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

Assignment 3 - Due date 02/01/24

Aditi Jackson

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github.

Once you have the file open on your local machine the first thing you will do is rename the file such that it includes your first and last name (e.g., "LuanaLima_TSA_A02_Sp24.Rmd"). Then change "Student Name" on line 4 with your name.

Then you will start working through the assignment by **creating code and output** that answer each question. Be sure to use this assignment document. Your report should contain the answer to each question and any plots/tables you obtained (when applicable).

Please keep this R code chunk options for the report. It is easier for us to grade when we can see code and output together. And the tidy.opts will make sure that line breaks on your code chunks are automatically added for better visualization.

When you have completed the assignment, **Knit** the text and code into a single PDF file. Submit this pdf using Sakai.

Questions

Consider the same data you used for A2 from the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumption
The data comes from the US Energy Information and Administration and corresponds to the December 2022 **Monthly** Energy Review. Once again you will work only with the following columns: Total 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.\

```
# checking working directory
getwd()
```

```
## [1] "/home/guest/ENV797_APJ_S24_NEW/Assignments/RMD"
```

```
# had issues knitting due to using relative path; changed to absolute path and it worked
# loading data using read.csv
renewable_energy_full <-
  read.csv(
  "/home/guest/ENV797_APJ_S24_NEW/Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source

# converting Month column to date object using lubridate
renewable_energy_full$Month <- ym(renewable_energy_full$Month)
```

```r
# renaming Month column to "Date"
colnames(renewable_energy_full)[colnames(renewable_energy_full) == "Month"] <- "Date"

# creating subset of data with Total Renewable Energy Production
## and Hydroelectric Power Consumption by date
renewable_energy_sub <- renewable_energy_full %>%
  select(
    Date,
    Total.Renewable.Energy.Production,
    Hydroelectric.Power.Consumption)

# Verifying data
head(renewable_energy_sub)
```

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

```r
# transforming data into ts object
# start date is Jan 1, 1973
# frequency is 12 (monthly data)
renewable_energy_ts <- ts(renewable_energy_sub,start=c(1973,1),frequency=12)
```

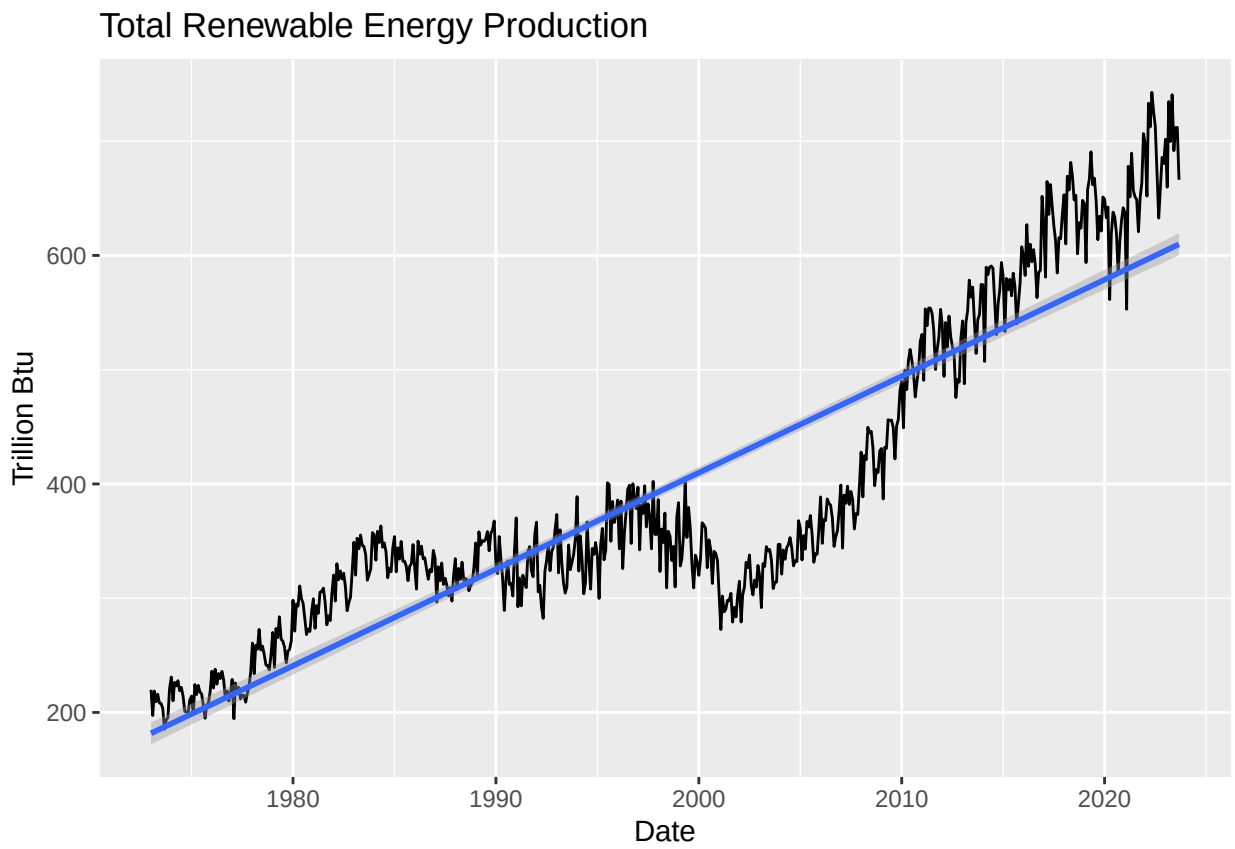
##Trend Component

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 same 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
# Total Renewable Energy Production plots

