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

Assignment 3 - Due date 02/04/25

Lucy Wang

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_Consumption 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.\

```
#Load/install required package here
library(lubridate)
library(ggplot2)
library(forecast)
library(readxl)
library(openxlsx)
library(tseries)
library(Kendall)
library(cowplot)
library(glue)
```

```
# Load data file
data_file <- read_excel(path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xl
                         skip = 12,
```

```
                           sheet="Monthly Data",col_names=FALSE)

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

#Assign the column names to the data set
colnames(data_file) <- read_col_names

#Visualize the first rows of the data set
cleaned_df <- subset(data_file, select = c('Month',
                                           'Total Renewable Energy Production',
                                           'Hydroelectric Power Consumption'))

cleaned_df <- as.data.frame(cleaned_df)

#Create time series
ts_data <- ts(cleaned_df[,2:3],start=c(1973,1),frequency=12)

nobs <- nrow(cleaned_df)
```

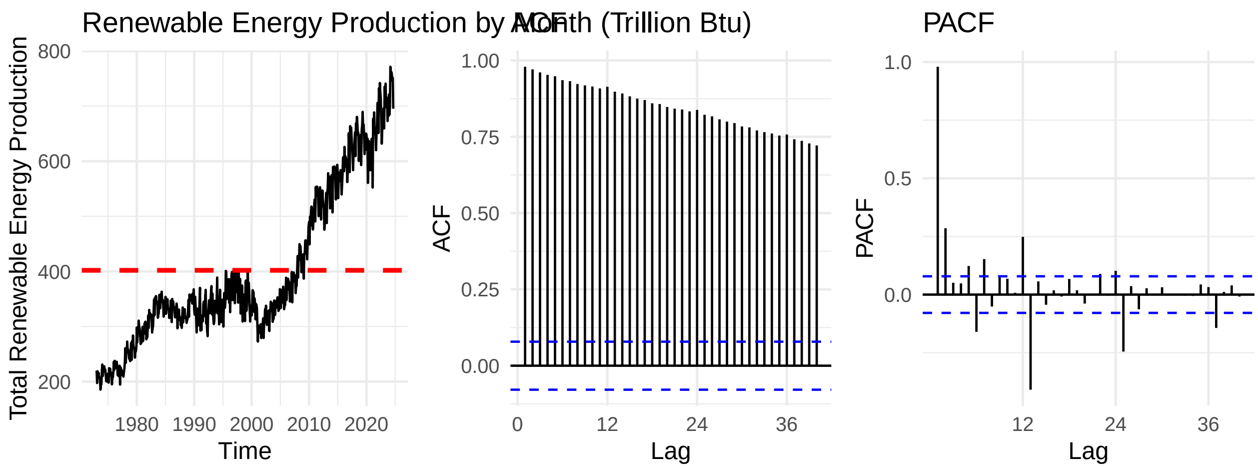
##Trend Component

Q1

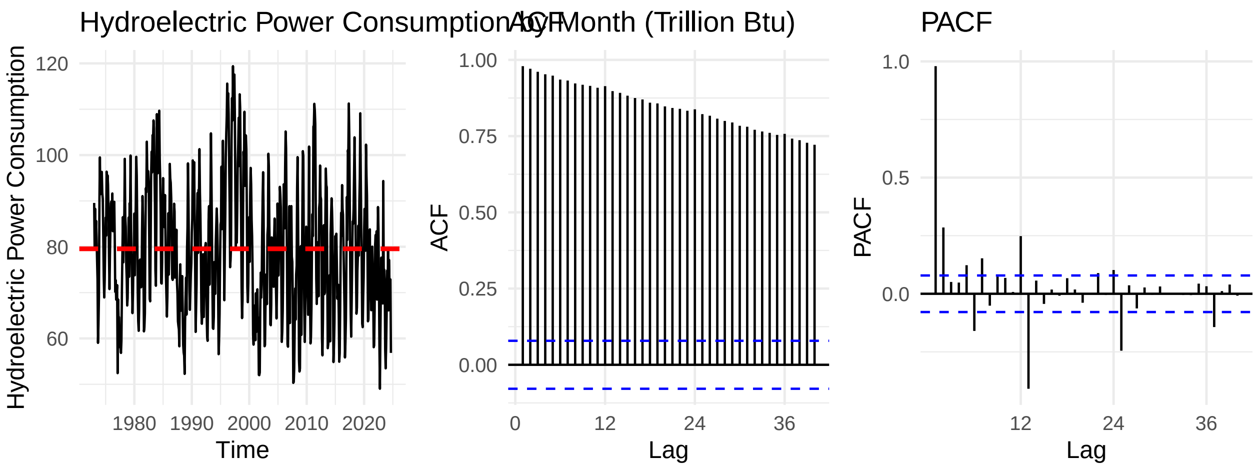
For each time series, i.e., Renewable Energy Production and Hydroelectric Consumption create three plots: one with time series, one with the ACF and with the PACF. You may use the some code from A2, but I want all the three plots side by side as in a grid. (Hint: use function `plot_grid()` from the `cowplot`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

```
Renewable_ts_plot <- autoplot(ts_data[,1]) +
  xlab("Time") +
  ylab("Total Renewable Energy Production") +
  labs(color="Reservoir")+
  geom_hline(yintercept = mean(ts_data[,1]),
             color = "red",
             linetype = "dashed",
             size = 1) +
  ggtitle("Renewable Energy Production by Month (Trillion Btu)") +
  theme_minimal()

# Create the ACF plot
Renewable_acf <- Acf(ts_data[,1], lag.max=40, type="correlation", plot=FALSE)
Renewable_acf_plot <- autoplot(Renewable_acf) +
  ggtitle("ACF") +
  theme_minimal()

# Create the PACF plot
Renewable_pacf <- Pacf(ts_data[,1], lag.max=40, plot=FALSE)
Renewable_pacf_plot <- autoplot(Renewable_pacf) +
  ggtitle("PACF") +
  theme_minimal()

# Arrange the plots side by side using plot_grid()
```

```
plot_grid(Renewable_ts_plot, Renewable_acf_plot, Renewable_pacf_plot, ncol = 3, align = 'h')
```




```
Hydro_ts_plot <- autoplot(ts_data[,2]) +
  xlab("Time") +
  ylab("Hydroelectric Power Consumption") +
  labs(color="Reservoir")+
  geom_hline(yintercept = mean(ts_data[,2]),
             color = "red",
             linetype = "dashed",
             size = 1) +
  ggtitle("Hydroelectric Power Consumption by Month (Trillion Btu)") +
  theme_minimal()

# Create the ACF plot
Hydro_acf <- Acf(ts_data[,2], lag.max=40, type="correlation", plot=FALSE)
Hydro_acf_plot <- autoplot(Renewable_acf) +
  ggtitle("ACF") +
  theme_minimal()

# Create the PACF plot
Hydro_pacf <- Pacf(ts_data[,2], lag.max=40, plot=FALSE)
Hydro_pacf_plot <- autoplot(Renewable_pacf) +
  ggtitle("PACF") +
  theme_minimal()

# Arrange the plots side by side using plot_grid()
plot_grid(Hydro_ts_plot, Hydro_acf_plot, Hydro_pacf_plot, ncol = 3, align = 'h')
```



