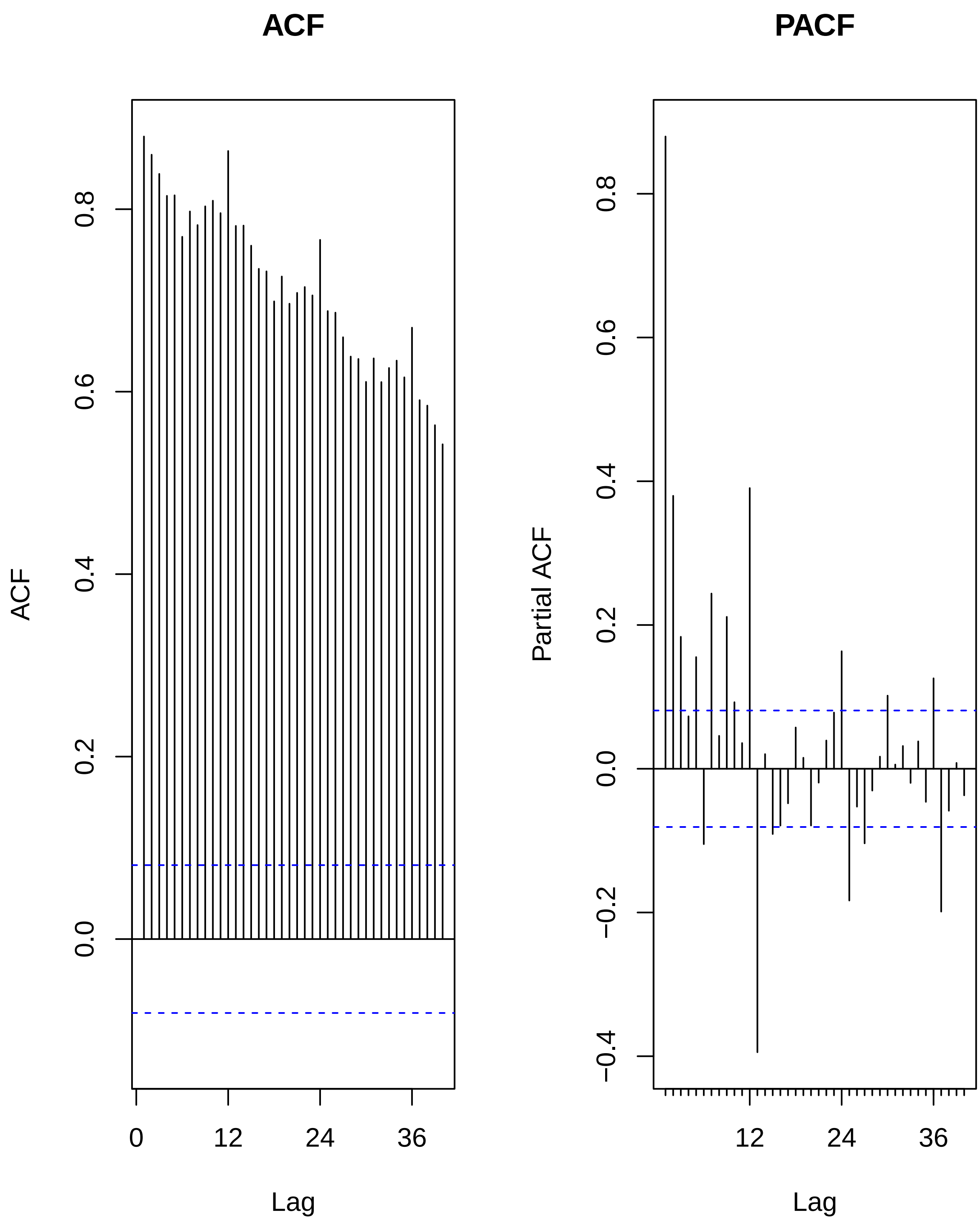
The plots did not change much from Q1. The PACF and ACF plot does not look very different at all.

```
# Divide window into 1 row 3 columns
par(mfrow=c(1, 2))

detrended_ts <- ts(cbind(be_detrend, re_detrend, hp_detrend),
                   frequency = 12,
                   start = c(1973, 1))

Acf(detrended_ts[,1],lag.max=40,
    main="ACF")

Pacf(detrended_ts[,1],lag.max=40,
     main="PACF")
```