## ts plot
Renewables_tsplot <-
  ggplot(renewable_energy_sub,aes(x=Date,y=Total.Renewable.Energy.Production))+
  geom_line(color="black")+
  geom_smooth(method="lm")+
  labs(x="Date",y="Trillion Btu",title="Total Renewable Energy Production")
Renewables_tsplot
```

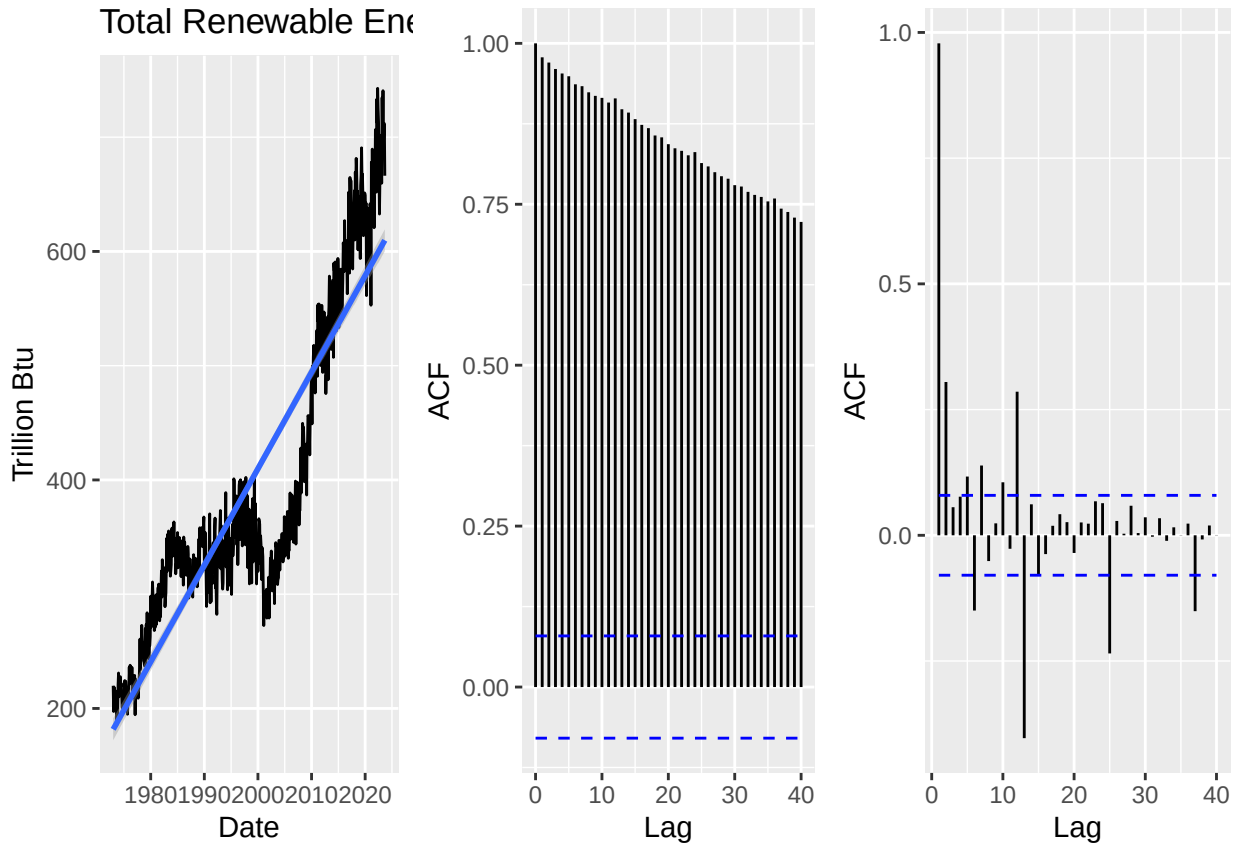
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
## acf plot
Renewables_acf <- Acf(renewable_energy_ts[,2],lag.max=40,plot=FALSE)

## pacf plot
Renewables_pacf <- Pacf(renewable_energy_ts[,2],lag.max=40,plot=FALSE)

## plotting renewable energy production on same grid
RenewableEnergy_allPlots <- plot_grid(
  autoplot(Renewables_tsplot),
  autoplot(Renewables_acf,main=NULL),
  autoplot(Renewables_pacf,main=NULL),
  nrow=1, ncol=3)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

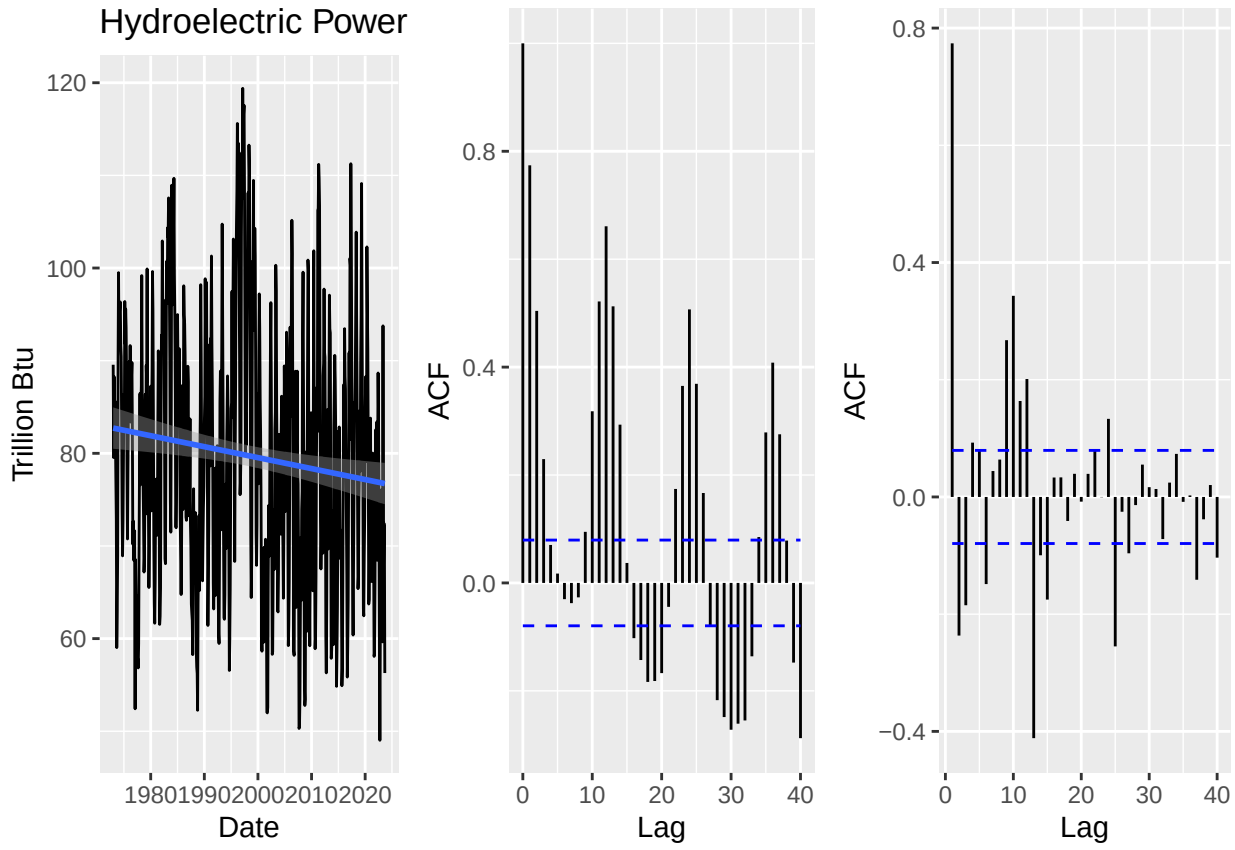
```
RenewableEnergy_allPlots
```




