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

Assignment 3 - Due date 02/04/25

Mazhar Bhuyan

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_Consumpti
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.\

```
library(forecast)
```

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

```
library(tseries)
library(Kendall)
install.packages("cowplot")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(cowplot)
library(readxl)
library(ggplot2)
install.packages("purrr")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(purrr)
install.packages("stargazer")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
getwd()
```

```
## [1] "/home/guest/TSA_Mazhar/Time_Series_Mazhar"
```

```
#Importing Dataset
Energy_Data <- read_excel("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx
  skip = 12,
  sheet = "Monthly Data",
  col_names = FALSE)
head(Energy_Data)

col_names <- read_excel("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
      skip=10,
      n_max = 1,
      sheet="Monthly Data",
      col_names = FALSE)
head(col_names)

Energy_Data_ts <- ts(Energy_Data[, 2:14], start = c(1973, 1), frequency = 12)
head(Energy_Data_ts)
```

##Trend Component

Q1

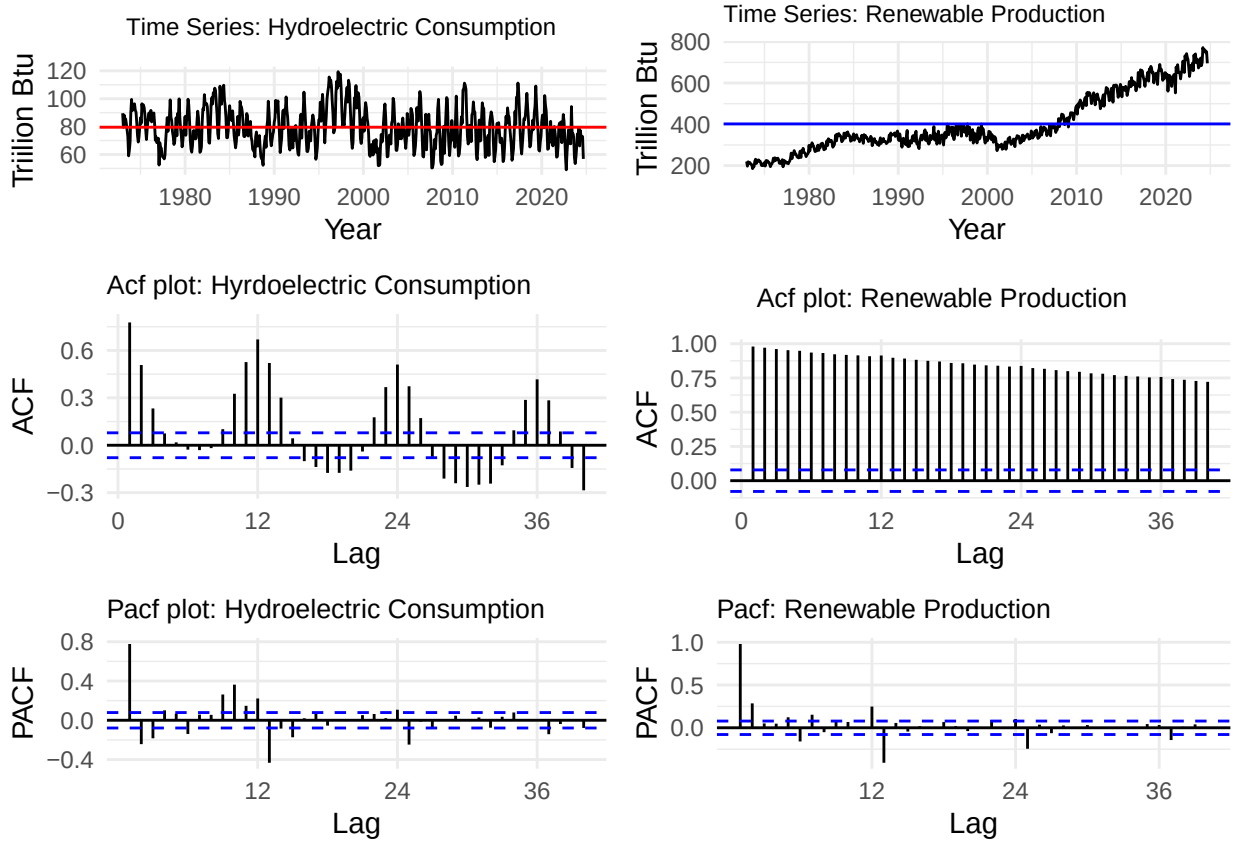
For each time series, i.e., Renewable Energy Production and Hydroelectric Consumption create three plots: one with time series, one with the ACF and with the PACF. You may use the some code form A2, but I want all the three plots side by side as in a grid. (Hint: use function `plot_grid()` from the `cowplot`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

```
Mean_value_Hydro <- mean(Energy_Data_ts[, 5]) #Means for stationarity
print(Mean_value_Hydro)
```

```
## [1] 79.55371
```

```
ts_hydro <- autoplot(Energy_Data_ts[, 5])+
  ggtitle(paste("Time Series:" , "Hydroelectric Consumption")) +
  xlab("Year") +
  ylab("Triilion Btu") +
  geom_hline(yintercept = Mean_value_Hydro,
             color = "red",
             linetype = "solid") +
  theme_minimal()
#print(ts_hydro)

Mean_value_renewable <- mean(Energy_Data_ts[, 4])
#print(Mean_value_renewable)

ts_renewables <- autoplot(Energy_Data_ts[, 4]) +
  ggtitle(paste("Time Series:" , "Renewable Production")) +
  xlab("Year") +
  ylab("Trillion Btu") +
  geom_hline(yintercept = Mean_value_renewable,
             color = "blue",
             linetype = "solid") +
  theme_minimal()
#print(ts_renewables)

acf_hydro <- ggAcf(Energy_Data_ts[, 5], lag.max = 40) +
  ggtitle("Acf plot: Hyrdoelectric Consumption") +
  theme_minimal()
#print(acf_hydro)

acf_renewables <- ggAcf(Energy_Data_ts[, 4], lag.max = 40) +
  ggtitle("Acf plot: Renewable Production") +
  theme_minimal()
#print(acf_renewables)

pacf_hydro <- ggPacf(Energy_Data_ts[, 5], lag.max = 40) +
  ggtitle("Pacf plot: Hydroelectric Consumption") +
  theme_minimal()
#print(pacf_hydro)

pacf_renewables <- ggPacf(Energy_Data_ts[, 4], lag.max = 40) +
  ggtitle("Pacf: Renewable Production") +
  theme_minimal()
#print(pacf_renewables)
```

```
##plot_grid(ts_hydro, ts_renewables, acf_hydro,
      #acf_renewables, pacf_hydro, pacf_renewables, nrow = 3, align = "v")
#plot overlaps.


plot_grid(ts_hydro +
          theme(plot.title = element_text(size = 9,
                                          margin = margin(5, 5, 10, 10))),
          ts_renewables +
          theme(plot.title = element_text(size = 9)),
          acf_hydro +
          theme(plot.title = element_text(size = 10)),
          acf_renewables +
          theme(plot.title = element_text(size = 10,
                                          margin = margin(5, 5, 10, 10))),
          pacf_hydro +
          theme(plot.title = element_text(size = 10)),
          pacf_renewables +
          theme(plot.title = element_text(size = 10)),
                                            nrow = 3,
                                            rel_heights = c(1, 1.2))
```