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 total renewable energy production time series shows **an overall upward trend** with slight fluctuations around 1990 to 2000. Its ACF decays slowly, suggesting non-stationarity with a trend. The PACF plot shows a strong lag-1 correlation, indicating the strong dependency on their past values.

The hydroelectric power consumption time series does not show long-term increasing or decreasing trends, which is more **stationary (no trend)**. Rather, it shows seasonal or cyclic variations. The ACF plot decays slowly, showing strong persistence. Its PACF includes several significant lags, indicating a potential seasonal or cyclical structure.

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.

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

#Fit a linear trend to TS of renewable
Renewable_linear_trend <- lm(cleaned_df[,2] ~ t)
print(summary(Renewable_linear_trend))
```

```
##
## Call:
## lm(formula = cleaned_df[, 2] ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -151.11  -37.84   13.53   41.76  149.42
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.87293    4.96189   35.65   <2e-16 ***
## t             0.72393    0.01382   52.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 61.75 on 619 degrees of freedom
## Multiple R-squared:  0.8159, Adjusted R-squared:  0.8156
## F-statistic:  2743 on 1 and 619 DF,  p-value: < 2.2e-16
```

```r
#Fit a linear trend to TS of hydroelectric
Hydro_linear_trend <- lm(cleaned_df[,3] ~ t)
print(summary(Hydro_linear_trend))
```

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

Renewable: The intercept is at 176.87, meaning that the renewable energy production at t=0 (the start of 1973) is 176.87 trillion Btu. The slope is 0.7239, meaning that the renewable energy production increases by 0.7239 trillion Btu per month. The adjusted R-squared of 0.8156 shows that 81.56% of the variation can be explained by the model, indicating a strong linear trend. The low p-value (less than 0.05) shows that the linear model is statistically significant.