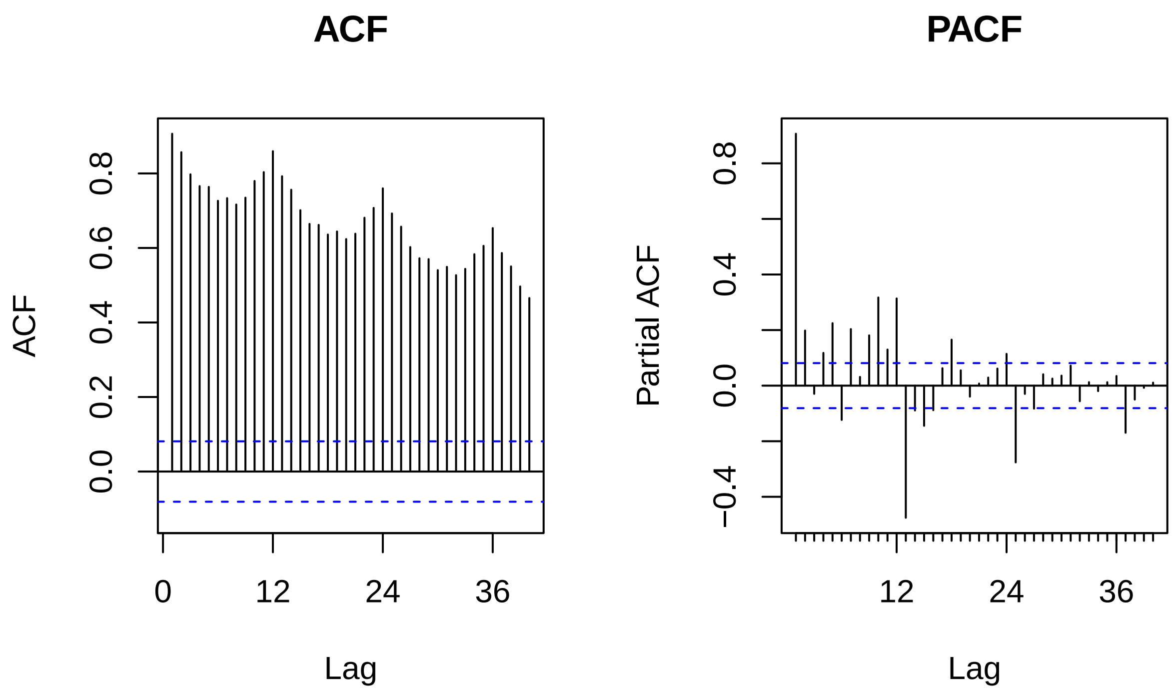
Total Renewable Energy Production:

For total renewable energy production, we can see that the ACF scalloping is more noticable for the detrended series. The PACF plot looks very similar from that in Q1.

```r
# Divide window into 1 row 3 columns
par(mfrow=c(1, 2))

Acf(detrended_ts[,2],lag.max=40,
    main="ACF")

Pacf(detrended_ts[,2],lag.max=40,
     main="PACF")
```



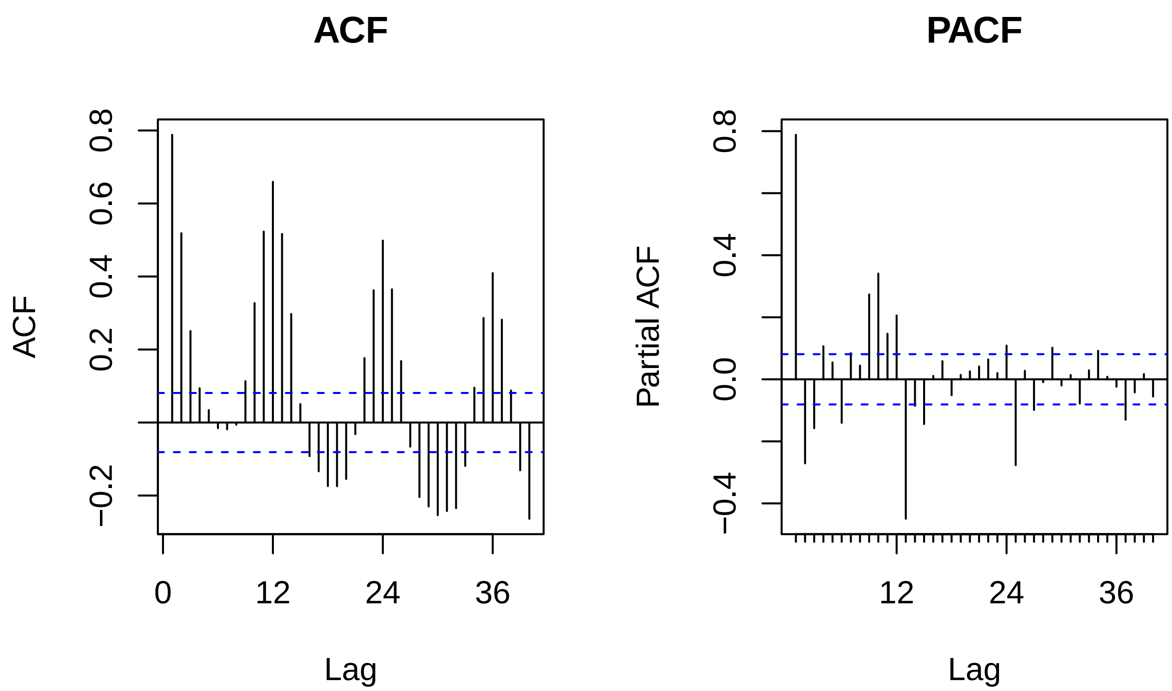
Hydroelectric Power Consumption:

For hydroelectric power, both the ACF and PACF plots look similar to those in Q1.

```r
# Divide window into 1 row 3 columns
par(mfrow=c(1, 2))

Acf(detrended_ts[,3],lag.max=40,
    main="ACF")

Pacf(detrended_ts[,3],lag.max=40,
     main="PACF")
```



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 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.

Total Biomass Energy:

We can see from the below seasonal means model that none of the regression coefficients are significant. This suggests that there is no seasonal trend for the Total Biomass Energy series. In terms of an interpretation, we can interpret the coefficient for the dummiesJan as follows: We expect that the total biomass energy for January is 1.498 lower than that in December (but this is not signficant). We can interpret the rest of the dummies in a similar way.

We can interpret the coefficient on the intercept as follows: At the baseline month of December, we expect the total biomass energy to be 284.241, and this coefficient is signficant.

```
#Use seasonal means model
#First create the seasonal dummies
dummies <- seasonaldummy(repc[,1])

#Then fit a linear model to the seasonal dummies
be_seasonal = lm(repc[,1]~dummies)
summary(be_seasonal)
```

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

```
## lm(formula = repc[, 1] ~ 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
# save coefficients
be_coefs <- coef(be_seasonal)
```

Total Renewable Energy:

We can see from the below seasonal means model that none of the regression coefficients are significant. This suggests that there is no seasonal trend for the Total Renewable Energy series. We can interpret the coefficient on the dummiesJan as follows: We expect the total renewable energy in January to be 11.793 higher than that in December (but this is not significant).

We can interpret the intercept as follows: At the baseline month of December, we expect the average total renewable energy to be 589.971, and this coefficient is significant.

```r
#Use seasonal means model
#First create the seasonal dummies
dummies <- seasonaldummy(repc[,2])

#Then fit a linear model to the seasonal dummies
re_seasonal = lm(repc[,2]~dummies)
summary(re_seasonal)
```

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

```
# save coefficients
re_coefs <- coef(re_seasonal)
```

Hydroelectric Power:

We can see from the below seasonal means model that many of the regression coefficients are significant. This suggests that there is a seasonal trend for the Hydroelectric power consumption series. We can interpret the January dummies coefficient as follows: We expect the average hydroelectric power consumption for January to be 13.558 higher than that in December (the baseline). We can interpret the rest of the coefficients in the same way. We can interpret the intercept as follows: We expect the average hydroelectric power consumption in December to be 237.841. The coefficient on the intercept is also significant.

```
#Use seasonal means model
#First create the seasonal dummies
dummies <- seasonaldummy(repc[,3])

#Then fit a linear model to the seasonal dummies
hp_seasonal = lm(repc[,3]~dummies)
summary(hp_seasonal)
```

```
##
## Call:
## lm(formula = repc[, 3] ~ 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
```

```r
# save coefficients
hp_coefs <- coef(hp_seasonal)
```

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?

```r
nobs <- nrow(repc)
```

Total Biomass Energy:

We can see that there is not much difference here between the deseaoned series (green) and the original series from Q1 (blue), except that the mean for the deseasoned series is near 0. This makes sense because in the previous question we found that the total biomass energy series did not have a seasonal trend.