```
# Hydroelectric Power Consumption plots

## ts plot
Hydro_tsplot <-
  ggplot(renewable_energy_sub,aes(x=Date,y=Hydroelectric.Power.Consumption))+
  geom_line(color="black")+
  geom_smooth(method="lm")+
  labs(x="Date",y="Trillion Btu",title="Hydroelectric Power Consumption")

## acf plot
HydroConsump_acf <- Acf(renewable_energy_ts[,3],lag.max=40,plot=FALSE)

## pacf plot
HydroConsump_pacf <- Pacf(renewable_energy_ts[,3],lag.max=40,plot=FALSE)

## plotting on same grid
Hydroelectric_allPlots <- plot_grid(
  autoplot(Hydro_tsplot),
  autoplot(HydroConsump_acf,main=NULL),
  autoplot(HydroConsump_pacf,main=NULL),
  nrow=1,ncol=3)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
Hydroelectric_allPlots
```



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?

> Renewable energy production has a notable increasing trend based on the time series plot, confirmed the highly correlative values in the ACF and spikes in PACF. Hydroelectric power consumption seems to have a slightly decreasing trend based on the downward-sloping line in the time series plot. There also appears to be a seasonal component based on the continuously alternating peaks and troughs in the ACF.

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.

```
# creating vector "t" to represent number of observations
num_obs <- nrow(renewable_energy_sub)
t <- 1:num_obs


# fitting linear regression for Renewable Energy Consumption time series
Renewables_linReg <-
  lm(Total.Renewable.Energy.Production ~ t, data = renewable_energy_sub)
summary(Renewables_linReg)
```

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

```
## lm(formula = Total.Renewable.Energy.Production ~ t, data = renewable_energy_sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -148.27  -35.63   11.58   41.51  144.27
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 180.98940    4.90151   36.92   <2e-16 ***
## t             0.70404    0.01392   50.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.41 on 607 degrees of freedom
## Multiple R-squared:  0.8081, Adjusted R-squared:  0.8078
## F-statistic:  2557 on 1 and 607 DF,  p-value: < 2.2e-16
```

```
# saving coefficients
Renew_beta0 <- Renewables_linReg$coefficients[1]
Renew_beta1 <- Renewables_linReg$coefficients[2]
```

Renewable energy production: The regression results tell us that total renewable energy production increases by 0.70 trillion Btu every year and that the regression coefficients are significant at a p value less than 0.001. Based on the R-Squared value, 80.8% of the variability in renewable energy production is explained by time. Overall there appears to be a strong correlation between the dataset and time.

```
# fitting linear regression for Hydropower Production time series
Hydro_linReg <-
  lm(Hydroelectric.Power.Consumption ~ t, data = renewable_energy_sub)
summary(Hydro_linReg)
```

```
##
## Call:
## lm(formula = Hydroelectric.Power.Consumption ~ t, data = renewable_energy_sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.818 -10.620  -0.669   9.357  39.528
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 82.734747   1.140265  72.557  < 2e-16 ***
## t           -0.009849   0.003239  -3.041  0.00246 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.05 on 607 degrees of freedom
## Multiple R-squared:  0.015,  Adjusted R-squared:  0.01338
## F-statistic: 9.247 on 1 and 607 DF,  p-value: 0.002461
```

