ENV 790.30 - Time Series Analysis for Energy Data | Spring 2022

Assignment 3 - Due date 02/08/22

Rob Kravec

Questions

Consider the same data you used for A2. The data comes from the US Energy Information and Administration and corresponds to the January 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(tidyverse)
library(forecast)
library(Kendall)
library(tseries)
library(readxl)
library(patchwork)
```

First, I'll create the time series object requested, pulling code from A02.

```
# Read in data
file_path = paste0('../Data/Table_10.1_Renewable_Energy_Production_and',
                   '_Consumption_by_Source.xlsx')
data <- read_excel(path = file_path, sheet = "Monthly Data", skip = 10,
                   na = "Not Available")

# Remove first row, which contains units for each column
data <- data[-1, ]

# Rename relevant columns
data <- data %>%
  rename(Biomass_prod = 'Total Biomass Energy Production',
         Renewable_prod = 'Total Renewable Energy Production',
         Hydro_consumption = 'Hydroelectric Power Consumption')

# Select columns
data_small <- data %>%
  select(Biomass_prod, Renewable_prod,
         Hydro_consumption)

# Convert data types to numeric
data_small <- sapply(data_small, as.numeric) %>%
  as_tibble()

# Create df for plotting
```

```r
plot_df <- cbind(Month = data$Month, data_small)

# Create ts object
data_ts <- ts(data = data_small, start = c(1973, 1), end = c(2021, 9),
              frequency = 12)

# Show first 6 rows of ts object
head(data_ts)
```

```
##          Biomass_prod Renewable_prod Hydro_consumption
## Jan 1973      129.787        403.981           272.703
## Feb 1973      117.338        360.900           242.199
## Mar 1973      129.938        400.161           268.810
## Apr 1973      125.636        380.470           253.185
## May 1973      129.834        392.141           260.770
## Jun 1973      125.611        377.232           249.859
```

Trend Component

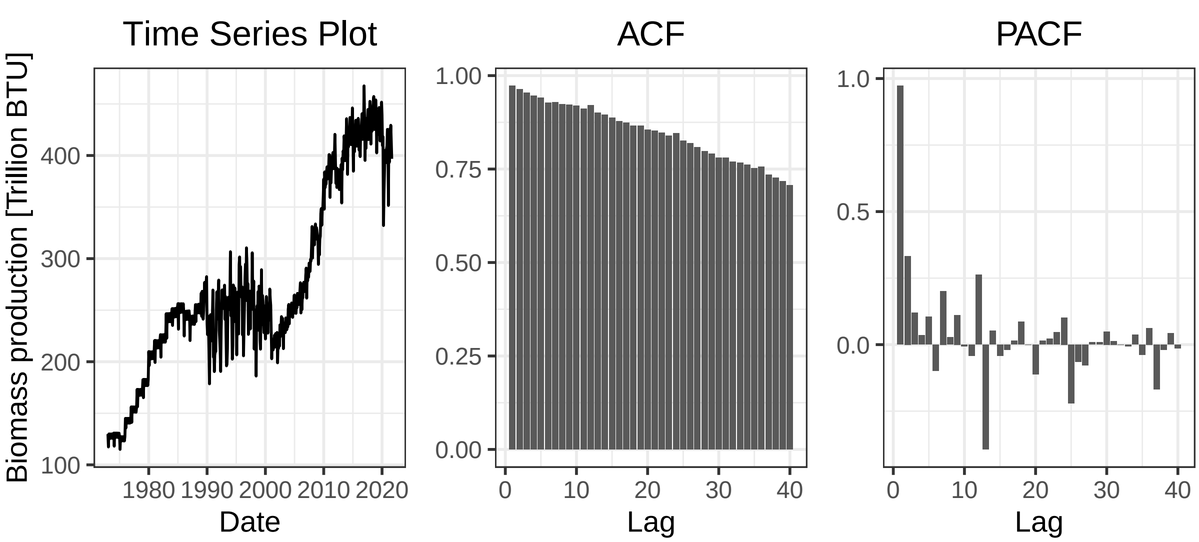
Q1

Create a plot window that has one row and three columns. And then for each object on your data frame, fill the plot window with time series plot, ACF and PACF. You may use the some code form A2, but I want all three plots on the same window this time. (Hint: use par() function)

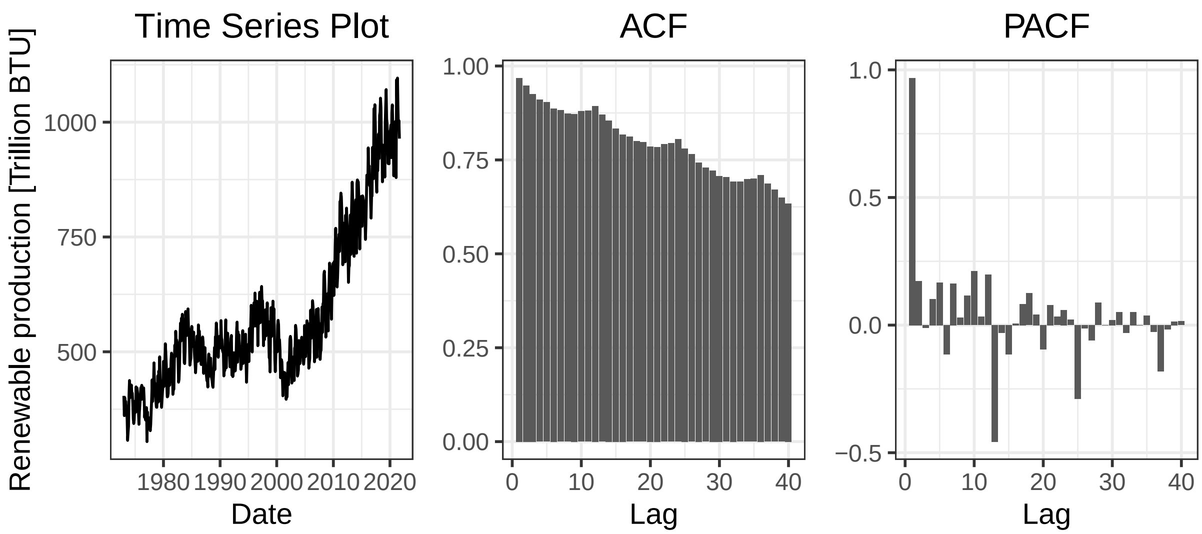
```r
# Define functions to create desired plot panel
plot_acf <- function(ts, lag_amt, title = "ACF") {
  # Prepare data
  acf_data <- data.frame(lag = 1:lag_amt,
                         acf = Acf(ts, lag.max = lag_amt,
                                   plot = F)$acf[2:(lag_amt + 1)])