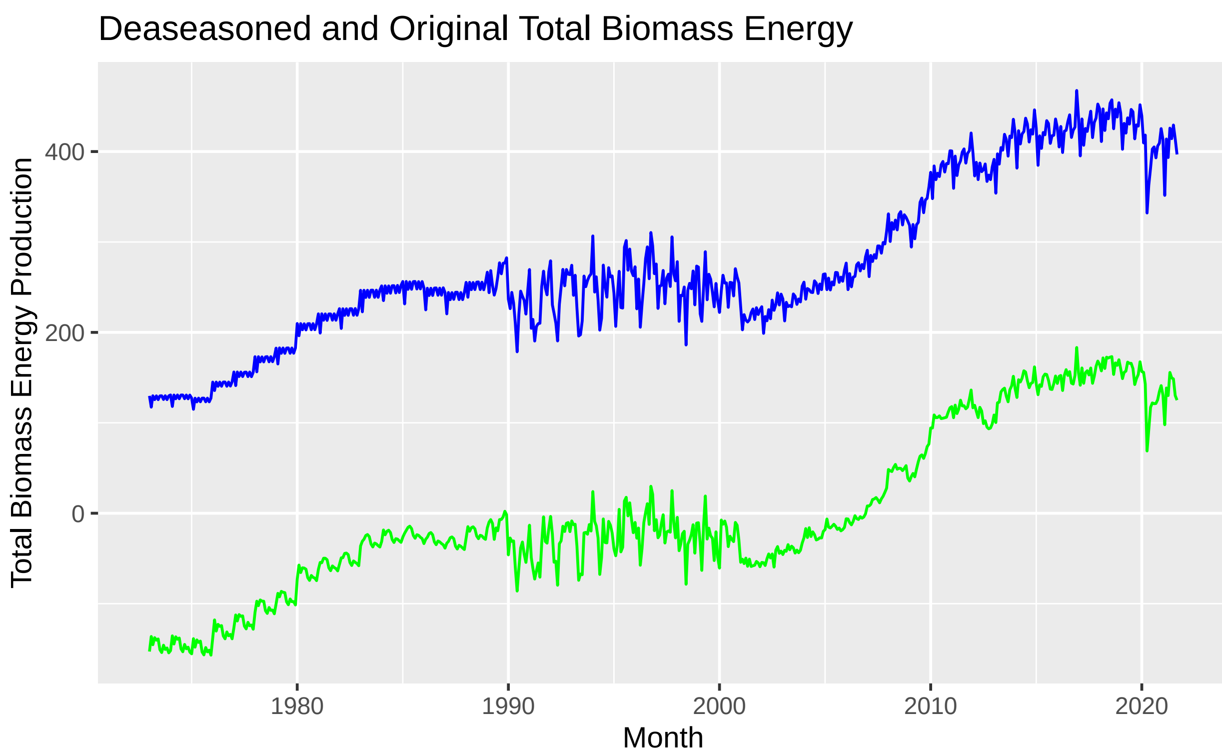
```r
dummies <- seasonaldummy(repc[,1])

#Store regression coefficients
beta_int=be_coefs[1]
beta_coeff=be_coefs[2:12]

#compute seasonal component
be_comp=array(0,nobs)
for(i in 1:nobs){
  be_comp[i]=(beta_int+beta_coeff%*%dummies[i,])
}

#Removing seasonal component
be_deseason <- repc[,1]-be_comp

#Understanding what we did
ggplot(repc_raw, aes(x=Month, y= `Total Biomass Energy Production`)) +
            geom_line(color="blue") +
            geom_line(aes(y=be_deseason), col="green") +
  labs(title = "Deaseasoned and Original Total Biomass Energy")
```



Total Renewable Energy

We can see that there is not much difference here between the deaseaoned series (green) and the original series from Q1 (blue), except that the mean for the deseasoned series is near 0. This makes sense because in the previous question we found that the total renewable energy series did not have a seasonal trend.

