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

Assignment 3 - Due date 02/01/24

Jenn McNeill

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.\

```
library(forecast)
library(tseries)
library(Kendall)
library(lubridate)
library(ggplot2)
library(cowplot)
library(dplyr)
getwd()
```

```
## [1] "/Users/jennifermcneill/TSA_Spring2024/TSA_Spring2024"
```

##Trend Component

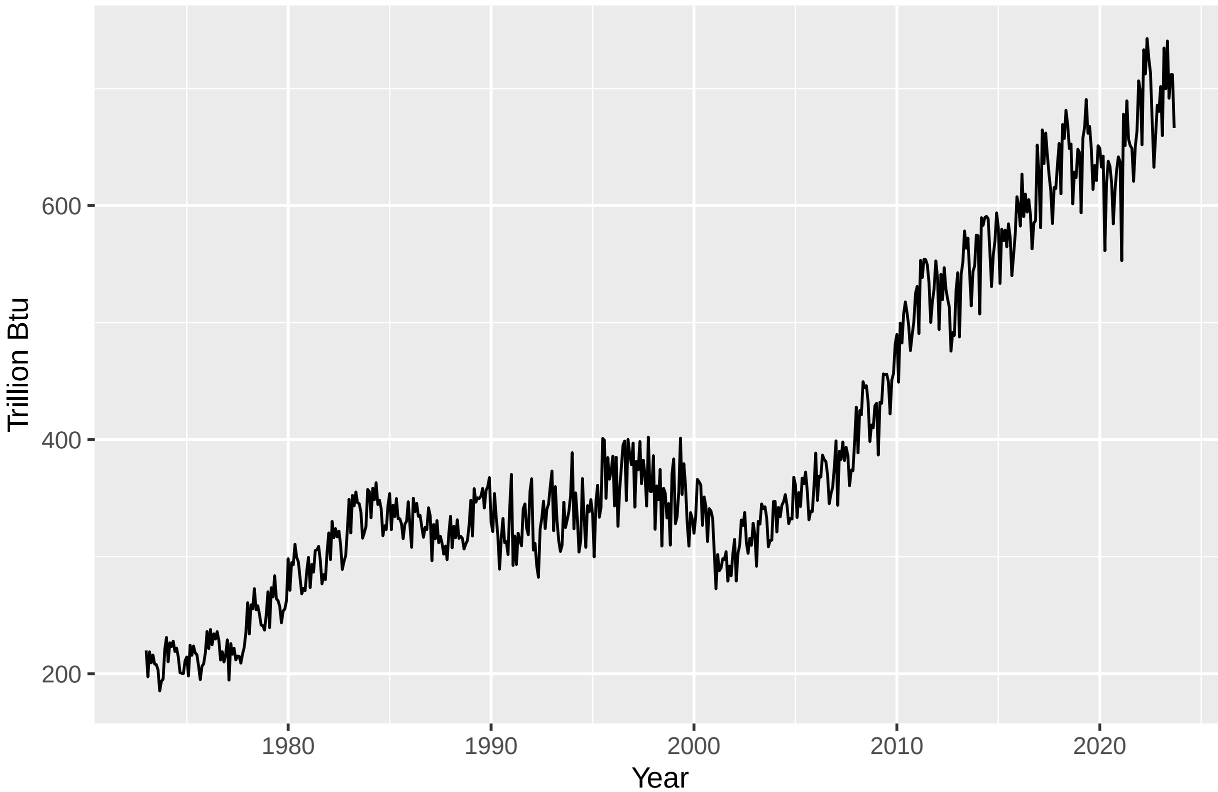
Q1

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)

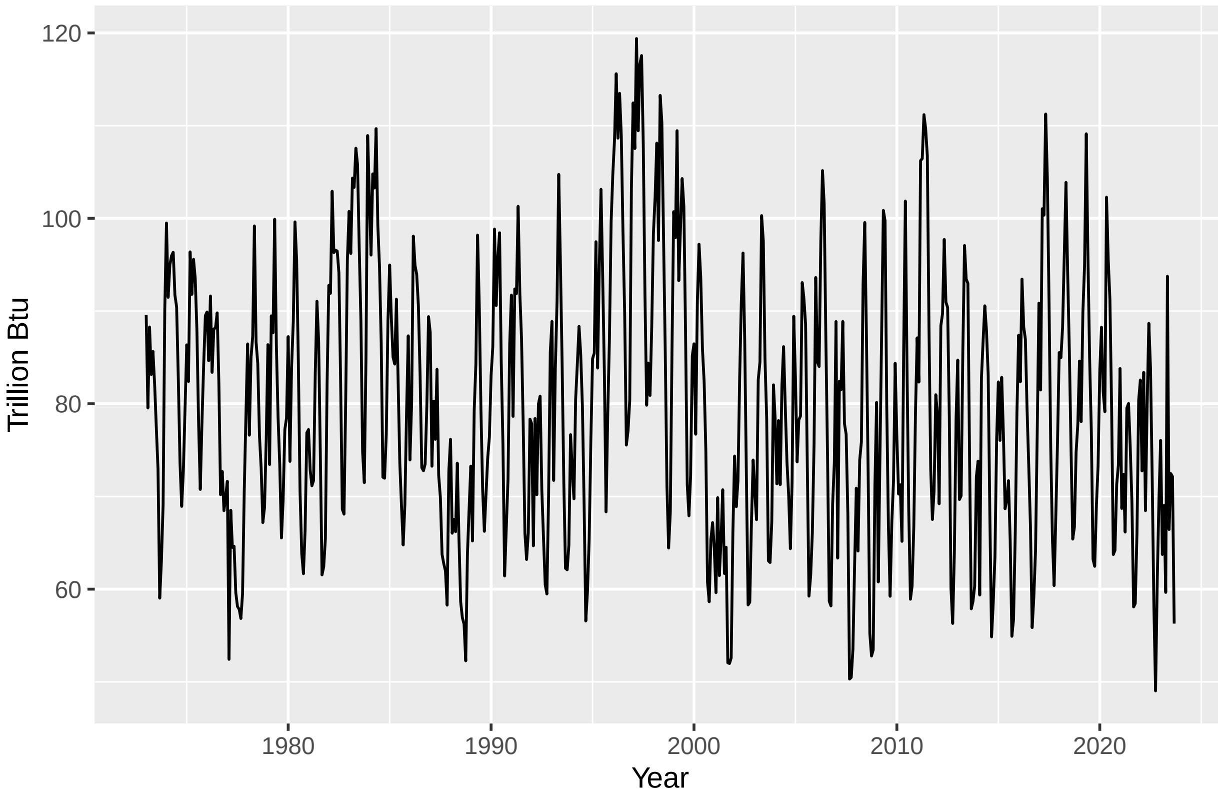
```
#Importing data set
renewable_energy_raw <- read.table(file="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_I

#Select columns of interest
renewable_energy <- select(renewable_energy_raw, Month, Total.Biomass.Energy.Production:Hydroelectric.Po

#Change the date to a date object
Date <- ym(renewable_energy$Month)

#Add the newly formatted date to a new dataframe
renewables <- cbind(Date,renewable_energy[,2:4])
head(renewables)
```

```
##         Date Total.Biomass.Energy.Production Total.Renewable.Energy.Production
## 1 1973-01-01                         129.787                           219.839
## 2 1973-02-01                         117.338                           197.330
## 3 1973-03-01                         129.938                           218.686
## 4 1973-04-01                         125.636                           209.330
## 5 1973-05-01                         129.834                           215.982
## 6 1973-06-01                         125.611                           208.249
##   Hydroelectric.Power.Consumption
## 1                          89.562
## 2                          79.544
## 3                          88.284
## 4                          83.152
## 5                          85.643
## 6                          82.060
```

