ENV 790.30 - Time Series Analysis for Energy Data | Spring 2025

Assignment 3 - Due date 02/04/25

Rachael Stephan

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_A03_Sp25.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_Consumpt The data comes from the US Energy Information and Administration and corresponds to the December 2024 **Monthly** Energy Review. Once again you will work only with the following columns: Total Renewable Energy Production and Hydroelectric Power Consumption. Create a data frame structure with these two 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.

```r
#Load/install required package here
library(forecast); library(tseries); library(Kendall); library(cowplot)
library(lubridate); library(tidyverse); library(openxlsx); library(knitr)

#set theme
mytheme <- theme_bw(base_size = 10)+
  theme(axis.title = element_text(size = 10, hjust = 0.5),
        plot.title.position = "panel",
        panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.25),
        plot.caption = element_text(hjust = 0),
        legend.box = "vertical",
        legend.location = "plot",
        axis.gridlines = element_line(color = "grey", linewidth = 0.25),
        axis.ticks = element_line(color = "black", linewidth = 0.5))
theme_set(mytheme)

#upload dataset
renewable_e_prod_consump <-
  read.xlsx("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
                                    sheet = "Monthly Data",
                                    startRow = 13,
                                    colNames = FALSE)

#get column names
col_units <-
  read.xlsx("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
            rows = 11:12,
            sheet="Monthly Data",
            colNames=FALSE)

#set col names
colnames(renewable_e_prod_consump) <- col_units[1,]

#fix dates
renewable_e_prod_consump$Month <- as_date(renewable_e_prod_consump$Month, origin = "1900-01-01")
renewable_e_prod_consump$Month <- paste(month(renewable_e_prod_consump$Month,
                                              label = TRUE,
                                              abbr = TRUE),
                                        year(renewable_e_prod_consump$Month))

#select for columns of interest
energy_matrix <- renewable_e_prod_consump %>%
  select(`Month`,
         `Total Renewable Energy Production`,
         `Hydroelectric Power Consumption`)

#get first few rows of each column to check structure and values
kable(head(energy_matrix),
      caption = "First few rows of the selected timeseries for analysis")
```

Table 1: First few rows of the selected timeseries for analysis

Month	Total Renewable Energy Production	Hydroelectric Power Consumption
Jan 1973	219.839	89.562
Feb 1973	197.330	79.544
Mar 1973	218.686	88.284
Apr 1973	209.330	83.152
May 1973	215.982	85.643
Jun 1973	208.249	82.060

```
str(energy_matrix)
```

```
## 'data.frame':    621 obs. of  3 variables:
##  $ Month                             : chr  "Jan 1973" "Feb 1973" "Mar 1973" "Apr 1973" ...
##  $ Total Renewable Energy Production: num  220 197 219 209 216 ...
##  $ Hydroelectric Power Consumption  : num  89.6 79.5 88.3 83.2 85.6 ...
```

```
#create time series object
energy_ts <- ts(energy_matrix[,2:3],
                start=c(1973,1),
                frequency=12)
```

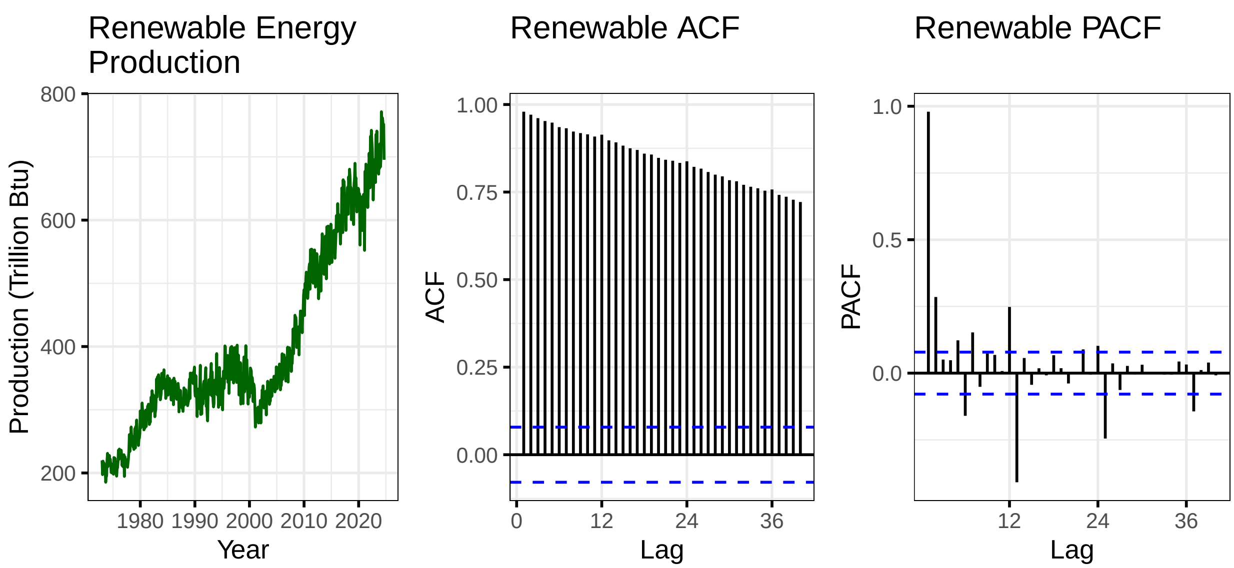
Trend Component

Q1

For each time series, i.e., Renewable Energy Production and Hydroelectric Consumption create three plots: one with time series, one with the ACF and with the PACF. You may use the some code form A2, but I want all the three plots side by side as in a grid. (Hint: use function `plot_grid()` *from the* `cowplot` *package)*