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?

Yes. Some of the plots show trends while others does not.

Hydroelectric energy consumption: TS The plot does not show any trend over time. It looks like a stationary plot.

Renewable energy production: TS The plot shows an upward trend. It does not fluctuate over a constant mean.

Hydroelectric energy consumption: Acf The Acf decays over time. There is strong seasonality as we can observe that after 40 lags the seasonality remains.

Renewable Energy Production: Acf The Acf decays over time. The plot is also significant after 40 lags. It indicates that renewable energy production is higly dependent on the previous values.

Hydroelectric Energy Consumption: Pacf Significant Pacf value in the first lag.

Renewable Energy Production: Pacf Significant first lag. Reduces over time.

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.

```
Hydro_consumption <- Energy_Data_ts[, 5]
lm_hydro <- lm(Hydro_consumption ~time(Hydro_consumption))
            summary(lm_hydro)
```

```
##
## Call:
## lm(formula = Hydro_consumption ~ time(Hydro_consumption))
##
## 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)              342.85614   75.06654   4.567 5.96e-06 ***
## time(Hydro_consumption)  -0.13173    0.03755  -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
```

```r
Renewable_production <- Energy_Data_ts[, 4]
lm_renewable <- lm(Renewable_production ~ time(Renewable_production))
            summary(lm_renewable)
```

```
##
## Call:
## lm(formula = Renewable_production ~ time(Renewable_production))
##
## 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)                 -1.696e+04  3.316e+02  -51.16   <2e-16 ***
## time(Renewable_production)  8.687e+00  1.659e-01   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
```

```r
coeff_hydro <- coef(lm_hydro)
coeff_renewable <- coef(lm_renewable)


stargazer(lm_hydro, lm_renewable, type = "text",
          title = "Regression Results",
          align = TRUE, star.cutoffs = c(0.1, 0.05, 0.01))
```

```
##
## Regression Results
## ================================================================================
##                                          Dependent variable:
##                                 ---------------------------------------
##                                 Hydro_consumption Renewable_production
##                                        (1)                (2)
## --------------------------------------------------------------------------------
## time(Hydro_consumption)              -0.132***
##                                       (0.038)
##
## time(Renewable_production)                                 8.687***
##                                                            (0.166)
##
## Constant                              342.856***        -16,962.280***
##                                       (75.067)           (331.560)
##
## --------------------------------------------------------------------------------
## Observations                            621                621
## R2                                     0.019              0.816
## Adjusted R2                            0.018              0.816
```

```
## Residual Std. Error (df = 619)      13.980              61.750
## F Statistic (df = 1; 619)          12.304***         2,742.930***
## ============================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

