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

Assignment 3 - Due date 02/04/25

Alex Lopez

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.\

```r
#Load/install required package here
library(forecast)
```

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

```r
library(tseries)
library(dplyr)
```

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

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

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

```
library(here)
```

```
## here() starts at /home/guest/ENERGY797
```

```
library(Kendall)
library(ggplot2)
library(cowplot)

#load packages to import Excel files
library(readxl)
library(openxlsx)

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

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

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

```
## New names:
## * `` -> `...1`
```

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

```r
#assign column names
colnames(energy_data) <- read_col_names

#select columns
energy_df <- energy_data %>%
  select(5, 6)

#check first few rows
head(energy_df)
```

```
## # A tibble: 6 x 2
##   `Total Renewable Energy Production` `Hydroelectric Power Consumption`
##                                 <dbl>                             <dbl>
## 1                                220.                              89.6
## 2                                197.                              79.5
## 3                                219.                              88.3
## 4                                209.                              83.2
## 5                                216.                              85.6
## 6                                208.                              82.1
```

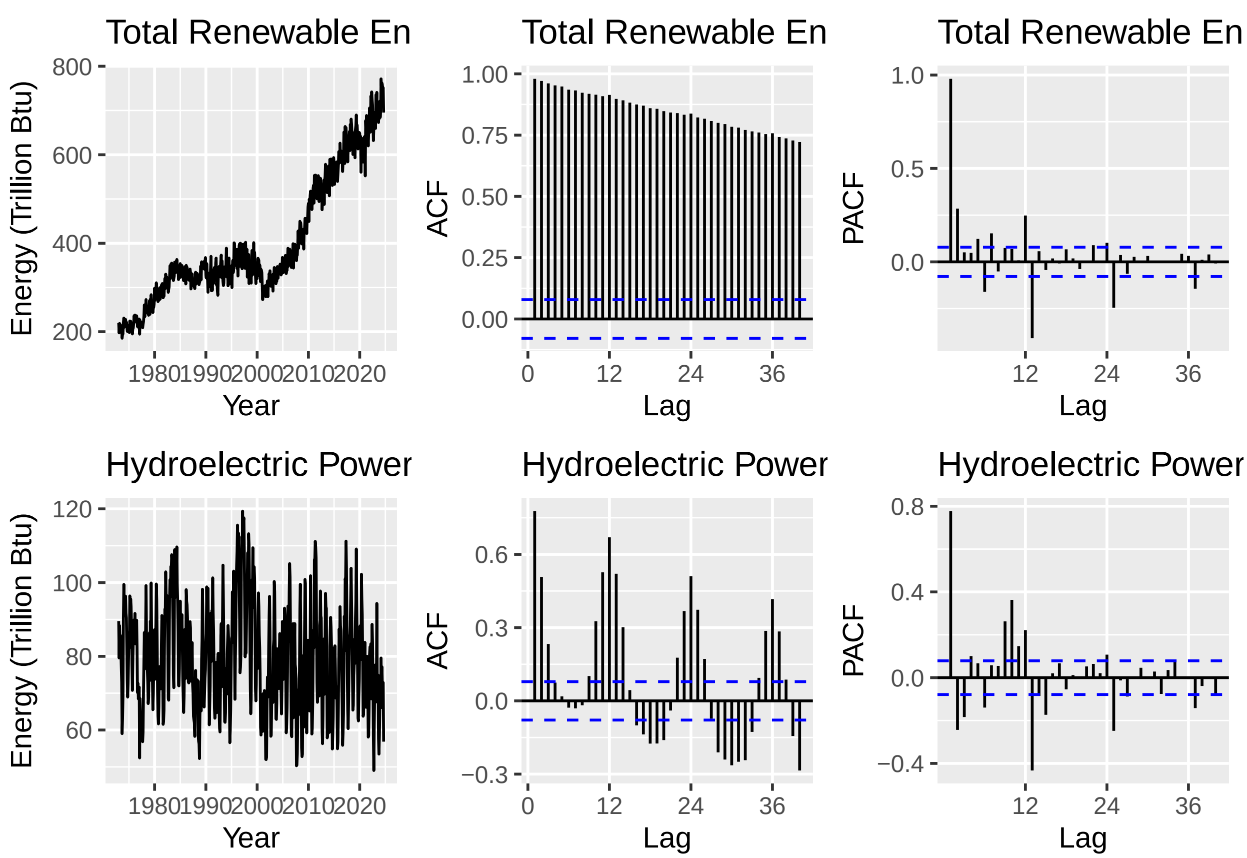
##Trend Component

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)