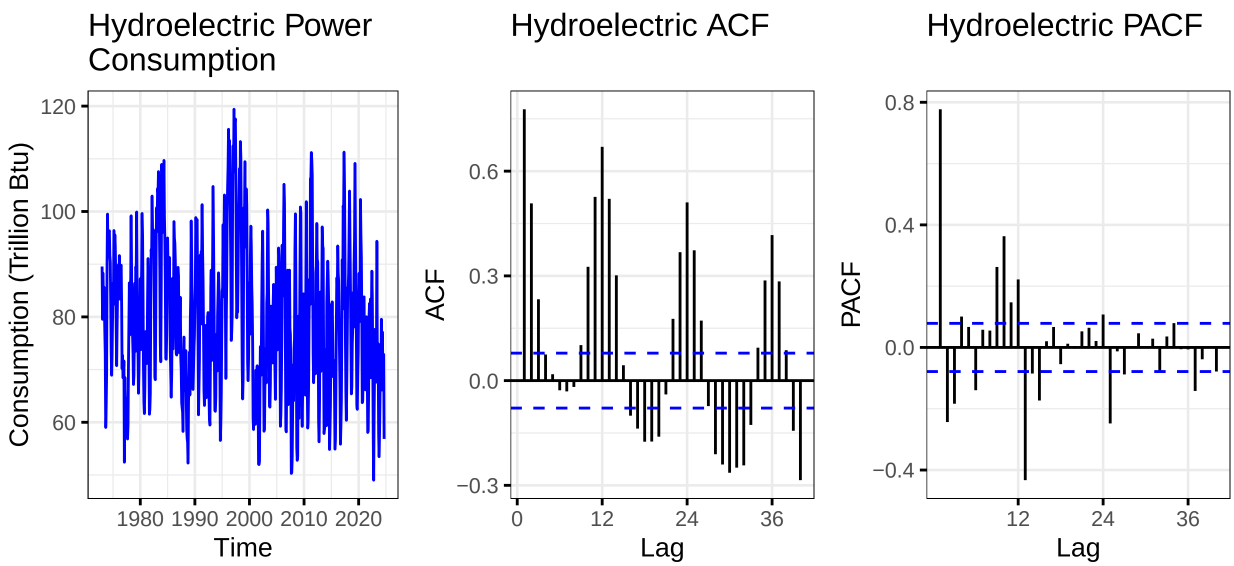
```
#create plots for renewable ts
#time series
energy_ts_plot_renewable <- autoplot(energy_ts[,1], col ="darkgreen")+
  labs(y = paste("Production",col_units[2,5], sep = " "),
       title = "Renewable Energy\nProduction",
       x = "Year")

#ACF
energy_acf_renewable <- Acf(energy_ts[,1],
                                  lag.max=40,
                                  type="correlation",
                                  plot=FALSE)

energy_acf_plot_renewable <- autoplot(energy_acf_renewable)+
  labs(title = "Renewable ACF")

#PACF
energy_pacf_renewable <- Pacf(energy_ts[,1],
                                  lag.max=40,
                                  plot=FALSE)
```

```r
energy_pacf_plot_renewable <- autoplot(energy_pacf_renewable)+
  labs(title = "Renewable PACF")

#plot the renewable grid
plot_grid(energy_ts_plot_renewable,
          energy_acf_plot_renewable,
          energy_pacf_plot_renewable,
          align = "h",
          nrow = 1)
```




```r
#plot renewable ts
#time series
energy_ts_plot_hydro <- autoplot(energy_ts[,2], col ="blue")+
  labs(y = paste("Consumption",col_units[2,6], sep = " "),
       title = "Hydroelectric Power\nConsumption")

#ACF
energy_acf_hydro <- Acf(energy_ts[,2],
                        lag.max=40,
                        type="correlation",
                        plot=FALSE)

energy_acf_plot_hydro <- autoplot(energy_acf_hydro)+
  labs(title = "Hydroelectric ACF")

#PACF
energy_pacf_hydro <- Pacf(energy_ts[,2],
                          lag.max=40,
                          plot=FALSE)

energy_pacf_plot_hydro <- autoplot(energy_pacf_hydro)+
  labs(title = "Hydroelectric PACF")

#plot the hydro plot grid
```

```r
plot_grid(energy_ts_plot_hydro,
          energy_acf_plot_hydro,
          energy_pacf_plot_hydro,
          align = "h",
          nrow = 1)
```



Q2

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

The renewable energy consumption autocorrelation plot begins with a high magnitude and positive autocorrelation value that decreases at a consistent rate with each lag. There is still a high magnitude and positive correlation at 40 lags.

The autocorrelation plot for the hydroelectric energy production oscillates between highly positive and highly negative correlations, with the positive magnitudes decreasing and the negative magnitudes increasing over the 40 lags.

Both of the partial autocorrelation plots have similar trends. The largest lags are observed at 12n+1 lags, with a general decay in amplitude as lags increase. The amount of values that exceed the blue line decreases with increasing lags. There is possibly a small oscillation between positive and negative PACF that may coincide with seasonality in the data.

Q3

Use the lm() function to fit a linear trend to the two 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 variable for number of temporal observations
time <- 1:nrow(energy_ts)

#Fit the linear trend for renewable energy
lm_renewable <- lm(energy_ts[,1] ~ time)
summary(lm_renewable)

#save renewable lm coefficients
lm_renewable_beta0 <- as.numeric(lm_renewable$coefficients[1])
lm_renewable_beta1 <- as.numeric(lm_renewable$coefficients[2])
```

```
##
## Call:
## lm(formula = energy_ts[, 1] ~ time)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -151.11  -37.84   13.53   41.76  149.42
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.87293    4.96189   35.65   <2e-16 ***
## time          0.72393    0.01382   52.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.75 on 619 degrees of freedom
## Multiple R-squared:  0.8159, Adjusted R-squared:  0.8156
## F-statistic:  2743 on 1 and 619 DF,  p-value: < 2.2e-16
```

The linear regression was found to be significant (p-value $<$ 2.2e-16), indicating that the linear model identified is significant. Within the model, both the intercept (p-value $<$ 2e-16) and slope (p-value $<$ 2e-16) of the linear regression was found to be significant. Therefore, there is a linear trend within the renewable energy production time series. The trend is positively linear according to the beta slope estimate.

```
#Fit the linear trend for hydro energy
lm_hydro <- lm(energy_ts[,2] ~ time)
summary(lm_hydro)

#save renewable lm coefficients
lm_hydro_beta0 <- as.numeric(lm_hydro$coefficients[1])
lm_hydro_beta1 <- as.numeric(lm_hydro$coefficients[2])
```