```r
#Fit a linear trend to TS of hydroelectric
Hydro_linear_trend <- lm(cleaned_df[,3] ~ t)
print(summary(Hydro_linear_trend))
```

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

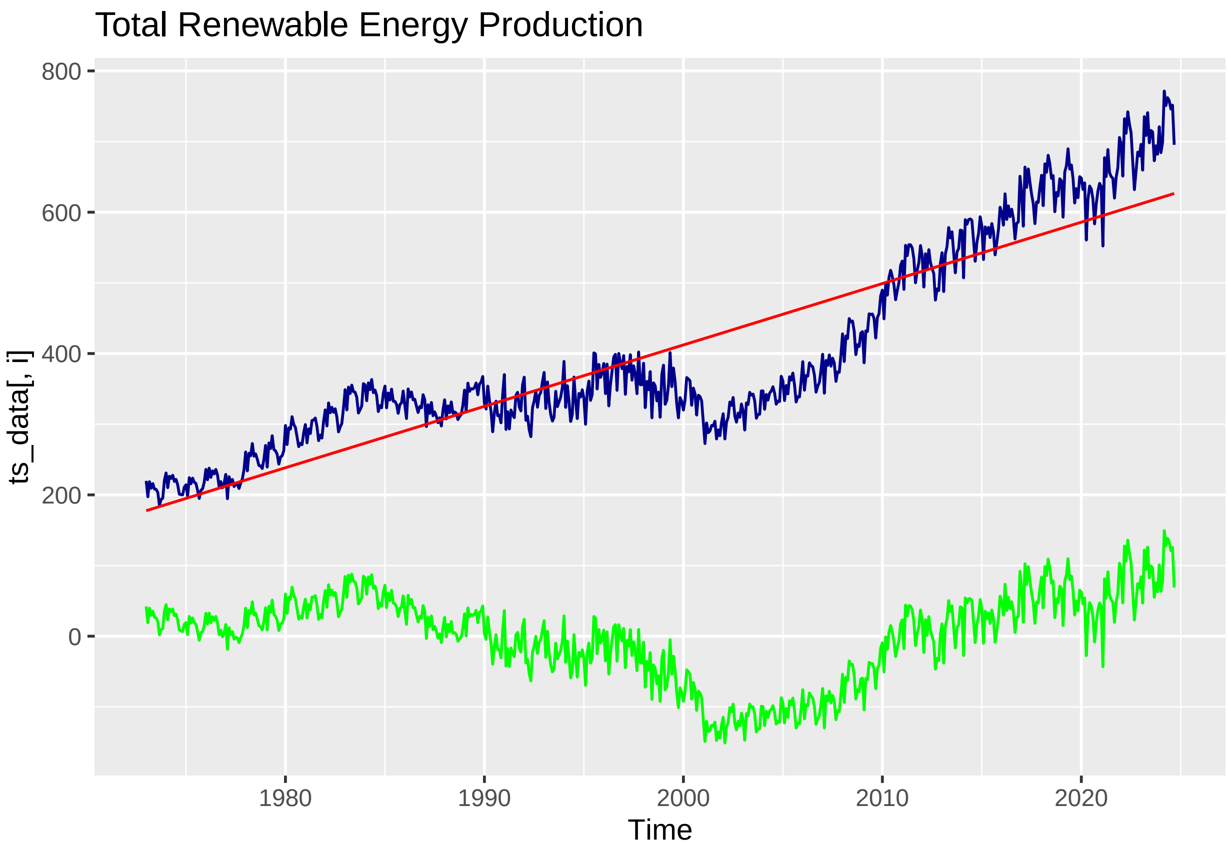
Hydroelectric: The intercept is at 82.97, meaning that the hydroelectric power consumption at t=0 (the start of 1973) is 82.97 trillion Btu. The slope is -0.01, meaning that the renewable energy production decreases by 0.01 trillion Btu per month. The adjusted R-squared of 0.01791 shows that only a small portion of the variation can be explained by the model, indicating the stationarity. The low p-value (less than 0.05) shows that the linear model is statistically significant.

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?

```
plot_detrend <- function(lm_model, i){
  # assign beta
  beta0 <- as.numeric(lm_model$coefficients[1])
  beta1 <- as.numeric(lm_model$coefficients[2])

  # detrend inflow
  linear_trend <- beta0 + beta1 * t
  ts_linear <- ts(linear_trend,start=c(1973,1),frequency=12)

  detrend_energy <- cleaned_df[,i+1] - linear_trend
  ts_detrend <- ts(detrend_energy, start = c(1973,1),frequency = 12)

  #Plot
  detrended_plot <- autoplot(ts_data[,i],color="darkblue")+
    autolayer(ts_detrend,series="Detrended",color="green")+
    autolayer(ts_linear,series="Linear Component",color="red")+
    ggtitle(colnames(ts_data)[i])