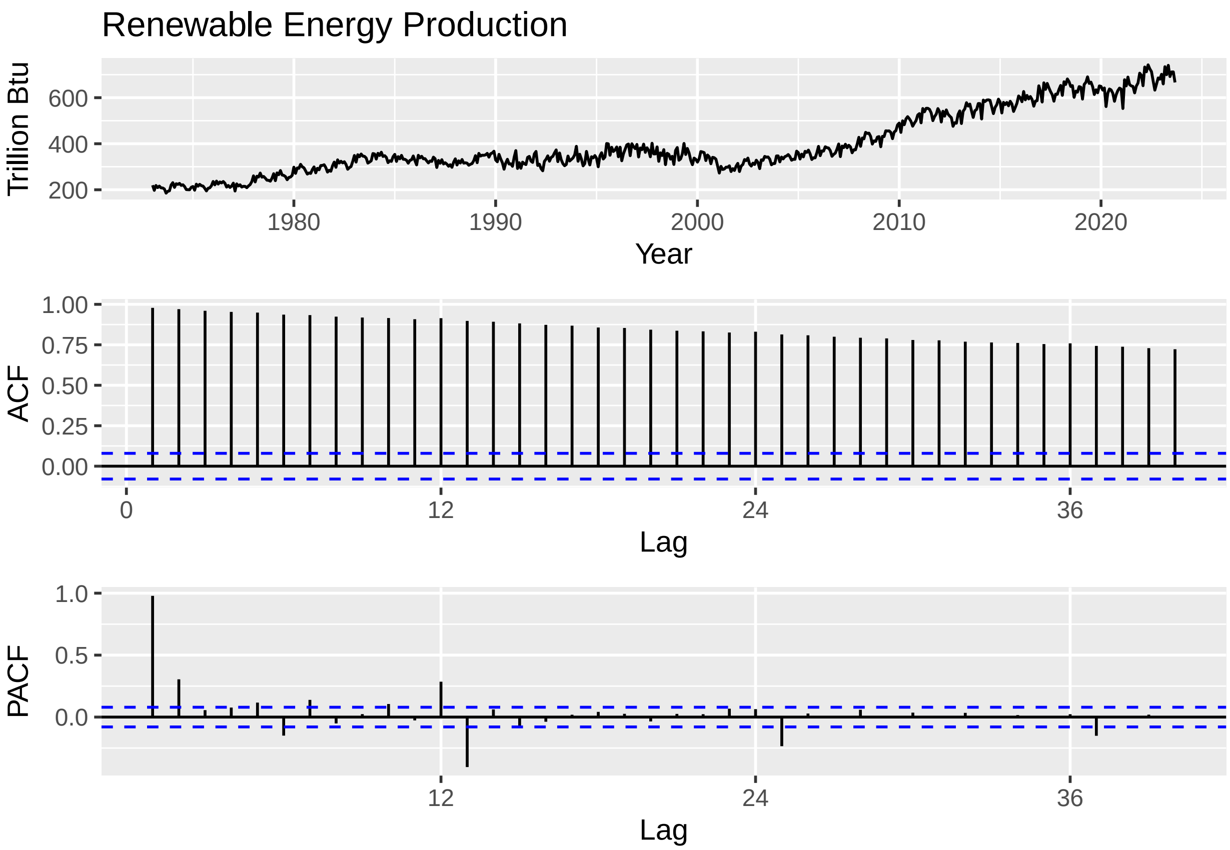
```
#Create time series objects
ts_rp <- ts(renewables[,3],start=c(1973,1), frequency=12)
ts_hc <- ts(renewables[,4],start=c(1973,1), frequency=12)

#Compute Acf for each time series
rp_acf = ggAcf(ts_rp,lag.max=40,plot=TRUE) + ggtitle(NULL)
hc_acf = ggAcf(ts_hc,lag.max=40,plot=TRUE) + ggtitle(NULL)

#Compute Pacf for each time series
rp_pacf = ggPacf(ts_rp,lag.max=40,plot=TRUE) + ggtitle(NULL)
hc_pacf = ggPacf(ts_hc,lag.max=40,plot=TRUE) + ggtitle(NULL)

#Plot the time series objects
rp_ts_plot <-
  autoplot(ts_rp)+
  xlab("Year")+
  ylab("Trillion Btu")
rp_ts_plot
```



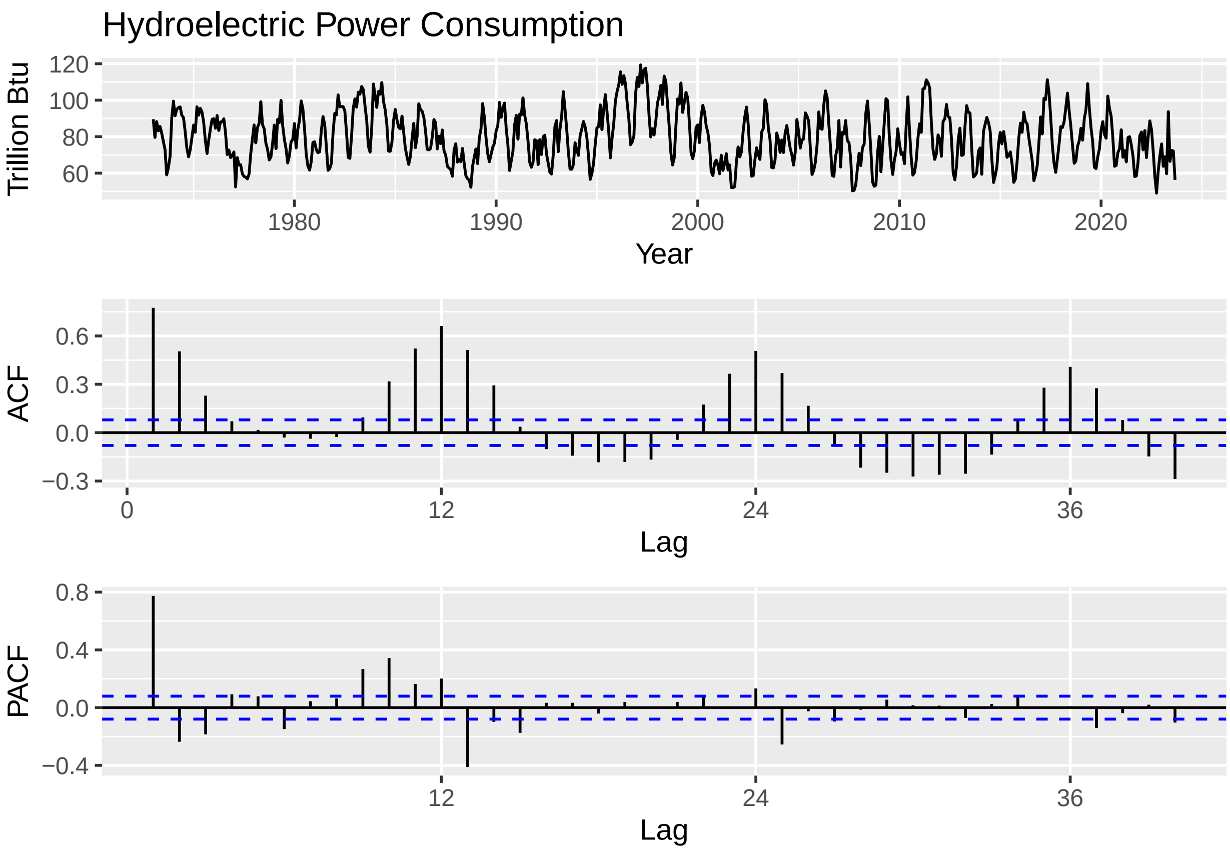

```
hc_ts_plot <-
  autoplot(ts_hc)+
  xlab("Year")+
  ylab("Trillion Btu")
hc_ts_plot
```




```r
#Plot all three plots side by side
plot_grid(rp_ts_plot + ggtitle("Renewable Energy Production"),
          rp_acf,
          rp_pacf,
          nrow = 3, align = "v")
```




```r
plot_grid(hc_ts_plot + ggtitle("Hydroelectric Power Consumption"),
          hc_acf,
          hc_pacf,
          nrow = 3, align = "v")
```



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 Production series appears to have a steady upward trend over the years from 1973 to 2024. The Hydroelectric Power Consumption series does not appear to trend upwards nor downwards.

