ENV 790.30 - Time Series Analysis for Energy Data | Spring 2024

Assignment 3 - Due date 02/01/24

Samantha Pace

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github.

Once you have the file open on your local machine the first thing you will do is rename the file such that it includes your first and last name (e.g., "LuanaLima_TSA_A02_Sp24.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_Consumpti The data comes from the US Energy Information and Administration and corresponds to the December 2022 **Monthly** Energy Review. Once again you will work only with the following columns: Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption. Create a data frame structure with these three time series only.

R packages needed for this assignment:"forecast","tseries", and "Kendall". Install these packages, if you haven't done yet. Do not forget to load them before running your script, since they are NOT default packages.\

```
#Load/install required package here

#install.packages("forecast")
#install.packages("tseries")
#install.packages("Kendall")
#install.packages("cowplot")
#install.packages("ggplot2")
#install.packages("dplyr")

library(forecast)
```

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

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

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
getwd()
```

```
## [1] "C:/Users/saman/OneDrive/Desktop/Duke Spring 24/GITHUB/TSA_Sp24"
```

```
#Importing data using read.xlsx
energy_data <- read_excel("Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
                          skip = 12,
                          sheet = "Monthly Data", col_names = FALSE)
```

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

```
# Getting column names from row 11
read_col_names <-
  read_excel(path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
             skip = 10,n_max = 1, sheet="Monthly Data",col_names=FALSE)
```

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

```r
energy_data <-
  read_excel(path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
             skip = 12, sheet="Monthly Data",col_names=FALSE)
```

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

```r
# inputting correct column names
colnames(energy_data) <- read_col_names
head(energy_data)
```

```
## # A tibble: 6 x 14
##   Month               `Wood Energy Production` `Biofuels Production`
##   <dttm>                                 <dbl> <chr>
## 1 1973-01-01 00:00:00                     130. Not Available
## 2 1973-02-01 00:00:00                     117. Not Available
## 3 1973-03-01 00:00:00                     130. Not Available
## 4 1973-04-01 00:00:00                     125. Not Available
## 5 1973-05-01 00:00:00                     130. Not Available
## 6 1973-06-01 00:00:00                     125. Not Available
## # i 11 more variables: `Total Biomass Energy Production` <dbl>,
## #   `Total Renewable Energy Production` <dbl>,
## #   `Hydroelectric Power Consumption` <dbl>,
## #   `Geothermal Energy Consumption` <dbl>, `Solar Energy Consumption` <chr>,
```

```
## #   'Wind Energy Consumption' <chr>, 'Wood Energy Consumption' <dbl>,
## #   'Waste Energy Consumption' <dbl>, 'Biofuels Consumption' <chr>,
## #   'Total Biomass Energy Consumption' <dbl>, ...
```

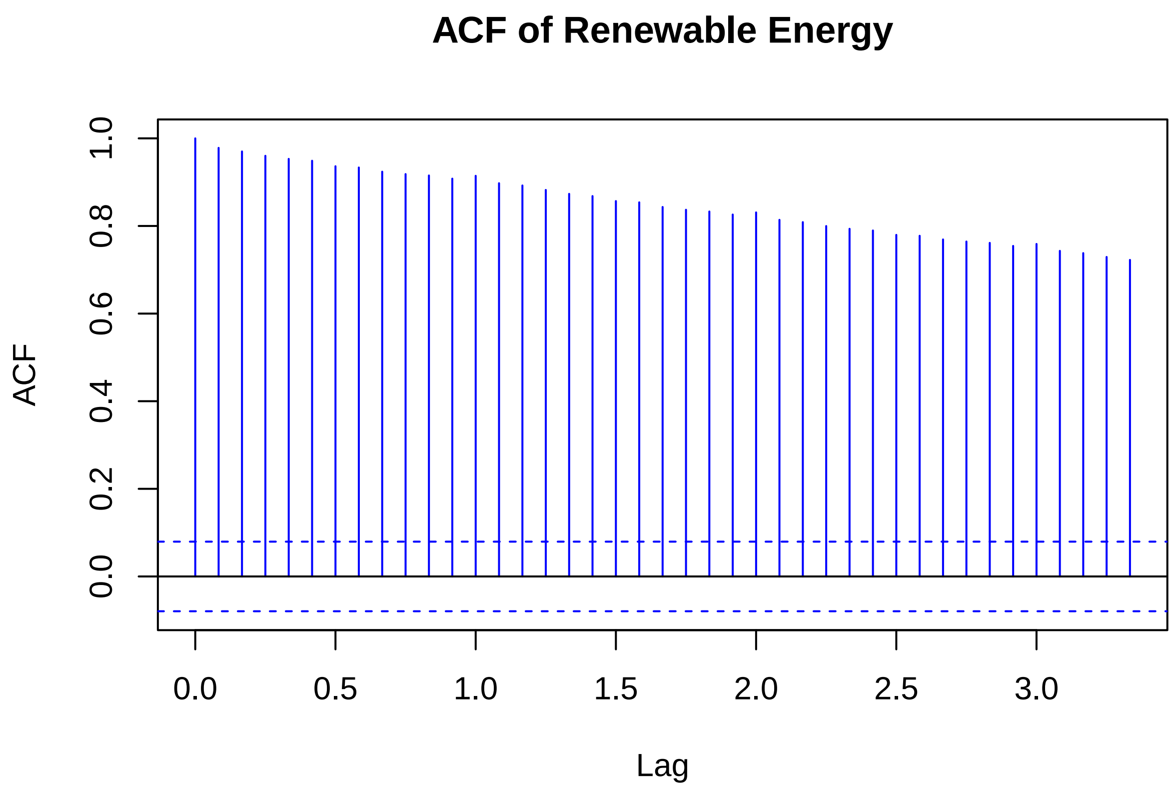
```
# selecting the 3 columns needed
energy_data <- energy_data %>% select(1, 5, 6)
```

##Trend Component

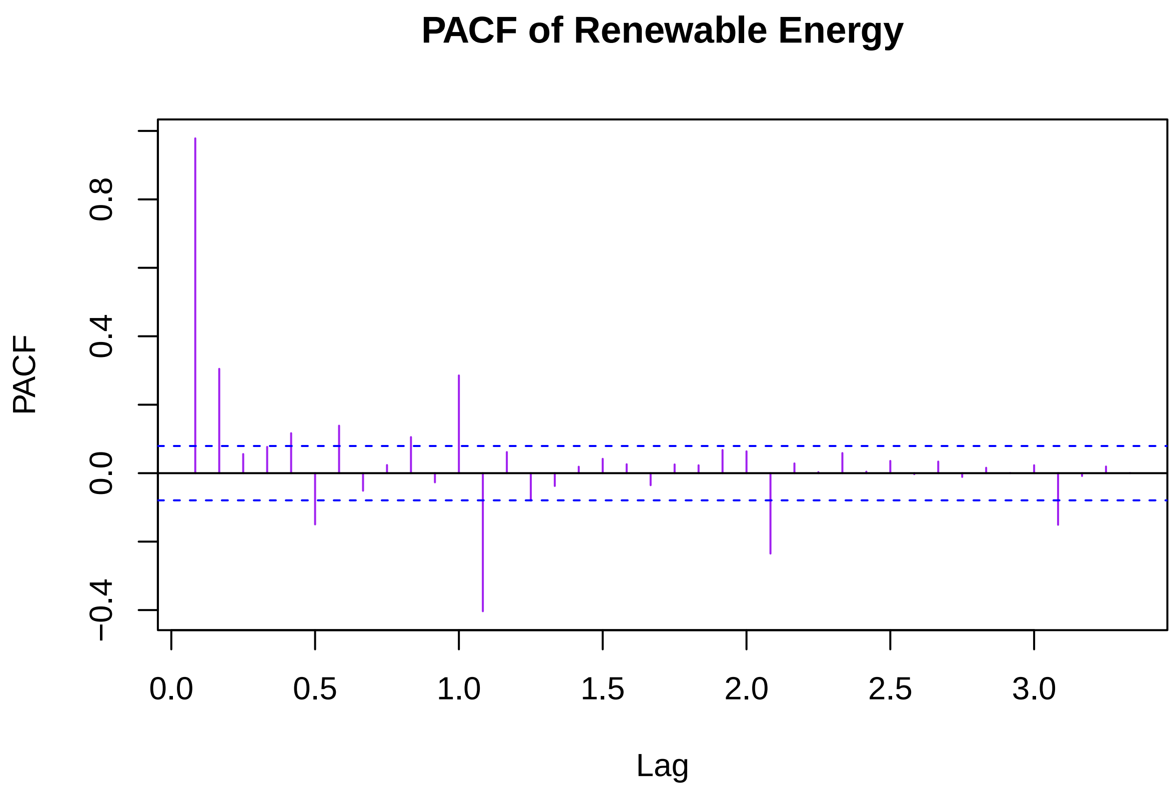
Q1

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)

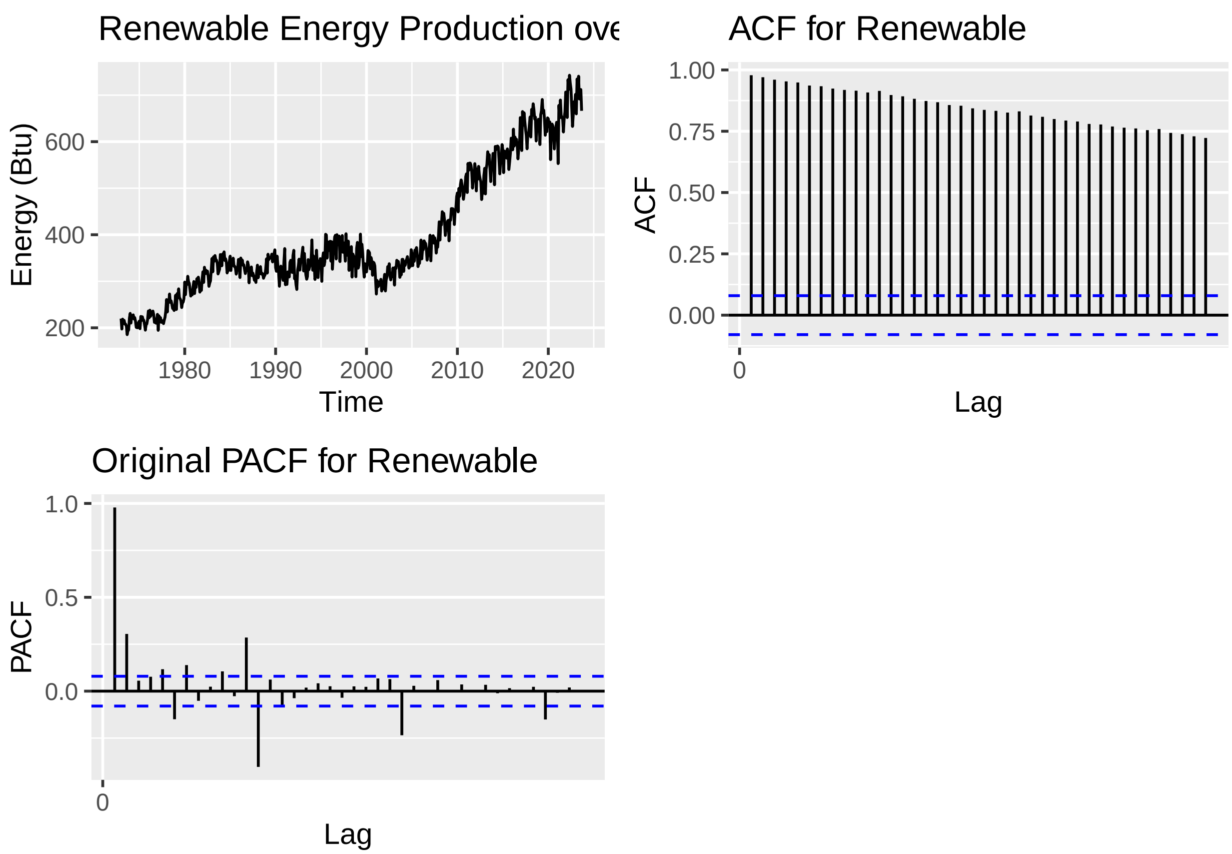
```
# transforming data frame in a time series object
REP_ts <- ts(energy_data$`Total Renewable Energy Production`,
             start = c(1973,1),
             frequency = 12)