  return(list(detrended_data = ts_detrend, plot = detrended_plot))
}
```

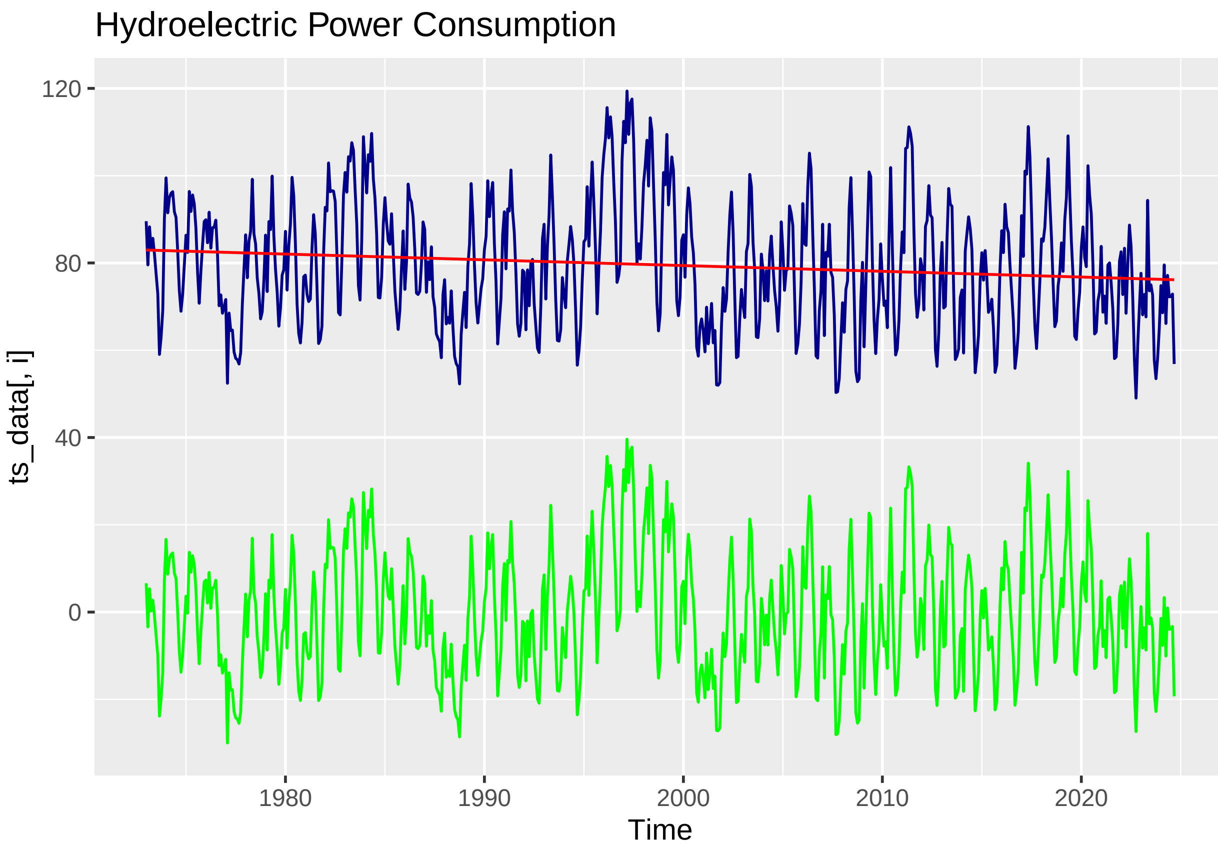
```
Renewable_detrend_ts <- plot_detrend(Renewable_linear_trend,1)$plot
Renewable_detrend_ts
```



The detrended series is presented by the green lines, while the original times series are shown in blue.

Renewable: The strong upward trend is now eliminated, leaving fluctuations around 0 and showing stationarity. With detrending, we are able to observe the seasonality.

```
Hydro_detrend_ts <-plot_detrend(Hydro_linear_trend,2)$plot
Hydro_detrend_ts
```



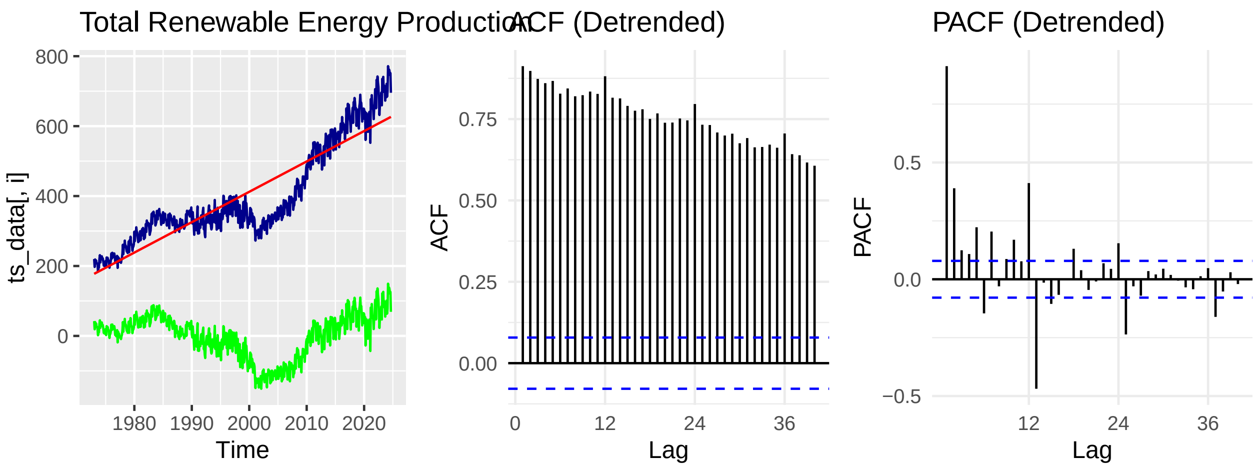
Hydroelectric: Since the linear trend is relatively flat showing non-stationarity, the detrended series shows similar pattern (including the seasonal pattern) with the original time series except for the shift around 0.

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?

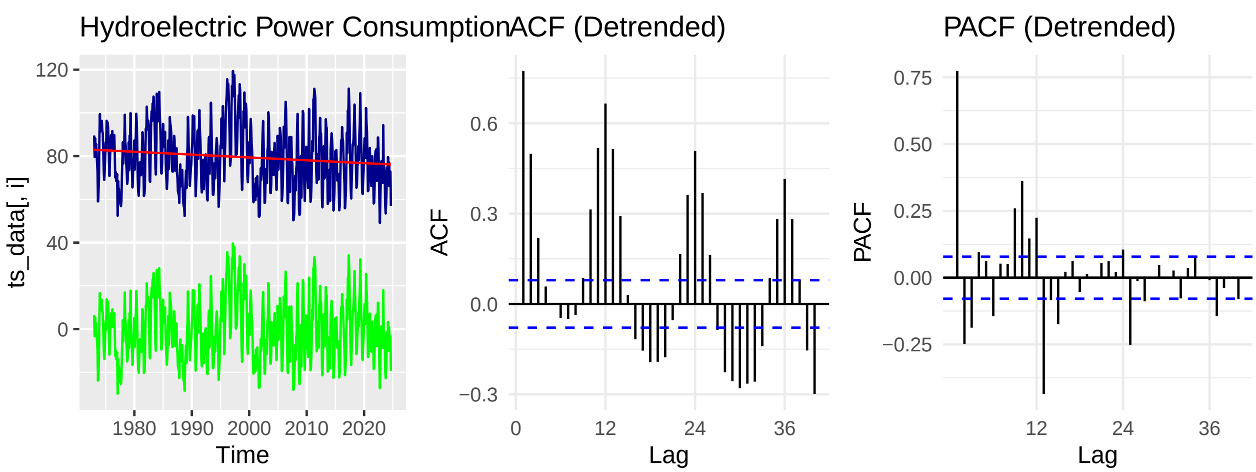
```r
plot_acf_pacf <- function(lm_model, i){
  detrended_data <- plot_detrend(lm_model,i)$detrended_data
  detrend_acf <- Acf(detrended_data,
                     lag.max=40, type="correlation",
                     plot=FALSE)
  acf_plot <- autoplot(detrend_acf) +
    ggtitle(("ACF (Detrended)")) +
    theme_minimal()
  detrend_pacf <- Pacf(detrended_data,
                       lag.max=40, plot=FALSE)
  pacf_plot <- autoplot(detrend_pacf) +
    ggtitle(glue("PACF (Detrended)")) +
    theme_minimal()
  return(list(acf_plot, pacf_plot))
}
```