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.

```r
#Create variables to use for indexing
nobs <- nrow(renewables)
t <- 1:nobs

#Fit a linear trend to the renewable production time series
rp_lm <- lm(renewables[,3]~t)
summary(rp_lm)
```

```
##
## Call:
## lm(formula = renewables[, 3] ~ 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
#Store slope and intercept coefficients
rp_beta0 <- rp_lm$coefficients[1]
rp_beta1 <- rp_lm$coefficients[2]

#Fit a linear trend to the hydroelectric consumption time series
hc_lm <- lm(renewables[,4]~t)
summary(hc_lm)
```

```
##
## Call:
## lm(formula = renewables[, 4] ~ 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
#Store slope and intercept coefficients
hc_beta0 <- hc_lm$coefficients[1]
hc_beta1 <- hc_lm$coefficients[2]

rp_beta0
```

```
## (Intercept)
##    180.9894
```

```r
rp_beta1
```

```
##         t
## 0.7040391
```

```r
hc_beta0
```

```
## (Intercept)
##    82.73475
```

```r
hc_beta1
```

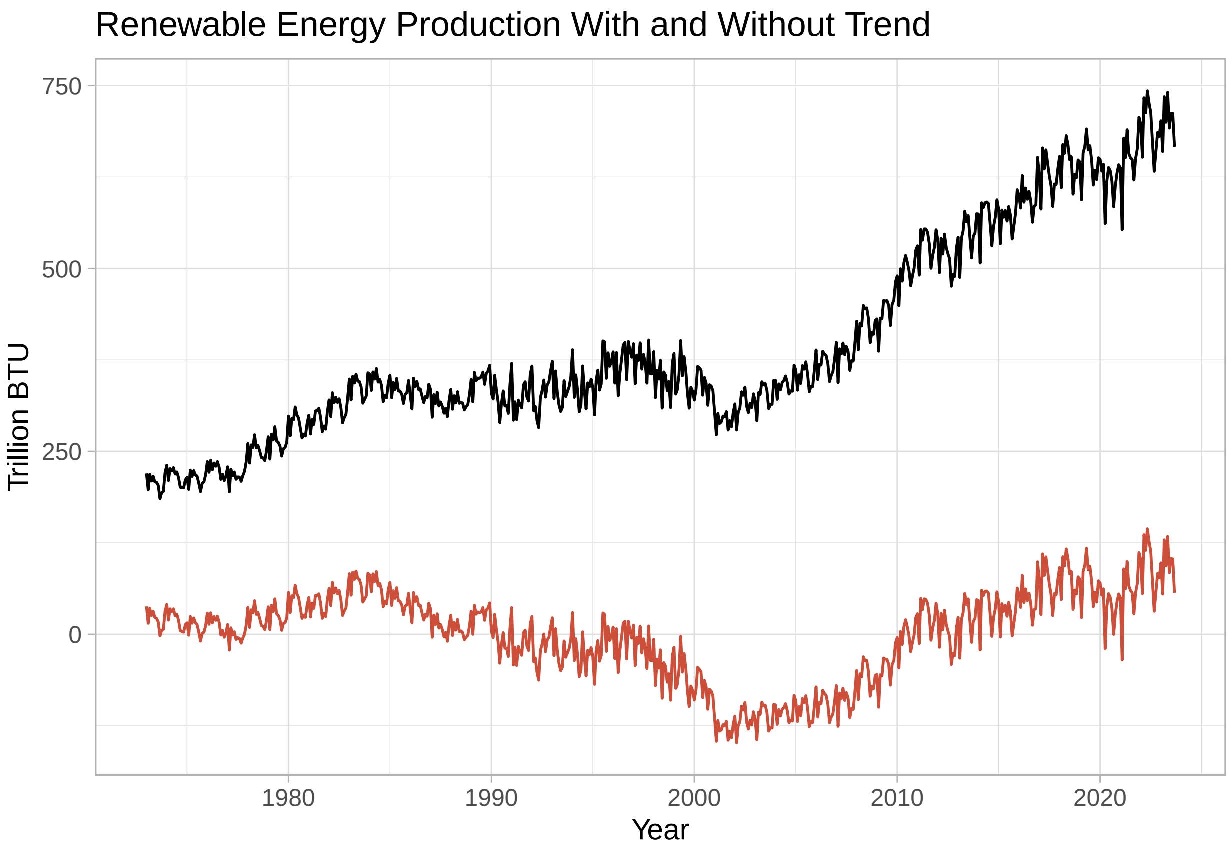
```
##            t
## -0.009849298
```

Beta0 is ~181 for renewable production, which means that the linear regression model starts at this point at the y-axis. Beta1 for renewable production being positive shows that production is increasing over time. The R^2 value of 0.81 means that 81% of the variability is explained by time. Beta0 is ~83 for hydroelectric consumption, which means that the linear regression model starts at this point at the y-axis. Beta1 for hydroelectric consumption being negative/practically zero shows that the consumption is not trending upwards or downwards in a significant direction. The R^2 value of 0.01 means that 1% of the variability is explained by time.

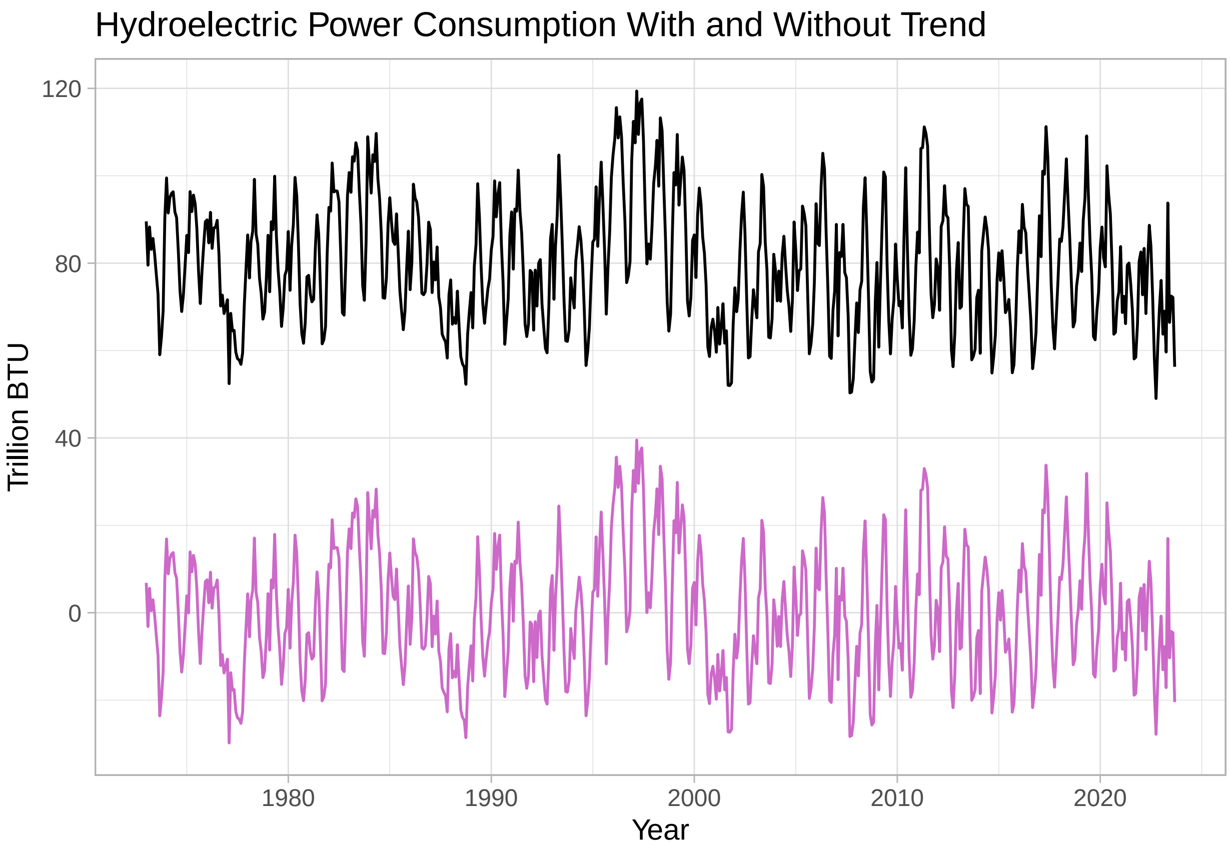
Q4

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
#Store the detrended renewable production series
rp_detrend <- renewables[,3] - (rp_beta0 + (rp_beta1*t))