  # Create plot
  acf_plt <- ggplot(data = acf_data, mapping = aes(x = lag, y = acf)) +
    geom_bar(stat = 'identity') +
    labs(x = 'Lag', y = '', title = title) +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5),
          axis.title.y = element_blank())

  # Return plot
  return(acf_plt)
}

plot_pacf <- function(ts, lag_amt, title = "PACF") {
  # Prepare data
  pacf_data <- data.frame(lag = 1:lag_amt,
                          pacf = pacf(ts,
                                      plot = F,
                                      lag.max = lag_amt)$acf)

  # Create plot
  pacf_plt <- ggplot(data = pacf_data, mapping = aes(x = lag, y = pacf)) +
    geom_bar(stat = 'identity') +
    labs(x = 'Lag', y = '', title = title) +
```

```
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5),
          axis.title.y = element_blank())

  # Return plot
  return(pacf_plt)
}


plt_three <- function(plot_df, col_num, lag_amt, ts_y_lab) {
  # Designate column of interest for time series plot
  plot_df <- plot_df %>%
    mutate(ts_plot = plot_df[, col_num])

  # Create time series plot
  ts_plt <- ggplot(data = plot_df, mapping = aes(x = Month, y = ts_plot)) +
    geom_line() +
    labs(x = 'Date', y = ts_y_lab,
         title = 'Time Series Plot') +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5))

  # Create ACF plot
  acf_plt <- plot_acf(plot_df[, col_num], lag_amt)

  # Create PACF plot
  pacf_plt <- plot_pacf(plot_df[, col_num], lag_amt)

  # Return all 3 plots in a single row using patchwork syntax
  return(ts_plt + acf_plt + pacf_plt)
}
```

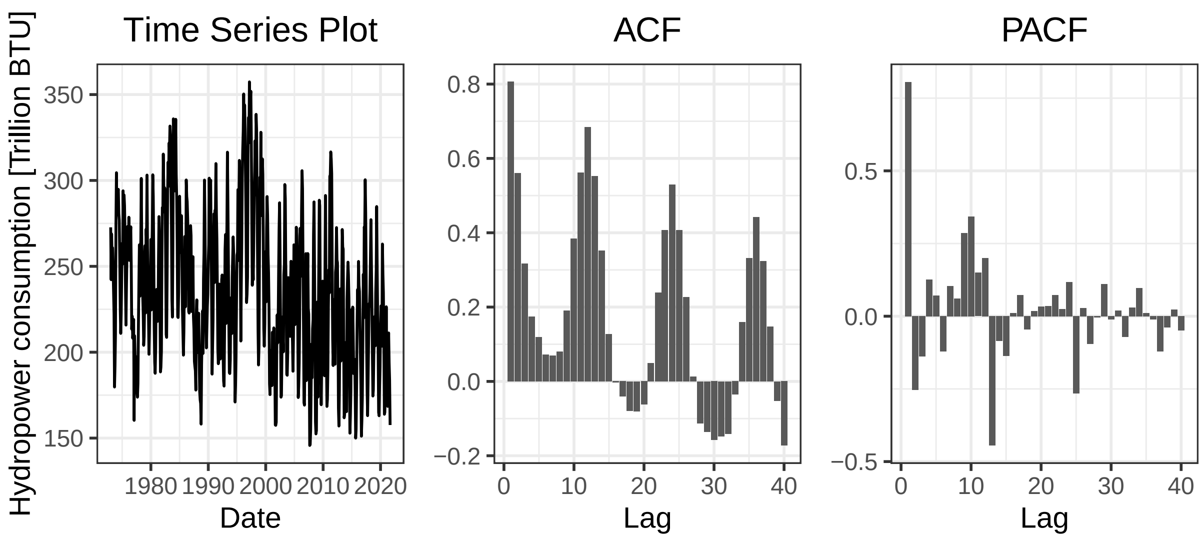
```
# Create requested plot panels
plt_three(plot_df, 2, 40, 'Biomass production [Trillion BTU]')
```