```r
Renewable_detrend_acf <- plot_acf_pacf(Renewable_linear_trend, 1)[[1]]
Renewable_detrend_pacf <- plot_acf_pacf(Renewable_linear_trend, 1)[[2]]

# Arrange the plots side by side using plot_grid()
plot_grid(Renewable_detrend_ts, Renewable_detrend_acf, Renewable_detrend_pacf, ncol = 3, align = 'h')
```



Renewable: After detrending, the ACF still shows autocorrelation patterns indicating the strong linear trend. For the detrended PACF plot, the significant lags have reduced, meaning less dependence on past values.

```r
Hydro_detrend_acf <- plot_acf_pacf(Hydro_linear_trend,2)[[1]]
Hydro_detrend_pacf <- plot_acf_pacf(Hydro_linear_trend,2)[[2]]

# Arrange the plots side by side using plot_grid()
plot_grid(Hydro_detrend_ts, Hydro_detrend_acf, Hydro_detrend_pacf, ncol = 3, align = 'h')
```



Hydroelectric: Comparing to the original ACF plot that shows gradual decay, the detrended ACF shows more short-term fluctuations indicating seasonality. The autocorrelation at higher lags has also decreased, suggesting the removed long-term trend component. In the detrended PACF plot, the values are much more stationary around zero, and some intermediate lags show significant partial autocorrelations.

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, the series seem to have seasonal trends. The time series plots show short-term periodic fluctuations.

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?

```
#Use seasonal means model
seas_model <- function(i){
  dummies <- seasonaldummy(ts_data[,i])
  seas_means_model <- lm(ts_data[,i] ~ dummies)
  return(seas_means_model)
}

R_seas_means_model <- seas_model(1)
summary(R_seas_means_model)
```

```
##
## Call:
## lm(formula = ts_data[, i] ~ dummies)
##
## 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 ***
## dummiesJan     1.973     28.468   0.069    0.945
## dummiesFeb   -34.348     28.468  -1.207    0.228
## dummiesMar     4.721     28.468   0.166    0.868
## dummiesApr    -7.896     28.468  -0.277    0.782
## dummiesMay     7.199     28.468   0.253    0.800
## dummiesJun    -3.394     28.468  -0.119    0.905
## dummiesJul     2.850     28.468   0.100    0.920
## dummiesAug    -4.430     28.468  -0.156    0.876
## dummiesSep   -29.037     28.468  -1.020    0.308
## dummiesOct   -20.205     28.606  -0.706    0.480
## dummiesNov   -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
```

Renewable: The intercept is at 410.598, meaning that the average renewable energy production in December is 410.598 trillion Btu. The seasonal dummy coefficients across the month shows random variation from December and therefore does not indicate a seasonal pattern. The low adjusted R-squared of -0.0092 suggests that less than 1% of the variability in renewable energy production can be explained by seasonality.
The p-values are greater than 0.05, showing NO evidence that the model is statistically significant.