#Plot
ggplot(renewables, aes(x=Date))+
  geom_line(aes(y=renewables[,3]), color = "black")+
  geom_line(aes(y=rp_detrend), color = "tomato3")+
  ggtitle("Renewable Energy Production With and Without Trend")+
  xlab("Year")+
  ylab("Trillion BTU")+
  theme_light()
```




```r
#Store the detrended hydroelectric production series
hc_detrend <- renewables[,4] - (hc_beta0 + (hc_beta1*t))

#Plot
ggplot(renewables, aes(x=Date))+
  geom_line(aes(y=renewables[,4]), color = "black")+
  geom_line(aes(y=hc_detrend), color = "orchid3")+
  ggtitle("Hydroelectric Power Consumption With and Without Trend")+
  xlab("Year")+
  ylab("Trillion BTU")+
  theme_light()
```

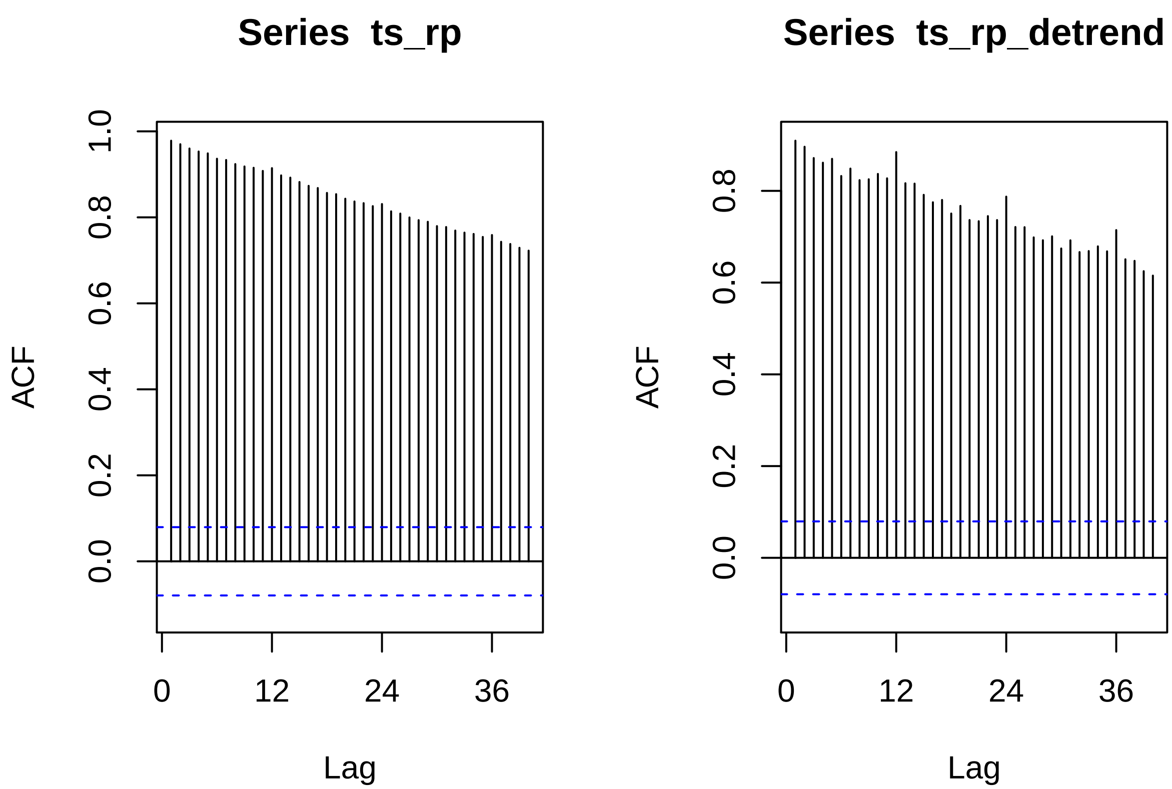


When you detrend both of these series, the production/consumption values in trillion btu decrease significantly. In the renewable energy production graph, we are removing the "upward" trend from the data, so we see the plot flatten out instead of steadily increasing upwards. In the hydroelectric power consumption graph, which did not show an upward or downward trend to begin with, we see that the shape of the graph stays the same. Since there was not a clear trend to remove, the plot just shifts downwards.

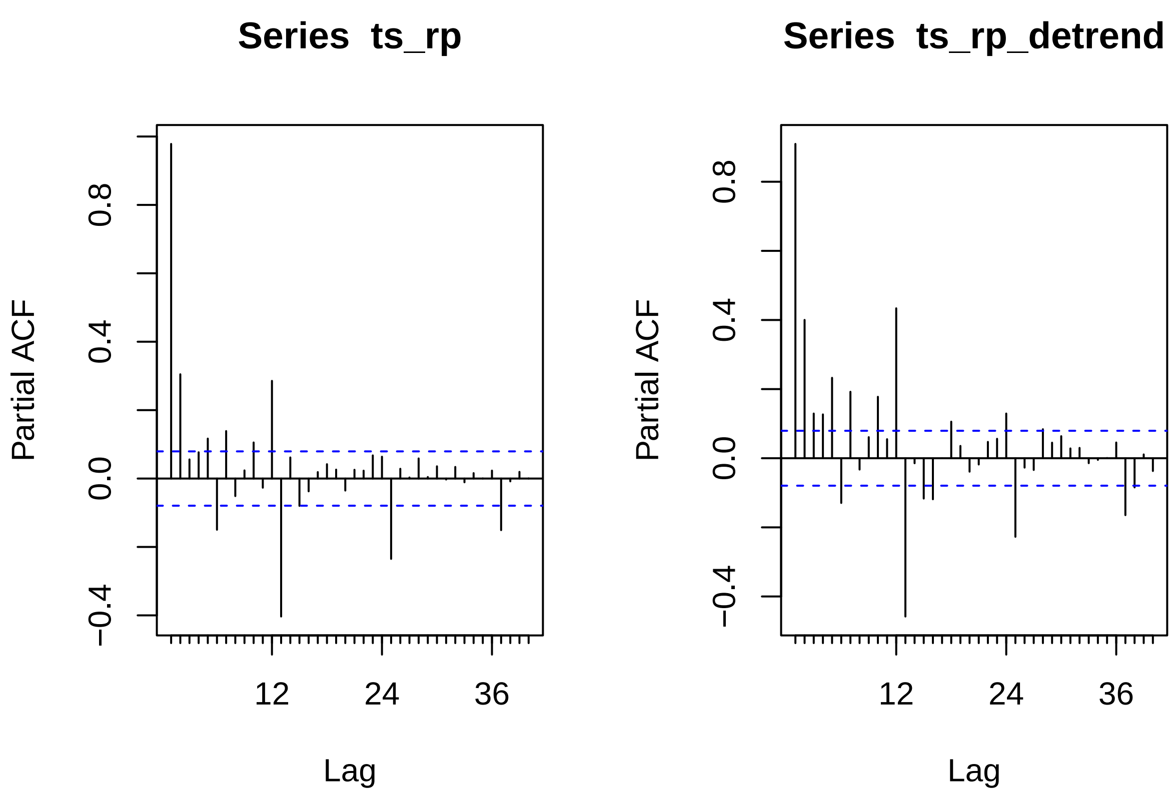
Q5

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?

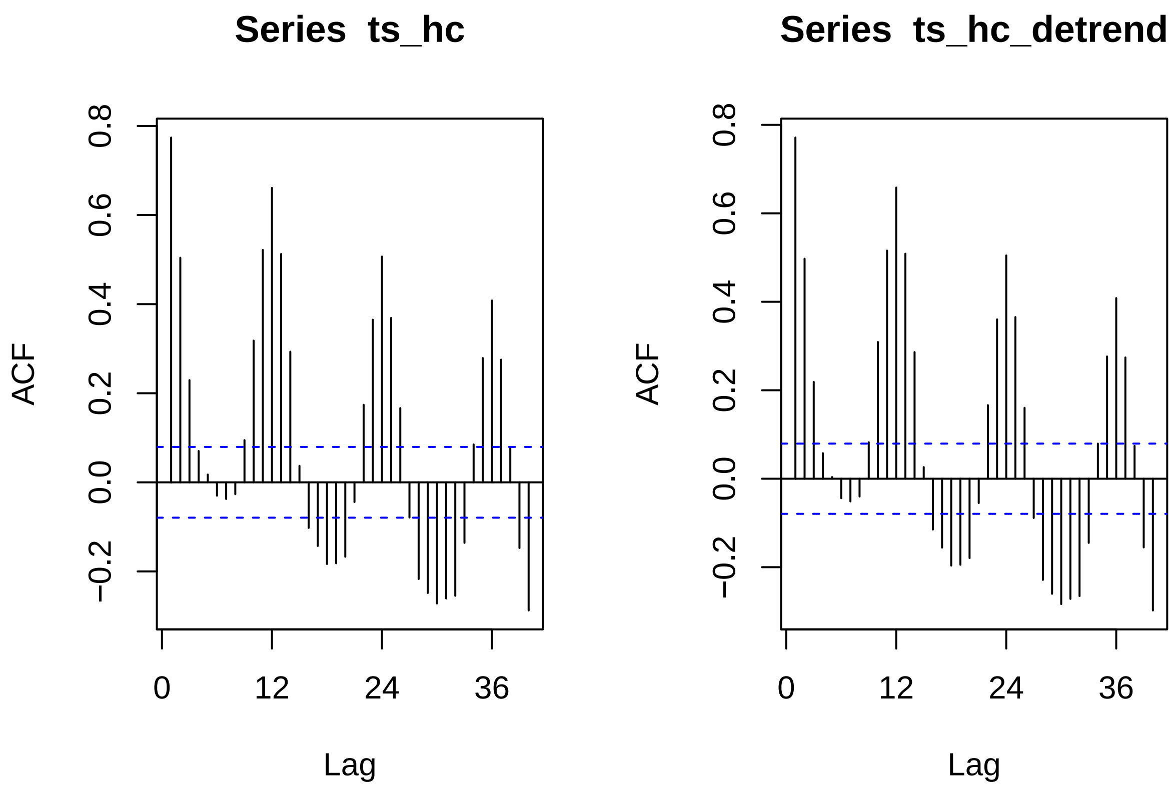
```r
#Make the detrended data into a time series object
ts_rp_detrend <- ts(rp_detrend, start=c(1973,1), frequency=12)
ts_hc_detrend <- ts(hc_detrend, start=c(1973,1), frequency=12)