```
##
## Call:
## lm(formula = energy_ts[, 2] ~ time)
##
## Residuals:
```

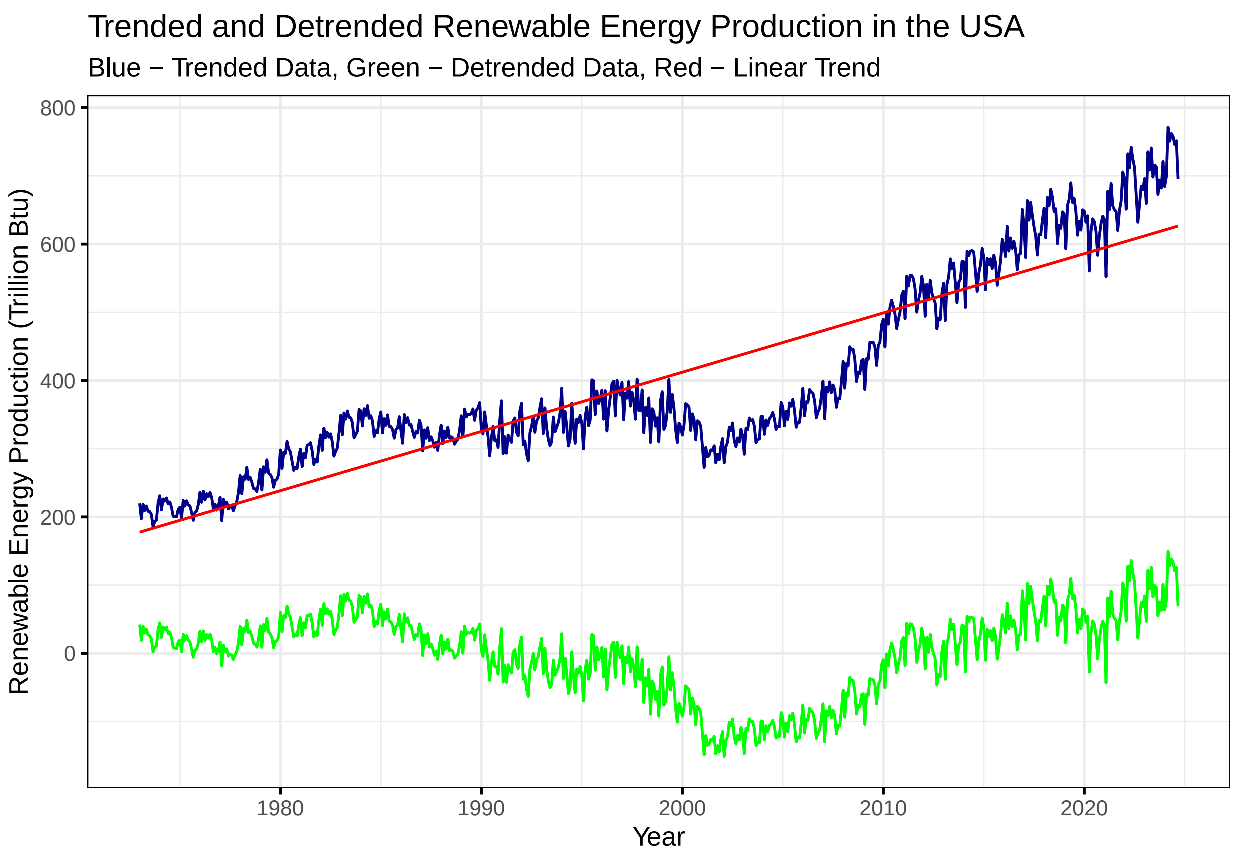
```
##     Min      1Q  Median      3Q     Max
## -29.995 -10.422  -0.720   9.161  39.624
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 82.96766    1.12339  73.855  < 2e-16 ***
## time        -0.01098    0.00313  -3.508 0.000485 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.98 on 619 degrees of freedom
## Multiple R-squared:  0.01949,    Adjusted R-squared:  0.01791
## F-statistic:  12.3 on 1 and 619 DF,  p-value: 0.0004848
```

The linear regression was found to be significant (p-value = 0.0004848), indicating that the linear model identified is significant. Within the model, both the intercept (p-value < 2e-16) and slope (p-value = 0.000485) of the linear regression was found to be significant. Therefore, there is a linear trend within the renewable energy production time series. The trend is negatively linear according to the beta slope estimate.

Q4

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

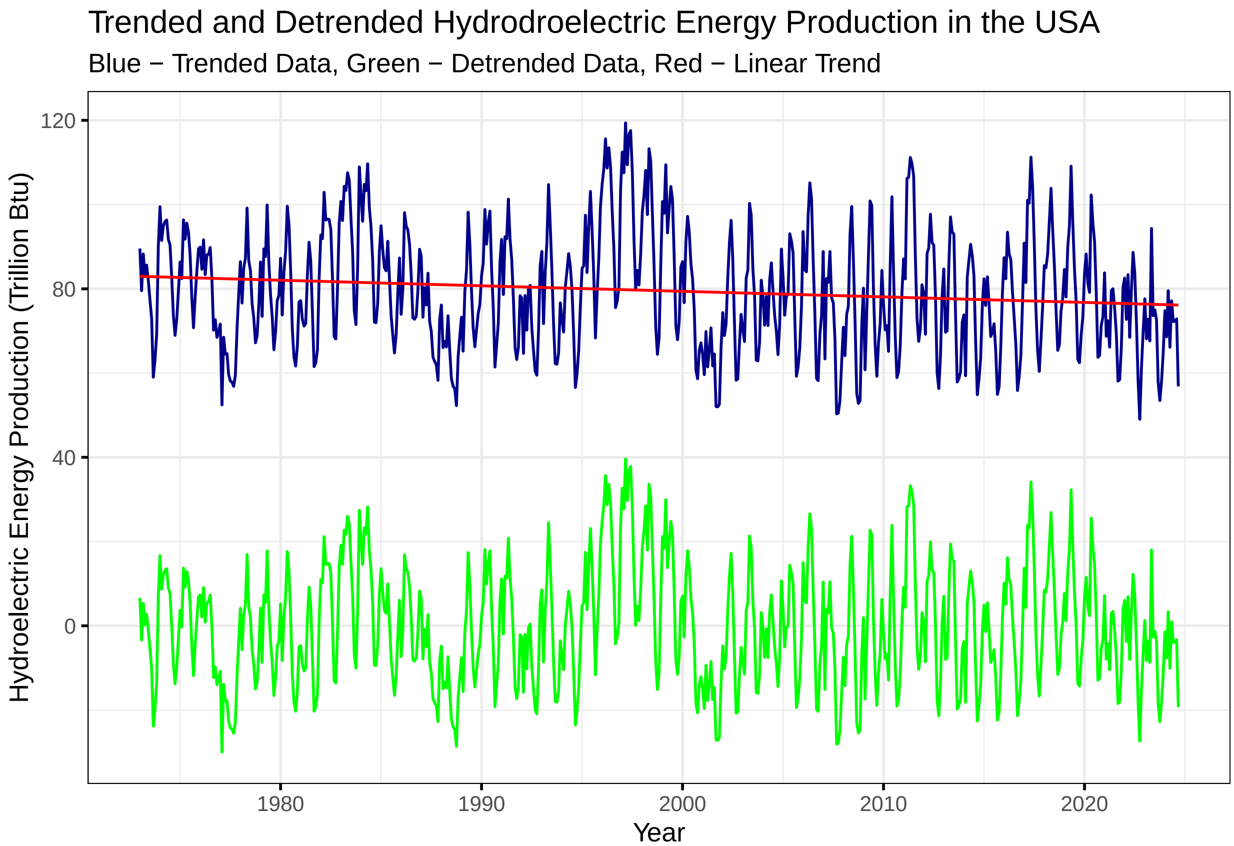
```r
#create renewable linear trend equation
lm_trend_renewable <- lm_renewable_beta0 + lm_renewable_beta1 * time

#use linear trend to detrend renewable ts
lm_detrend_renewable <- energy_ts[,1] - lm_trend_renewable


#create timeseries trend
ts_trend_renewable <- ts(lm_trend_renewable,
                         start=c(1973,1),
                         frequency=12)

ts_detrend_renewable <- ts(lm_detrend_renewable,
                           start = c(1973,1),
                           frequency = 12)

#plot ts and trends
autoplot(energy_ts[,1],color="darkblue")+
  autolayer(ts_detrend_renewable, series ="Detrended",color="green")+
  autolayer(ts_trend_renewable, series ="Linear Component",color="red")+
  labs(title = "Trended and Detrended Renewable Energy Production in the USA",
       y = paste("Renewable Energy Production",col_units[2,6], sep = " "),
       x= "Year",
       subtitle = "Blue - Trended Data, Green - Detrended Data, Red - Linear Trend")
```



There is a difference between the trended and detrended data sets. The detrended dataset(green) removes the linear trend (red) from the original dataset (blue). This removes the linear increase from the dataset. So the patterns of the peaks and valleys remain, but the upwards trend does not.