```
H_seas_means_model <- seas_model(2)
summary(H_seas_means_model)
```

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

```
## lm(formula = ts_data[, i] ~ dummies)
##
## 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 ***
## dummiesJan     4.941      2.043   2.418 0.015880 *
## dummiesFeb    -2.513      2.043  -1.230 0.219219
## dummiesMar     7.053      2.043   3.452 0.000595 ***
## dummiesApr     5.399      2.043   2.642 0.008441 **
## dummiesMay    13.907      2.043   6.807 2.40e-11 ***
## dummiesJun    10.729      2.043   5.251 2.09e-07 ***
## dummiesJul     4.033      2.043   1.974 0.048845 *
## dummiesAug    -5.399      2.043  -2.643 0.008436 **
## dummiesSep   -16.596      2.043  -8.123 2.54e-15 ***
## dummiesOct   -16.370      2.053  -7.973 7.66e-15 ***
## dummiesNov   -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
```

Hydroelectric: The intercept is at 79.981, meaning that the average hydroelectric power consumption in December is 79.981 trillion Btu. The seasonal dummy coefficients across the month shows stepped variations from December. The adjusted R-squared suggests that 46% of the variability in hydroelectric consumption can be explained by seasonality.
The p-values for all months except for February are less than 0.05, showing the significant seasonal effect associated with most months (with February as the exception).

Therefore, hydroelectric power consumption shows a seasonal trend while renewable energy production does not. The renewable series does not match with my answer in Q6.

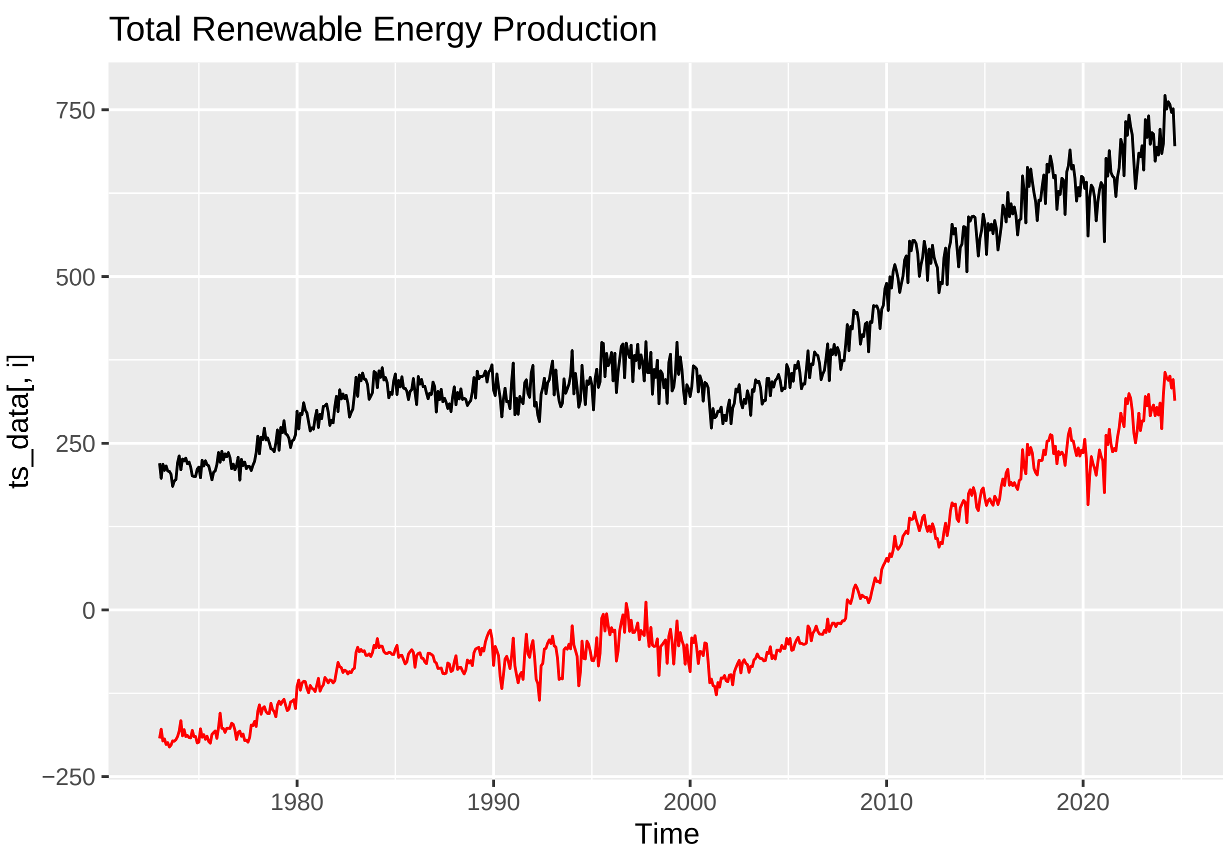
Q8

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
deseason_series <- function(seas_means_model,i){
  beta_intercept <-seas_means_model$coefficients[1]
  beta_coeff <-seas_means_model$coefficients[2:12]

  dummies <- seasonaldummy(ts_data[,i])

  inflow_seas_comp <- array(0,nobs)
  for(n in 1:nobs){
    inflow_seas_comp[n] <- beta_intercept + beta_coeff %*% dummies[n,]
  }

  deseason_data <- ts_data[,i] - inflow_seas_comp

  ts_deseason_data <- ts(deseason_data,start=c(1973,1),
```