```r
plt_three(plot_df, 3, 40, 'Renewable production [Trillion BTU]')
```




```r
plt_three(plot_df, 4, 40, 'Hydropower consumption [Trillion BTU]')
```



Q2

From the plot in Q1, do the series Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption appear to have a trend? If yes, what kind of trend?

`Biomass production` and `Renewable production` appear to have a strong positive trend, while `Hydropower consumption` appears to have a weak negative trend.

I can use stationarity tests to determine what types of trends the data contains, starting with the augmented Dickey-Fuller test to check for stochastic trends.

I only reject the null hypothesis (at a cutoff of $\alpha = 0.05$) for `Hydropower consumption`, suggesting that `Biomass production` and `Renewable production` contain a unit root (i.e., have a stochastic trend).

```
name_list <- c("Biomass", "Renewables", "Hydro")
for (i in 1:3) {
  print(name_list[i])
  print(adf.test(data_ts[, i],alternative = "stationary"))
}
```

```
## [1] "Biomass"
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts[, i]
## Dickey-Fuller = -1.6325, Lag order = 8, p-value = 0.7339
## alternative hypothesis: stationary
##
## [1] "Renewables"
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts[, i]
## Dickey-Fuller = -1.4383, Lag order = 8, p-value = 0.8161
## alternative hypothesis: stationary
##
## [1] "Hydro"
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts[, i]
## Dickey-Fuller = -4.947, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

Next, I'll use the Seasonal Mann-Kendall test to check for a deterministic trend in the `Hydropower consumption` series.

Clearly, there is a deterministic trend for the `Hydropower consumption` series.

```
print(summary(SeasonalMannKendall(data_ts[, 3])))
```

```
## Score =  -4394 , Var(Score) = 159104
## denominator =  13968
## tau = -0.315, 2-sided pvalue =< 2.22e-16
## NULL
```

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.

```
# Define time vector
t <- 1:nrow(data)

# Initialize dataframe with output from regressions
df_q3 <- data.frame(Intercept = c(0, 0, 0),
                    Slope = c(0, 0, 0),
                    row.names = c("Biomass", "Renewables", "Hydro"))
```

```r
# Perform regressions and print summaries, as instructed
for (i in 1:3) {
  model <- lm(data_ts[, i] ~ t)
  print(colnames(data_ts)[i])
  print(summary(model))
  df_q3[i, 1] <- summary(model)$coefficients[1]
  df_q3[i, 2] <- summary(model)$coefficients[2]
}
```

```
## [1] "Biomass_prod"
##
## Call:
## lm(formula = data_ts[, i] ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -101.892  -24.306    4.932   33.103   82.292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.348e+02  3.282e+00   41.07   <2e-16 ***
## t           4.744e-01  9.705e-03   48.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.64 on 583 degrees of freedom
## Multiple R-squared:  0.8039, Adjusted R-squared:  0.8035
## F-statistic:  2389 on 1 and 583 DF,  p-value: < 2.2e-16
##
## [1] "Renewable_prod"
##
## Call:
## lm(formula = data_ts[, i] ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -230.488  -57.869    5.595   62.090  261.349
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 323.18243    8.02555   40.27   <2e-16 ***
## t             0.88051    0.02373   37.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96.93 on 583 degrees of freedom
## Multiple R-squared:  0.7025, Adjusted R-squared:  0.702
## F-statistic:  1377 on 1 and 583 DF,  p-value: < 2.2e-16
##
## [1] "Hydro_consumption"
##
## Call:
## lm(formula = data_ts[, i] ~ t)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -94.892 -31.300  -2.414  27.876 121.263
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 259.18303    3.47464  74.593  < 2e-16 ***
## t            -0.07924    0.01027  -7.712 5.36e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.97 on 583 degrees of freedom
## Multiple R-squared:  0.09258,    Adjusted R-squared:  0.09103
## F-statistic: 59.48 on 1 and 583 DF,  p-value: 5.364e-14
```

```
# Display results in tabular format
df_q3
```

```
##            Intercept       Slope
## Biomass     134.7897  0.47438290
## Renewables  323.1824  0.88050647
## Hydro       259.1830 -0.07924154
```

Regression interpretation:

- **Biomass**: Starts at 134.79 Trillion BTU in Jan 1973 and increases by 0.474 Trillion BTU each month
- **Renewables**: Starts at 323.182 Trillion BTU in Jan 1973 and increases by 0.881 Trillion BTU each month
- **Hydropower**: Starts at 259.183 Trillion BTU in Jan 1973 and decreases by 0.079 Trillion BTU each month