```
#stargazer function gives a nice tidy look of the regression result.
```

Hydroelectric Consumption: The regression estimates an intercept of 342.856 indicates that at time zero the consumption is 342.856 trillion Btu but the coefficient is negative indicating that over time the consumption is reduced.

Renewable Production: The regression estimates an intercept of -16,962 indicates that the time series started after the time period. The coefficieint is positive and shows that renewable production is increasing by 8.68 trillion btu every year.

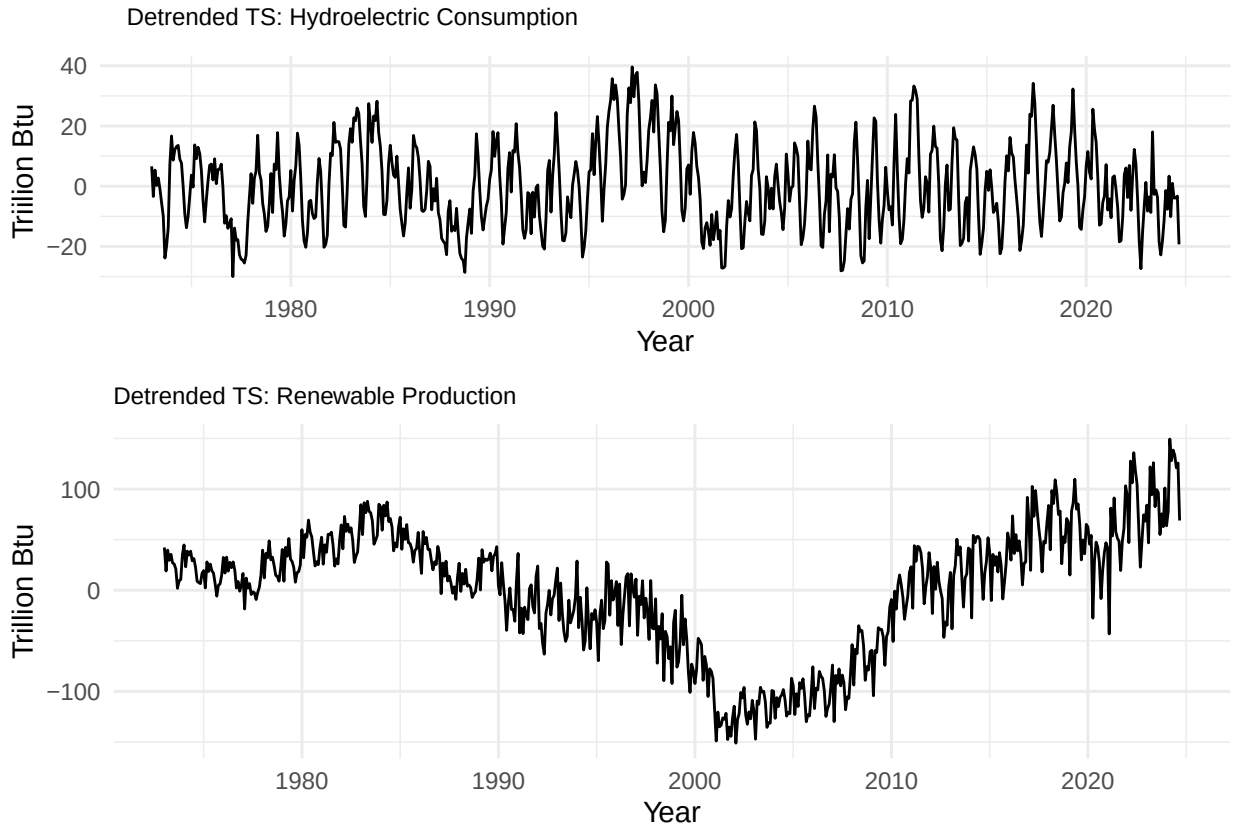
Significance Test: In both model the p-value is below 0.05 which means that the results are significant at 95% level.

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?

```
time_series <- time(Energy_Data_ts)

trend_hydro <-coeff_hydro[1] + coeff_hydro[2]*time_series
trend_renewable <- coeff_renewable[1] + coeff_renewable[2]*time_series

detrended_hydro <- Hydro_consumption - trend_hydro
detrended_renewable <- Renewable_production - trend_renewable

#detrended_ts <- ts(cbind(detrended_hydro, detrended_renewable),
#start = start(Energy_Data_ts), frequency = frequency(Energy_Data_ts))

detrended_hydro_plot <- autoplot(detrended_hydro) +
  ggtitle(paste("Detrended TS:" , "Hydroelectric Consumption")) +
  xlab("Year") +
  ylab("Triilion Btu") +
  theme_minimal()
#print(detrended_hydro_plot)

detrended_renewable_plot <- autoplot(detrended_renewable) +
  ggtitle(paste("Detrended TS:", "Renewable Production")) +
  xlab("Year") +
  ylab("Trillion Btu") +
  theme_minimal()
#print(detrended_renewable_plot)


plot_grid(detrended_hydro_plot +
          theme(plot.title = element_text(size = 9,
```

```
                                        margin = margin(5, 5, 10, 10))),
          detrended_renewable_plot +
          theme(plot.title = element_text(size = 9)),
          nrow = 2, rel_heights = c(1, 1.2))
```

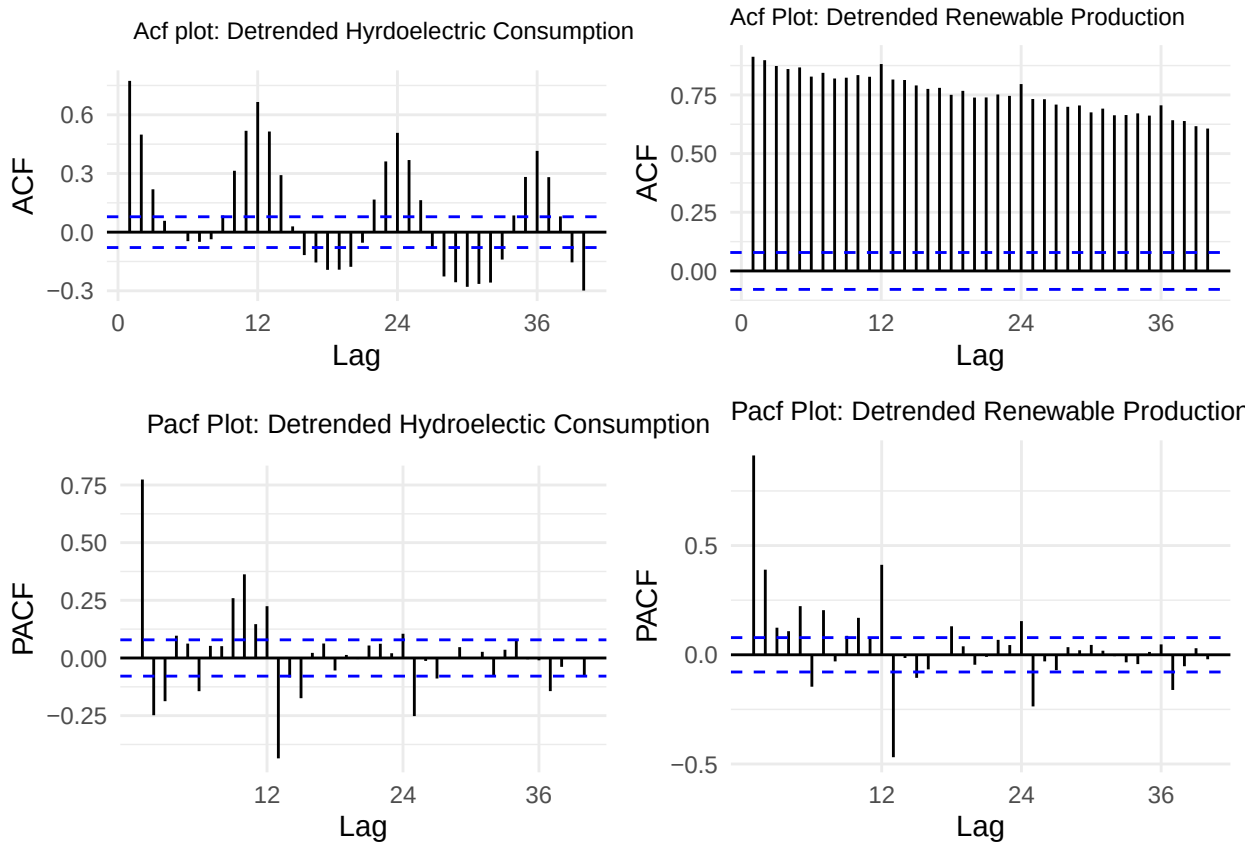


Q5

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?