HEC_ts <- ts(energy_data$`Hydroelectric Power Consumption`,
             start = c(1973,1),
             frequency = 12)

# creating individual plots for Renewable Energy Production
REP_plot1 <- autoplot(REP_ts,
                  ylab= "Energy (Btu)",
                  main = "Renewable Energy Production over Time")

REP_ACF <- acf(REP_ts,
               col = 'blue',
               lag.max = 40,
               ylab = "ACF",
               main = "ACF of Renewable Energy")
```



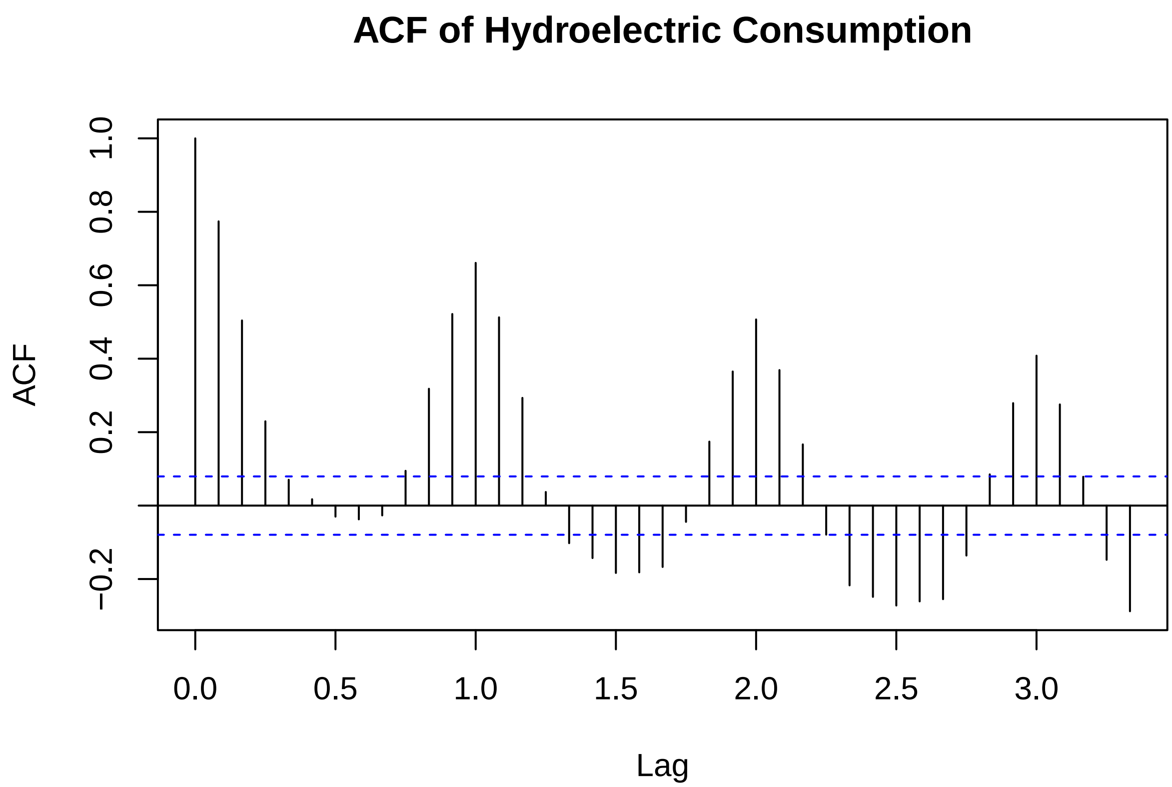

```r
REP_ACF_autoplot <- autoplot(REP_ACF)+
  ggtitle("ACF for Renewable")

REP_PACF <- pacf(REP_ts,
                 col = 'purple',
                 lag.max = 40,
                 ylab = 'PACF',
                 main = "PACF of Renewable Energy")
```



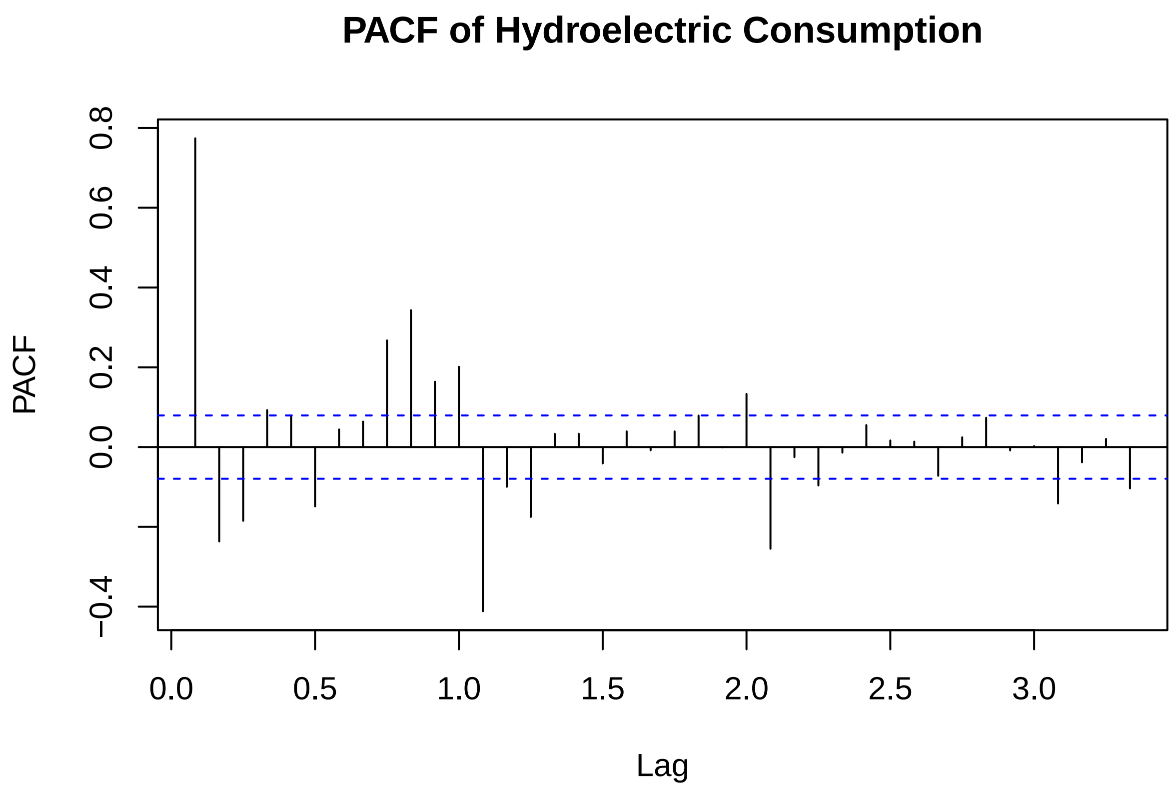

```
REP_PACF_autoplot <- autoplot(REP_PACF)+
  ggtitle("Original PACF for Renewable")

# combining REP individual with plot grid
plot_grid(REP_plot1, REP_ACF_autoplot, REP_PACF_autoplot)
```



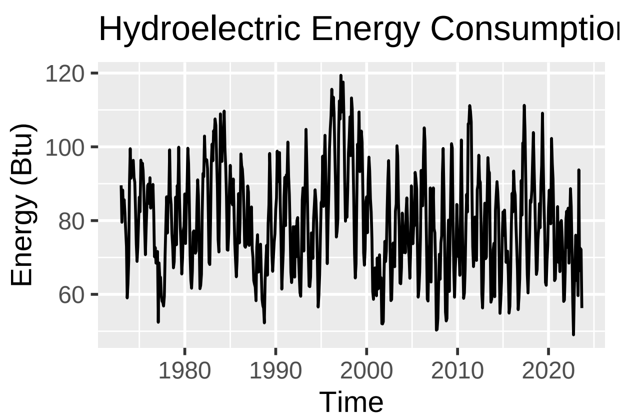

```r
# creating individual plots for Hydroelectric Consumption
HEC_plot1 <- autoplot(HEC_ts,
                 ylab = "Energy (Btu)",
                 main = "Hydroelectric Energy Consumption over Time")

HEC_ACF <- acf(HEC_ts,
               lag.max = 40,
               ylab = "ACF",
               main = "ACF of Hydroelectric Consumption")
```

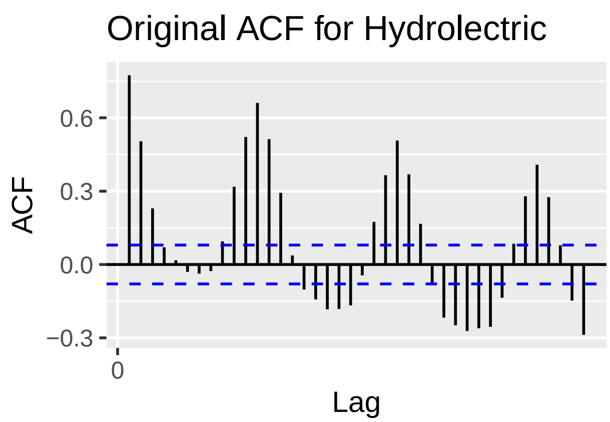


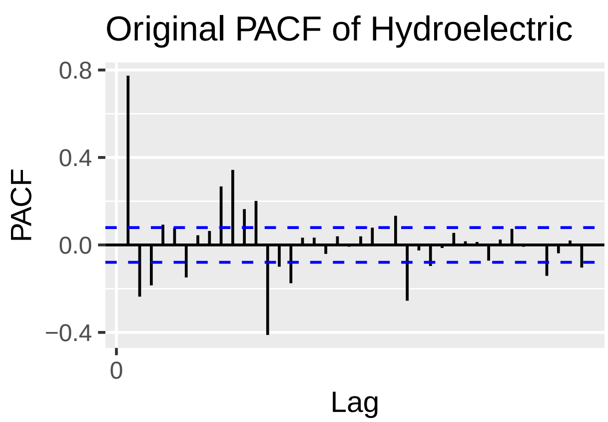

```
HEC_ACF_autoplot <- autoplot(HEC_ACF) +
  ggtitle("Original ACF for Hydrolectric")

HEC_PACF <- pacf(HEC_ts,
                 lag.max = 40,
                 ylab = "PACF",
                 main = "PACF of Hydroelectric Consumption")
```




```
HEC_PACF_autoplot <- autoplot(HEC_PACF) +
  ggtitle("Original PACF of Hydroelectric")

# combining with plotgrid
plot_grid(HEC_plot1, HEC_ACF_autoplot, HEC_PACF_autoplot)
```







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?

Answer: The Renewable Energy data appears to have an increasing trend based on the time series plot. The Hydroelectric Energy data appears to have a slight decreasing trend, but is difficult to discern. The Hydroelectric ACF graph clearly shows a regularly spaced wave component, indicating the presence of a seasonality component, separate from the trend component.

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.