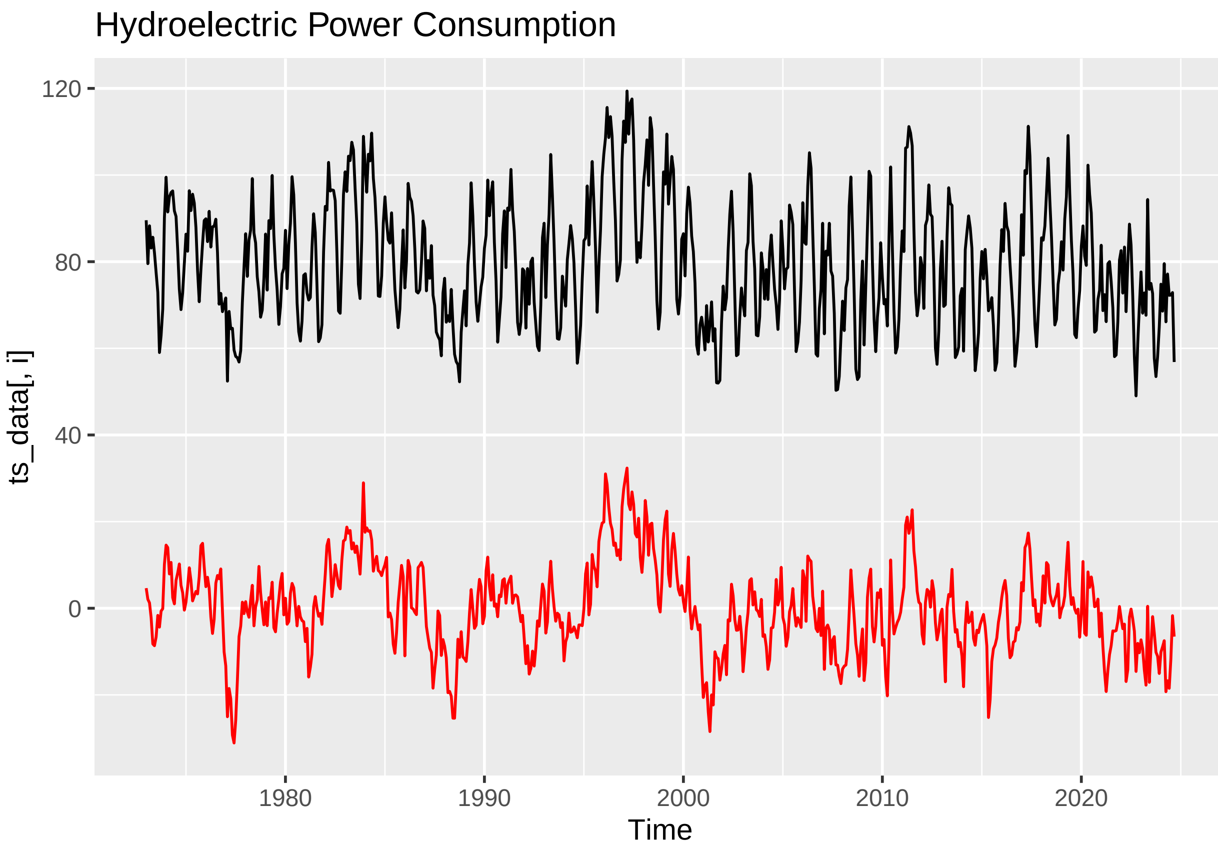
```
                                    frequency = 12)

  deseason_plot <-autoplot(ts_data[,i])+
    autolayer(ts_deseason_data,color="red") +
    ggtitle(colnames(ts_data)[i])

  return(list(deseason_ts = ts_deseason_data, plot = deseason_plot))
}
```

```
R_deseason_ts <- deseason_series(R_seas_means_model,1)$deseason_ts
R_deseason_plot <- deseason_series(R_seas_means_model,1)$plot
R_deseason_plot
```



Renewable: The deseason series has smoothed the periodic fluctuations slightly, making the long–term upward trend more observable.

```
H_deseason_ts <- deseason_series(H_seas_means_model,2)$deseason_ts
H_deseason_plot <- deseason_series(H_seas_means_model,2)$plot
H_deseason_plot
```



Hydroelectric: The deseason series has smoothed the seasonal fluctuations (short-term, periodic variations). The series becomes more irregular with few observable trend.