```r
#transform data frame to time series object
ts_energy <- ts(energy_df, start = c(1973,1), frequency = 12)

#create empty lists to store plots for Renewable Energy and Hydroelectric Power Consumption
time_series_plots <- list()
acf_plots <- list()
pacf_plots <- list()

#loop through both columns
for (i in 1:2) {
```

```
  #time series
  time_series_plots[[i]] <- autoplot(ts_energy[, i]) +
    ggtitle(colnames(ts_energy)[i]) +
    xlab('Year') +
    ylab('Energy (Trillion Btu)')

  #ACF
  acf_plots[[i]] <- autoplot(Acf(ts_energy[, i], lag.max = 40, plot = FALSE)) +
    ggtitle(paste(colnames(ts_energy)[i]))

  #PACF
  pacf_plots[[i]] <- autoplot(Pacf(ts_energy[, i], lag.max = 40, plot = FALSE)) +
    ggtitle(paste(colnames(ts_energy)[i]))
}

# Combine all plots in a grid: Time series, ACF, PACF for each variable side by side
combined_plots <- plot_grid(
  time_series_plots[[1]], acf_plots[[1]], pacf_plots[[1]],
  time_series_plots[[2]], acf_plots[[2]], pacf_plots[[2]],
  ncol = 3, align = "v"
)

print(combined_plots)
```



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?

There is an overall rising trend in energy production from renewable energy sources in the last 50 years. However, there is no overall trend for hydroelectric power consumption in the last 50 years - instead, there are fluctuations throughout the time period.

Q3

Use the *lm()* function to fit a linear trend to the two time series. Ask R to print the summary of the regression. Interpret the regression output, i.e., slope and intercept. Save the regression coefficients for further analysis.

```
#create a time vector
nobs <- nrow(ts_energy)
t <- 1:nobs

#initialize lists to store results
linear_models <- list()
coefficients_list <- list()

for (i in 1:2) {
  #fit a linear trend model for each time series
  linear_models[[i]] <- lm(ts_energy[, i] ~ t)

  #print the summary of the regression
  print(summary(linear_models[[i]]))

  #store the coefficients
  beta0 <- as.numeric(linear_models[[i]]$coefficients[1])  #intercept
  beta1 <- as.numeric(linear_models[[i]]$coefficients[2])  #slope

  #save the coefficients for further analysis
  coefficients_list[[colnames(ts_energy)[i]]] <- c(Intercept = beta0, Slope = beta1)
}
```

```
##
## Call:
## lm(formula = ts_energy[, i] ~ 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
##
##
## Call:
## lm(formula = ts_energy[, i] ~ 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
```