```
# storing vectors
nobs <-nrow(energy_data)
t <- 1:nobs

# fitting linear trend to Renewable Energy Production Data
REP_linear_trend <- lm(energy_data$`Total Renewable Energy Production`~t)
summary(REP_linear_trend)
```

```
##
```

```
## Call:
## lm(formula = energy_data$‘Total Renewable Energy Production‘ ~
##     t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -148.27  -35.63   11.58   41.51  144.27
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 180.98940    4.90151   36.92   <2e-16 ***
## t             0.70404    0.01392   50.57   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 60.41 on 607 degrees of freedom
## Multiple R-squared:  0.8081, Adjusted R-squared:  0.8078
## F-statistic:  2557 on 1 and 607 DF,  p-value: < 2.2e-16
```

```r
# saving the regression coefficients for REP
REP_beta0 <- REP_linear_trend$coefficients[1]
REP_beta1 <- REP_linear_trend$coefficients[2]

# fitting linear trend to Hydroelectric Consumption (HEC) Data
HEC_linear_trend <- lm(energy_data$`Hydroelectric Power Consumption`~t)
summary(HEC_linear_trend)
```

```
##
## Call:
## lm(formula = energy_data$‘Hydroelectric Power Consumption‘ ~
##     t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.818 -10.620  -0.669   9.357  39.528
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 82.734747   1.140265  72.557  < 2e-16 ***
## t           -0.009849   0.003239  -3.041  0.00246 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 14.05 on 607 degrees of freedom
## Multiple R-squared:  0.015,  Adjusted R-squared:  0.01338
## F-statistic: 9.247 on 1 and 607 DF,  p-value: 0.002461
```

```r
# saving the regression coefficients
HEC_beta0 <- HEC_linear_trend$coefficients[1]
HEC_beta1 <- HEC_linear_trend$coefficients[2]
```

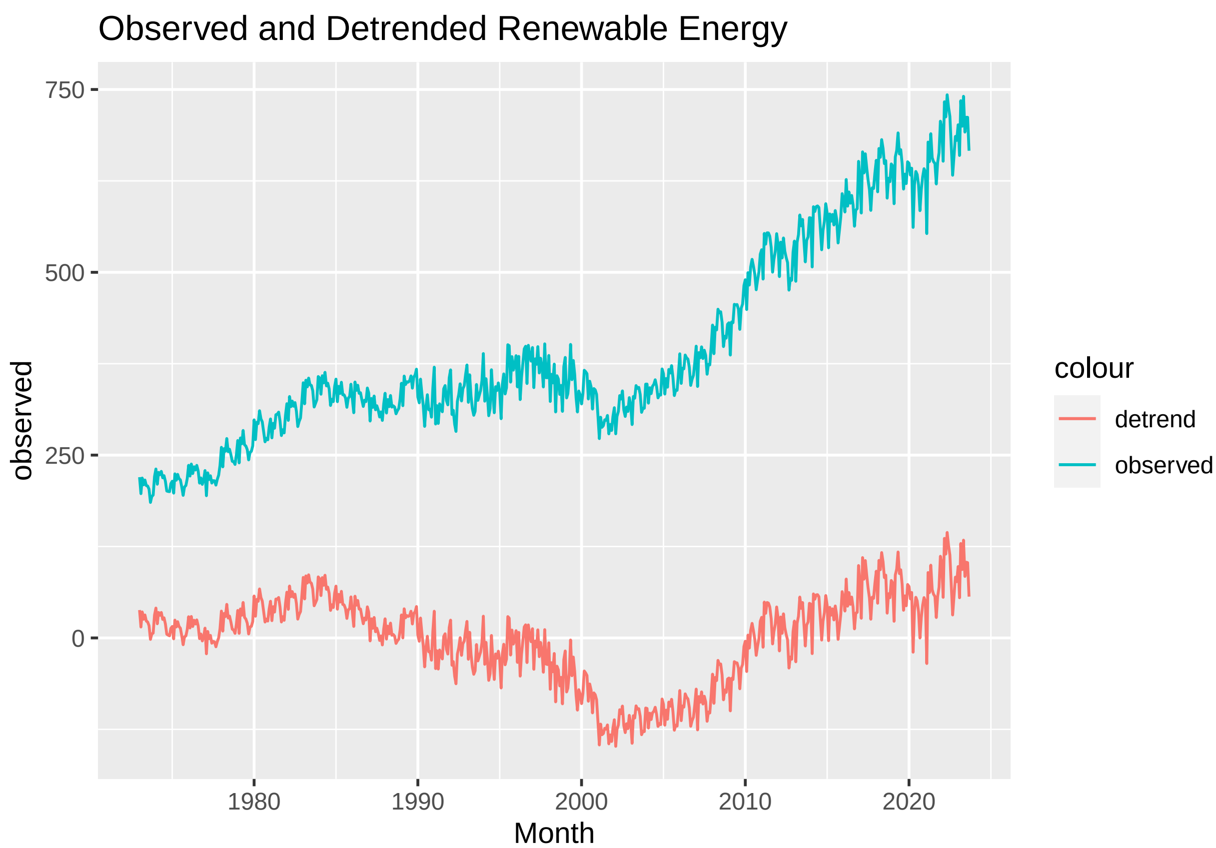
Answer: For Renewable Energy, the coefficient of t is 0.704, and has a positive slope, which means the linear model is increasing over time. The intercept is 180.989. The p-value is less than 0.05, meaning the values are statistically significant. The adjusted R squared value is 0.808, which means that this model, specifically the component of time, accounts for 80.8% of the variation of the data.

For Hydroelectric Energy, the coefficient of t is -0.00985, which is slightly decreasing because the sign is negative. The intercept is 82.73. The p value is less than 0.05, so the findings are statistically significant. The Adjusted R-squared value is 0.013, which means this model, specifically the component of time, only accounts for 1.3% of the variability of the data, which is very low. This linear trend does not fit the data well.

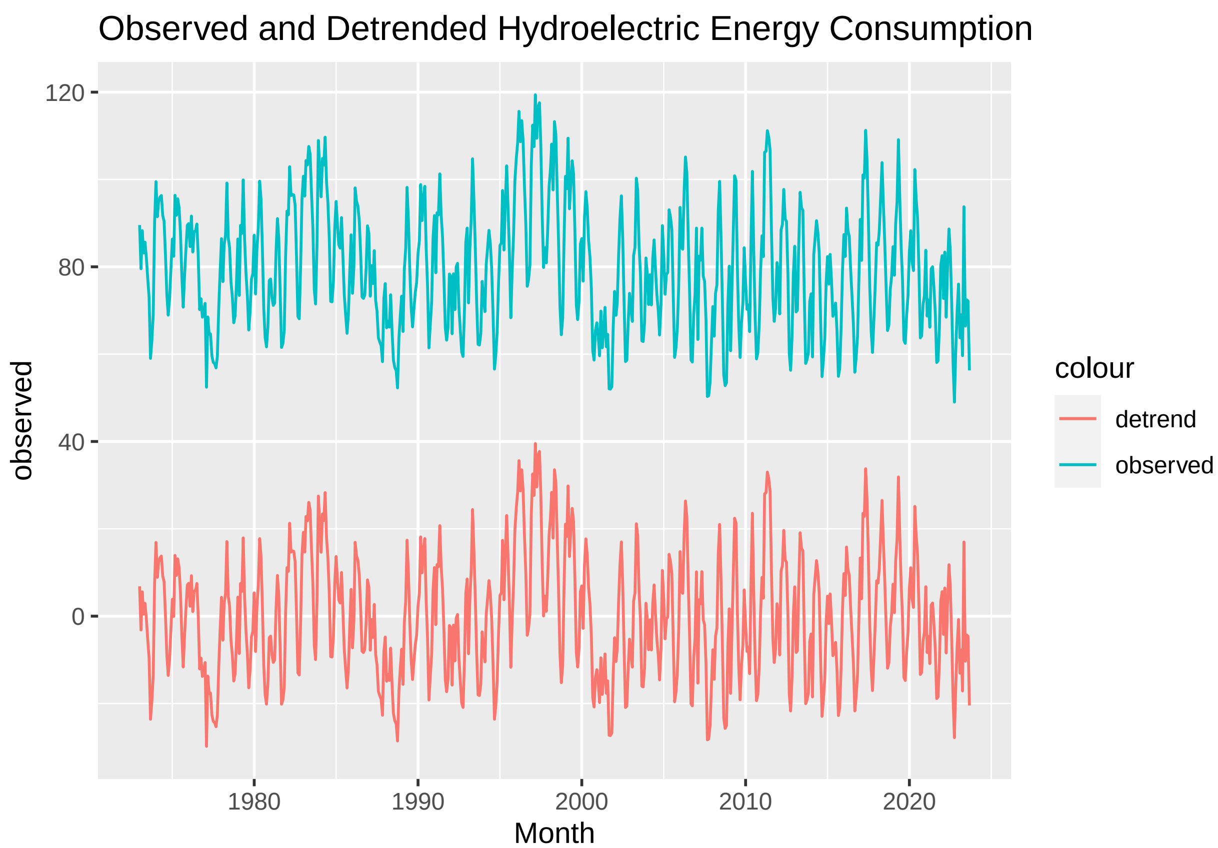
Q4

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
# Creating REP detrended series
REP_detrend <- energy_data[,2] - (REP_beta0 + REP_beta1*t)

colnames(REP_detrend)[1]<-"detrend"

REP_detrend_df <- data.frame("Month" = energy_data$Month,
                             "Observed" = energy_data[,2],
                             "Detrend" = REP_detrend)

colnames(REP_detrend_df)[2] <- "observed"

#plotting observed and detrended REP series
ggplot(REP_detrend_df, aes(x=Month)) +
  geom_line(aes(y=observed, color = "observed")) +
  geom_line(aes(y=detrend, color = "detrend")) +
  ggtitle("Observed and Detrended Renewable Energy")
```




```
# Creating HEC detrended series
HEC_detrend <- energy_data[,3] - (HEC_beta0 + HEC_beta1*t)

colnames(HEC_detrend)[1]<-"detrend"

HEC_detrend_df <- data.frame("Month" = energy_data$Month,
                             "Observed" = energy_data[,3],
                             "Detrend" = HEC_detrend)

colnames(HEC_detrend_df)[2]<-"observed"

#plotting observed and detrended HEC series
  ggplot(HEC_detrend_df, aes(x=Month)) +
    geom_line(aes(y=observed, color = "observed")) +
    geom_line(aes(y=detrend, color = "detrend")) +
    ggtitle("Observed and Detrended Hydroelectric Energy Consumption")
```