#Compare the Acf for renewable production with and without trend
par(mfrow=c(1,2))
  Acf(ts_rp,lag.max=40,plot=TRUE)
  Acf(ts_rp_detrend,lag.max=40,plot=TRUE)
```



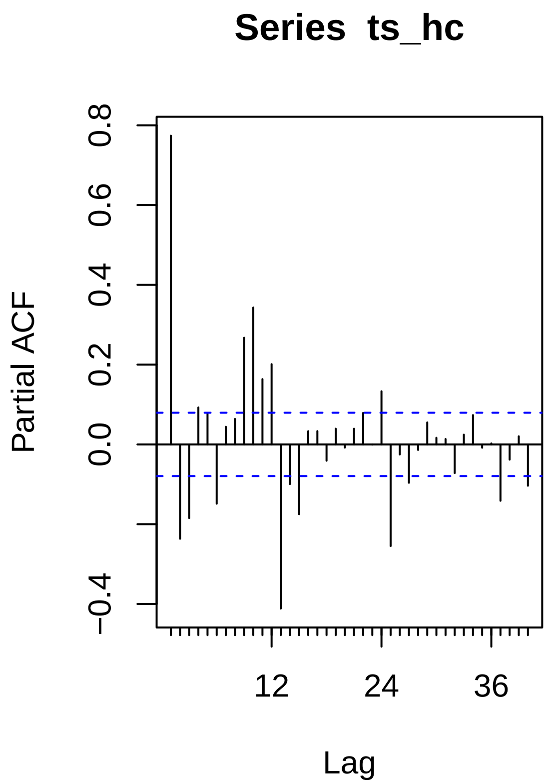

```r
#Compare the Pacf for renewable production with and without trend
par(mfrow=c(1,2))
  Pacf(ts_rp,lag.max=40,plot=TRUE)
  Pacf(ts_rp_detrend,lag.max=40,plot=TRUE)
```

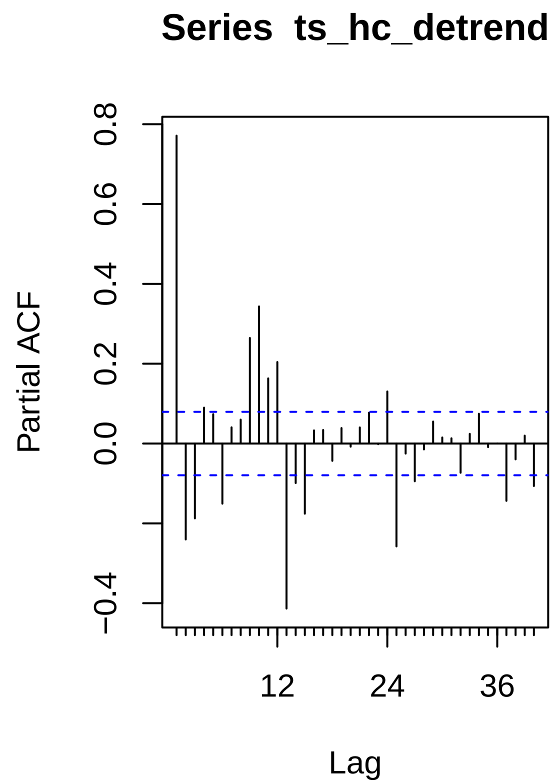


```r
#Compare the Acf for hydroelectric consumption with and without trend
par(mfrow=c(1,2))
  Acf(ts_hc,lag.max=40,plot=TRUE)
  Acf(ts_hc_detrend,lag.max=40,plot=TRUE)
```




```r
#Compare the Pacf for hydroelectric consumption with and without trend
par(mfrow=c(1,2))
  Pacf(ts_hc,lag.max=40,plot=TRUE)
  Pacf(ts_hc_detrend,lag.max=40,plot=TRUE)
```





The ACF and PACF for the renewable energy production series changed a little bit when detrended. When the trend was removed from this dataset, the ACF and PACF show stronger correlations at lags 1, 13, 25, 37 as they did with the trended time series. Without the upward trend, it is easier to correlate data from future years back to the prior years, which makes sense. Since the hydroelectric power consumption data never showed much of a trend to begin with, detrending the time series did not change either its ACF nor its PACF.

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 renewable energy production time series does not appear to have a significant seasonal trend, and the hydroelectric power consumption time series displays dramatic seasonal peaks and falls. Assuming hydroelectric power consumption relies on seasonal factors, like how much power is needed for heating/cooling of homes, for example, the seasonality of this time series data set makes sense to me.

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
#Start with renewable production
#Create dummies
dummies_rp <- seasonaldummy(ts_rp)

#Regress on dummies
ts_rp_lm <- lm(renewables[,3]~dummies_rp)
summary(ts_rp_lm)
```

```
##
## Call:
## lm(formula = renewables[, 3] ~ dummies_rp)
##
## 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 ***
## dummies_rpJan    2.962     27.546   0.108    0.914
## dummies_rpFeb  -34.476     27.546  -1.252    0.211
## dummies_rpMar    3.929     27.546   0.143    0.887
## dummies_rpApr   -8.695     27.546  -0.316    0.752
## dummies_rpMay    6.645     27.546   0.241    0.809
## dummies_rpJun   -4.198     27.546  -0.152    0.879
## dummies_rpJul    2.460     27.546   0.089    0.929
## dummies_rpAug   -5.026     27.546  -0.182    0.855
## dummies_rpSep  -29.119     27.546  -1.057    0.291
## dummies_rpOct  -20.068     27.682  -0.725    0.469
## dummies_rpNov  -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
#Continue with hydroelectric consumption
#Create dummies
dummies_hc <- seasonaldummy(ts_hc)

#Regress on dummies
ts_hc_lm <- lm(renewables[,4]~dummies_hc)
summary(ts_hc_lm)
```

```
##
## Call:
## lm(formula = renewables[, 4] ~ dummies_hc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.323  -5.849  -0.468   6.243  32.290
##
```