```r
#print the coefficients for both time series
coefficients_list
```

```
## $`Total Renewable Energy Production`
##   Intercept       Slope
## 176.8729264   0.7239349
##
## $`Hydroelectric Power Consumption`
##   Intercept       Slope
## 82.96766541 -0.01097734
```

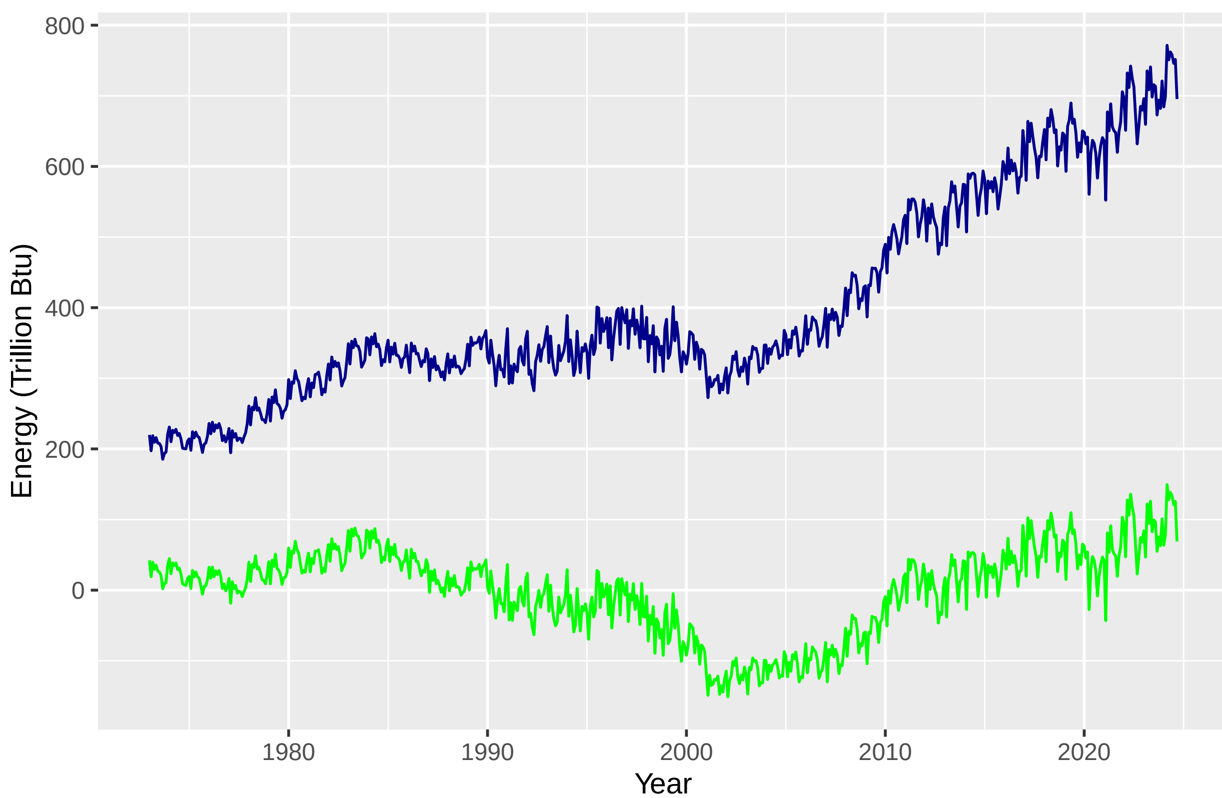
According to the summary of the regression of both time series, in January 1973 (when $t = 0$), total renewable energy production and hydroelectric power consumption were approximately 177 and 83 trillion Btu (these are the intercept values, beta0), respectively. For each additional month, total renewable energy production increased by approximately 0.72 trillion Btu and hydroelectric consumption decreased by approximately 0.01 trillion Btu. These changes per month are the slopes, or beta1, of each time series.

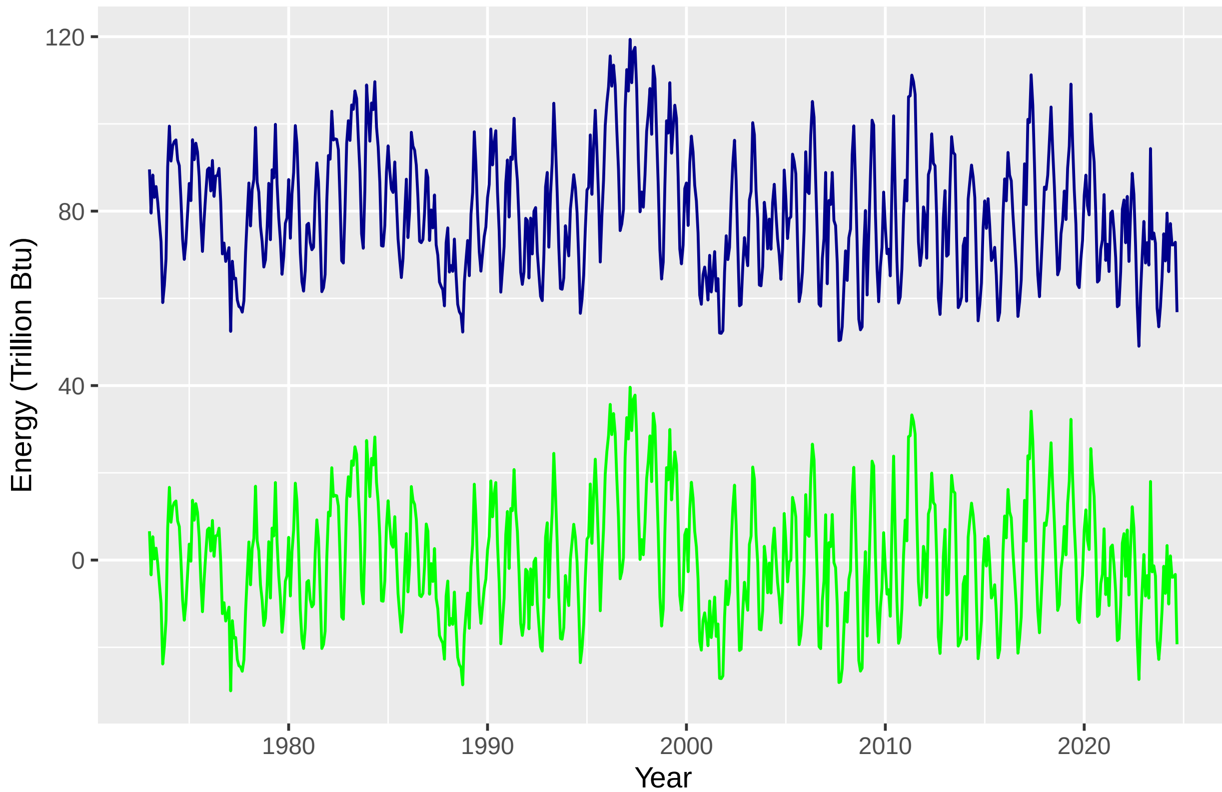
Q4

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
#initialize lists to store detrended series
ts_detrended_list <- list()

for (i in 1:2) {
  #calculate the linear trend using the regression coefficients (beta0, beta1)
  linear_trend <- coefficients_list[[colnames(ts_energy)[i]]][1] +
    coefficients_list[[colnames(ts_energy)[i]]][2] * t
```

```r
  #create a time series object for the linear trend
  ts_linear <- ts(linear_trend, start = c(1973, 1), frequency = 12)

  #detrend the time series by subtracting the linear trend from the original series
  detrended_series <- ts_energy[, i] - linear_trend

  #create a time series object for the detrended series
  ts_detrended <- ts(detrended_series, start = c(1973, 1), frequency = 12)

  #store the detrended series for later plotting
  ts_detrended_list[[colnames(ts_energy)[i]]] <- ts_detrended
}

#plot the original time series and detrended series for comparison
for (i in 1:2) {
  print(
    autoplot(ts_energy[, i], color = "darkblue") +
      autolayer(ts_detrended_list[[colnames(ts_energy)[i]]], color = "green") +
      xlab("Year") +
      ylab("Energy (Trillion Btu)")
  )
}
```





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?

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.

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?

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?

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?