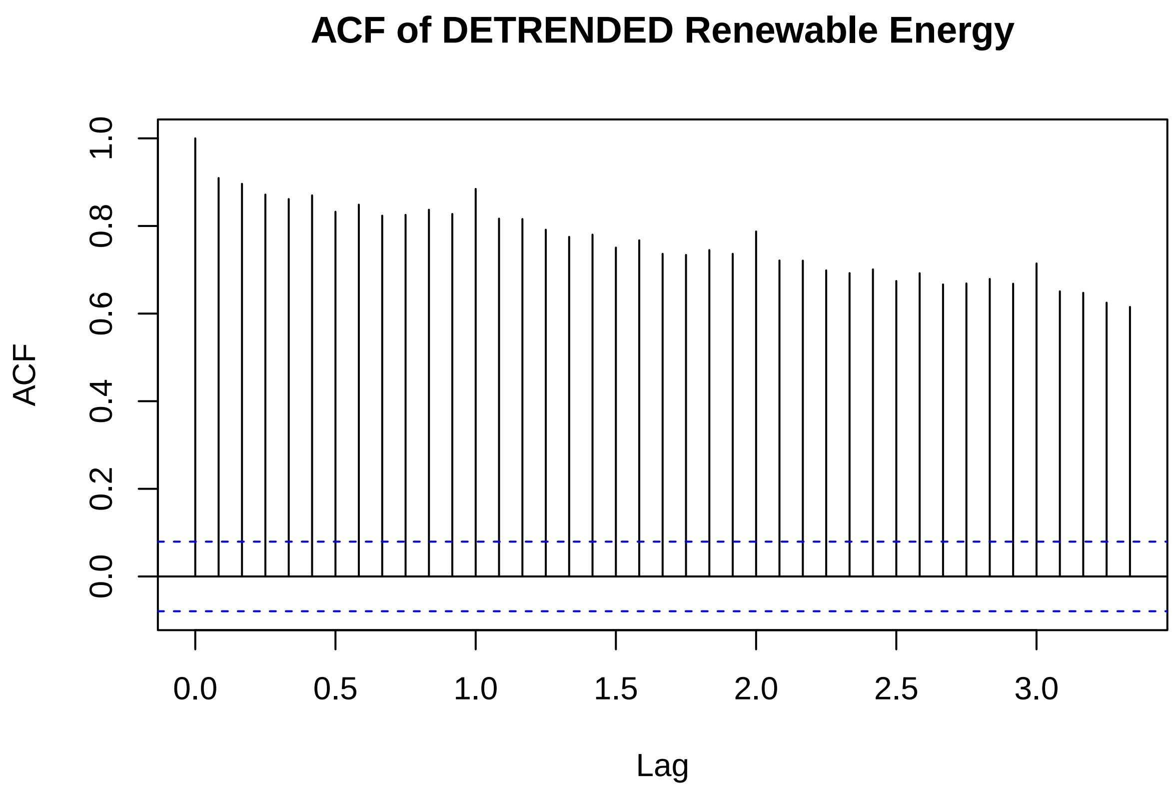
Answer: For the Renewable Energy plot, the data is now centered on zero since the intercept was removed. The positive linear model was also removed, making the slope of the detrended data less steep. The observations for the detrended data now range from about -125 Btu to 125 Btu. There is still variability in this detrended dataset.

For the Hydroelectric Consumption comparison of observed and detrended, since there was very little slope, the main noticeable change is the removal of the intercept from the detrended series which centers the data around zero. The rest of the variability of the dataset looks to be nearly the same; there is still a lot of variability in the dataset.

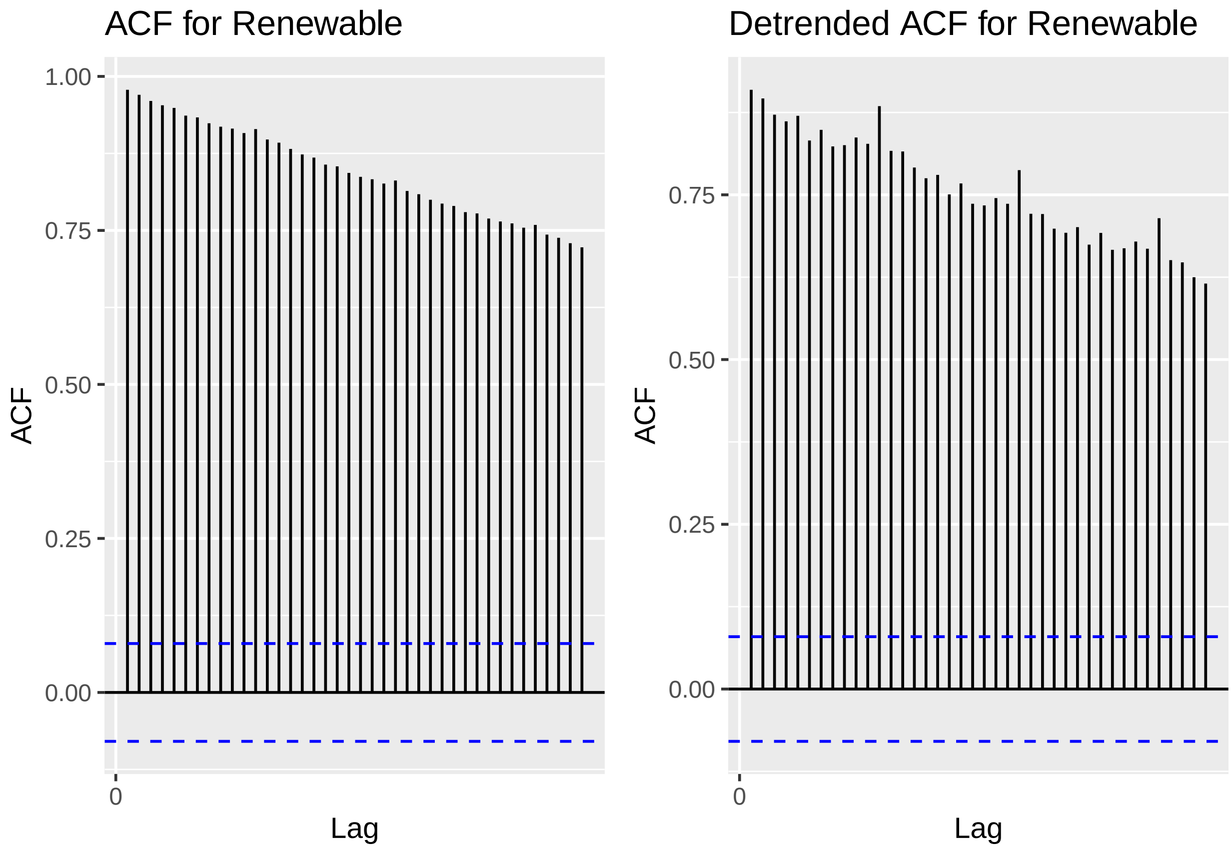
Q5

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. nut mot mandatory. Did the plots change? How?

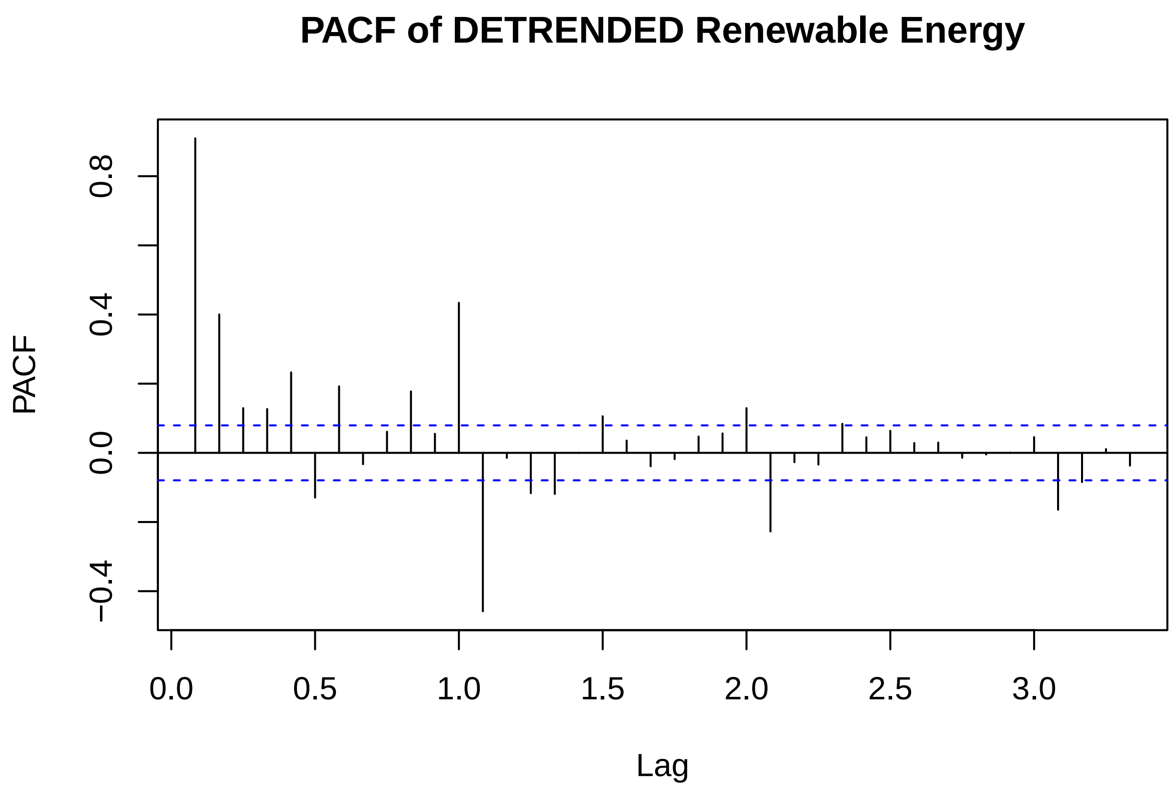
```
# transforming data frame in a time series object
REP_detrend_ts <- ts(REP_detrend_df$detrend, start = c(1973,1), frequency = 12)
HEC_detrend_ts <- ts(HEC_detrend_df$detrend, start = c(1973,1), frequency = 12)

# plotting detrended ACF and PACF for Renewable Energy
REP_detrend_ACF <- acf(REP_detrend_ts,
                lag.max = 40,
                ylab = "ACF",
                main = "ACF of DETRENDED Renewable Energy")
```



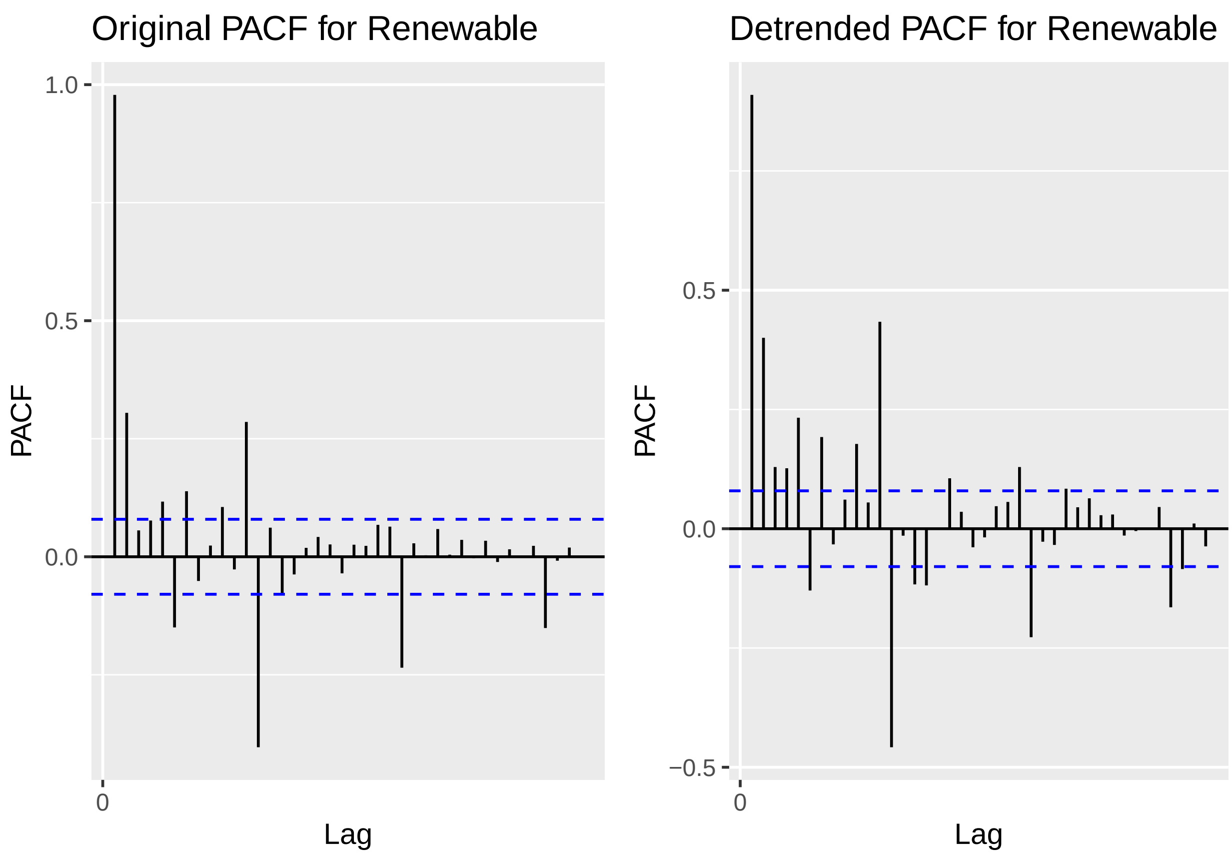

```
REP_ACF_detrend_autoplot <- autoplot(REP_detrend_ACF) +
  ggtitle("Detrended ACF for Renewable")

plot_grid(REP_ACF_autoplot, REP_ACF_detrend_autoplot)
```