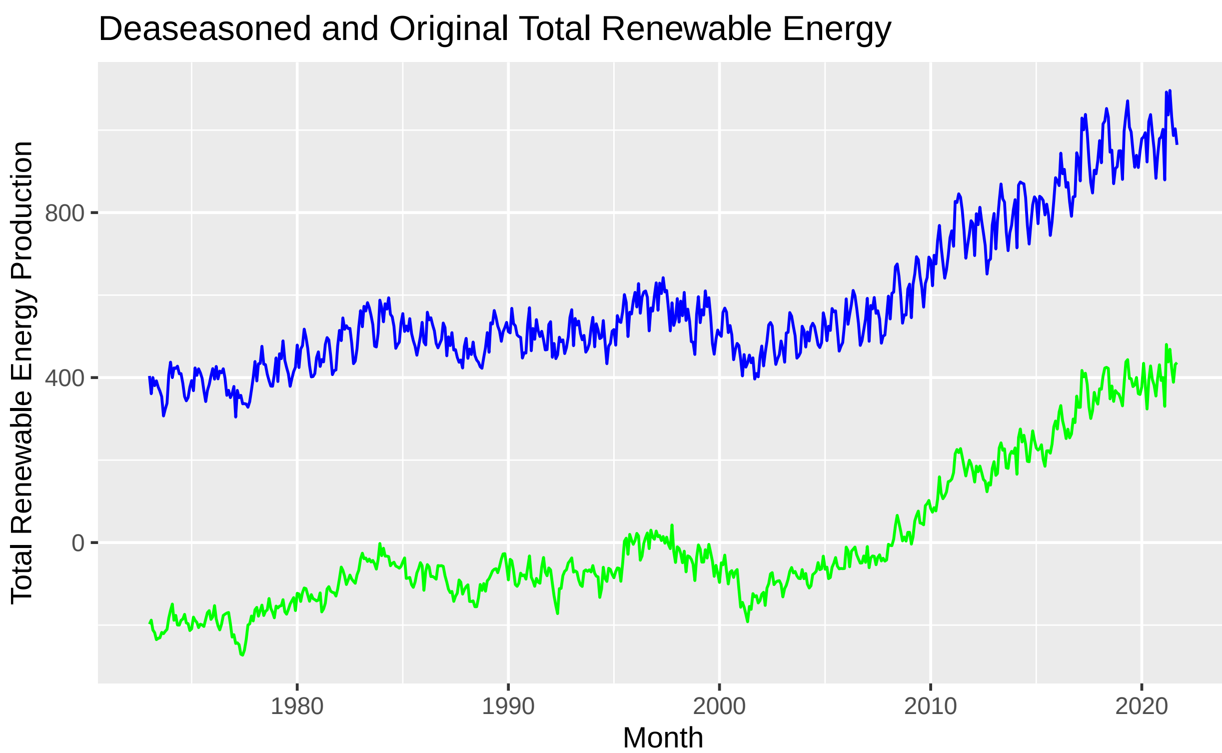
```
dummies <- seasonaldummy(repc[,2])

#Store regression coefficients
beta_int=re_coefs[1]
beta_coeff=re_coefs[2:12]

#compute seasonal component
re_comp=array(0,nobs)
for(i in 1:nobs){
  re_comp[i]=(beta_int+beta_coeff%*%dummies[i,])
}

#Removing seasonal component
re_deseason <- repc[,2]-re_comp

#Understanding what we did
ggplot(repc_raw, aes(x=Month, y= `Total Renewable Energy Production`)) +
            geom_line(color="blue") +
            geom_line(aes(y=re_deseason), col="green") +
  labs(title = "Deaseasoned and Original Total Renewable Energy")
```



Hydroelection Power Consumption:

We can see that there is some difference between the deaseaoned series (green) and the original series from Q1 (blue), and the mean for the deseasoned series is near 0. Specifically, there is less variation in the series within each year. This makes sense because in the previous question we found that the hydroelectric power consumption series did not have a seasonal trend.