```
#create hydro linear trend equation
lm_trend_hydro <- lm_hydro_beta0 + lm_hydro_beta1 * time

#use linear trend to detrend hydro ts
lm_detrend_hydro <- energy_ts[,2] - lm_trend_hydro


#create timeseries trend
ts_trend_hydro <- ts(lm_trend_hydro,
                        start=c(1973,1),
                        frequency=12)

ts_detrend_hydro <- ts(lm_detrend_hydro,
                          start = c(1973,1),
                          frequency = 12)

#plot ts and trends
autoplot(energy_ts[,2],color="darkblue")+
  autolayer(ts_detrend_hydro, series ="Detrended",color="green")+
  autolayer(ts_trend_hydro, series ="Linear Component",color="red")+
  labs(title = "Trended and Detrended Hydrodroelectric Energy Production in the USA",
       y = paste("Hydroelectric Energy Production",col_units[2,6], sep = " "),
```

```
       x= "Year",
       subtitle = "Blue - Trended Data, Green - Detrended Data, Red - Linear Trend")
```



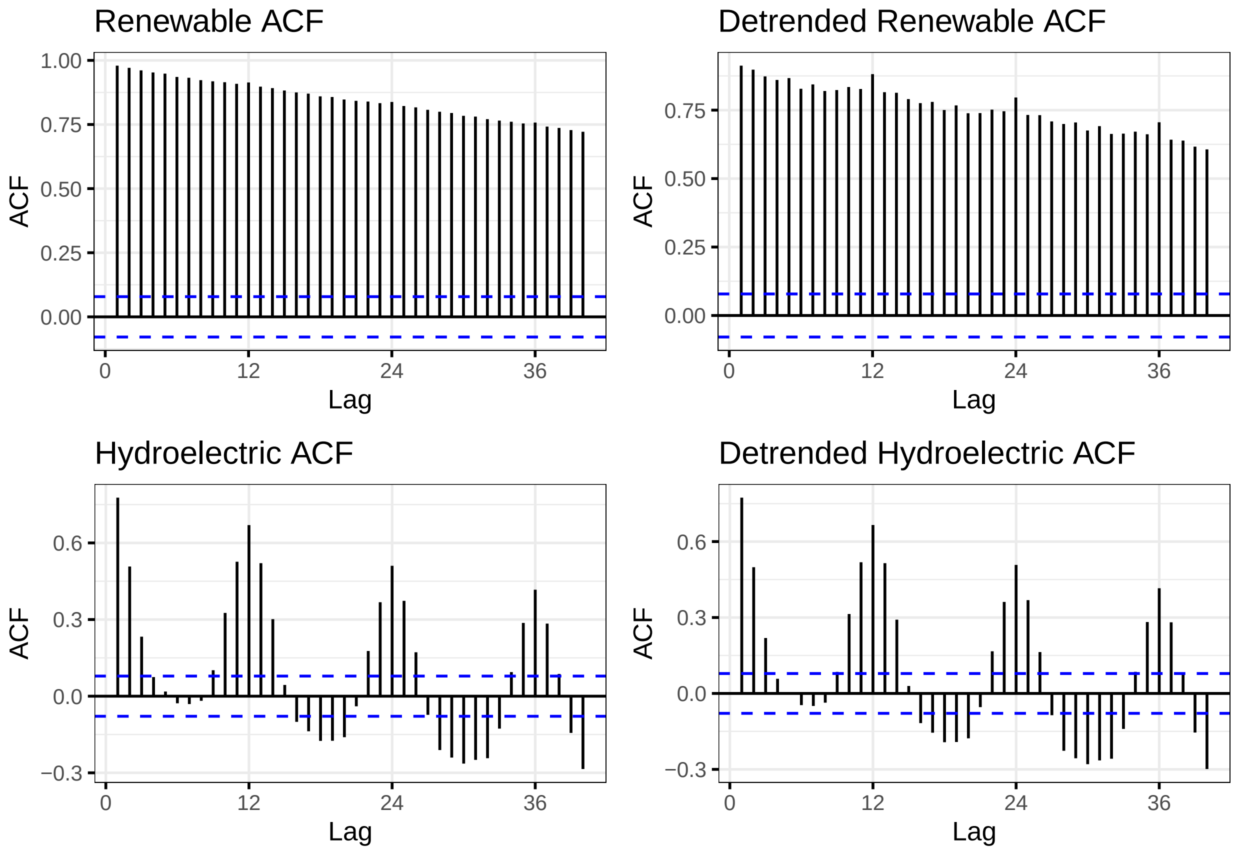
There is a less noticeable difference between the trended and detrended hydroelectric data sets. The detrended dataset(green) also removes the linear trend (red) from the original dataset (blue), eliminating the linear decrease from the dataset. Since the slope is smaller, this change is less prevalent.

Q5

Plot ACF and PACF for the detrended series and compare with the plots from Q1. You may use plot_grid() again to get them side by side, but not mandatory. Did the plots change? How?

```
#determine ACF, PACF for renewable energy
#ACF
acf_detrend_renewable <- Acf(ts_detrend_renewable,
                  lag.max=40,
                  type="correlation",
                  plot=FALSE)

acf_detrend_plot_renewable <- autoplot(acf_detrend_renewable)+
  labs(title = "Detrended Renewable ACF")