Q4

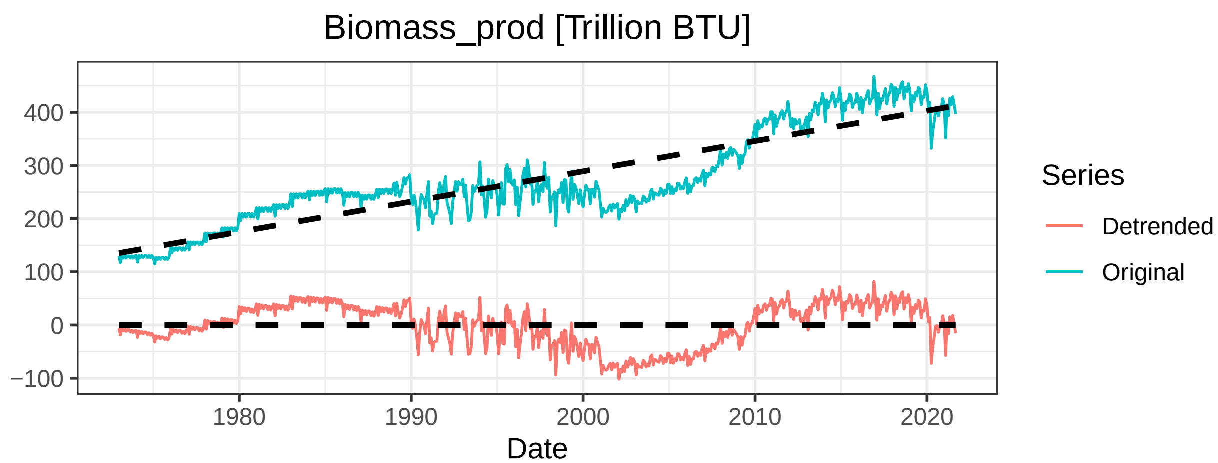
Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

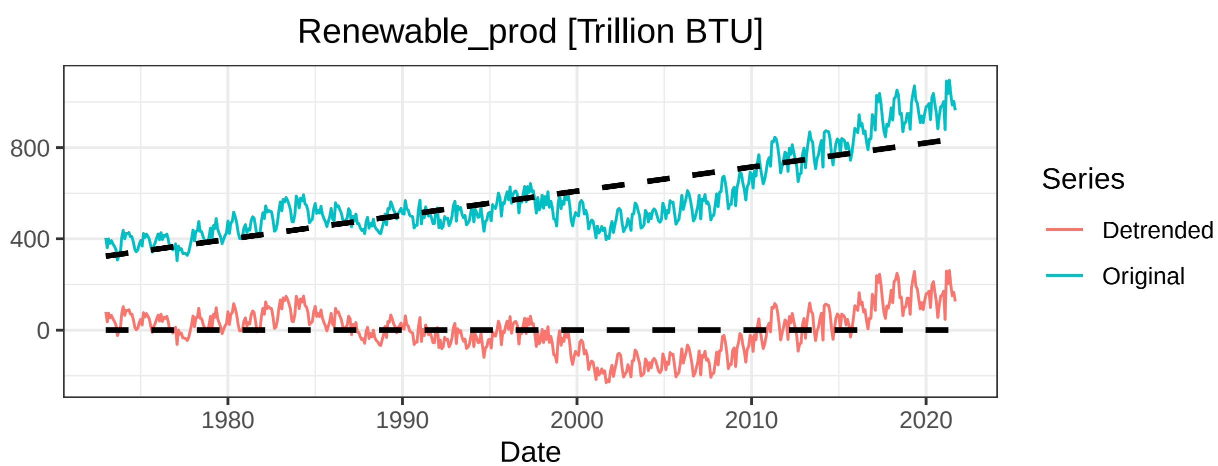
Below, I plot the original time series and detrended time series (both with trend lines) for the 3 variables of interest. For each, the detrended series have approximately horizontal trend lines near y = 0.