```
# saving coefficients
Hydro_beta0 <- Hydro_linReg$coefficients[1]
Hydro_beta1 <- Hydro_linReg$coefficients[2]
```

Hydroelectric power consumption: The regression results show that hydroelectric power consump-

tion decreases by 0.0098 trillion Btu each year. Beta0 is significant at a p value of 0.001 while Beta1 is less significant at a p value of 0.005. Based on the regression's R-Squared, only 1.5% of the variability in hydroelectric power consumption is explained by time. Overall there appears to be a weak correlation between the data set and time.

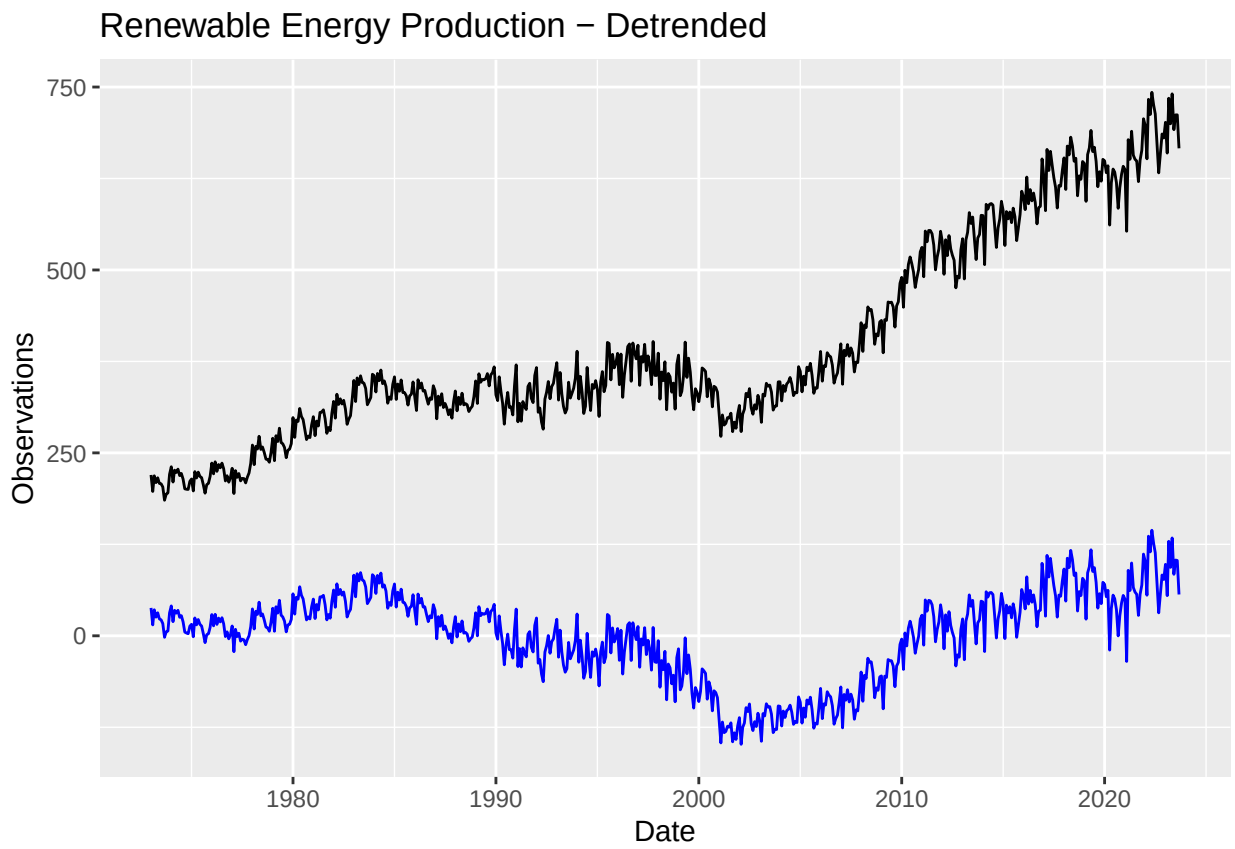
Q4

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

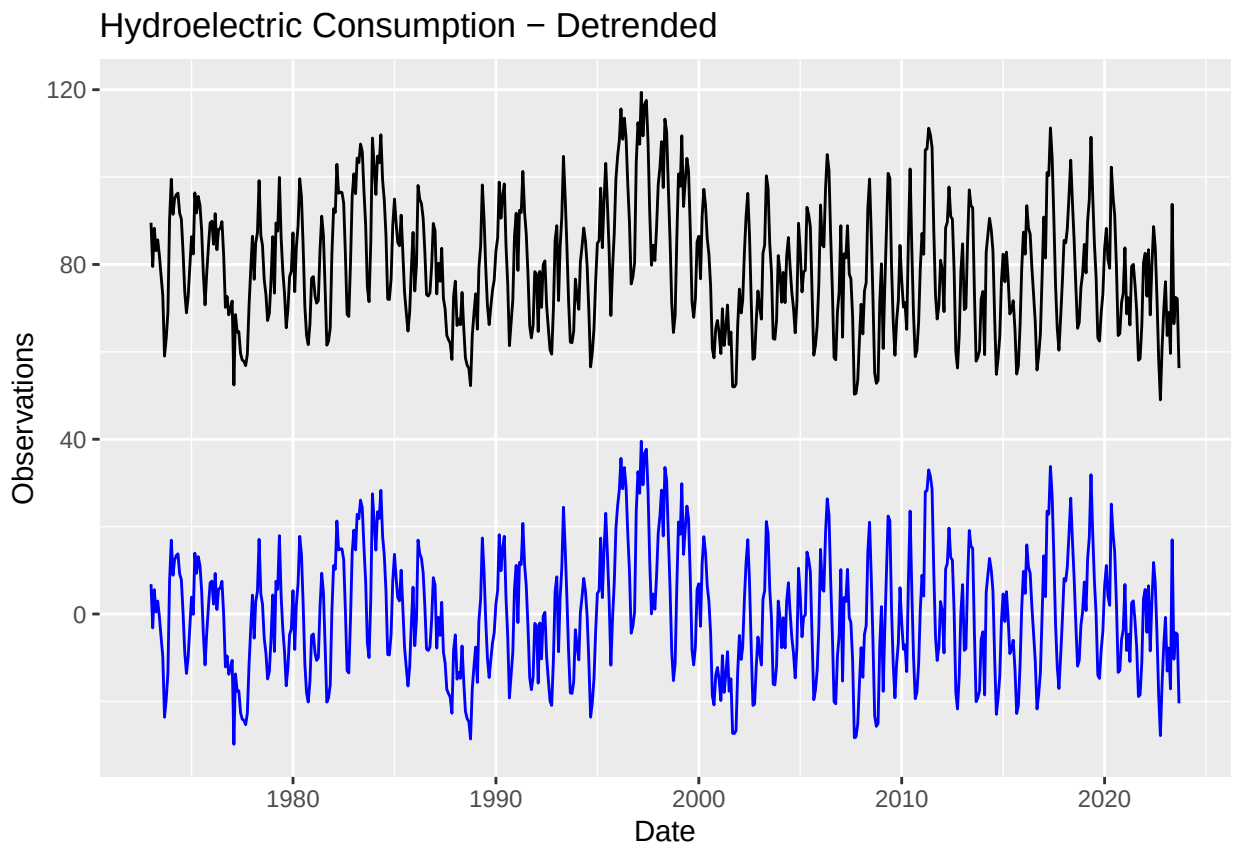
```
# Renewable Energy Production