```
acf_detrended_hydro <- ggAcf(detrended_hydro, lag.max = 40) +
  ggtitle("Acf plot: Detrended Hyrdoelectric Consumption") +
  theme_minimal()
#print(acf_detrended_hydro)

acf_detrended_renewable <- ggAcf(detrended_renewable, lag.max = 40) +
  ggtitle("Acf Plot: Detrended Renewable Production") +
  theme_minimal()
#print(acf_detrended_renewable)

pacf_detrended_hydro <- ggPacf(detrended_hydro, lag.max = 40) +
  ggtitle("Pacf Plot: Detrended Hydroelectic Consumption") +
  theme_minimal()
#print(pacf_detrended_hydro)
```

```r
pacf_detrended_renewable <- ggPacf(detrended_renewable, lag.max = 40) +
  ggtitle("Pacf Plot: Detrended Renewable Production") +
  theme_minimal()
#print(pacf_detrended_renewable)

plot_grid(acf_detrended_hydro +
            theme(plot.title = element_text(size = 9,
                                            margin = margin(5, 5, 10, 10))),
          acf_detrended_renewable +
            theme(plot.title = element_text(size = 9)),
          pacf_detrended_hydro +
            theme(plot.title = element_text(size = 10,
                                            margin = margin(5, 5, 10, 10))),
          pacf_detrended_renewable +
            theme(plot.title = element_text(size = 10)),
          nrow = 2, rel_heights = c(1, 1.2))
```



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.

Yes, Hydroelectric consumption seems to have seasonality. As we can see, time series shows a repeated pattern. Same goes with the acf plot of the hydroelectric consumption.

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?

```
hydro_seasonal<- seasonaldummy(Hydro_consumption)
str(hydro_seasonal)
```

Hydro_Seasonal Fit Model

```
##  num [1:621, 1:11] 1 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:11] "Jan" "Feb" "Mar" "Apr" ...
```

```
#converting to dataframe
hydro_seasonal_df <- data.frame(hydro_seasonal)
# now we have a seasonal dummy containing data frame but this "hydro_seasonal_df" does not have values
hydro_seasonal_cbind_df <- data.frame(
       Hydro_Consumption = as.numeric(Hydro_consumption),
     hydro_seasonal_df
)

lm_hydro_seasonal_model <- lm(Hydro_Consumption ~
                                 ., data = hydro_seasonal_cbind_df)
summary(lm_hydro_seasonal_model)
```

```
##
## Call:
## lm(formula = Hydro_Consumption ~ ., data = hydro_seasonal_cbind_df)
##
## 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 ***
## Jan            4.941      2.043   2.418 0.015880 *
## Feb           -2.513      2.043  -1.230 0.219219
## Mar            7.053      2.043   3.452 0.000595 ***
```

```
## Apr            5.399      2.043   2.642 0.008441 **
## May           13.907      2.043   6.807 2.40e-11 ***
## Jun           10.729      2.043   5.251 2.09e-07 ***
## Jul            4.033      2.043   1.974 0.048845 *
## Aug           -5.399      2.043  -2.643 0.008436 **
## Sep          -16.596      2.043  -8.123 2.54e-15 ***
## Oct          -16.370      2.053  -7.973 7.66e-15 ***
## Nov          -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
```

```
renewable_seasonal<- seasonaldummy(Renewable_production)
str(renewable_seasonal)
```

Renewable_Seasonal Fit Model

```
##  num [1:621, 1:11] 1 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:11] "Jan" "Feb" "Mar" "Apr" ...
```

```
#converting to dataframe
renewable_seasonal_df <- data.frame(renewable_seasonal)

renewable_seasonal_cbind_df <- data.frame(
       Renewable_Production = as.numeric(Renewable_production),
     renewable_seasonal_df
)

lm_renewable_seasonal_model <- lm(Renewable_Production ~
                                    ., data = renewable_seasonal_cbind_df)
summary(lm_renewable_seasonal_model)
```