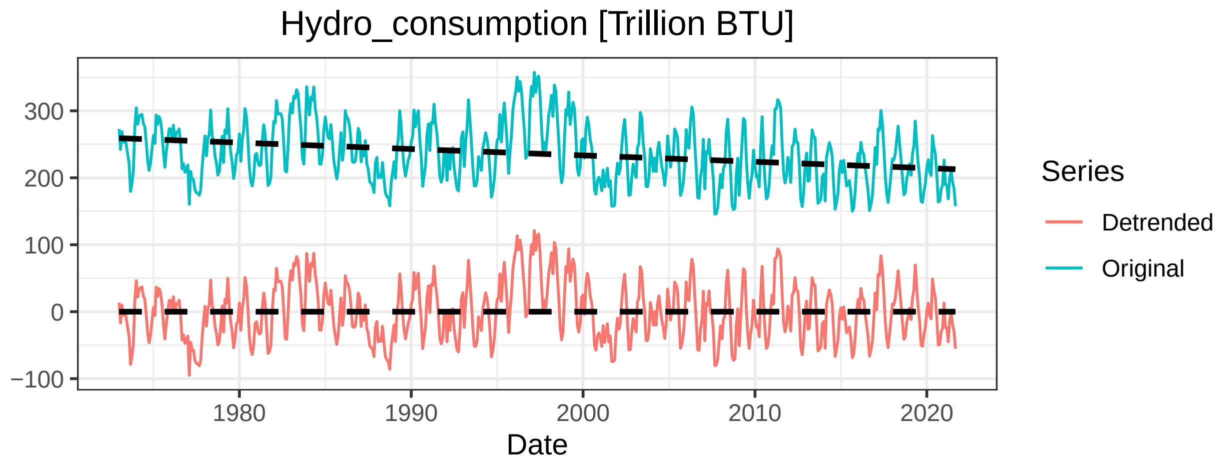
There still do appear to be some patterns in the detrended data (e.g., cycles over long time horizons for `biomass production`, seasonal variation for `hydropower consumption`), but there is at least no longer a clear up/downward trend in each series

```
# Generate plots in a loop
for (i in 1:3) {

  # Detrend series (and save results for next exercise)
  detrended <- data_ts[, i] - (t * df_q3[i, 2] + df_q3[i, 1])
  detrended_name <- paste0("detrended", i)
  assign(detrended_name, detrended)

  # Create dataframe for plotting
  plot_df_4 <- data.frame(Month = data$Month,
                          Original = data_ts[, i] %>% as.numeric(),
                          Detrended = detrended %>% as.numeric())

  # Generate plot
  plt <- ggplot(data = plot_df_4, mapping = aes(x = Month)) +
```

```
    geom_line(mapping = aes(y = Original, color = "Original")) +
    geom_smooth(mapping = aes(y = Original), method = 'lm', color = 'black',
                se = F, linetype = 2) +
    geom_line(mapping = aes(y = Detrended, color = "Detrended")) +
    geom_smooth(mapping = aes(y = Detrended), method = 'lm', color = 'black',
                se = F, linetype = 2) +
    labs(x = "Date",
         title = paste0(colnames(data_small[, i]), " [Trillion BTU]"),
         color = "Series") +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5), axis.title.y = element_blank())

  # Display plot
  print(plt)
}
```







Q5

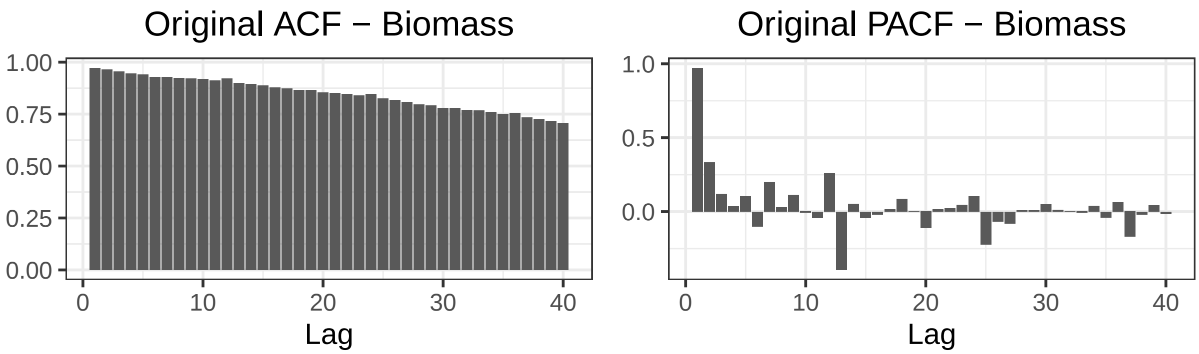
Plot ACF and PACF for the detrended series and compare with the plots from Q1. Did the plots change? How?

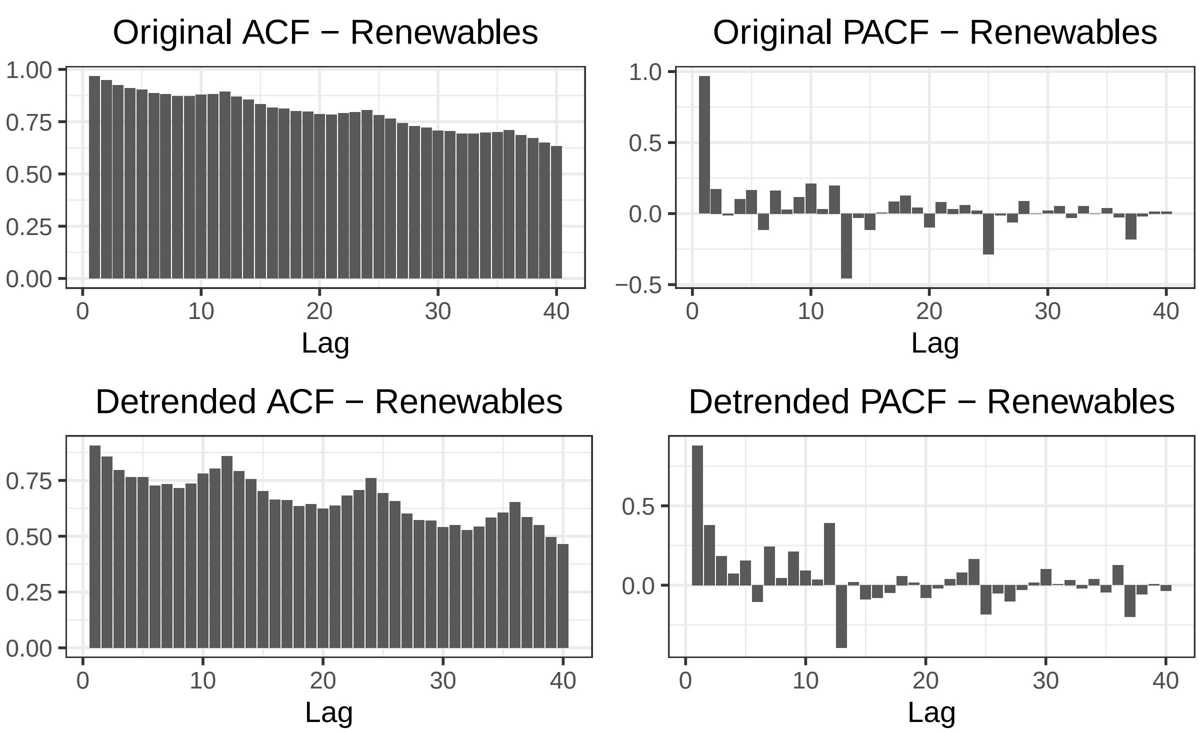
For this question, I create a 2x2 grid of plots for each variable of interest, where the top row of each grid shows the original ACF and PACF plots, while the bottom row shows the detrended ACF and PACF plots.