#PACF
pacf_detrend_renewable <- Pacf(ts_detrend_renewable,
                  lag.max=40,
```

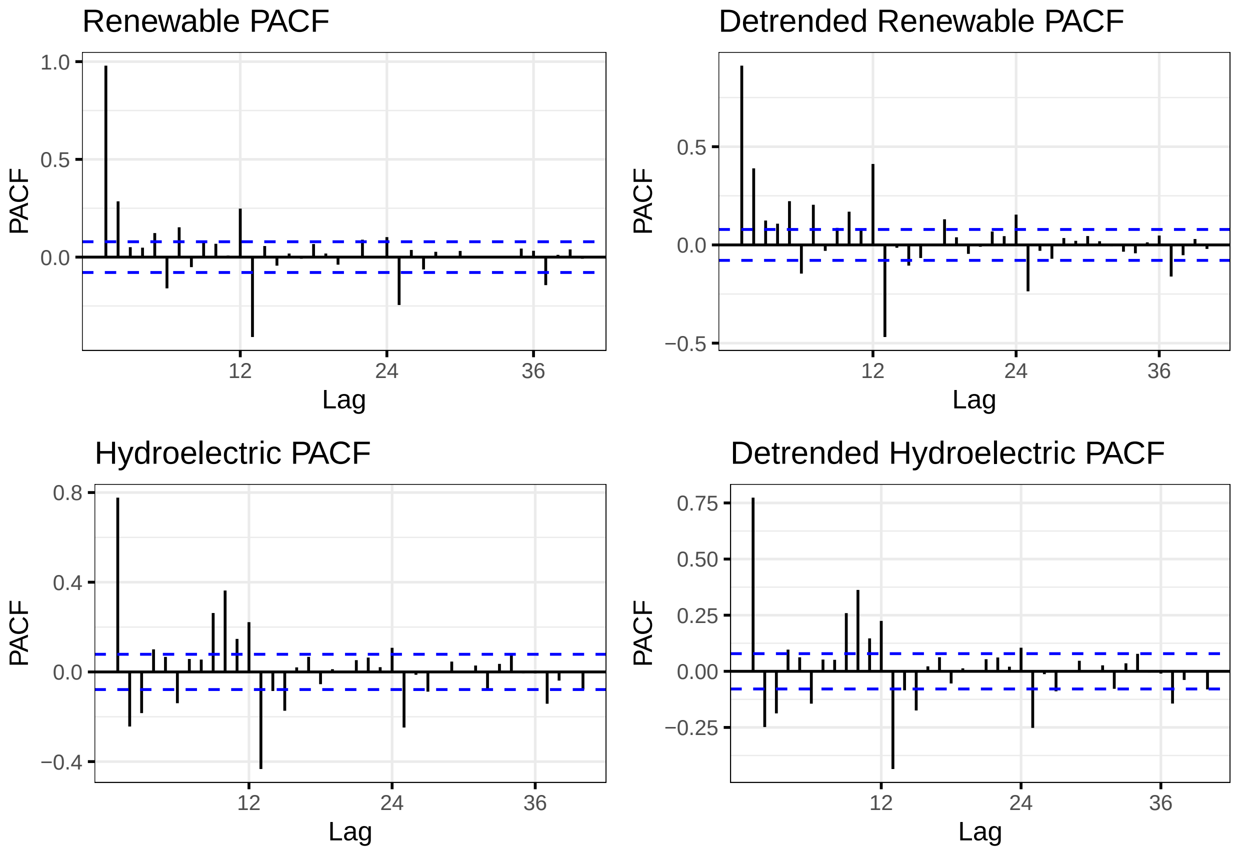
```r
                  plot=FALSE)

pacf_detrend_plot_renewable <- autoplot(pacf_detrend_renewable)+
  labs(title = "Detrended Renewable PACF")

#determine ACF, PACF for hydro energy
#ACF
acf_detrend_hydro <- Acf(ts_detrend_hydro,
                  lag.max=40,
                  type="correlation",
                  plot=FALSE)

acf_detrend_plot_hydro <- autoplot(acf_detrend_hydro)+
  labs(title = "Detrended Hydroelectric ACF")

#PACF
pacf_detrend_hydro <- Pacf(ts_detrend_hydro,
                  lag.max=40,
                  plot=FALSE)

pacf_detrend_plot_hydro <- autoplot(pacf_detrend_hydro)+
  labs(title = "Detrended Hydroelectric PACF")

#plot ACF comparisons in a plot
plot_grid(energy_acf_plot_renewable,
          acf_detrend_plot_renewable,
          energy_acf_plot_hydro,
          acf_detrend_plot_hydro,
          align = "h",
          nrow = 2)
```



The detrended hydroelectric ACF did not have a visible difference to the trended hydroelectric ACF. The values and the patterns were similar, and any change in the values was not uniform.

The renewable energy ACF did have a visible difference between the trended and detrended plots. The detrended data had slight peaks at lags at multiples of twelve with a (possibly curved) depression in between. The magnitudes were also slightly smaller. Otherwise, the consistent decrease in ACF remained. This indicates the possible presence of seasonality.

```
#Plot PACF comparisons in a grid
plot_grid(energy_pacf_plot_renewable,
          pacf_detrend_plot_renewable,
          energy_pacf_plot_hydro,
          pacf_detrend_plot_hydro,
          align = "h",
          nrow = 2)
```



The detrended hydroelectric PACF did not have a visible difference to the trended hydroelectric PACF. The values and the patterns were similar, and any change in the values was not uniform.

The renewable energy PACF did have a visible difference between the trended and detrended plots. The detrended data generally had larger magnitudes and more significant correlations in the PACF plots. However, the general trend of positive/negative and small/large magnitudes in the PACF values remained the same.

Seasonal Component

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

Q6

Just by looking at the time series and the acf plots, do the series seem to have a seasonal trend? No need to run any code to answer your question. Just type in you answer below.

The hydroelectric timeseries plot and ACF plot both strongly suggest a seasonal component. Both of these plots have visually evident oscillations in there values that occur over 12 lags or time points (i.e., a year).

There also appears to be slight seasonal trends in certain sections of the renewable energy ts, but they are much more difficult to identify. There appears to be a general oscillation around the trend each year, but there are also more prominent deviations to the seasonal component than in the hydroelectric ts.

Q7

Use function lm() to fit a seasonal means model (i.e. using the seasonal dummies) the two time series. Ask R to print the summary of the regression. Interpret the regression output. From the results which series have a seasonal trend? Do the results match you answer to Q6?

```r
#Use seasonal means model on the trended renewable dataset
ts_dummies_renewable <- seasonaldummy(energy_ts[,1])

lm_seasonal_renewable <- lm(energy_ts[,1] ~ ts_dummies_renewable)
summary(lm_seasonal_renewable)
```

```
##
## Call:
## lm(formula = energy_ts[, 1] ~ ts_dummies_renewable)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -205.65  -91.59  -54.59  117.87  356.19
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              410.598     20.228  20.299   <2e-16 ***
## ts_dummies_renewableJan    1.973     28.468   0.069    0.945
## ts_dummies_renewableFeb  -34.348     28.468  -1.207    0.228
## ts_dummies_renewableMar    4.721     28.468   0.166    0.868
## ts_dummies_renewableApr   -7.896     28.468  -0.277    0.782
## ts_dummies_renewableMay    7.199     28.468   0.253    0.800
## ts_dummies_renewableJun   -3.394     28.468  -0.119    0.905
## ts_dummies_renewableJul    2.850     28.468   0.100    0.920
## ts_dummies_renewableAug   -4.430     28.468  -0.156    0.876
## ts_dummies_renewableSep  -29.037     28.468  -1.020    0.308
## ts_dummies_renewableOct  -20.205     28.606  -0.706    0.480
## ts_dummies_renewableNov  -20.706     28.606  -0.724    0.469
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 144.5 on 609 degrees of freedom
## Multiple R-squared:  0.008696,   Adjusted R-squared:  -0.009209
## F-statistic: 0.4857 on 11 and 609 DF,  p-value: 0.9126
```

```r
lm_seasonal_beta_int_renewable <-lm_seasonal_renewable$coefficients[1]
lm_seasonal_beta_coeff_renewable <-lm_seasonal_renewable$coefficients[2:12]
```

The intercept of the seasonal lm model was significant (p value $<$ 2e-16), but none of the other coefficients or the model were significant. This indicates that there was no monthly seasonal trend identified in the renewable energy production. The intercept captures the seasonal component of December and the y-intercept of the ts. This could influence the calculated significance of this value. If there is a seasonal variation, it may not be captured in 12 bins but in fewer. This analysis could also be impacted by the presence of the trend that is not accounted for in this model.