```
##
## Call:
## lm(formula = Renewable_Production ~ ., data = renewable_seasonal_cbind_df)
##
## 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 ***
## Jan            1.973     28.468   0.069    0.945
## Feb          -34.348     28.468  -1.207    0.228
```

```
## Mar            4.721     28.468   0.166    0.868
## Apr           -7.896     28.468  -0.277    0.782
## May            7.199     28.468   0.253    0.800
## Jun           -3.394     28.468  -0.119    0.905
## Jul            2.850     28.468   0.100    0.920
## Aug           -4.430     28.468  -0.156    0.876
## Sep          -29.037     28.468  -1.020    0.308
## Oct          -20.205     28.606  -0.706    0.480
## Nov          -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
stargazer(lm_hydro_seasonal_model, lm_renewable_seasonal_model,
          type = "text",
          title = "Regression Results Seasonal",
          align = TRUE, star.cutoffs = c(0.1, 0.05, 0.01))
```

Summary Output

```
##
## Regression Results Seasonal
## ================================================================
##                                        Dependent variable:
##                                    -----------------------------
##                                   Hydro_Consumption Renewable_Production
##                                          (1)               (2)
## ----------------------------------------------------------------
## Jan                                     4.941**           1.973
##                                         (2.043)          (28.468)
##
## Feb                                     -2.513           -34.348
##                                         (2.043)          (28.468)
##
## Mar                                    7.053***           4.721
##                                         (2.043)          (28.468)
##
## Apr                                    5.399***           -7.896
##                                         (2.043)          (28.468)
##
## May                                    13.907***           7.199
##                                         (2.043)          (28.468)
##
## Jun                                    10.729***           -3.394
##                                         (2.043)          (28.468)
##
## Jul                                     4.033**           2.850
##                                         (2.043)          (28.468)
```

```
##
## Aug                                -5.399***           -4.430
##                                     (2.043)           (28.468)
##
## Sep                               -16.596***           -29.037
##                                     (2.043)           (28.468)
##
## Oct                               -16.370***           -20.205
##                                     (2.053)           (28.606)
##
## Nov                               -10.805***           -20.706
##                                     (2.053)           (28.606)
##
## Constant                           79.981***          410.598***
##                                     (1.452)           (20.228)
##
## ------------------------------------------------------------------
## Observations                          621                621
## R2                                   0.470              0.009
## Adjusted R2                          0.460              -0.009
## Residual Std. Error (df = 609)       10.367             144.453
## F Statistic (df = 11; 609)          49.003***            0.486
## ==================================================================
## Note:                                    *p<0.1; **p<0.05; ***p<0.01
```

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?

```
#saving coefficients from seasonal fit model

#coeff_hydro_seasonal <- coef(lm_hydro_seasonal_model)
#coeff_renewable_seasonal <- coef(lm_renewable_seasonal_model)
```

```
#deseasoning the hydro series
seasonal_fit_hydro <- fitted(lm_hydro_seasonal_model) #extracting fitted component from regression

#subtracting seasonal component from original time series

deseasoned_hydro <- Hydro_consumption -seasonal_fit_hydro

#converting deseasoned series to a time series

deseasoned_hydro_ts <- ts(deseasoned_hydro, frequency = 12,
                          start = start(Hydro_consumption))

deseasoned_hydro_ts_mean <- mean(deseasoned_hydro_ts)

#Why Use start = start(Hydro_consumption)?
#Maintains the Correct Time Index:
#Your original Hydro_consumption time series has a defined start time (e.g., January 1980).
#If you don't specify start = start(Hydro_consumption), R will treat the new series as if it starts at
```

```r
#deseasoning the renewable series

seasonal_fit_renewable <- fitted(lm_renewable_seasonal_model)

deseasoned_renewable <- Renewable_production -seasonal_fit_renewable

deseasoned_renewable_ts <- ts(deseasoned_renewable,
                              frequency = 12,
                              start = start(Renewable_production))

deseasoned_renewable_ts_mean <- mean(deseasoned_renewable_ts)
```

```r
# Comparing
ts_deseasoned_hydro <- autoplot(deseasoned_hydro_ts)+
  ggtitle(paste("Deseasoned Hydroelectric Consumption")) +
  xlab("Year") +
  ylab("Triilion Btu") +
  geom_hline(yintercept = deseasoned_hydro_ts_mean,
             color = "blue",
             linetype = "solid")
  theme_minimal()
```

```
## List of 136
##  $ line                            :List of 6
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                            :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                            :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                           : NULL
```

```
##  $ aspect.ratio                    : NULL
##  $ axis.title                      : NULL
##  $ axis.title.x                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom             : NULL
##  $ axis.title.y                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left               : NULL
##  $ axis.title.y.right              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num -90
```

```
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x                   :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom            : NULL
##  $ axis.text.y                   :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
```

```
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left                : NULL
##  $ axis.text.y.right               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.theta                 : NULL
##  $ axis.text.r                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x                    : NULL
##  $ axis.ticks.x.top                : NULL
##  $ axis.ticks.x.bottom             : NULL
##  $ axis.ticks.y                    : NULL
##  $ axis.ticks.y.left               : NULL
##  $ axis.ticks.y.right              : NULL
##  $ axis.ticks.theta                : NULL
##  $ axis.ticks.r                    : NULL
##  $ axis.minor.ticks.x.top          : NULL
##  $ axis.minor.ticks.x.bottom       : NULL
##  $ axis.minor.ticks.y.left         : NULL
##  $ axis.minor.ticks.y.right        : NULL
##  $ axis.minor.ticks.theta          : NULL
##  $ axis.minor.ticks.r              : NULL
```