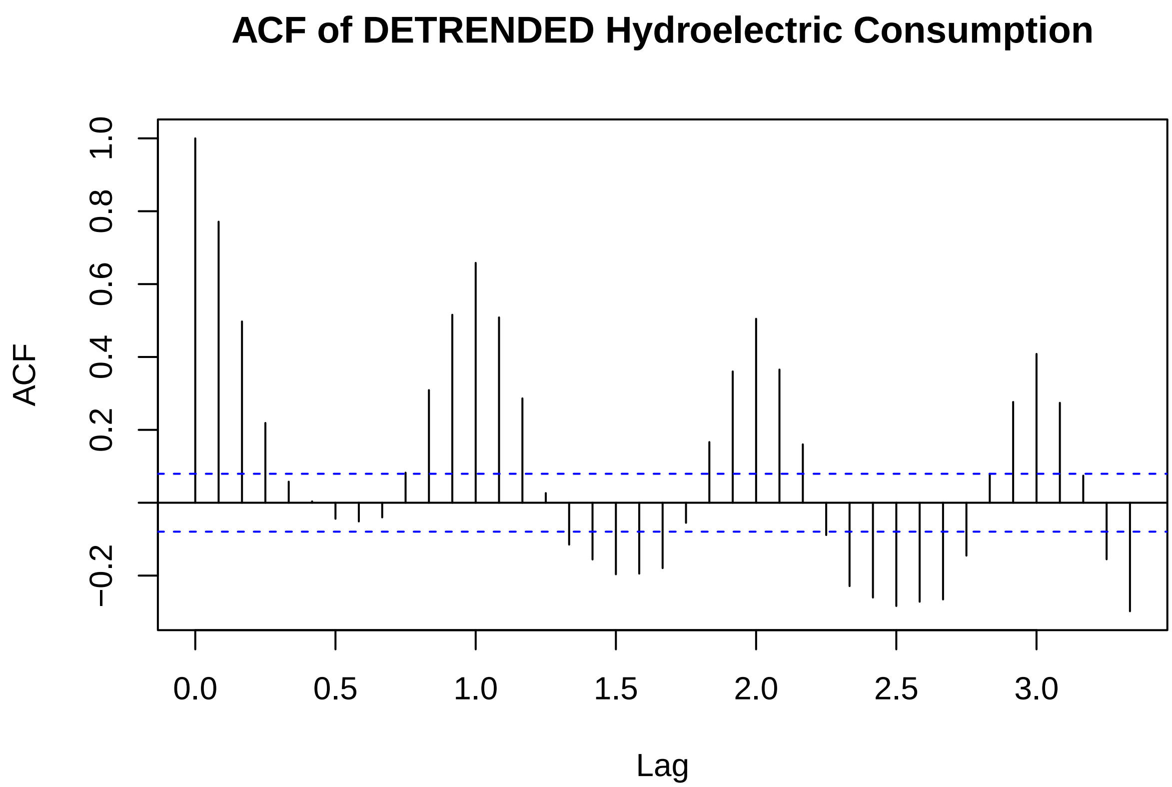

```
REP_detrend_PACF <- pacf(REP_detrend_ts,
                  lag.max = 40,
                  ylab = 'PACF',
                  main = "PACF of DETRENDED Renewable Energy")
```



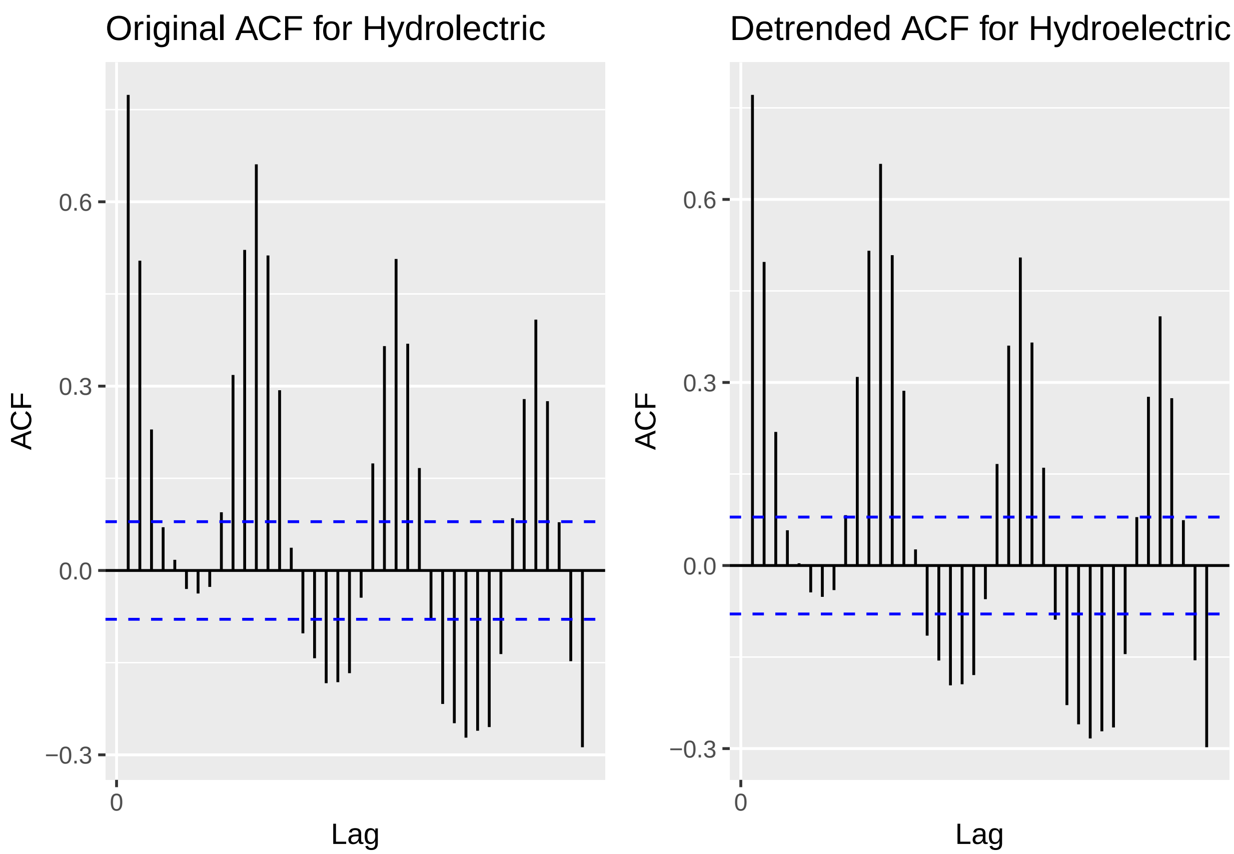

```
REP_PACF_detrend_autoplot <- autoplot(REP_detrend_PACF)+
  ggtitle("Detrended PACF for Renewable")

plot_grid(REP_PACF_autoplot, REP_PACF_detrend_autoplot)
```



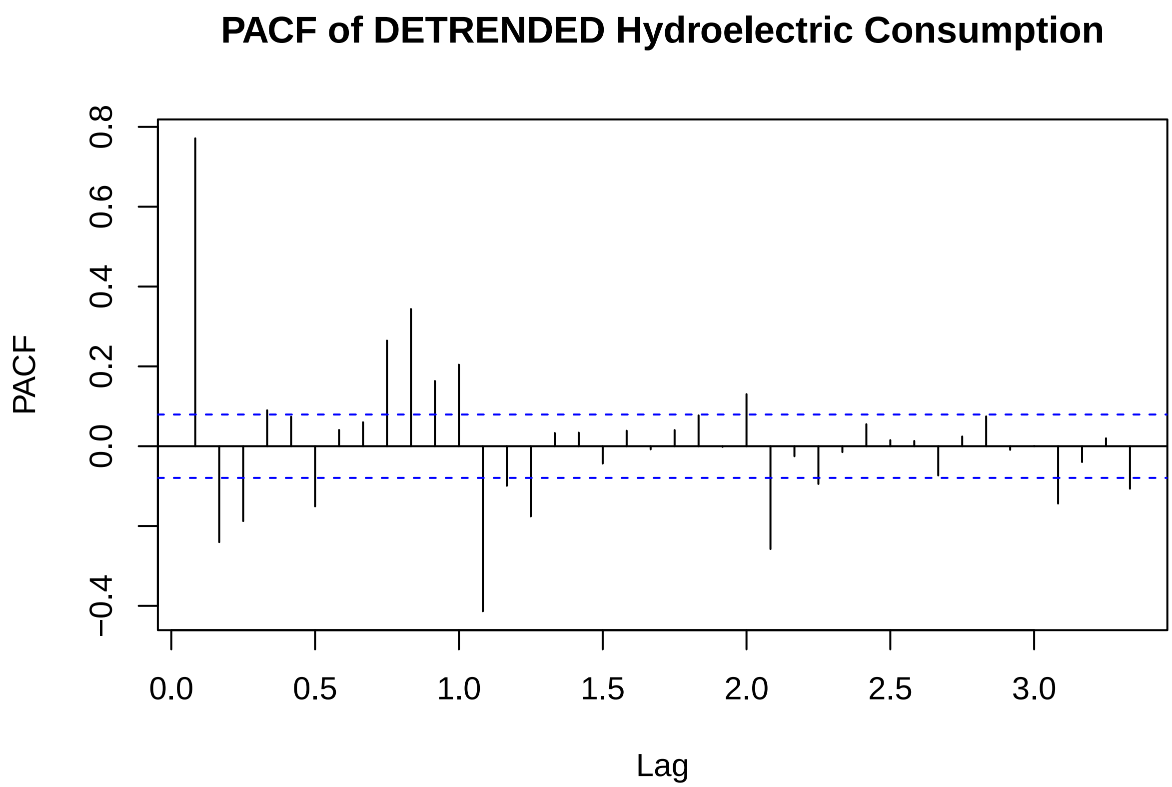

```r
# creating individual plots for Hydroelectric Consumption
HEC_detrend_ACF <- acf(HEC_detrend_ts,
                lag.max = 40,
                ylab = "ACF",
                main = "ACF of DETRENDED Hydroelectric Consumption")
```