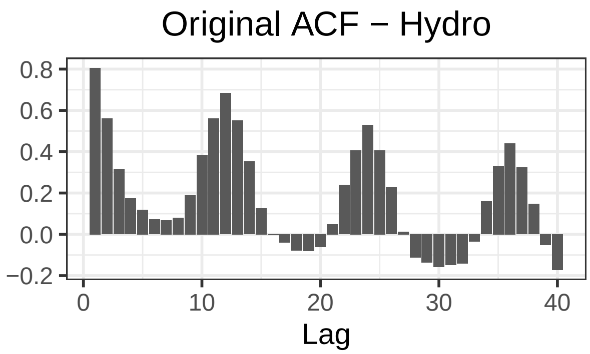
Some observations:

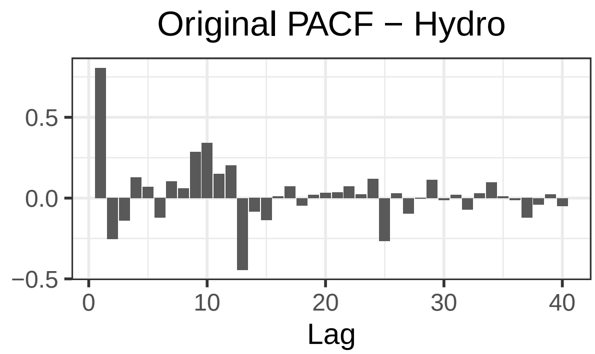
- For all three variables, detrended ACF values appear to be slightly lower than original ACF values
- It is more challenging to diagnose a consistent difference in PACF (aside from lag 1). I can at least say that the PACF plots look different after detrending and that I'm unable to pick up on a pattern
- Seasonal components appear more pronounced after detrending. This effect is particularly clear for the detrended `renewables` plot but is present for all three variables

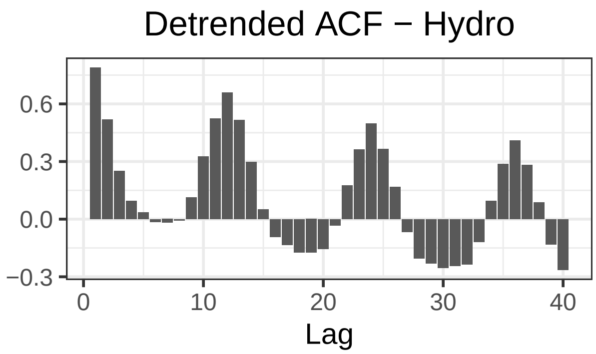
```
# Create lists that enable plotting in a loop
dfs_5 <- list(detrended1, detrended2, detrended3)

# Make plots
for (i in 1:3) {
  print((plot_acf(data_ts[, i], 40, paste0("Original ACF - ", name_list[i])) +
      plot_pacf(data_ts[, i], 40, paste0("Original PACF - ", name_list[i]))) /
     (plot_acf(dfs_5[[i]], 40, paste0("Detrended ACF - ", name_list[i])) +
        plot_pacf(dfs_5[[1]], 40, paste0("Detrended PACF - ", name_list[i]))))
}
```

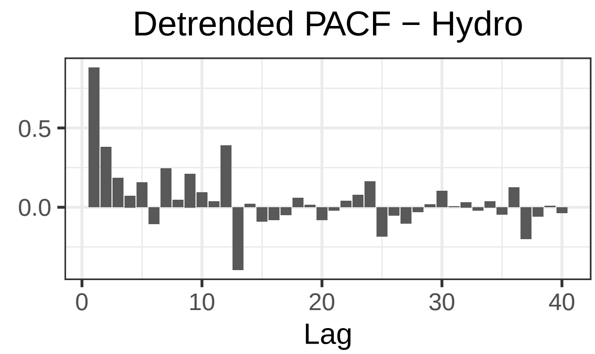

Original ACF − Biomass
Original PACF − Biomass
Detrended ACF − Biomass
Detrended PACF − Biomass
1.00
0.75
0.50
0.25
0.00
1.0
0.5
0.0
0
10
20
30
40
Lag


Original ACF − Renewables
Original PACF − Renewables
Detrended ACF − Renewables
Detrended PACF − Renewables
1.00
0.75
0.50
0.25
0.00
1.0
0.5
0.0
−0.5
0
10
20
30
40
Lag









Seasonal Component

Set aside the detrended series and consider the original series again from Q1 to answer Q6 to Q8.