```r
#Use seasonal means model on the trended hydro dataset
ts_dummies_hydro <- seasonaldummy(energy_ts[,2])

lm_seasonal_hydro <- lm(energy_ts[,2] ~ ts_dummies_hydro)
summary(lm_seasonal_hydro)
```

```
##
## Call:
## lm(formula = energy_ts[, 2] ~ ts_dummies_hydro)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.101  -6.241  -0.444   6.410  32.363
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           79.981      1.452  55.094  < 2e-16 ***
## ts_dummies_hydroJan    4.941      2.043   2.418 0.015880 *
## ts_dummies_hydroFeb   -2.513      2.043  -1.230 0.219219
## ts_dummies_hydroMar    7.053      2.043   3.452 0.000595 ***
## ts_dummies_hydroApr    5.399      2.043   2.642 0.008441 **
## ts_dummies_hydroMay   13.907      2.043   6.807 2.40e-11 ***
## ts_dummies_hydroJun   10.729      2.043   5.251 2.09e-07 ***
## ts_dummies_hydroJul    4.033      2.043   1.974 0.048845 *
## ts_dummies_hydroAug   -5.399      2.043  -2.643 0.008436 **
## ts_dummies_hydroSep  -16.596      2.043  -8.123 2.54e-15 ***
## ts_dummies_hydroOct  -16.370      2.053  -7.973 7.66e-15 ***
## ts_dummies_hydroNov  -10.805      2.053  -5.263 1.97e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.37 on 609 degrees of freedom
## Multiple R-squared:  0.4695, Adjusted R-squared:  0.4599
## F-statistic:    49 on 11 and 609 DF,  p-value: < 2.2e-16
```

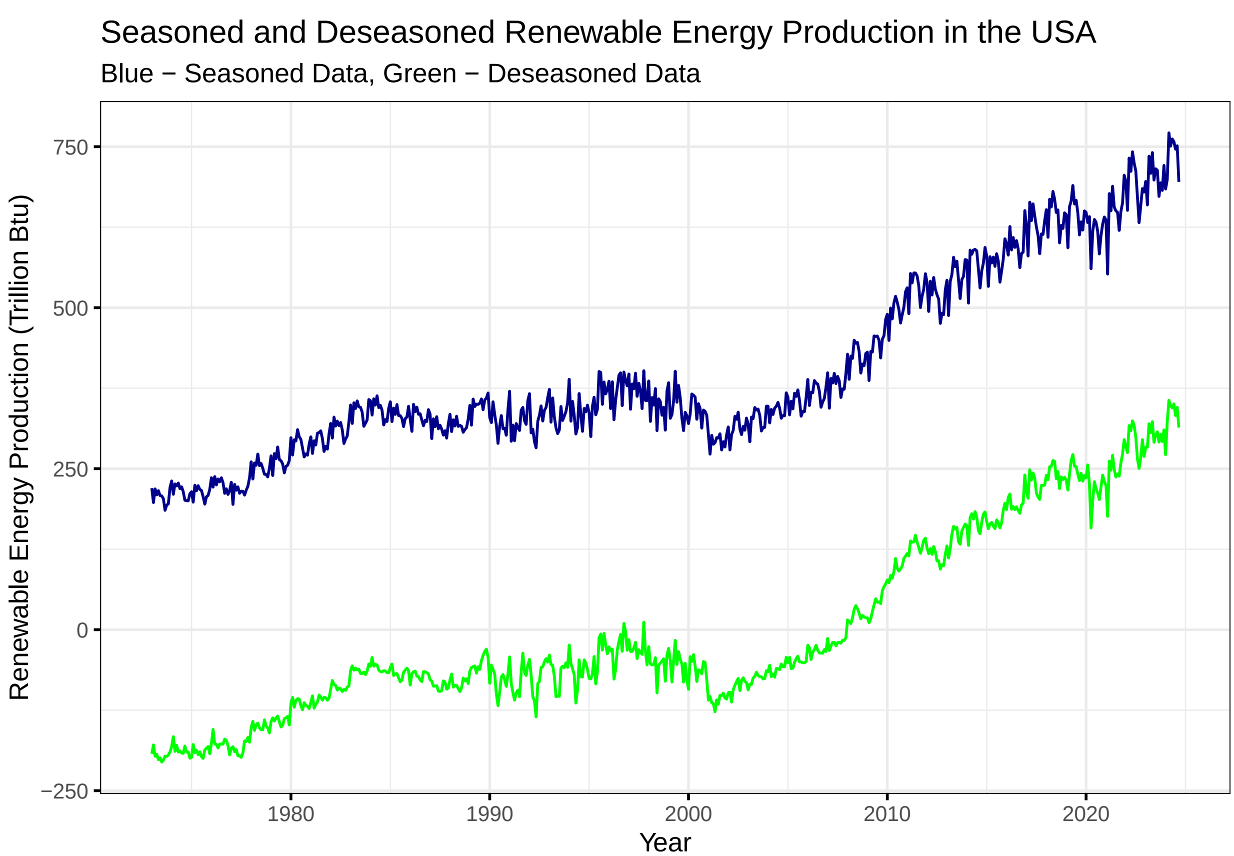
```r
lm_seasonal_beta_int_hydro <-lm_seasonal_hydro$coefficients[1]
lm_seasonal_beta_coeff_hydro <-lm_seasonal_hydro$coefficients[2:12]
```

Both the hydroelectric model and all of the coefficients, except for February, were calculated to be significant (most p-values < 0.01). This indicates that there is a seasonal component to the hydroelectric timeseries that is monthly. The values of each season indicate how much more or less energy is produced in each month compared to the intercept (December). So, the most energy is produced in may and the least amount is produced in September.

Q8

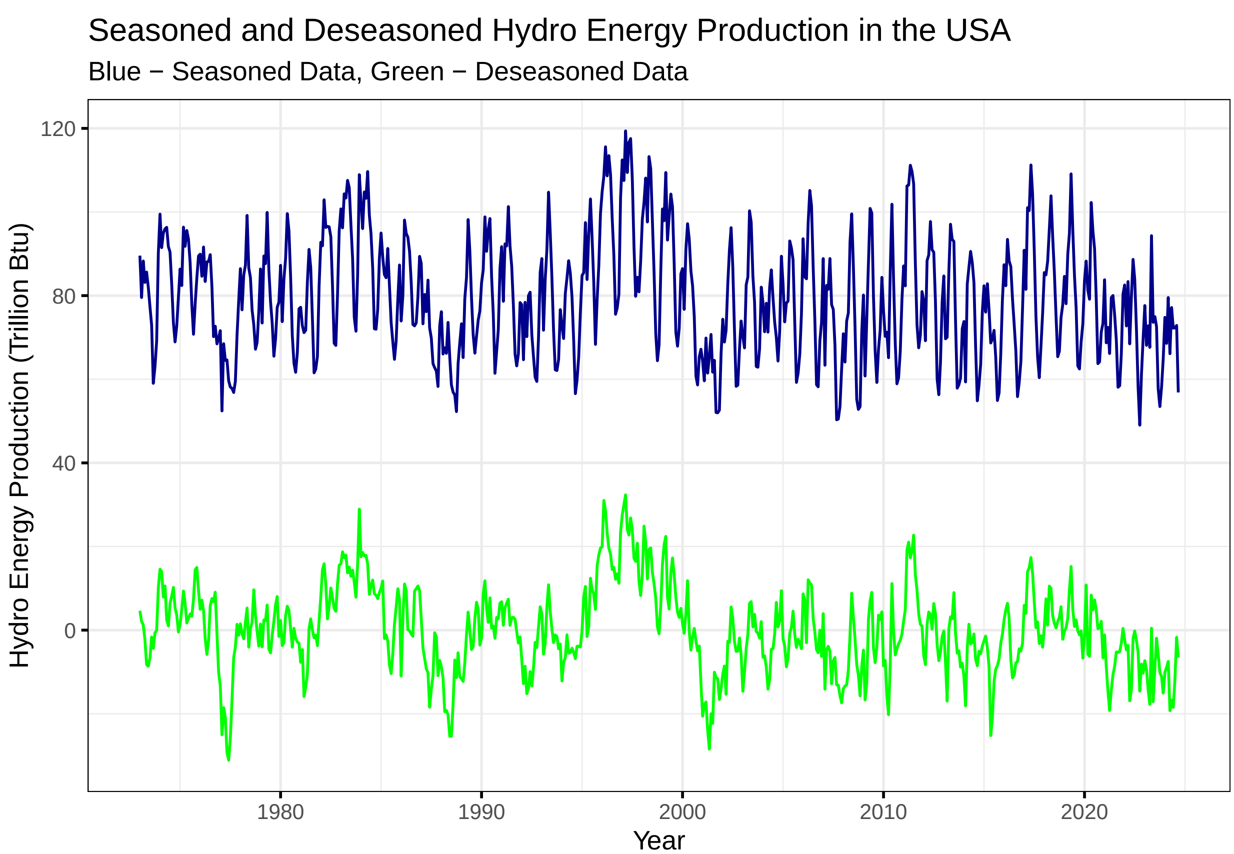
Use the regression coefficients from Q7 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