# de-trending renewable energy production time series
Renew_detrend <- renewable_energy_sub[,2] - (Renew_beta0+Renew_beta1*t)

# creating data frame in order to plot
df_renew_detrend <-
  data_frame("date"=renewable_energy_sub$Date,
             "observed"=renewable_energy_sub[,2],
             "detrend"=Renew_detrend)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## i Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# plotting detrended series
ggplot(df_renew_detrend,aes(x=date))+
  geom_line(aes(y=observed),color="black")+
  geom_line(aes(y=detrend),color="blue")+
  labs(x="Date",
       y="Observations",
       title="Renewable Energy Production - Detrended")
```




```
# Hydroelectric Power Consumption

# de-trending hydroelectric power consumption time series
Hydro_detrend <- renewable_energy_sub[,3] - (Hydro_beta0+Hydro_beta1*t)

# creating data frame in order to plot
df_hydro_detrend <-
  data_frame("date"=renewable_energy_sub$Date,
             "observed"=renewable_energy_sub[,3],"detrend"=Hydro_detrend)

# plotting detrended series
ggplot(df_hydro_detrend,aes(x=date))+
  geom_line(aes(y=observed),color="black")+
  geom_line(aes(y=detrend),color="blue")+
  labs(x="Date",y="Observations",title="Hydroelectric Consumption - Detrended")
```

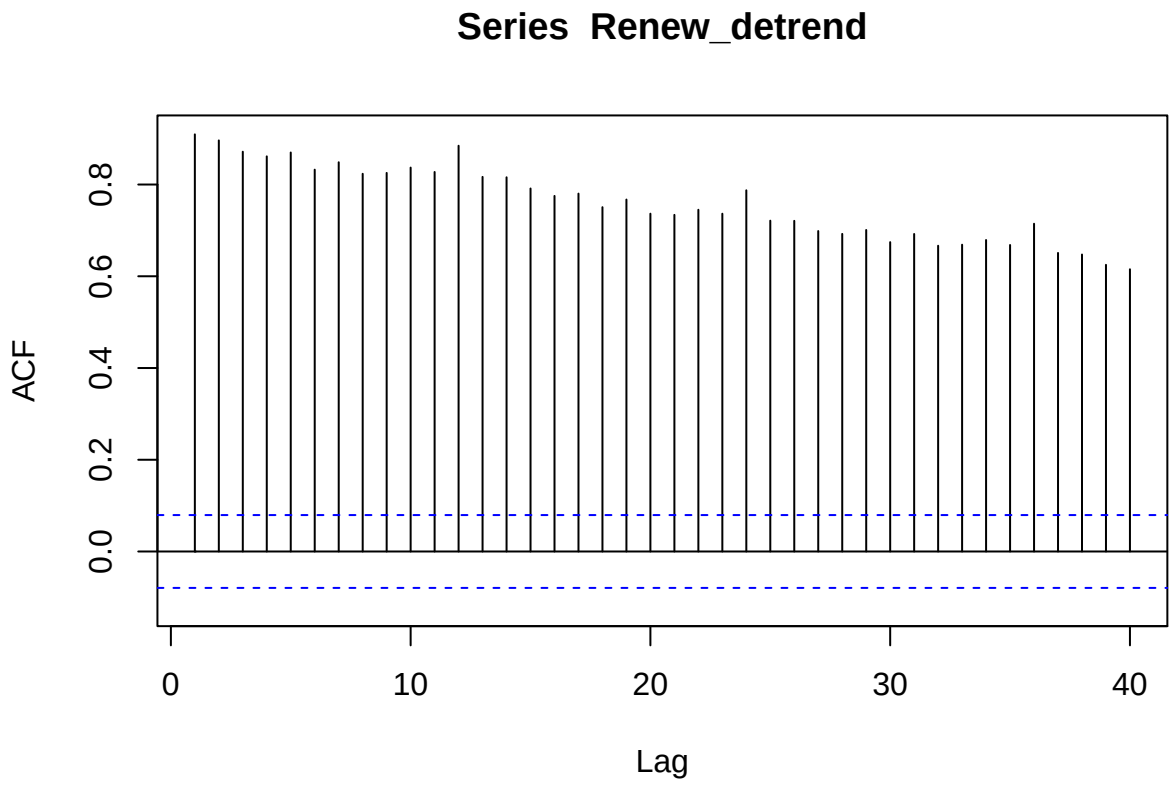


Comparing the detrended plots to the originals in Q1, the detrended lines (in blue) have a mean of zero. They look slightly different however there still appears to be a trend present.

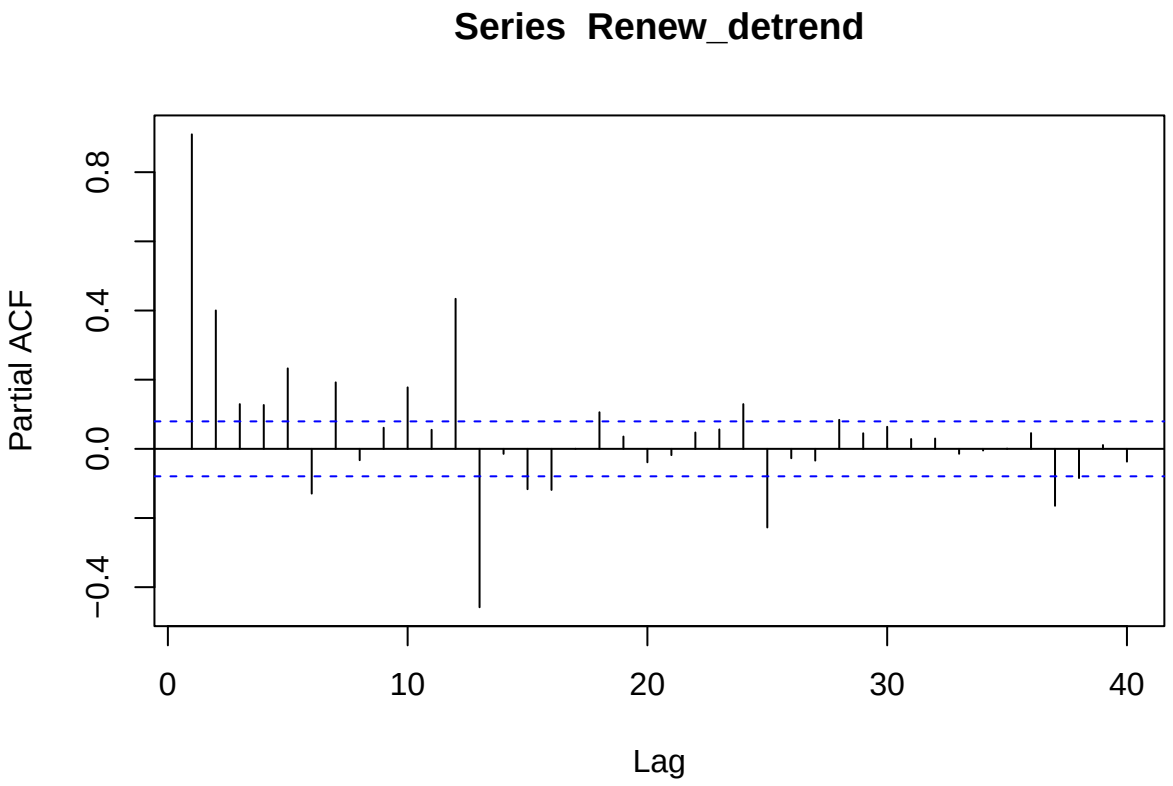
Q5

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 mot mandatory. Did the plots change? How?

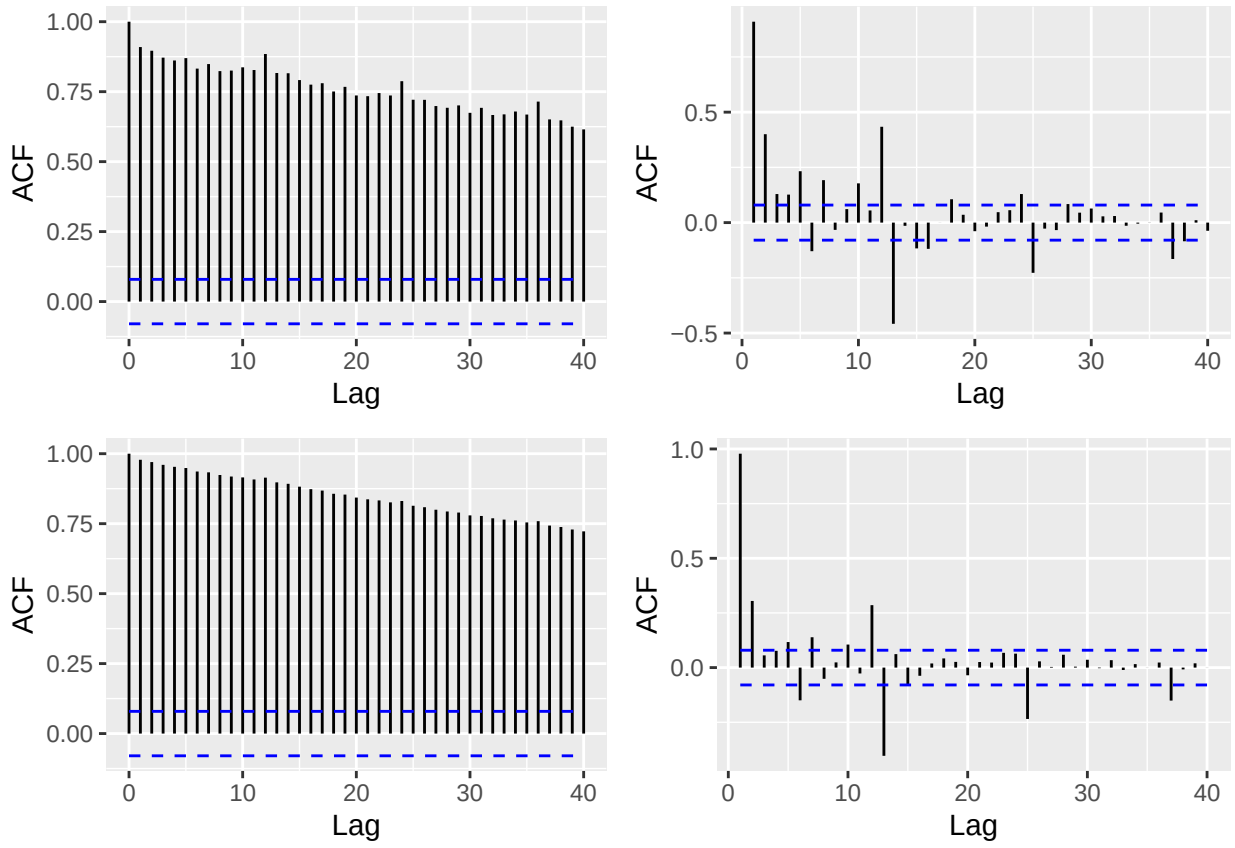
```
# Renewables - detrend ACF and PACF
Renewables_detrend_acf <- Acf(Renew_detrend,lag.max = 40)
```



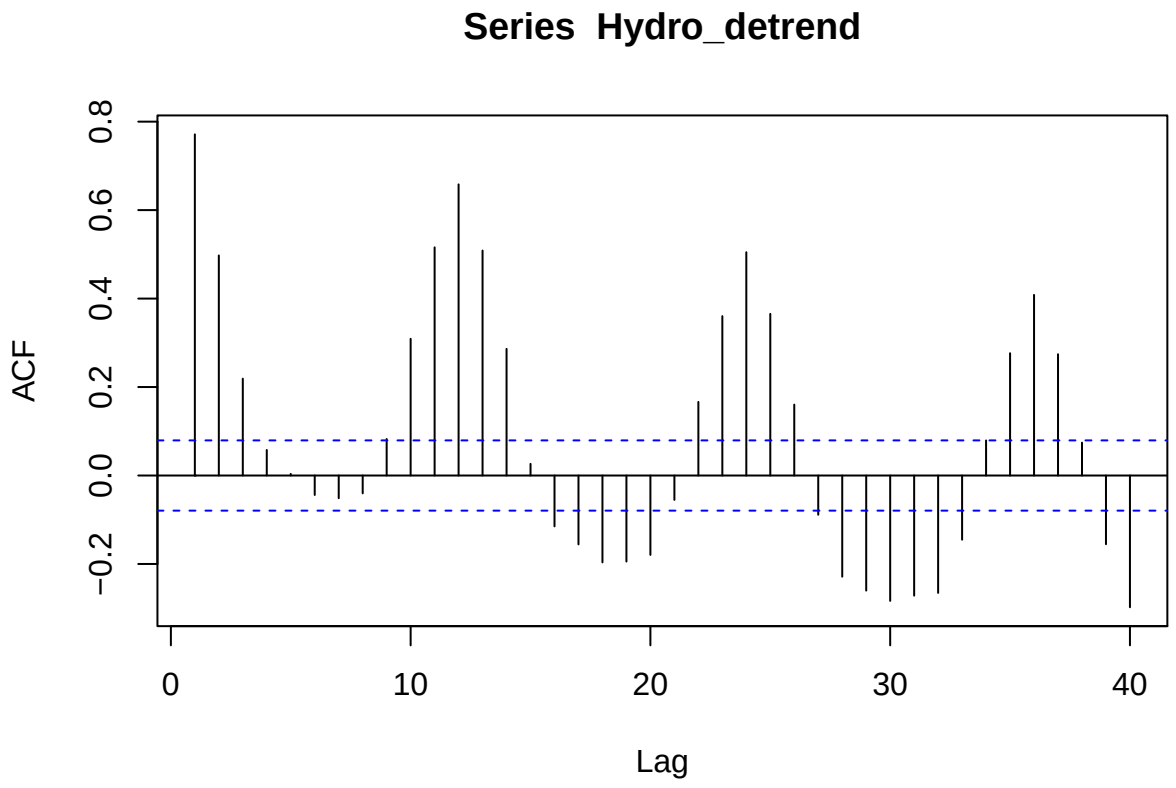