Q6

Do the series seem to have a seasonal trend? Which series? Use function *lm()* to fit a seasonal means model (i.e. using the seasonal dummies) to this/these time series. Ask R to print the summary of the regression. Interpret the regression output. Save the regression coefficients for further analysis.

Based on the original ACF plots, it's clear that `Hydropower` contains a seasonal trend. I originally wasn't sure whether `Renewables` contained a seasonal trend, so I fit a seasonal means model (not shown), and none of the coefficients had a p-value less than 0.05, indicating an absence of seasonal trend. I then repeated the exercise for `Biomass`, really just for fun, and found the same result. Therefore, the output and plots shown below pertain to the `Hydropower` series.

Additionally, I noted in `Q5` that the detrended series appeared to have more accentuated seasonal trends than the original series. I generated seasonal means models for detrended `Renewables` and `Biomass` series (also not shown), and some of the coefficients had significant p-values.

Interpreting the `Hydropower` regression output, the `Intercept` term corresponds to the mean value for December, and all other coefficients are adjustments from that baseline. Using a p-value threshold of $\alpha = 0.05$, we observe hydropower consumption significantly above the December baseline in January and Mar-Jun. We also observe hydropower consumption significantly below the December baseline in Aug-Nov. These results indicate the likely presence of a seasonal trend.

```
# Fit model, and display summary
i <- 3
dummies <- seasonaldummy(data_ts[, i])
model <- lm(data_ts[, i] ~ dummies)
summary(model)
```

```
##
## Call:
## lm(formula = data_ts[, i] ~ dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.253 -23.017  -3.042  21.487  99.478
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  237.841      4.892  48.616  < 2e-16 ***
## dummiesJan    13.558      6.883   1.970  0.04936 *
## dummiesFeb    -8.090      6.883  -1.175  0.24037
## dummiesMar    20.067      6.883   2.915  0.00369 **
## dummiesApr    16.619      6.883   2.414  0.01607 *
## dummiesMay    39.961      6.883   5.805 1.06e-08 ***
## dummiesJun    31.315      6.883   4.549 6.57e-06 ***
## dummiesJul    10.511      6.883   1.527  0.12732
## dummiesAug   -17.853      6.883  -2.594  0.00974 **
## dummiesSep   -49.852      6.883  -7.242 1.43e-12 ***
## dummiesOct   -48.086      6.919  -6.950 9.96e-12 ***
## dummiesNov   -32.187      6.919  -4.652 4.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.89 on 573 degrees of freedom
## Multiple R-squared:  0.4182, Adjusted R-squared:  0.4071
## F-statistic: 37.45 on 11 and 573 DF,  p-value: < 2.2e-16
```

```r
# Save regression coefficients
q6_coef <- summary(model)$coefficients[,1]
```

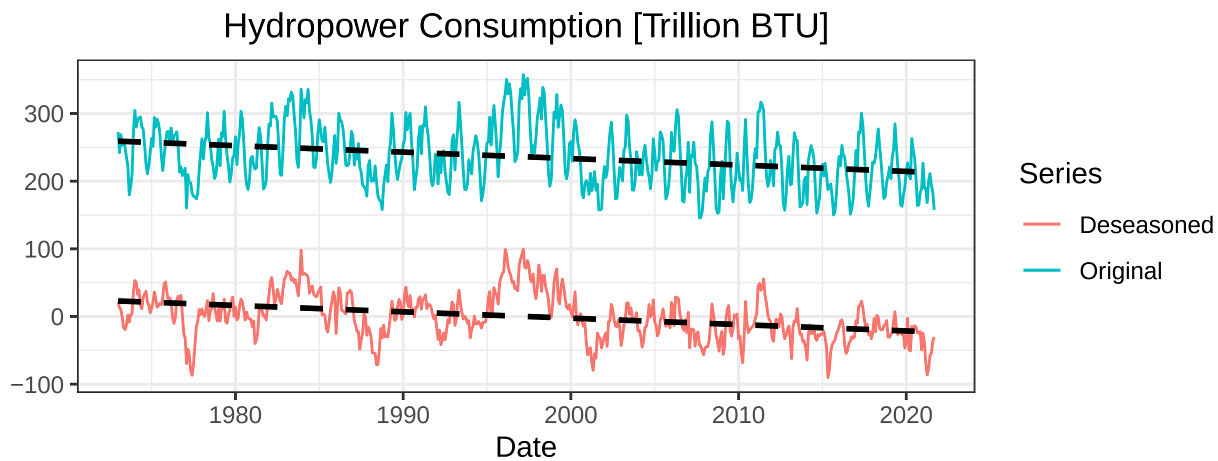
Q7

Use the regression coefficients from Q6 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