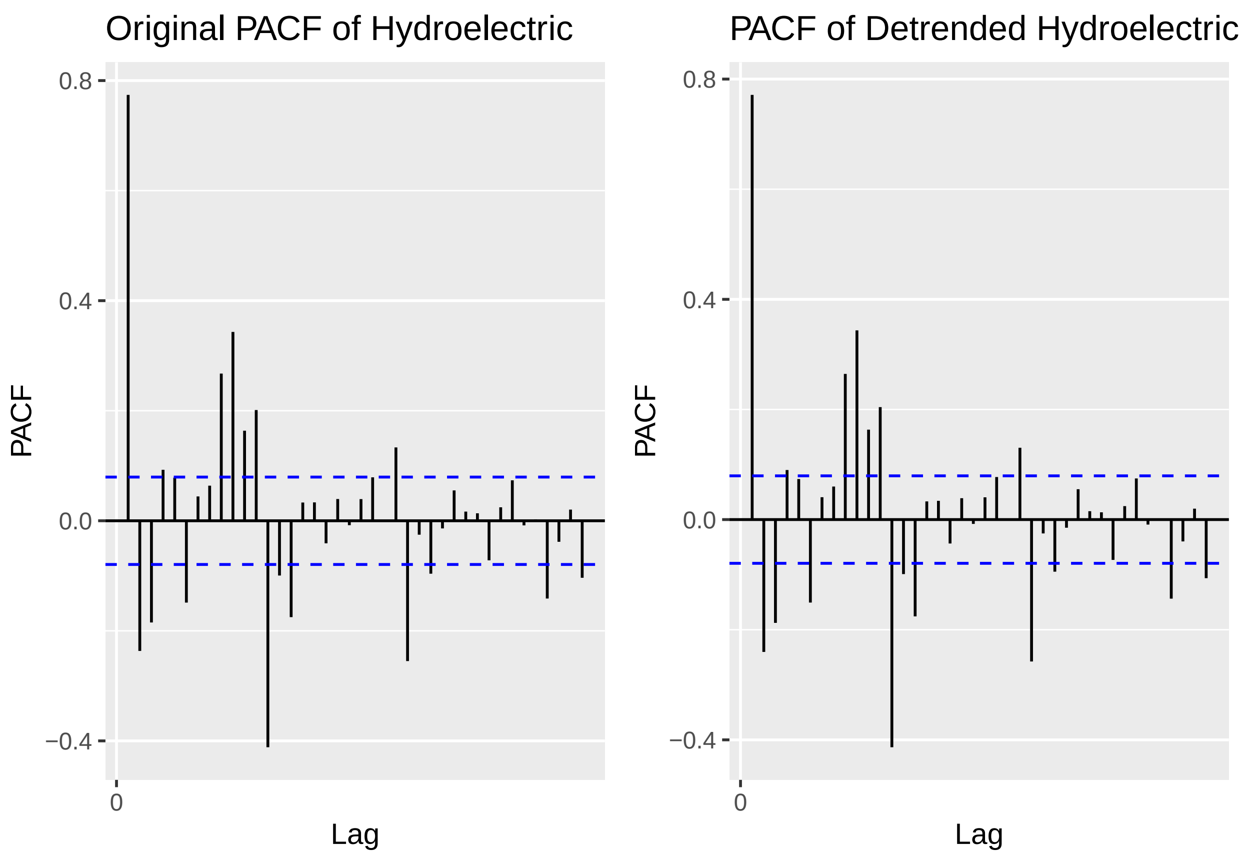

```r
HEC_ACF_detrend_autoplot <- autoplot(HEC_detrend_ACF)+
  ggtitle("Detrended ACF for Hydroelectric")

plot_grid(HEC_ACF_autoplot, HEC_ACF_detrend_autoplot)
```




```
HEC_detrend_PACF <- pacf(HEC_detrend_ts,
                    lag.max = 40,
                    ylab = "PACF",
                    main = "PACF of DETRENDED Hydroelectric Consumption")
```




```
HEC_PACF_detrend_autoplot <- autoplot(HEC_detrend_PACF) +
  ggtitle("PACF of Detrended Hydroelectric")

plot_grid(HEC_PACF_autoplot, HEC_PACF_detrend_autoplot)
```



Answer: The Renewable Energy's detrended compared to the original ACF shows a slightly lower and more irregular dependence on t from one lag to the next. The PACF of the observed and detrended Renewable Energy looks fairly similar, with a similar pattern of future lags having a coefficient still, however the intensity/magnitude of the coefficients is lower than the original PACF.

For the Hydroelectric Energy Consumption, there looks to be the same patterns in the ACF for the original and detrended data. The results are similar for the PACF - there is no noticeable difference between the PACF of the original and detrended data.

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.

Answer: From the original time series and ACF plots, it is difficult to tell if there is a seasonal trend in the Renewable Energy data. The PACF has a somewhat regular negative correlation coefficient pattern which suggests there may be some seasonality, but it is hard to detect.

The Hydroelectric Energy ACF plot shows a distinct wave pattern in the ACF that suggests there is a seasonal trend present. The PACF shows a somewhat regular trend of negative correlation coefficients in the lags, which suggests that there is a seasonal trend.

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
# fitting a seasonal means model
# creating dummies for REP
REP_dummies <- seasonaldummy(REP_ts)

# fitting linear model to the dummies, summary
REP_seasonal_means_model <- lm(REP_ts~REP_dummies)
summary(REP_seasonal_means_model)
```

```
##
## Call:
## lm(formula = REP_ts ~ REP_dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -199.19  -86.35  -48.84  113.18  331.58
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     404.526     19.574  20.666   <2e-16 ***
## REP_dummiesJan    2.962     27.546   0.108    0.914
## REP_dummiesFeb  -34.476     27.546  -1.252    0.211
## REP_dummiesMar    3.929     27.546   0.143    0.887
## REP_dummiesApr   -8.695     27.546  -0.316    0.752
## REP_dummiesMay    6.645     27.546   0.241    0.809
## REP_dummiesJun   -4.198     27.546  -0.152    0.879
## REP_dummiesJul    2.460     27.546   0.089    0.929
## REP_dummiesAug   -5.026     27.546  -0.182    0.855
## REP_dummiesSep  -29.119     27.546  -1.057    0.291
## REP_dummiesOct  -20.068     27.682  -0.725    0.469
## REP_dummiesNov  -20.346     27.682  -0.735    0.463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 138.4 on 597 degrees of freedom
## Multiple R-squared:  0.009296,   Adjusted R-squared:  -0.008958
## F-statistic: 0.5093 on 11 and 597 DF,  p-value: 0.8976
```

```r
# creating dummies for HEC
HEC_dummies <- seasonaldummy(HEC_ts)

# fitting linear model to the dummies and summary
HEC_seasonal_means_model <- lm(HEC_ts~HEC_dummies)
summary(HEC_seasonal_means_model)
```

```
##
## Call:
## lm(formula = HEC_ts ~ HEC_dummies)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.323  -5.849  -0.468   6.243  32.290
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      80.282      1.470  54.601  < 2e-16 ***
## HEC_dummiesJan    4.807      2.069   2.323  0.02050 *
## HEC_dummiesFeb   -2.725      2.069  -1.317  0.18831
## HEC_dummiesMar    6.825      2.069   3.298  0.00103 **
## HEC_dummiesApr    5.319      2.069   2.571  0.01039 *
## HEC_dummiesMay   13.922      2.069   6.729 4.02e-11 ***
## HEC_dummiesJun   10.650      2.069   5.147 3.60e-07 ***
## HEC_dummiesJul    3.912      2.069   1.891  0.05914 .
## HEC_dummiesAug   -5.677      2.069  -2.744  0.00626 **
## HEC_dummiesSep  -16.797      2.069  -8.118 2.72e-15 ***
## HEC_dummiesOct  -16.468      2.079  -7.920 1.17e-14 ***
## HEC_dummiesNov  -10.885      2.079  -5.235 2.29e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.4 on 597 degrees of freedom
## Multiple R-squared:  0.4697, Adjusted R-squared:  0.4599
## F-statistic: 48.07 on 11 and 597 DF,  p-value: < 2.2e-16
```

Answer: The p value on the Renewable Energy regression is above 0.05, so the coefficients generated are not significant. The adjusted R squared value is also very close to zero, which means that this model does not explain much at all of the variability of the data. It looks to be that the Renewable Energy Production data does not have a seasonality component. I was unable to detect a seasonality pattern in the REP original data, I thought it was possible there was one. This seasonal means model confirms my answer from Q6.

The Hydroelectric Power Consumption data looks to have a seasonal component. The adjusted R squared value is 0.4599, which means that this model accounts for about 46% of the variability. The p value is less than 0.05 for all the coefficients except February and July, which means it is significant. These results match my answer to Q6.

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?

```r
# storing REP coefficients
REP_deseas_beta_int <- REP_seasonal_means_model$coefficients[1]
REP_deseas_beta_coeff <- REP_seasonal_means_model$coefficients[2:12]

# storing HEC coefficients
HEC_deseas_beta_int <- HEC_seasonal_means_model$coefficients[1]
HEC_deseas_beta_coeff <- HEC_seasonal_means_model$coefficients[2:12]

nobs <- nrow(energy_data)

#computing REP seasonal component
REP_seas_comp <- array(0, nobs)
```

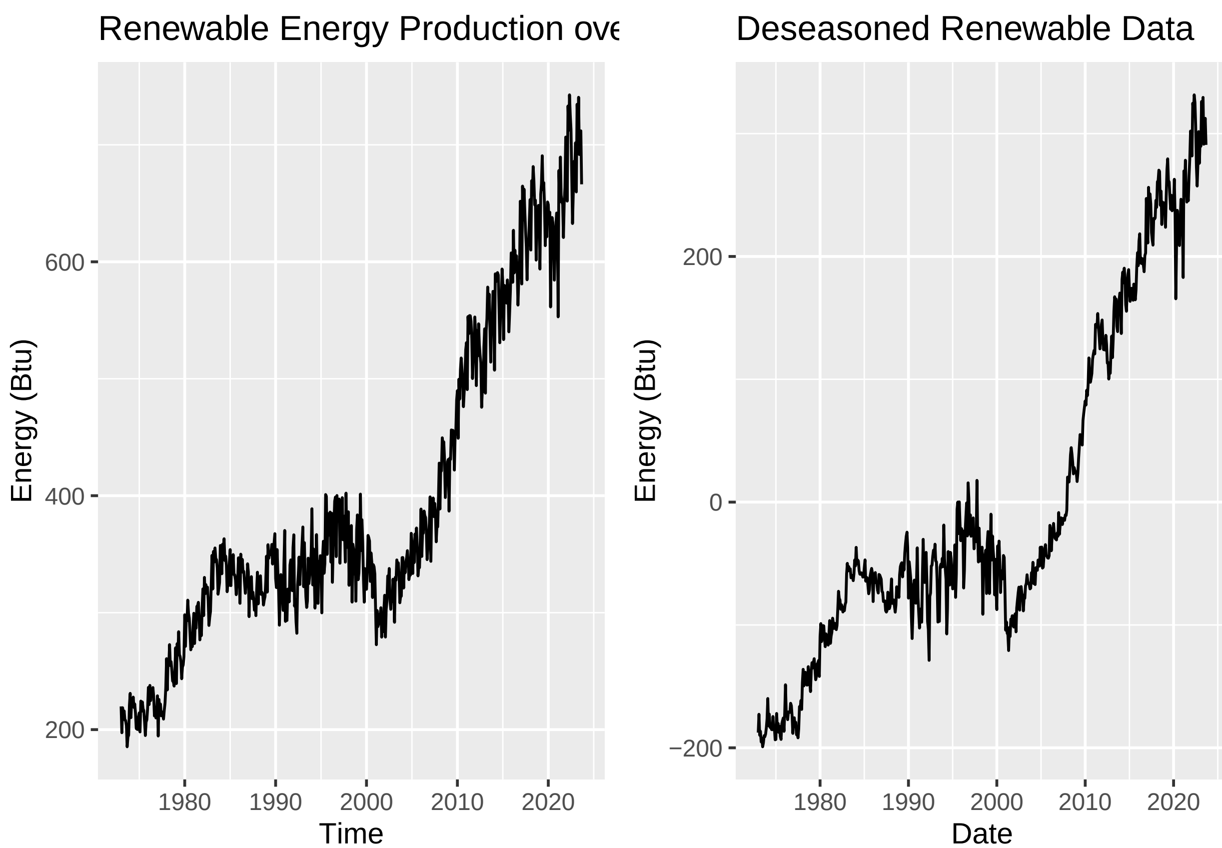
```r
for (i in 1:nobs) {
  REP_seas_comp[i] <-
    (REP_deseas_beta_int+REP_deseas_beta_coeff %*% REP_dummies[i,])
}