```
Renewables_detrend_pacf <- Pacf(Renew_detrend, lag.max = 40)
```



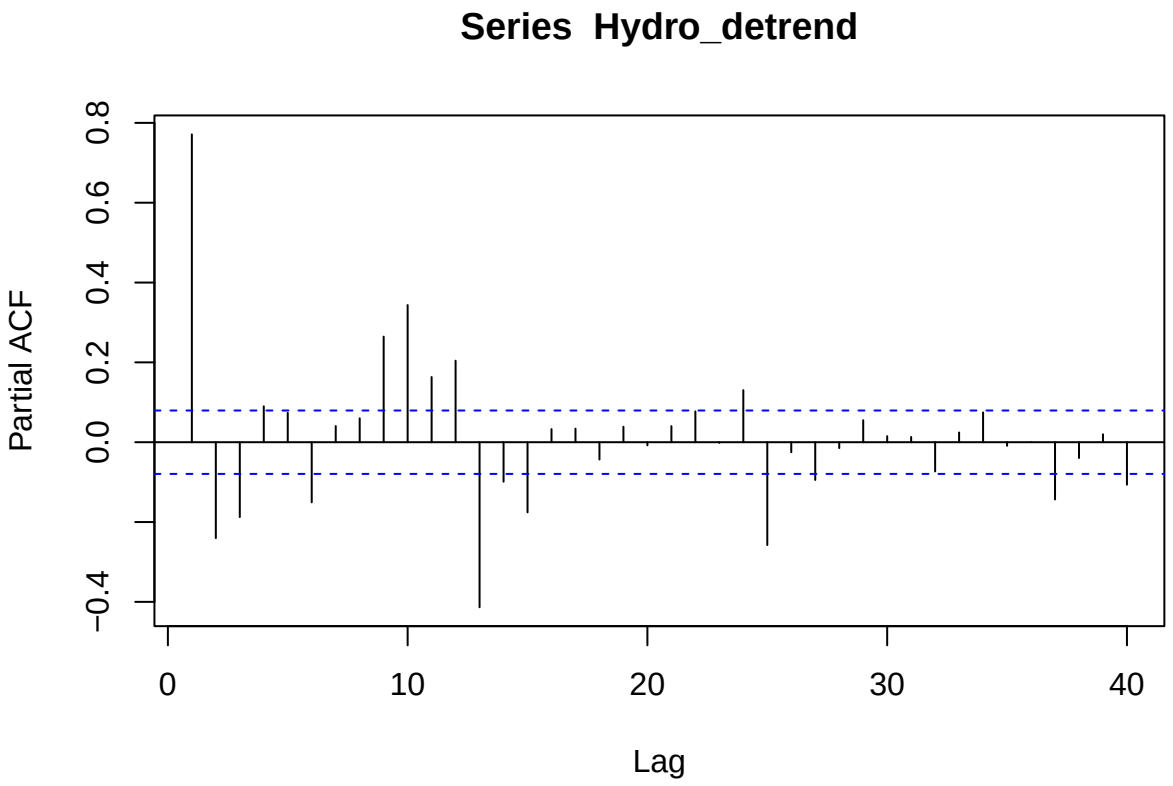