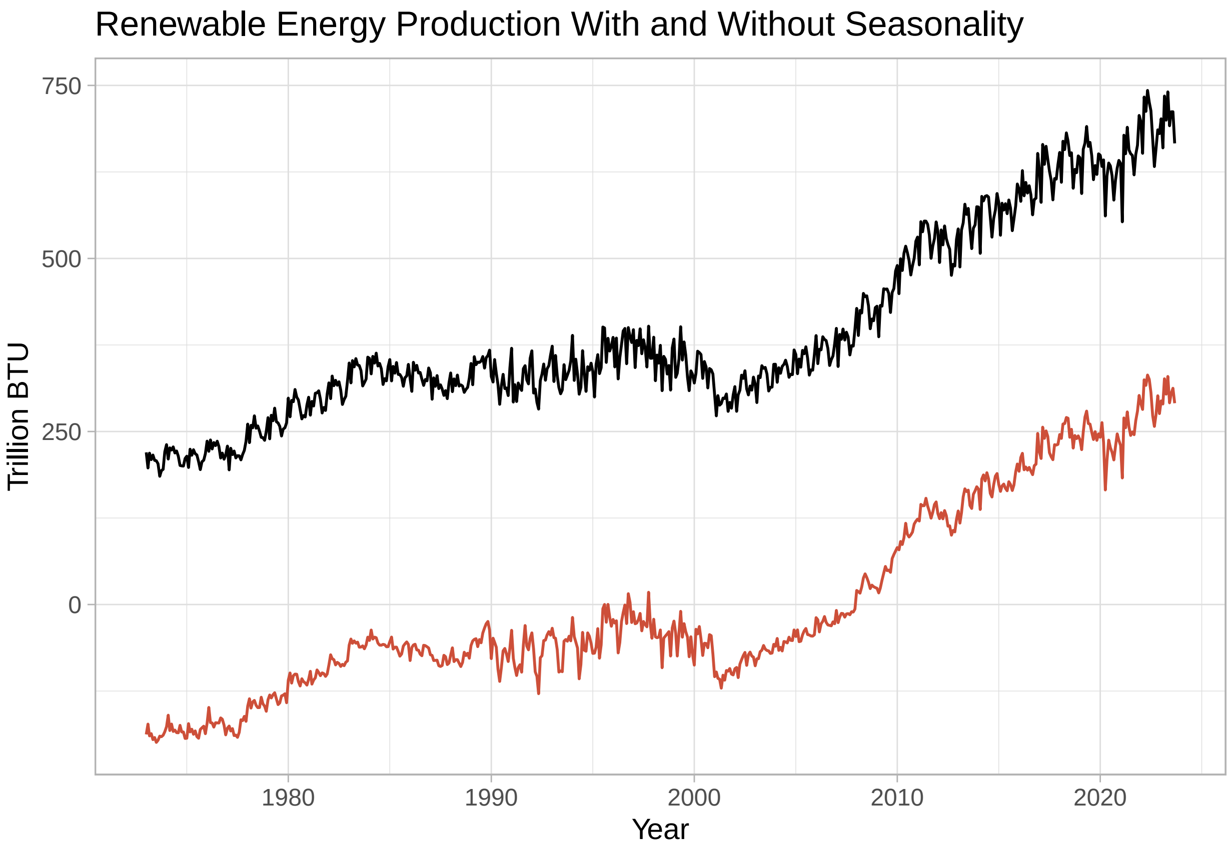
```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)     80.282      1.470  54.601  < 2e-16 ***
## dummies_hcJan    4.807      2.069   2.323  0.02050 *  
## dummies_hcFeb   -2.725      2.069  -1.317  0.18831    
## dummies_hcMar    6.825      2.069   3.298  0.00103 ** 
## dummies_hcApr    5.319      2.069   2.571  0.01039 *  
## dummies_hcMay   13.922      2.069   6.729 4.02e-11 ***
## dummies_hcJun   10.650      2.069   5.147 3.60e-07 ***
## dummies_hcJul    3.912      2.069   1.891  0.05914 .  
## dummies_hcAug   -5.677      2.069  -2.744  0.00626 ** 
## dummies_hcSep  -16.797      2.069  -8.118 2.72e-15 ***
## dummies_hcOct  -16.468      2.079  -7.920 1.17e-14 ***
## dummies_hcNov  -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
```

According to the linear regressions on the dummy matrix, the renewable energy production's linear regression R^2 value of -0.009 means that only -0.9% of renewable energy production is explained by seasonality. On the other hand, the hydroelectric power consumption's linear regression R^2 value of 0.459 means that 46% of hydroelectric power consumption is explained by seasonality. These values align with my prediction that seasonality played a role only in hydroelectric power consumption.

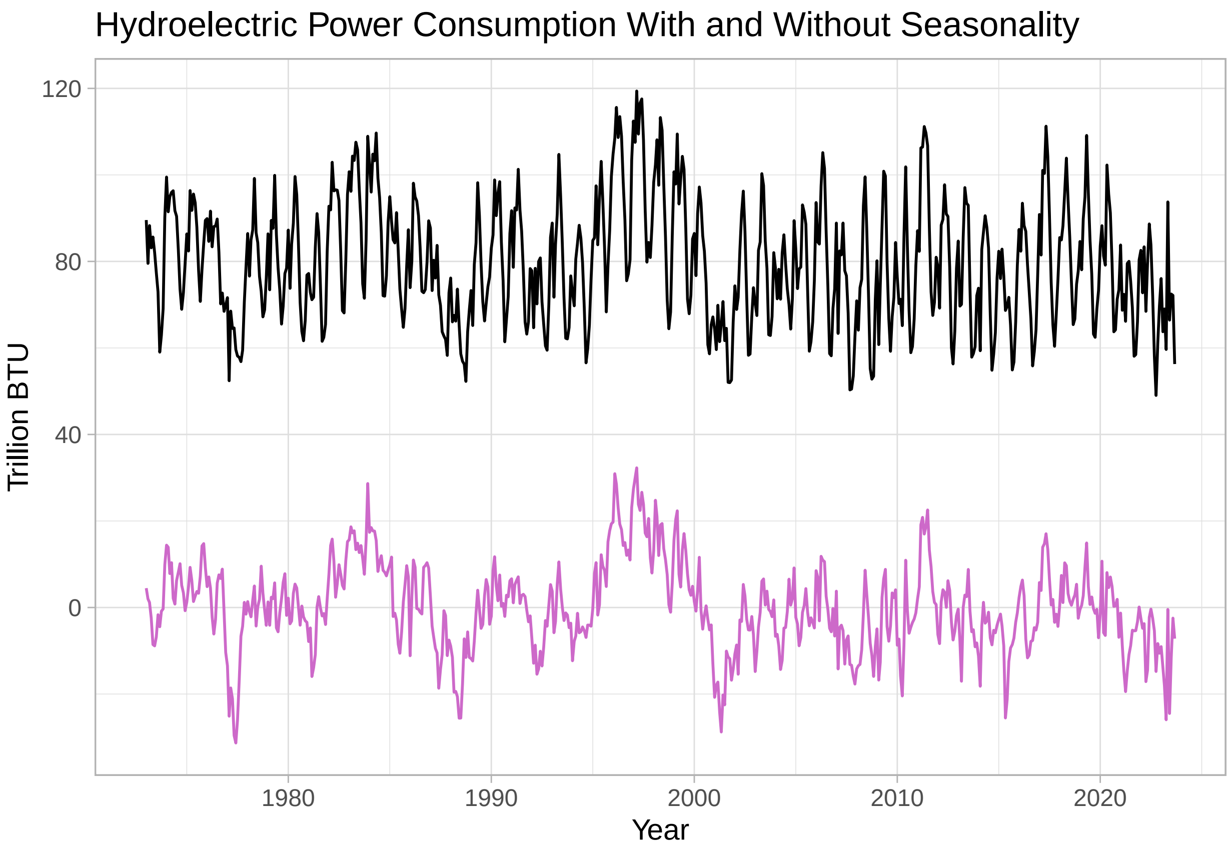
Q8

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
#Start with renewable production
#Store coefficients
beta0_rp <- ts_rp_lm$coefficients[1]
beta1_rp <- ts_rp_lm$coefficients[2:12]