```
##  $ axis.ticks.length              : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x            : NULL
##  $ axis.ticks.length.x.top        : NULL
##  $ axis.ticks.length.x.bottom     : NULL
##  $ axis.ticks.length.y            : NULL
##  $ axis.ticks.length.y.left       : NULL
##  $ axis.ticks.length.y.right      : NULL
##  $ axis.ticks.length.theta        : NULL
##  $ axis.ticks.length.r            : NULL
##  $ axis.minor.ticks.length        : 'rel' num 0.75
##  $ axis.minor.ticks.length.x      : NULL
##  $ axis.minor.ticks.length.x.top  : NULL
##  $ axis.minor.ticks.length.x.bottom: NULL
##  $ axis.minor.ticks.length.y      : NULL
##  $ axis.minor.ticks.length.y.left  : NULL
##  $ axis.minor.ticks.length.y.right : NULL
##  $ axis.minor.ticks.length.theta   : NULL
##  $ axis.minor.ticks.length.r      : NULL
##  $ axis.line                      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x                    : NULL
##  $ axis.line.x.top                : NULL
##  $ axis.line.x.bottom             : NULL
##  $ axis.line.y                    : NULL
##  $ axis.line.y.left               : NULL
##  $ axis.line.y.right              : NULL
##  $ axis.line.theta                : NULL
##  $ axis.line.r                    : NULL
##  $ legend.background              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin                  : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing                 : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x               : NULL
##  $ legend.spacing.y               : NULL
##  $ legend.key                     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size                : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height              : NULL
##  $ legend.key.width               : NULL
##  $ legend.key.spacing             : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.key.spacing.x           : NULL
##  $ legend.key.spacing.y           : NULL
##  $ legend.frame                   : NULL
##  $ legend.ticks                   : NULL
##  $ legend.ticks.length            : 'rel' num 0.2
##  $ legend.axis.line               : NULL
##  $ legend.text                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
```

```
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.position            : NULL
##  $ legend.title                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.position           : NULL
##  $ legend.position                 : chr "right"
##  $ legend.position.inside          : NULL
##  $ legend.direction                : NULL
##  $ legend.byrow                    : NULL
##  $ legend.justification            : chr "center"
##  $ legend.justification.top        : NULL
##  $ legend.justification.bottom     : NULL
##  $ legend.justification.left       : NULL
##  $ legend.justification.right      : NULL
##  $ legend.justification.inside     : NULL
##  $ legend.location                 : NULL
##  $ legend.box                      : NULL
##  $ legend.box.just                 : NULL
##  $ legend.box.margin               : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing              : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##   [list output truncated]
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

```r
#print(ts_deseasoned_hydro)

ts_deseasoned_renewable <- autoplot(deseasoned_renewable_ts) +
  ggtitle(paste("Deseasoned Renewable Production")) +
  xlab("Year") +
```

```
  ylab("Trillion Btu") +
  geom_hline(yintercept = deseasoned_renewable_ts_mean,
             color = "blue",
             linetype = "solid")
  theme_minimal()
```

```
## List of 136
##  $ line                            :List of 6
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                            :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                            :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                           : NULL
##  $ aspect.ratio                    : NULL
##  $ axis.title                      : NULL
##  $ axis.title.x                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
```

```
##  $ axis.title.x.top               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom            : NULL
##  $ axis.title.y                   :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left              : NULL
##  $ axis.title.y.right             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                      :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
```

```
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom             : NULL
##  $ axis.text.y                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left               : NULL
##  $ axis.text.y.right              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
```