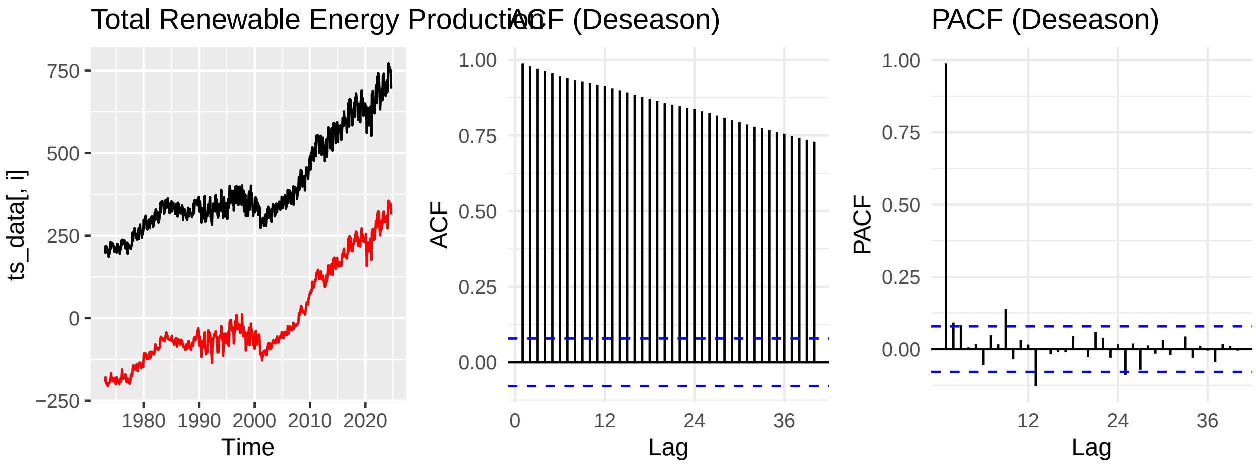
Q9

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?

```
deseason_acf_pacf <- function(seas_means_model, i){
  deseason_ts <- deseason_series(seas_means_model,i)$deseason_ts
  deseason_acf <- Acf(deseason_ts,
                              lag.max=40, type="correlation",
                              plot=FALSE)
  acf_plot <- autoplot(deseason_acf) +
    ggtitle(("ACF (Deseason)")) +
    theme_minimal()
  deseason_pacf <- Pacf(deseason_ts,
                        lag.max=40, plot=FALSE)
  pacf_plot <- autoplot(deseason_pacf) +
    ggtitle(glue("PACF (Deseason)")) +
    theme_minimal()
  return(list(acf_plot, pacf_plot))
}
```

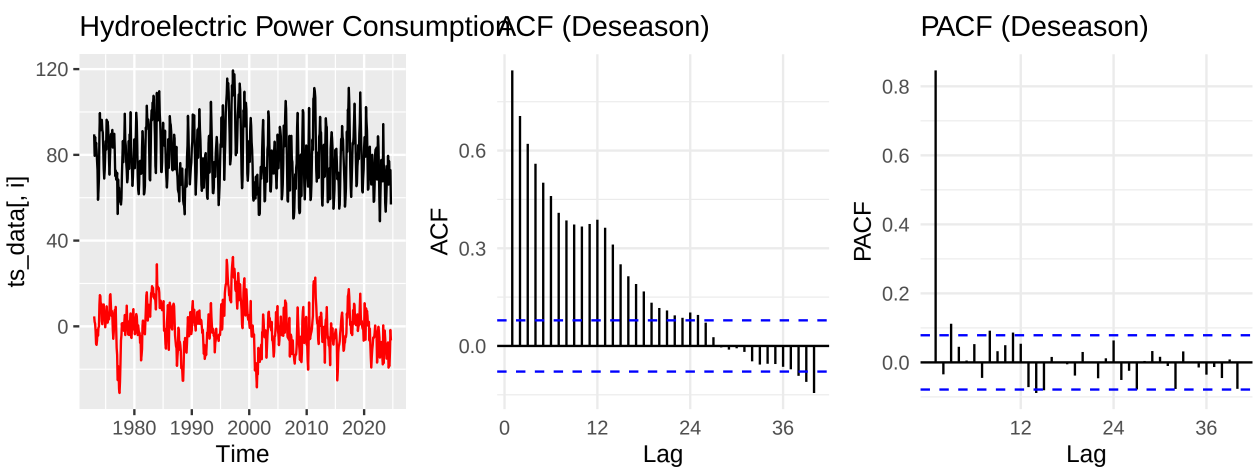
```
Renewable_deseason_acf <- deseason_acf_pacf(R_seas_means_model, 1)[[1]]
Renewable_deseason_pacf <- deseason_acf_pacf(R_seas_means_model, 1)[[2]]

# Arrange the plots side by side using plot_grid()
```

```r
plot_grid(R_deseason_plot, Renewable_deseason_acf, Renewable_deseason_pacf, ncol = 3, align = 'h')
```



Renewable: The autocorrelation of the deseason ACF reduced slightly comparing to the original ACF. The remaining correlation indicates potential non-seasonal considerations. The deseason PACF generally shows smaller spikes while lag 1 remains strong, suggesting the existence of short-term autocorrelation.

```r
Hydro_deseason_acf <- deseason_acf_pacf(H_seas_means_model,2)[[1]]
Hydro_deseason_pacf <- deseason_acf_pacf(H_seas_means_model,2)[[2]]

# Arrange the plots side by side using plot_grid()
plot_grid(H_deseason_plot, Hydro_deseason_acf, Hydro_deseason_pacf, ncol = 3, align = 'h')
```



Hydroelectric: The autocorrelation of the deseason ACF decays more quickly than the original ACF. The deseason PACF shows smaller spikes, indicating less seasonal dependence.