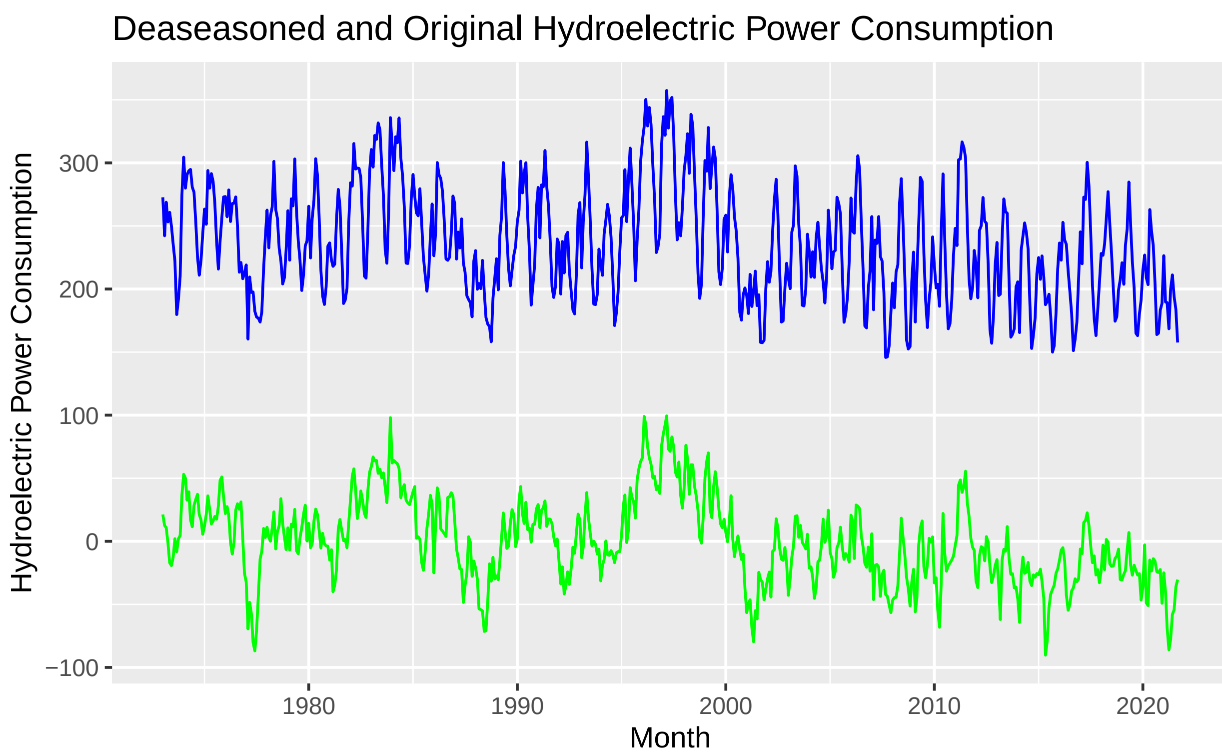
```
dummies <- seasonaldummy(repc[,3])

#Store regression coefficients
beta_int=hp_coefs[1]
beta_coeff=hp_coefs[2:12]

#compute seasonal component
hp_comp=array(0,nobs)
for(i in 1:nobs){
  hp_comp[i]=(beta_int+beta_coeff%*%dummies[i,])
}

#Removing seasonal component
hp_deseason <- repc[,3]-hp_comp

#Understanding what we did
ggplot(repc_raw, aes(x=Month, y= `Hydroelectric Power Consumption`)) +
            geom_line(color="blue") +
            geom_line(aes(y=hp_deseason), col="green") +
  labs(title = "Deaseasoned and Original Hydroelectric Power Consumption")
```



Q8

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

Total Biomass Energy Production:

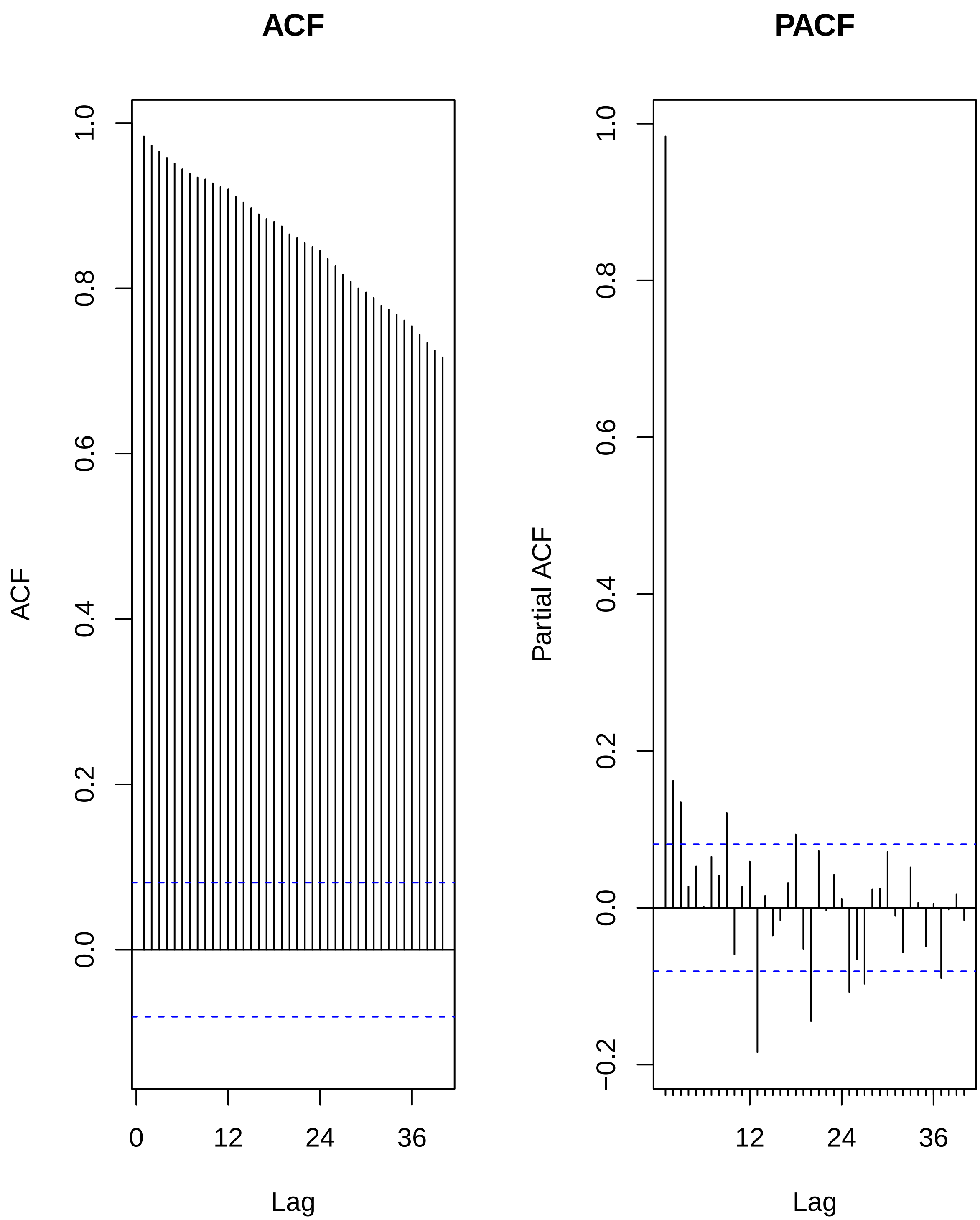
We can see that the ACF plot changed a small amount. Specifically, the very minimal scalloping pattern we saw in the ACF for Q1 has been completely erased in the below ACF plot. Additionally, for the PACF, we can see that the values after the first lag are less extreme in the below plot compared to the PACF from Q1.

```
# Divide window into 1 row 3 columns
par(mfrow=c(1, 2))

deseason_ts <- ts(cbind(be_deseason, re_deseason, hp_deseason),
                  frequency = 12,
                  start = c(1973, 1))

Acf(deseason_ts[,1],lag.max=40,
    main="ACF")

Pacf(deseason_ts[,1],lag.max=40,
     main="PACF")
```



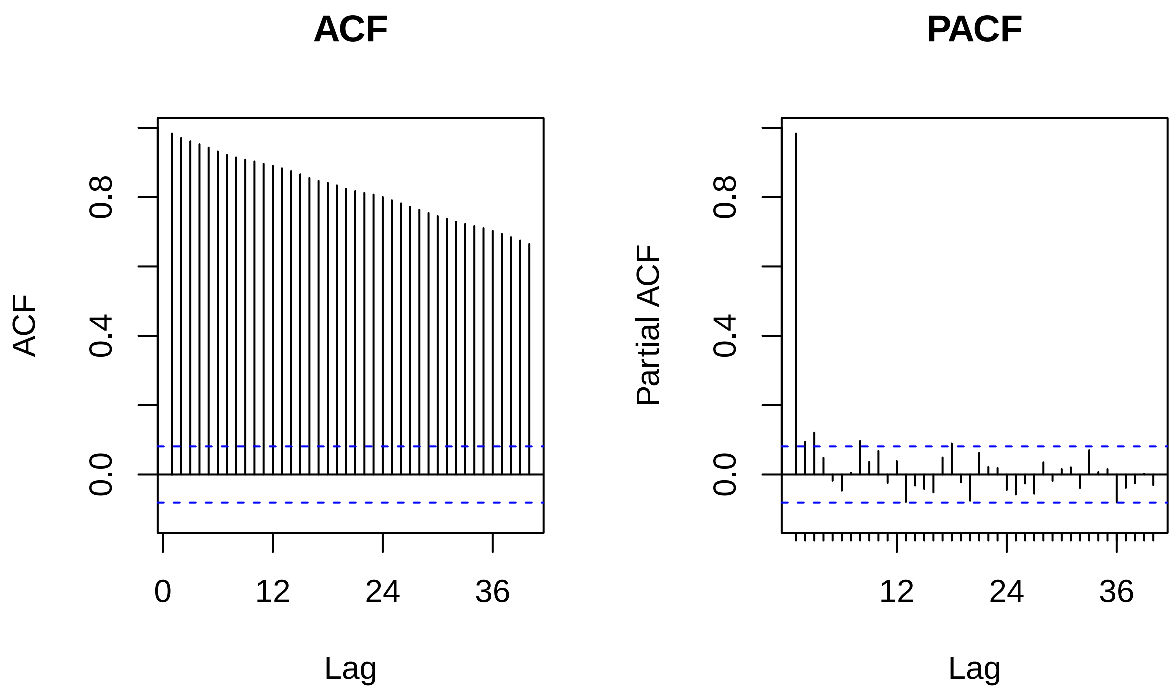
Total Renewable Energy Production:

We can see that the ACF plot changed from Q1. Specifically, the scalloping pattern we saw in the ACF for Q1 has been erased in the below ACF plot. Additionally, for the PACF, we can see that the values after the first lag are less extreme in the below plot compared to the PACF from Q1.

```r
# Divide window into 1 row 3 columns
par(mfrow=c(1, 2))

Acf(deseason_ts[,2],lag.max=40,
    main="ACF")

Pacf(deseason_ts[,2],lag.max=40,
     main="PACF")
```



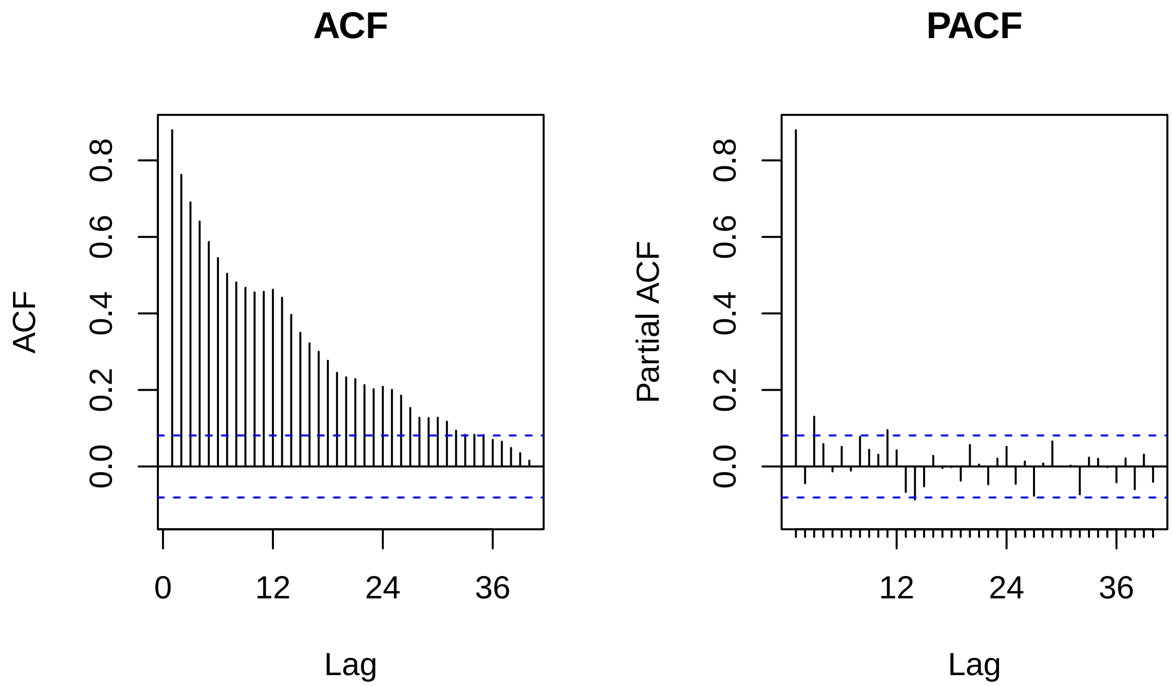
Hydroelectric Power Consumption:

For hydroelectric power, the ACF plot looks completely different below compared to the ACF plot in Q1. The positive and negative pattern we saw in Q1 has changed to a downward trend in the ACF below, with all positive values. The PACF plots actually look very similar.

```r
# Divide window into 1 row 3 columns
par(mfrow=c(1, 2))

Acf(deseason_ts[,3],lag.max=40,
    main="ACF")

Pacf(deseason_ts[,3],lag.max=40,
     main="PACF")
```


ACF
PACF
ACF
Partial ACF
0.0
0.2
0.4
0.6
0.8
0
12
24
36
Lag
Lag