#Seasonal component
rp_seasonal_component <- array(0,nobs)
for(i in 1:nobs){
  rp_seasonal_component[i] <- beta0_rp + beta1_rp %*% dummies_rp[i,]
}

#Remove seasonal component
rp_deseason <- renewables[,3] - rp_seasonal_component

#Plot original time series and deseasoned time series
ggplot(renewables, aes(x=Date))+
  geom_line(aes(y=renewables[,3]), color = "black")+
  geom_line(aes(y=rp_deseason), color = "tomato3")+
  ggtitle("Renewable Energy Production With and Without Seasonality")+
  xlab("Year")+
  ylab("Trillion BTU")+
  theme_light()
```




```r
#Continue with hydroelectric consumption
#Store coefficients
beta0_hc <- ts_hc_lm$coefficients[1]
beta1_hc <- ts_hc_lm$coefficients[2:12]

#Seasonal component
hc_seasonal_component <- array(0,nobs)
for(i in 1:nobs){
  hc_seasonal_component[i] <- beta0_hc + beta1_hc %*% dummies_hc[i,]
}

#Remove seasonal component
hc_deseason <- renewables[,4] - hc_seasonal_component

#Plot original time series and deseasoned time series
ggplot(renewables, aes(x=Date))+
  geom_line(aes(y=renewables[,4]), color = "black")+
  geom_line(aes(y=hc_deseason), color = "orchid3")+
  ggtitle("Hydroelectric Power Consumption With and Without Seasonality")+
  xlab("Year")+
  ylab("Trillion BTU")+
  theme_light()
```

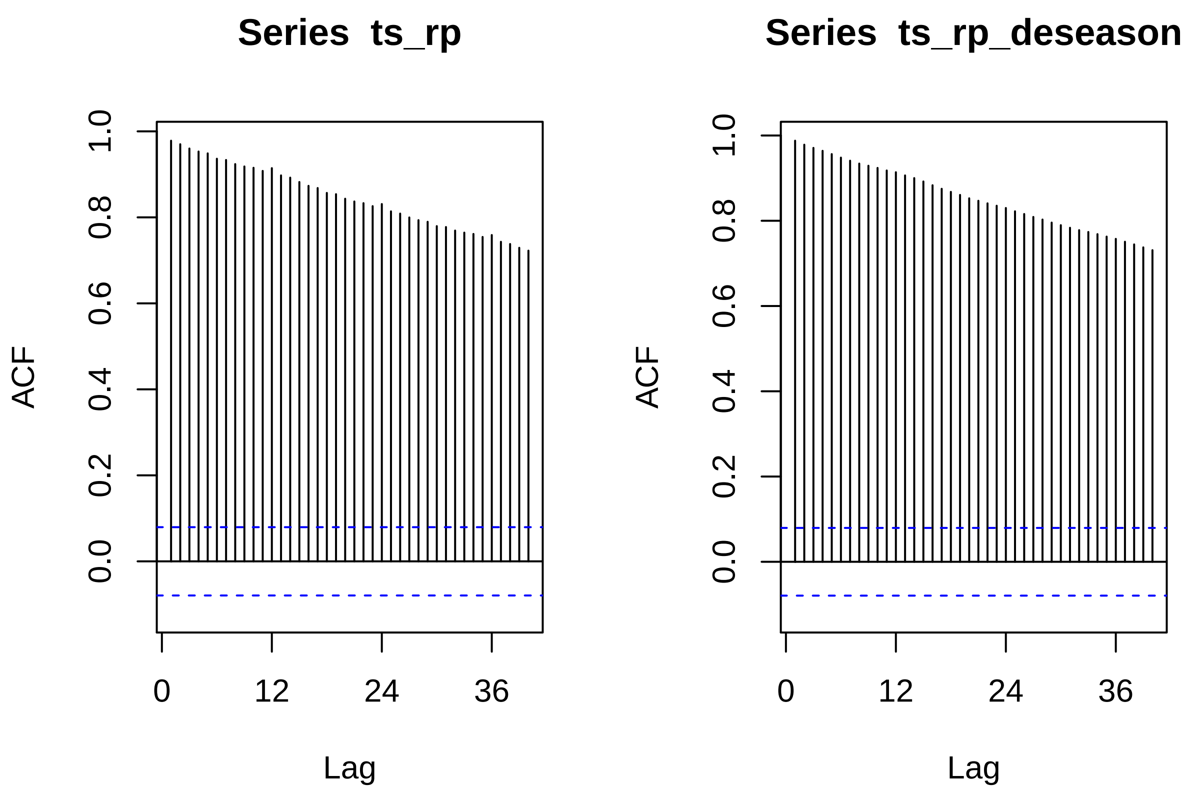


When seasonality is removed, the plot for renewable energy production shifts downward and the peaks and falls decrease slightly in amplitude. The hydroelectric power consumption plot displays a much more significant change when seasonality is removed because the dramatic rises and falls of the time series data decrease significantly in amplitude, which causes the new plot to take on a new shape.

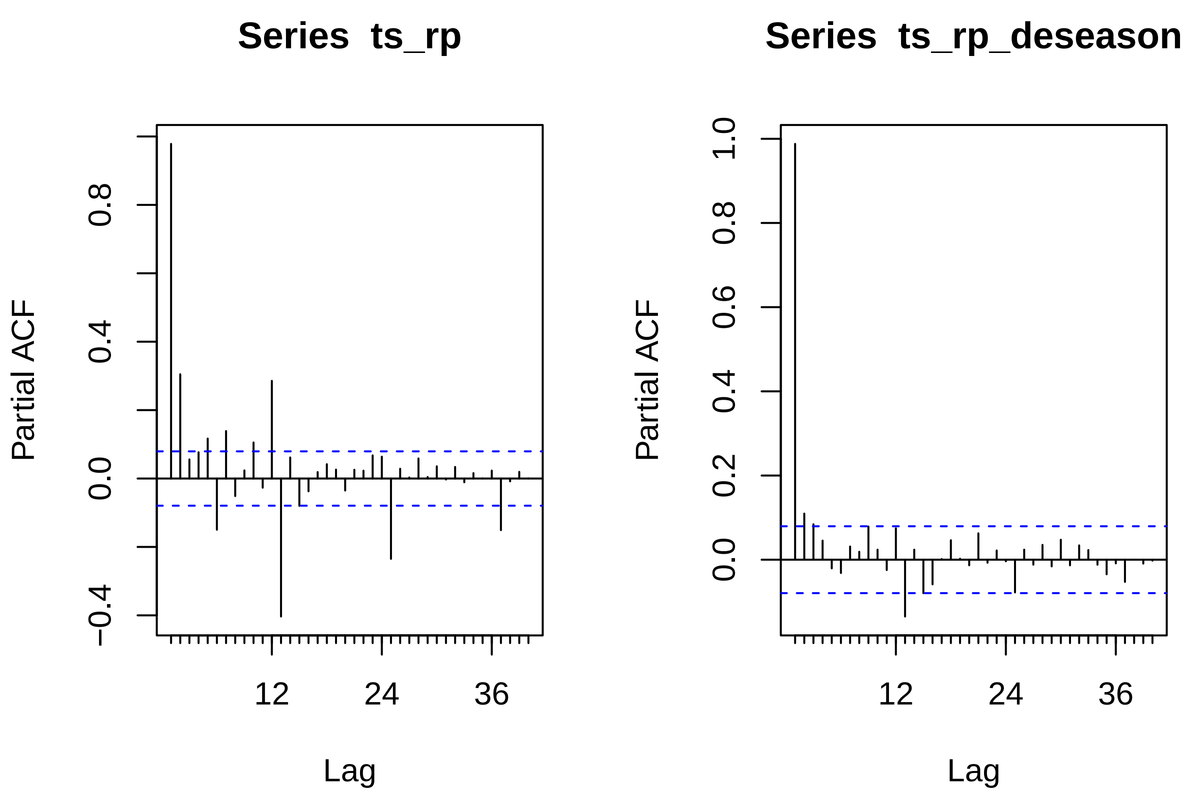
Q9

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?

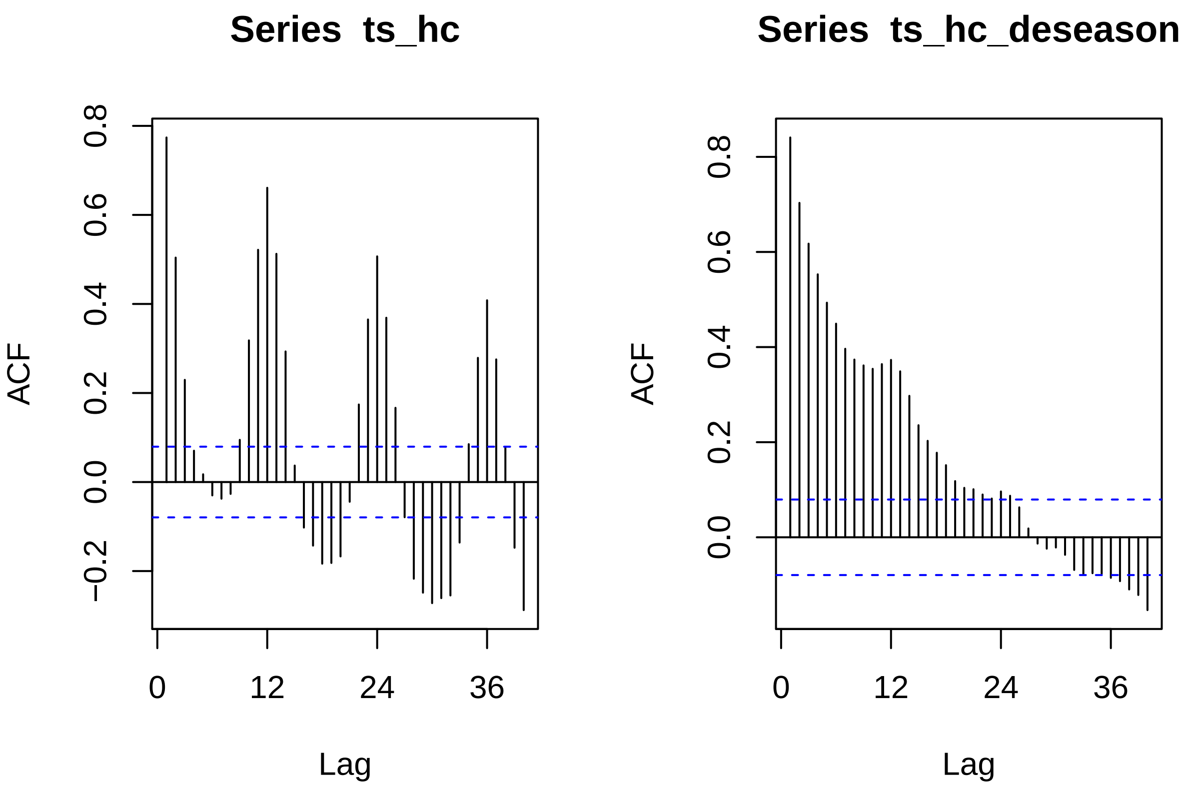
```
#Make the deseasoned data into a time series object
ts_rp_deseason <- ts(rp_deseason, start=c(1973,1), frequency=12)
ts_hc_deseason <- ts(hc_deseason, start=c(1973,1), frequency=12)