# computing HEC seasonal component
HEC_seas_comp <- array(0, nobs)
for (i in 1:nobs) {
  HEC_seas_comp[i] <-
    (HEC_deseas_beta_int+HEC_deseas_beta_coeff %*% HEC_dummies[i,])
}

# Removing seasonal component for REP & HEC
REP_deseason <- energy_data[,2] - REP_seas_comp
HEC_deseason <- energy_data[,3] - HEC_seas_comp

# converting deasoned data to a time series object
REP_deseason_ts <- ts(REP_deseason[1],
                    start = c(1973,1),
                     frequency = 12)

HEC_deseason_ts <- ts(HEC_deseason$`Hydroelectric Power Consumption`,
                      start = c(1973,1),
                      frequency = 12)

# creating plots
REP_deseason_plot <- autoplot(REP_deseason_ts,
         ylab = "Energy (Btu)",
         xlab = "Date",
         plot = FALSE,
         main = "Deseasoned Renewable Data")
```

```
## Warning in ggplot2::geom_line(na.rm = TRUE, ...): Ignoring unknown parameters:
## ‘plot‘
```

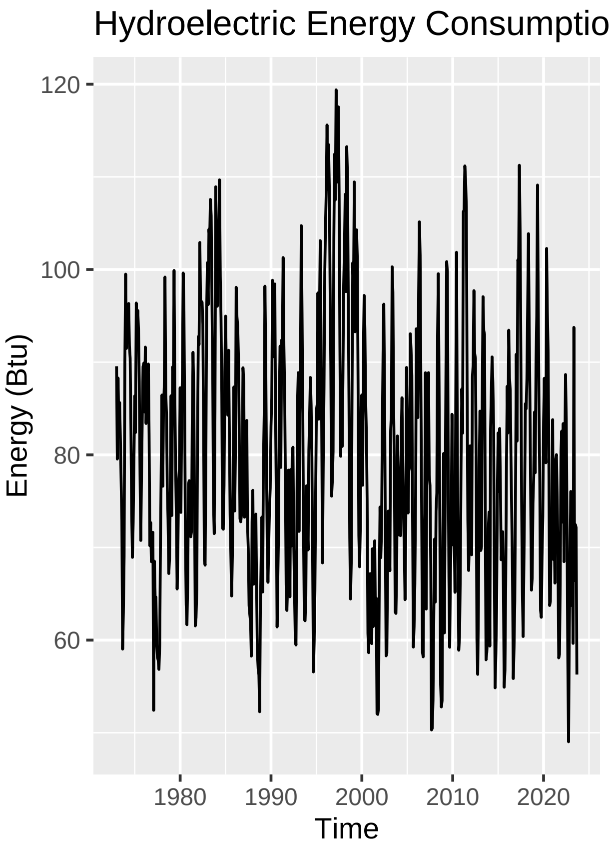
```r
HEC_deseason_plot <- autoplot(HEC_deseason_ts,
         ylab = "Energy (Btu)",
         xlab = "Date",
         plot = FALSE,
         main = "Deseasoned Hydroelectric Data")
```

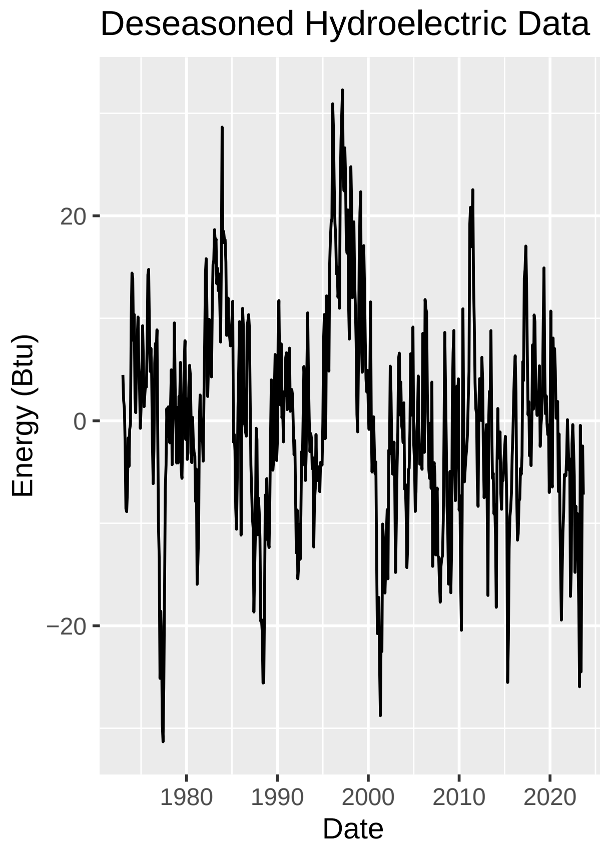
```
## Warning in ggplot2::geom_line(na.rm = TRUE, ...): Ignoring unknown parameters:
## ‘plot‘
```

```r
plot_grid(REP_plot1, REP_deseason_plot)
```




```r
plot_grid(HEC_plot1, HEC_deseason_plot)
```





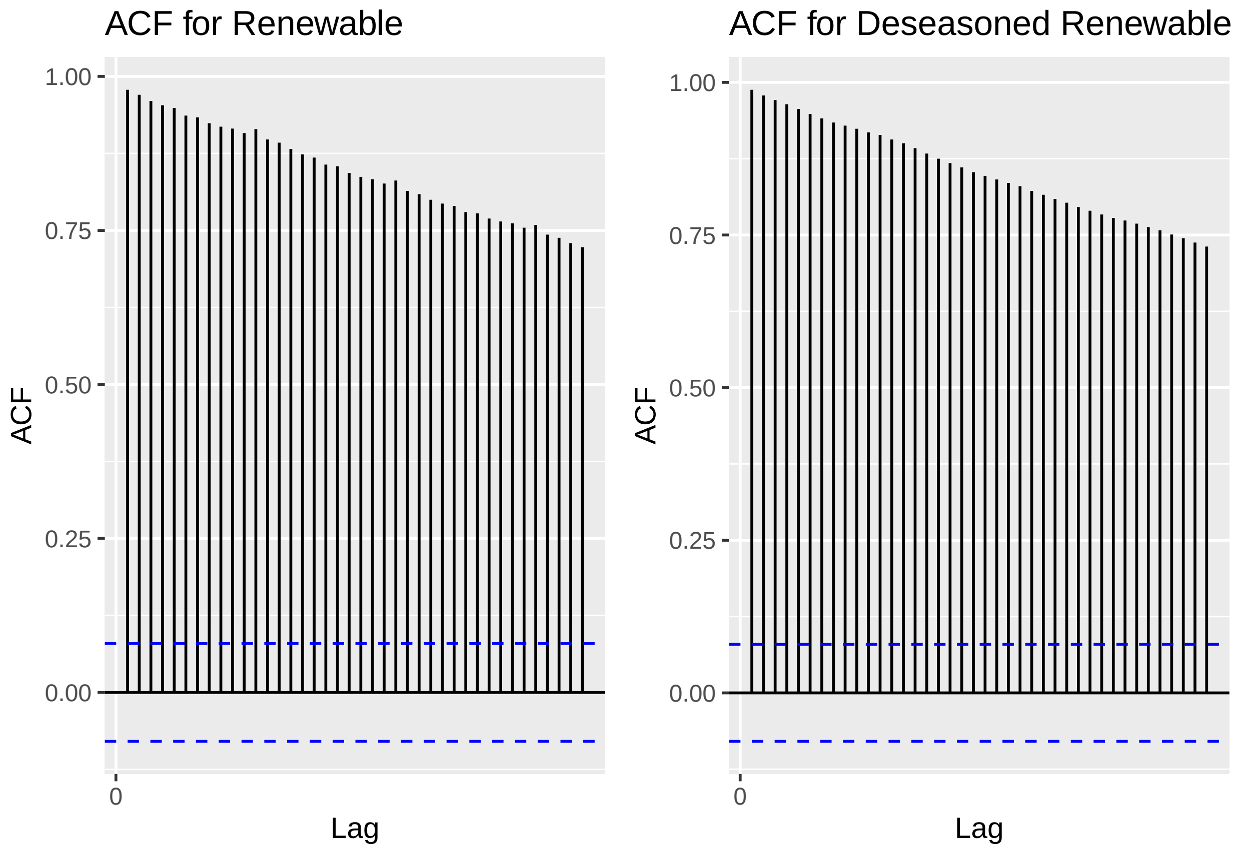
Answer: The deseasoned renewable energy data ranges from about -200 to 350 trillion Btu compared with the original data's range of 200 to 750 trillion Btu. The slope appears to be about the same. There looks to be less upward/downward wave patterns in the individual years of the detrended data.

For the hydroelectric consumption data, the deseasoned data showed a significant reduction in range from the original data. The deseasoned one ranges from about -30 to 30 trillion Btu, compared to the original data which ranged from 50 to 120 trillion Btu. The deseasoned data is centered on zero, and has significantly less of the evenly spaced up/down wave pattern in the data.

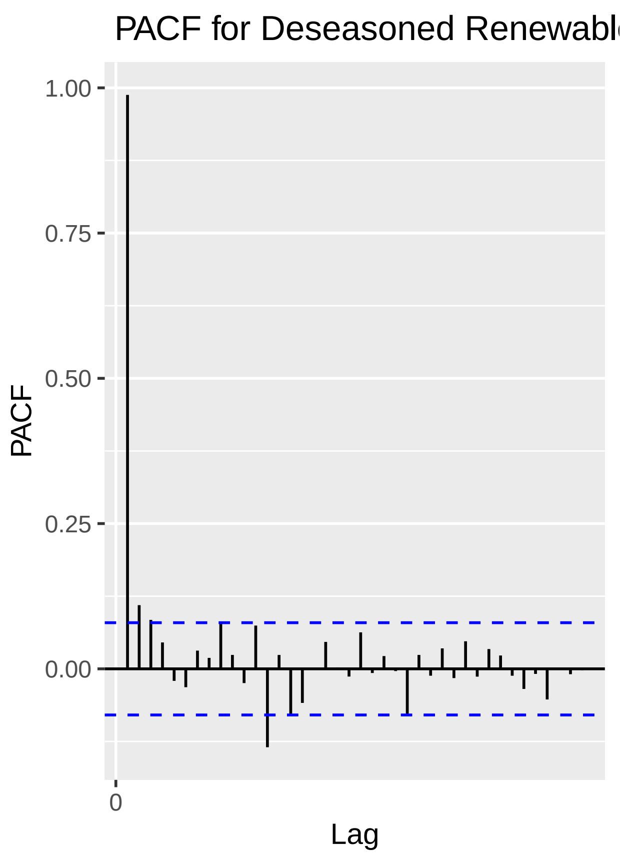
Q9

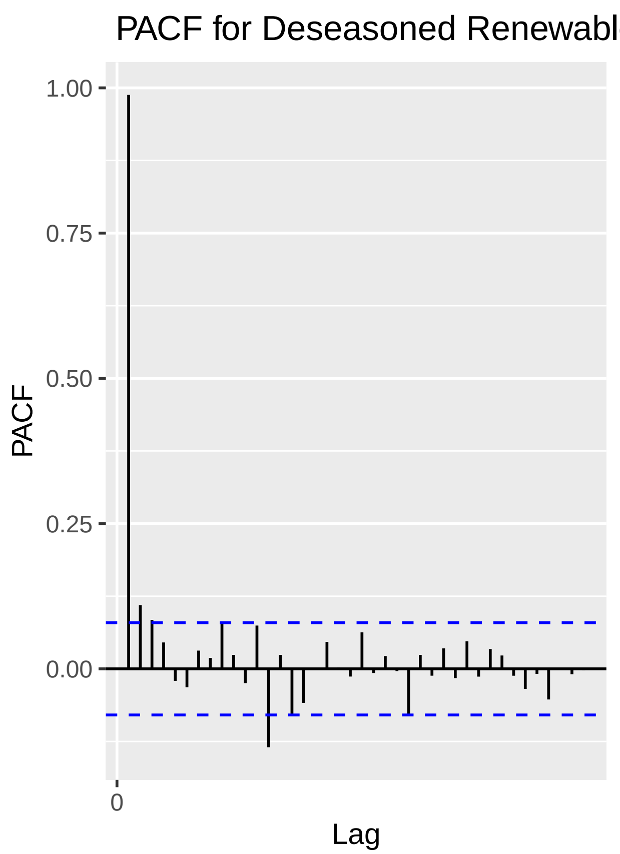
Plot ACF and PACF for the deseason series and compare with the plots from Q1. You may use plot_grid() again to get them side by side. nut mot mandatory. Did the plots change? How?