```
seasonal_comp_renewable <- array(0, nrow(energy_ts))


for(i in time){
  seasonal_comp_renewable[i] <- lm_seasonal_beta_int_renewable +
    lm_seasonal_beta_coeff_renewable %*% ts_dummies_renewable[i,]
}

lm_deseasoned_renewable <- energy_ts[,1] - seasonal_comp_renewable

ts_deseason_renewable <- ts(lm_deseasoned_renewable,
                              start=c(1973,1),
                              frequency = 12)

#plot ts and deseasoned data
autoplot(energy_ts[,1],color="darkblue")+
  autolayer(ts_deseason_renewable, series ="Deseasoned",color="green")+
  labs(title = "Seasoned and Deseasoned Renewable Energy Production in the USA",
       y = paste("Renewable Energy Production",col_units[2,6], sep = " "),
       x= "Year",
       subtitle = "Blue - Seasoned Data, Green - Deseasoned Data")
```



There are some changes to the deseasoned data set. Some of the oscillations around the trend have been flattened, in accordance with attempting to remove the sesaonality. The time series has also been decreased by the value of the intercept.

```r
seasonal_comp_hydro <- array(0, nrow(energy_ts))


for(i in time){
  seasonal_comp_hydro[i] <- lm_seasonal_beta_int_hydro +
    lm_seasonal_beta_coeff_hydro %*% ts_dummies_hydro[i,]
}

lm_deseasoned_hydro <- energy_ts[,2] - seasonal_comp_hydro

ts_deseason_hydro <- ts(lm_deseasoned_hydro,
                              start=c(1973,1),
                              frequency = 12)