#Compare the Acf for renewable production with and without seasonality
par(mfrow=c(1,2))
  Acf(ts_rp,lag.max=40,plot=TRUE)
  Acf(ts_rp_deseason,lag.max=40,plot=TRUE)
```



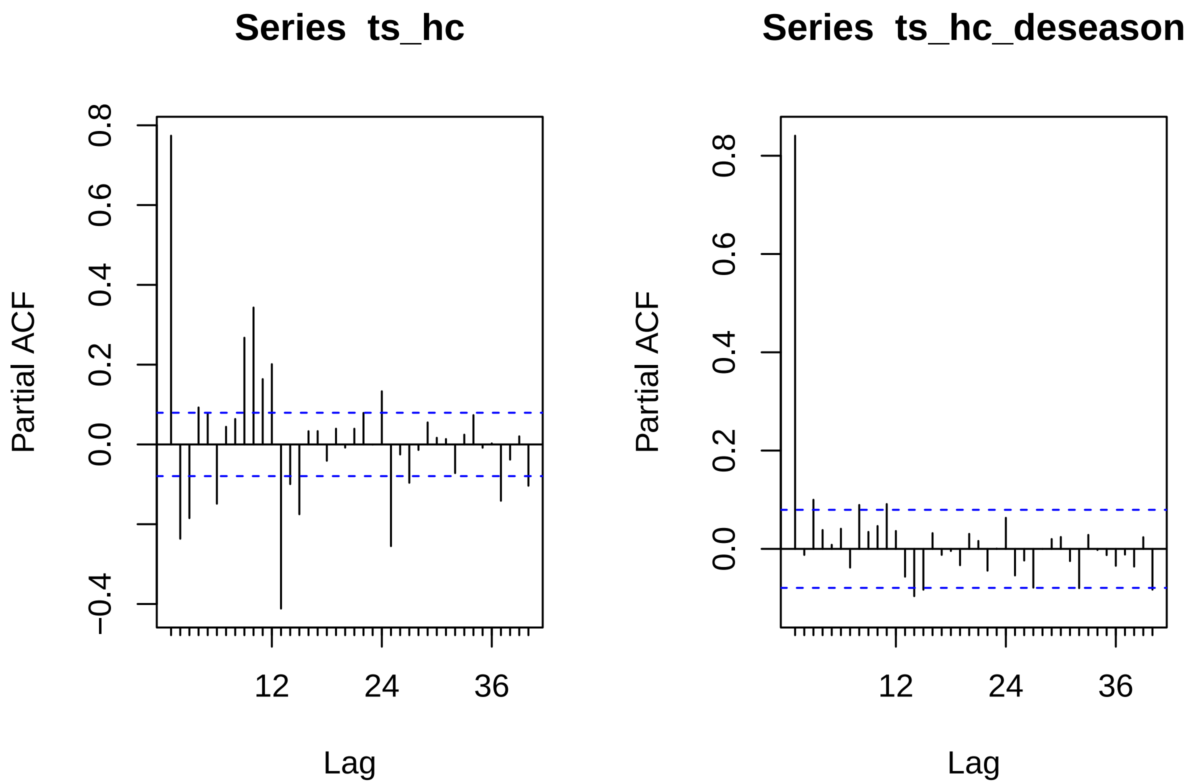

```r
#Compare the Pacf for renewable production with and without seasonality
par(mfrow=c(1,2))
  Pacf(ts_rp,lag.max=40,plot=TRUE)
  Pacf(ts_rp_deseason,lag.max=40,plot=TRUE)
```



```
#Compare the Acf for hydroelectric consumption with and without seasonality
par(mfrow=c(1,2))
  Acf(ts_hc,lag.max=40,plot=TRUE)
  Acf(ts_hc_deseason,lag.max=40,plot=TRUE)
```




```
#Compare the Pacf for hydroelectric consumption with and without seasonality
par(mfrow=c(1,2))
  Pacf(ts_hc,lag.max=40,plot=TRUE)
  Pacf(ts_hc_deseason,lag.max=40,plot=TRUE)
```



The ACF plots for renewable production did not change. The PACF plot for renewable production show slightly less significant partial autocorrelation for lags 13, 25, and 27 after removing seasonality. The ACF plot for hydroelectric consumption completely changes shape when the seasonality is removed, which shows that the data can no longer be strongly correlated based on a seasonal lag. The PACF likewise shows less significant partial autocorrelation for lags 13, 25, and 37 after removing seasonality.