The deseasoned `hydropower` series definitely has less of the predictable choppiness that characterizes the seasonal trend in the original series. The overall shape (e.g., sustained higher levels 1995-2000) and negative trend of the original series is maintained in the deseasoned version.

```r
# Deseason hydropower consumption time series
seasonal_component <- dummies %*% q6_coef[2:12] + q6_coef[1]
deseasoned <- data_small[, 3] - seasonal_component
colnames(deseasoned) <- "Deseasoned"

# Create dataframe for plotting
plot_df_7 <- cbind(plot_df, deseasoned)

# Plot time series on same plot
ggplot(data = plot_df_7, mapping = aes(x = Month)) +
    geom_line(mapping = aes(y = Hydro_consumption, color = "Original")) +
    geom_smooth(mapping = aes(y = Hydro_consumption),
                method = 'lm', color = 'black',
                se = F, linetype = 2) +
    geom_line(mapping = aes(y = Deseasoned, color = "Deseasoned")) +
```

```
    geom_smooth(mapping = aes(y = Deseasoned), method = 'lm', color = 'black',
                se = F, linetype = 2) +
    labs(x = "Date",
         title = "Hydropower Consumption [Trillion BTU]",
         color = "Series") +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5), axis.title.y = element_blank())
```

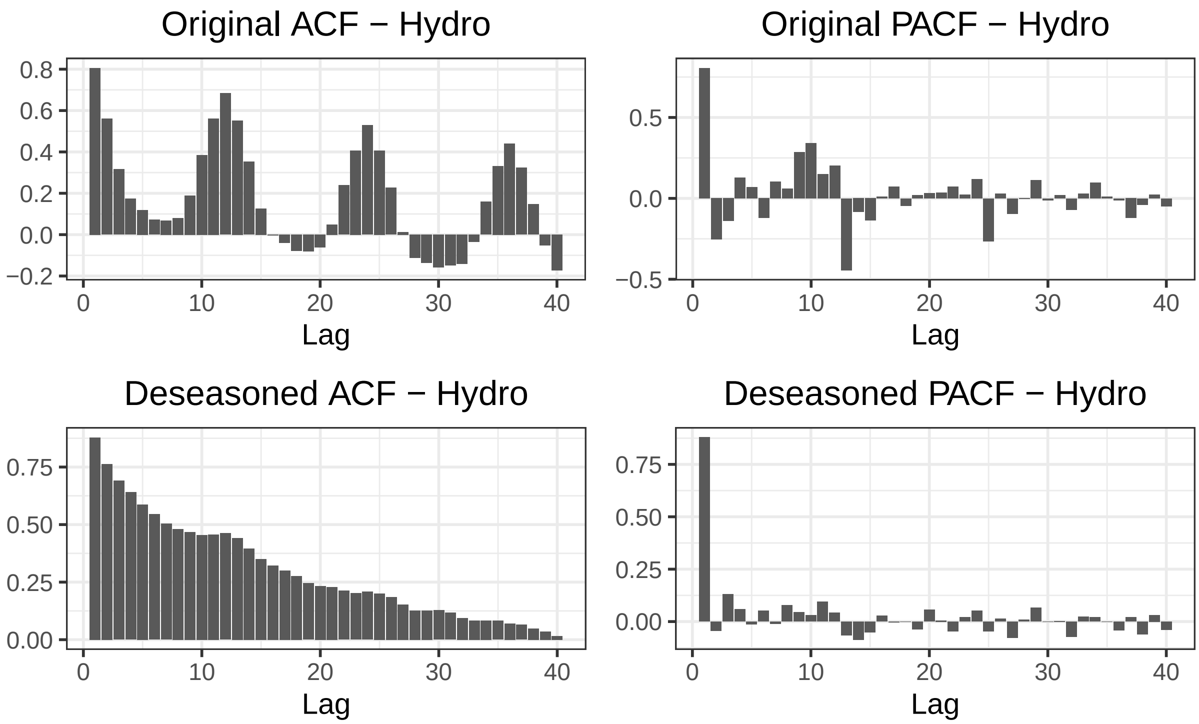


Q8

Plot ACF and PACF for the deseason series and compare with the plots from Q1. Did the plots change? How?

Yes, the plots changed a lot! The wave-like appearance in the original ACF plot is essentially gone in the deseasoned ACF plot. The deseasoned series has lower PACF values overall without any discernable wave-like pattern, suggesting that the vast majority of partial autocorrelation is coming from lag one.

```
(plot_acf(plot_df_7[, 4], 40, "Original ACF - Hydro") +
     plot_pacf(plot_df_7[, 4], 40, "Original PACF - Hydro")) /
    (plot_acf(plot_df_7[, 5], 40, "Deseasoned ACF - Hydro") +
       plot_pacf(plot_df_7[, 5], 40, "Deseasoned PACF - Hydro"))
```


Original ACF − Hydro
Original PACF − Hydro
Deseasoned ACF − Hydro
Deseasoned PACF − Hydro
Lag
Lag
Lag
Lag