```r
# comparing ACFs for deseason and original Renewable data
REP_detrend_ACF <- acf(REP_deseason_ts,
               lag.max = 40,
               ylab = "ACF",
               plot = FALSE,
               main = "Deseasoned Renewable ACF")

REP_deseason_ACF_autoplot <- autoplot(REP_detrend_ACF)+
  ggtitle("ACF for Deseasoned Renewable")

plot_grid(REP_ACF_autoplot, REP_deseason_ACF_autoplot)
```

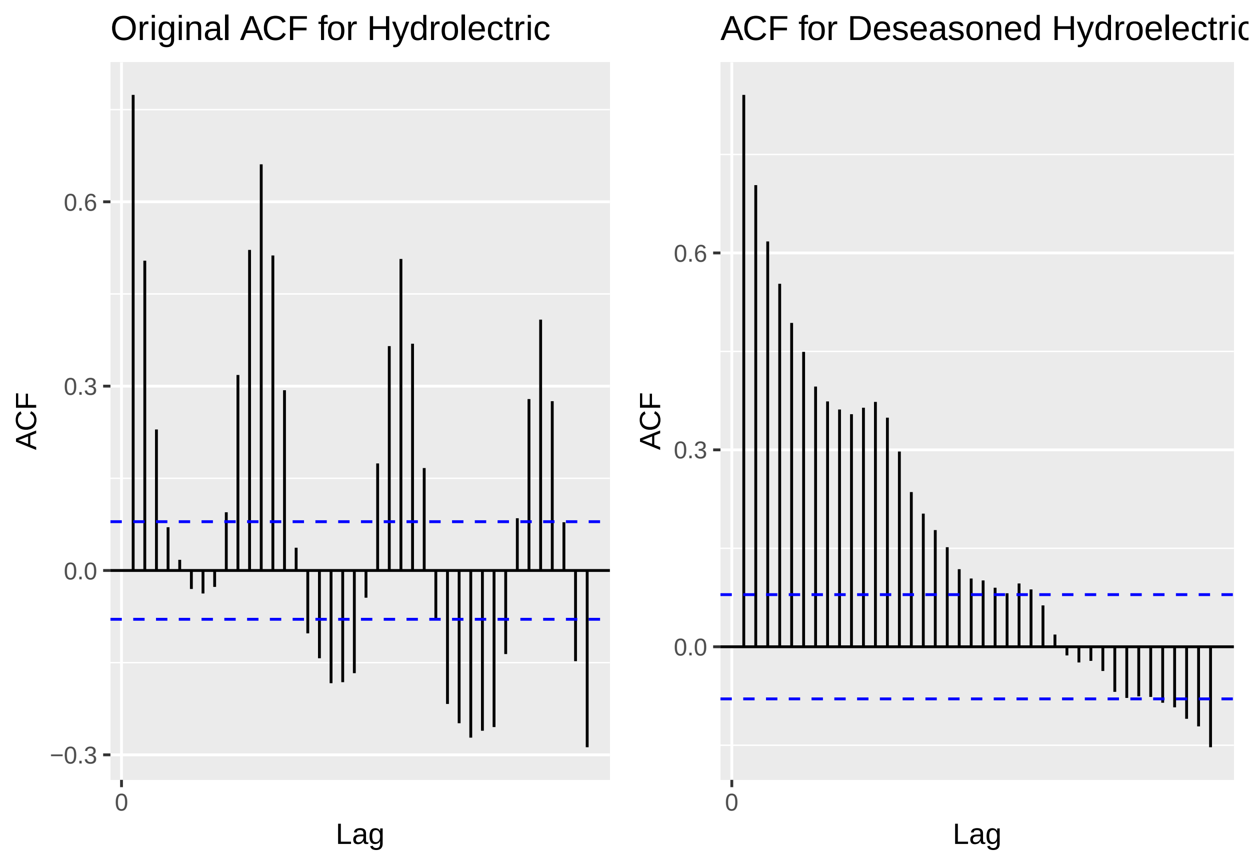



```r
# comparing PACFs for deseason and original Renewable data
REP_detrend_PACF <- pacf(REP_deseason_ts,
                  plot = FALSE,
                  lag.max = 40,
                  ylab = 'PACF',
                  main = "PACF for Deseasoned Renewable")

REP_PACF_autoplot <- autoplot(REP_detrend_PACF)+
  ggtitle(" PACF for Deseasoned Renewable")

plot_grid(REP_PACF_autoplot, REP_PACF_autoplot)
```

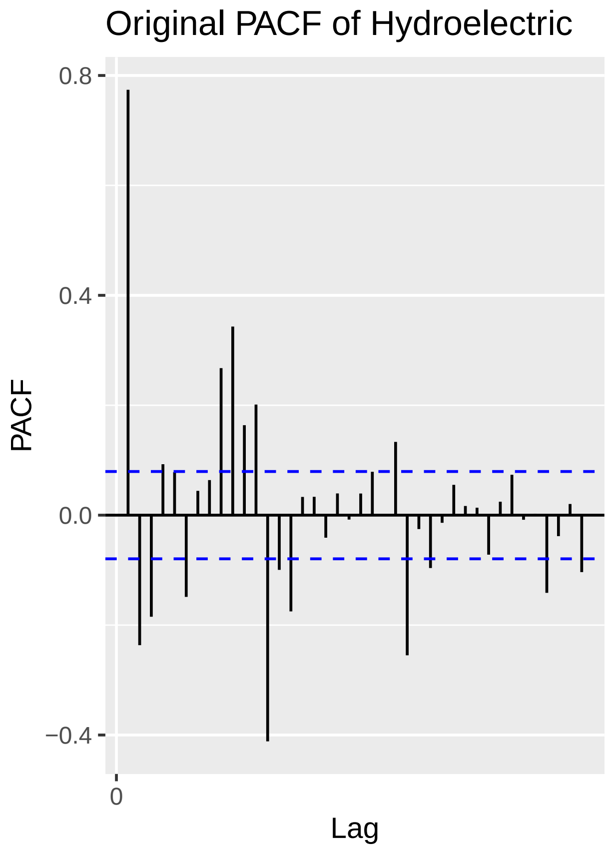


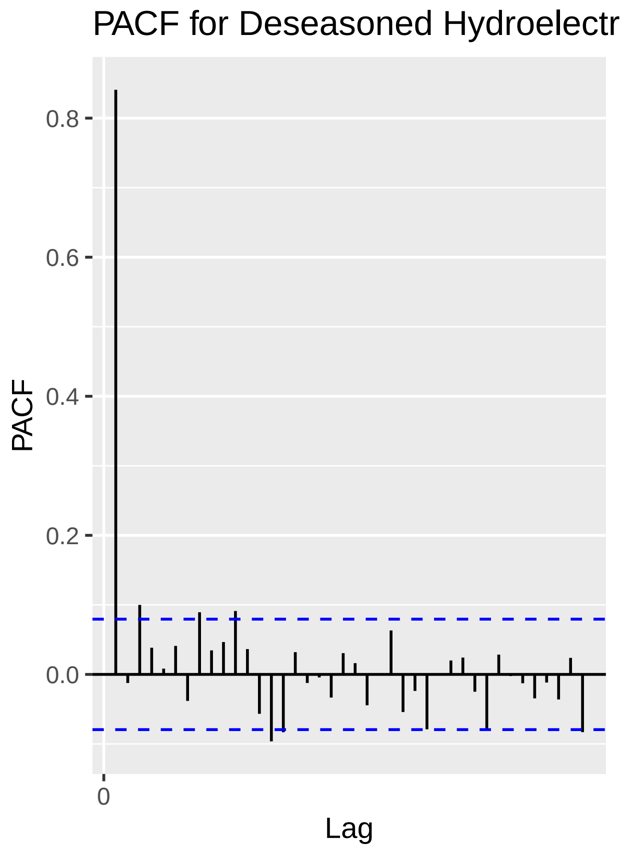



```
# comparing ACFs for deseason and original Hydroelectric data
HEC_detrend_ACF <- acf(HEC_deseason_ts,
               lag.max = 40,
               ylab = "ACF",
               plot = FALSE,
               main = "Deseasoned Hydroelectric ACF")

HEC_deseason_ACF_autoplot <- autoplot(HEC_detrend_ACF)+
  ggtitle("ACF for Deseasoned Hydroelectric")

plot_grid(HEC_ACF_autoplot, HEC_deseason_ACF_autoplot)
```




```r
# comparing PACFs for deseason and original Hydroelectric data
HEC_detrend_PACF <- pacf(HEC_deseason_ts,
                  plot = FALSE,
                  lag.max = 40,
                  ylab = 'PACF',
                  main = "PACF for Deseasoned Hydroelectric")

HEC_PACF_detrend_autoplot <- autoplot(HEC_detrend_PACF)+
  ggtitle("PACF for Deseasoned Hydroelectric")

plot_grid(HEC_PACF_autoplot, HEC_PACF_detrend_autoplot)
```





Answer: For the Renewable data, the ACF plots appear to have no change between the original and deseasoned data. Similiarly, for the PACF plots, there appears to be no change. For the Hydroelectric data, the ACF changed significantly between the original and deseasoned. The up/down wave pattern seen through the original ACF was removed from the deseasoned PACF. The deseasoned ACF shows a roughly downward linear trend. Similarly, for the PACFs, the first lag still shows to have high correlation, but the remaining lags are almost all within the dotted blue lines, showing that the deseasoning removed a lot of the magnitude and lessened the dependence on lags.