#plot ts and deseasoned data
autoplot(energy_ts[,2],color="darkblue")+
  autolayer(ts_deseason_hydro, series ="Deseasoned",color="green")+
  labs(title = "Seasoned and Deseasoned Hydro Energy Production in the USA",
       y = paste("Hydro Energy Production",col_units[2,6], sep = " "),
       x= "Year",
       subtitle = "Blue - Seasoned Data, Green - Deseasoned Data")
```

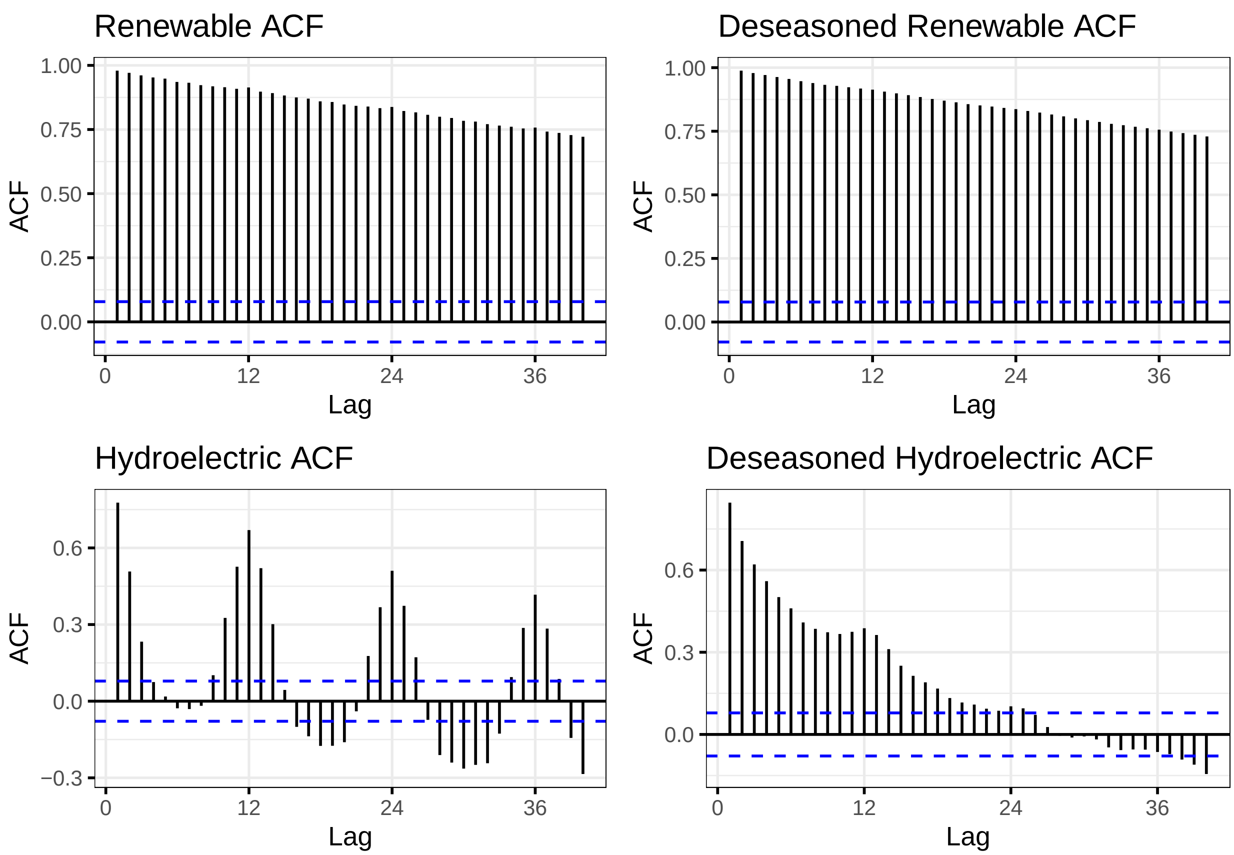


The hydroelectric deseasoned ts has changed than the renewable energy ts. There is less variation between adjacent months due to the removal of the seasonality. However, the increases and decreases that span over longer periods of time (i.e., years) remain present.

Q9

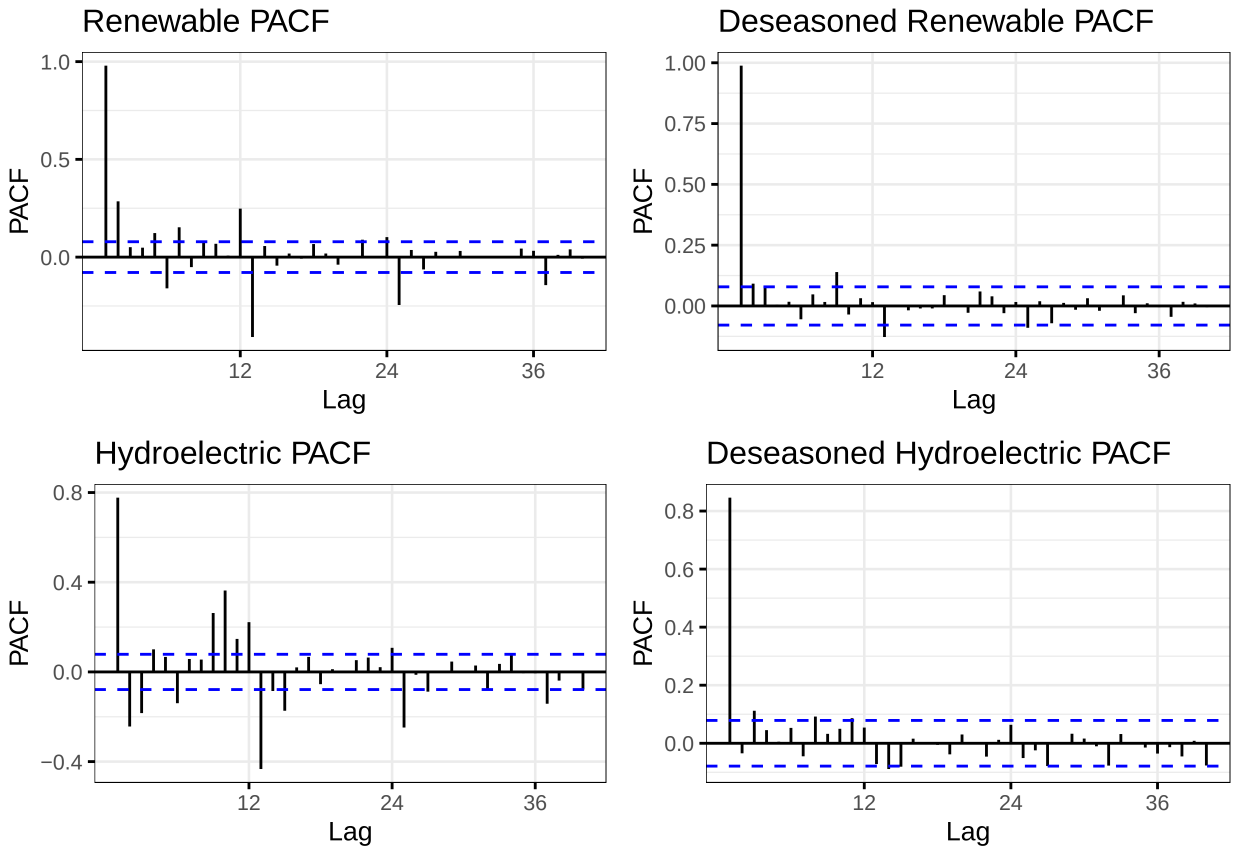
Plot ACF and PACF for the deseason series and compare with the plots from Q1. You may use plot_grid() again to get them side by side, but not mandatory. Did the plots change? How?

```r
#determine ACF, PACF for renewable energy
#ACF
acf_deseason_renewable <- Acf(ts_deseason_renewable,
                  lag.max=40,
                  type="correlation",
                  plot=FALSE)

acf_deseason_plot_renewable <- autoplot(acf_deseason_renewable)+
  labs(title = "Deseasoned Renewable ACF")

#PACF
pacf_deseason_renewable <- Pacf(ts_deseason_renewable,
                  lag.max=40,
                  plot=FALSE)

pacf_deseason_plot_renewable <- autoplot(pacf_deseason_renewable)+
  labs(title = "Deseasoned Renewable PACF")

#determine ACF, PACF for hydro energy
#ACF
acf_deseason_hydro <- Acf(ts_deseason_hydro,
                  lag.max=40,
                  type="correlation",
                  plot=FALSE)

acf_deseason_plot_hydro <- autoplot(acf_deseason_hydro)+
  labs(title = "Deseasoned Hydroelectric ACF")

#PACF
pacf_deseason_hydro <- Pacf(ts_deseason_hydro,
                  lag.max=40,
                  plot=FALSE)

pacf_deseason_plot_hydro <- autoplot(pacf_deseason_hydro)+
  labs(title = "Deseasoned Hydroelectric PACF")

#plot ACF comparisons in a plot
plot_grid(energy_acf_plot_renewable,
          acf_deseason_plot_renewable,
          energy_acf_plot_hydro,
          acf_deseason_plot_hydro,
          align = "h",
          nrow = 2)
```



The deseasoned ACF for renewable and hydroelectric energy has more consistency between adjacent months. The renewable energy deseasoned ACF plot removes the slight peaks and depressions in the original time series, leaving a relatively linear decrease. The deseasoned hydroelectric ACF plot removes the oscillations that occur over the course of a year. This leaves a fairly consistent decrease, with a brief increase around lag 12.

```r
#Plot PACF comparisons in a grid
plot_grid(energy_pacf_plot_renewable,
          pacf_deseason_plot_renewable,
          energy_pacf_plot_hydro,
          pacf_deseason_plot_hydro,
          align = "h",
          nrow = 2)
```



The deseasoned PACF plots for both ts removed most of the high magnitude lags. Only the first lag retains a high magnitude for both hydroelectric and renewable energy time series. The general trend of which lags had positive or negative values remained the same, but now most were within the blue dashed lines.