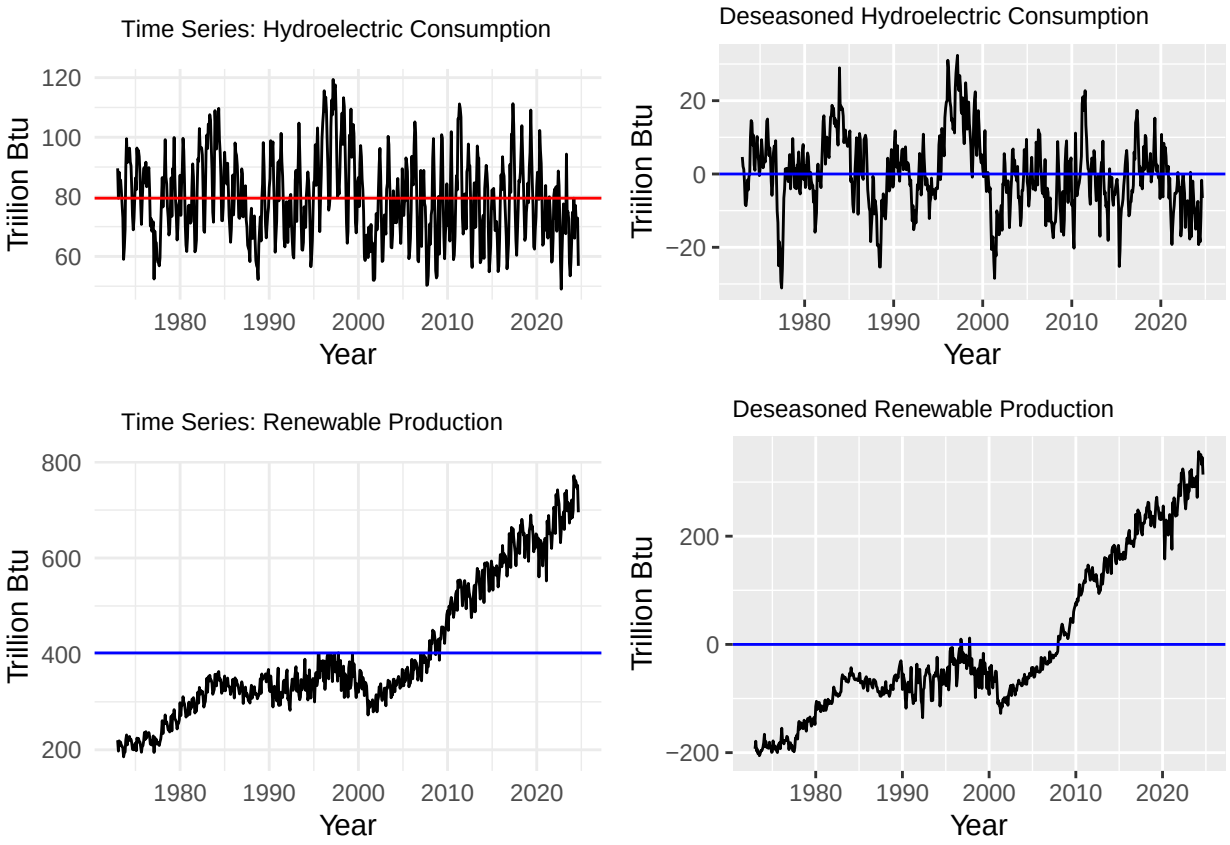
```
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.theta                 : NULL
##  $ axis.text.r                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x                    : NULL
##  $ axis.ticks.x.top                : NULL
##  $ axis.ticks.x.bottom             : NULL
##  $ axis.ticks.y                    : NULL
##  $ axis.ticks.y.left               : NULL
##  $ axis.ticks.y.right              : NULL
##  $ axis.ticks.theta                : NULL
##  $ axis.ticks.r                    : NULL
##  $ axis.minor.ticks.x.top          : NULL
##  $ axis.minor.ticks.x.bottom       : NULL
##  $ axis.minor.ticks.y.left         : NULL
##  $ axis.minor.ticks.y.right        : NULL
##  $ axis.minor.ticks.theta          : NULL
##  $ axis.minor.ticks.r              : NULL
##  $ axis.ticks.length               : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x             : NULL
##  $ axis.ticks.length.x.top         : NULL
##  $ axis.ticks.length.x.bottom      : NULL
##  $ axis.ticks.length.y             : NULL
##  $ axis.ticks.length.y.left        : NULL
##  $ axis.ticks.length.y.right       : NULL
##  $ axis.ticks.length.theta         : NULL
##  $ axis.ticks.length.r             : NULL
##  $ axis.minor.ticks.length         : 'rel' num 0.75
##  $ axis.minor.ticks.length.x       : NULL
##  $ axis.minor.ticks.length.x.top   : NULL
##  $ axis.minor.ticks.length.x.bottom: NULL
##  $ axis.minor.ticks.length.y       : NULL
##  $ axis.minor.ticks.length.y.left  : NULL
```

```
##  $ axis.minor.ticks.length.y.right : NULL
##  $ axis.minor.ticks.length.theta   : NULL
##  $ axis.minor.ticks.length.r       : NULL
##  $ axis.line                       : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x                     : NULL
##  $ axis.line.x.top                 : NULL
##  $ axis.line.x.bottom              : NULL
##  $ axis.line.y                     : NULL
##  $ axis.line.y.left                : NULL
##  $ axis.line.y.right               : NULL
##  $ axis.line.theta                 : NULL
##  $ axis.line.r                     : NULL
##  $ legend.background               : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin                   : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing                  : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x                : NULL
##  $ legend.spacing.y                : NULL
##  $ legend.key                      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size                 : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height               : NULL
##  $ legend.key.width                : NULL
##  $ legend.key.spacing              : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.key.spacing.x            : NULL
##  $ legend.key.spacing.y            : NULL
##  $ legend.frame                    : NULL
##  $ legend.ticks                    : NULL
##  $ legend.ticks.length             : 'rel' num 0.2
##  $ legend.axis.line                : NULL
##  $ legend.text                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.position            : NULL
##  $ legend.title                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
```

```
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.position           : NULL
##  $ legend.position                 : chr "right"
##  $ legend.position.inside          : NULL
##  $ legend.direction                : NULL
##  $ legend.byrow                    : NULL
##  $ legend.justification            : chr "center"
##  $ legend.justification.top        : NULL
##  $ legend.justification.bottom     : NULL
##  $ legend.justification.left       : NULL
##  $ legend.justification.right      : NULL
##  $ legend.justification.inside     : NULL
##  $ legend.location                 : NULL
##  $ legend.box                      : NULL
##  $ legend.box.just                 : NULL
##  $ legend.box.margin               : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing              : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##   [list output truncated]
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

```
#print(ts_deseasoned_renewable)
```

```
plot_grid(ts_hydro +
          theme(plot.title = element_text(size = 9,
                                          margin = margin(5, 5, 10, 10))),
          ts_deseasoned_hydro +
          theme(plot.title = element_text(size = 9)),

          ts_renewables +
          theme(plot.title = element_text(size = 9,
                                          margin = margin(5, 5, 10, 10))),
          ts_deseasoned_renewable +
          theme(plot.title = element_text(size = 9)),

          nrow = 2,
          rel_heights = c(1, 1.2))
```



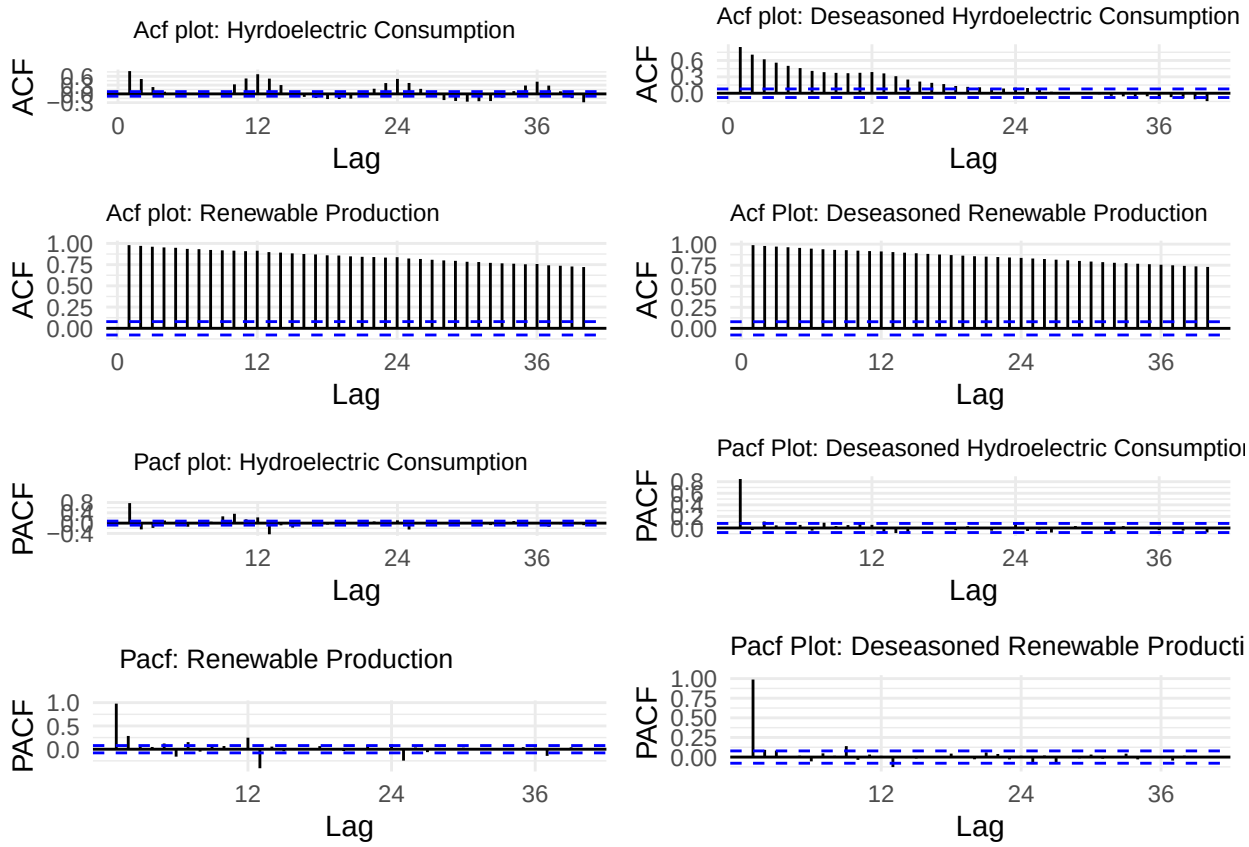
Hydroelectric Consumption: With the first series we can observe seasonality and the mean fluctuates around a stable mean. When the seasonality is removed and fluctuates around zero.

Renewable Production Since we have not observed seasonality in original time series, deseasoning did not have a great impact like hydroelectric consumption. In renewable production seasonality has been removed. ### 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?

```
# Comparing
acf_deseasoned_hydro <- ggAcf(deseasoned_hydro_ts, lag.max = 40) +
  ggtitle("Acf plot: Deseasoned Hyrdoelectric Consumption") +
  theme_minimal()
#print(acf_deseasoned_hydro)

acf_deseasoned_renewable <- ggAcf(deseasoned_renewable_ts, lag.max = 40) +
  ggtitle("Acf Plot: Deseasoned Renewable Production ") +
  theme_minimal()
#print(acf_deseasoned_renewable)

pacf_deseasoned_hydro <- ggPacf(deseasoned_hydro_ts, lag.max = 40) +
  ggtitle("Pacf Plot: Deseasoned Hydroelectric Consumption") +
  theme_minimal()
#print(pacf_deseasoned_hydro)
```

```
pacf_deseasoned_renewable <- ggPacf(deseasoned_renewable_ts, lag.max = 40) +
  ggtitle("Pacf Plot: Deseasoned Renewable Production") +
  theme_minimal()
#print(pacf_deseasoned_renewable)

plot_grid(acf_hydro +
          theme(plot.title = element_text(size = 9,
                                          margin = margin(5, 5, 10, 10))),
          acf_deseasoned_hydro +
              theme(plot.title = element_text(size = 9)),
          acf_renewables +
             theme(plot.title = element_text(size = 9)),
          acf_deseasoned_renewable +
             theme(plot.title = element_text(size = 9)),
          pacf_hydro +
              theme(plot.title = element_text(size = 9, margin = margin(5, 5, 10, 10))),
          pacf_deseasoned_hydro +
              theme(plot.title = element_text(size = 9)),
          pacf_renewables +
              theme(plot.title = element_text(size = 10, margin = margin(5, 5, 10, 10))),
          pacf_deseasoned_renewable +
              theme(plot.title = element_text(size = 10)),
          nrow = 4, rel_heights = c(1, 1.2))
```



Acf After removing the seasonality from the Hydroelectric series we observe that seasonality has reduced significantly. Earlier seasonality was observed in every six lags which were significant. After removing the seasonlaity the slight seasonality is observed after 24 lags.

Renewable production did not observe strong seasonality as we can observe from the plot that, after removing seasonality the plot did not change much. ####Pacf

After removing seasonality pacf of the Hydroelectric consumption has become less significant in all the lags except for the first one. It indicates that the seasonality has been removed.

Renewable production did not show a strong seasonality. The result was not changed as dramatically as in the case of hydroelectric consumption.