```
plot_grid(autoplot(Renewables_detrend_acf),
          autoplot(Renewables_detrend_pacf),
          autoplot(Renewables_acf),
          autoplot(Renewables_pacf))
```



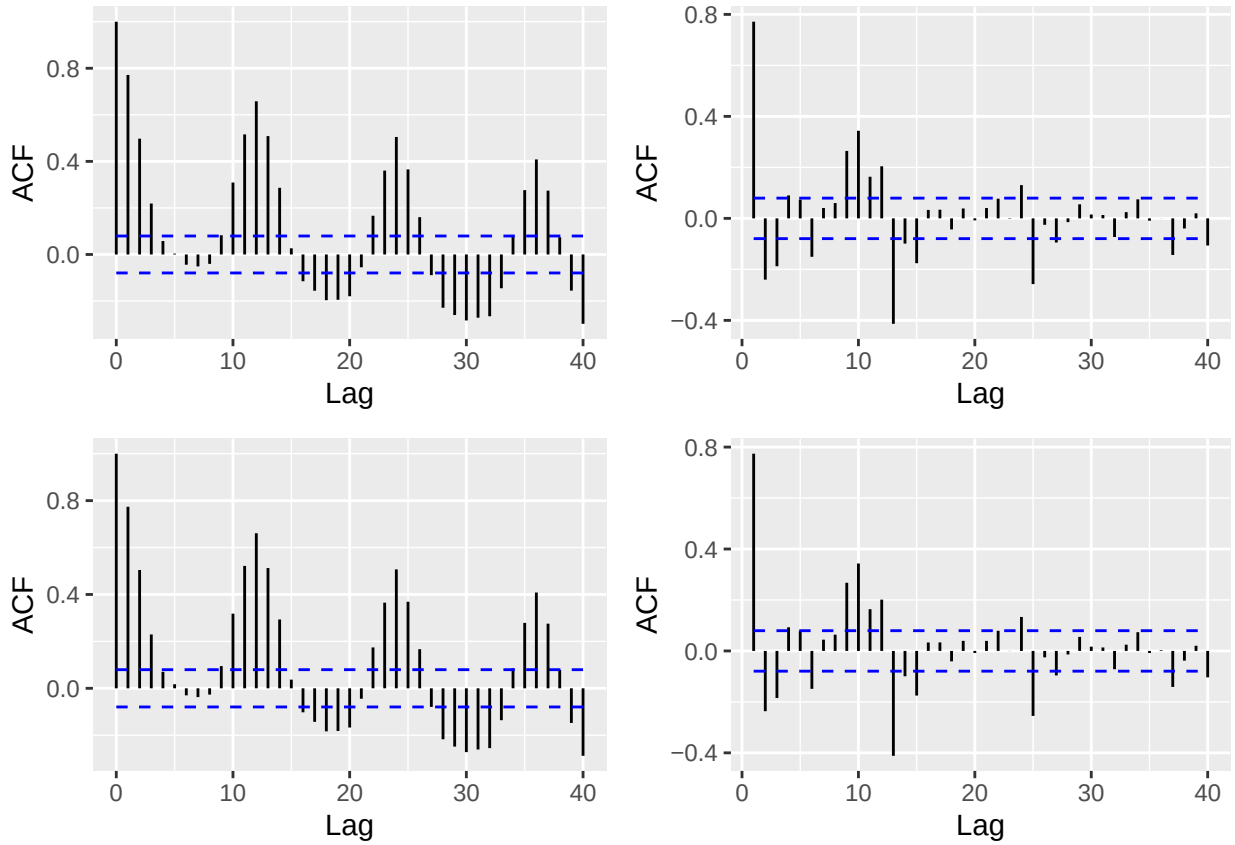

```
# Hydroelectric - detrend ACF and PACF
HydroConsump_detrend_acf <- Acf(Hydro_detrend,lag.max = 40)
```




```
HydroConsump_detrend_pacf <- Pacf(Hydro_detrend,lag.max = 40)
```




```
plot_grid(autoplot(HydroConsump_detrend_acf),
          autoplot(HydroConsump_detrend_pacf),
          autoplot(HydroConsump_acf),
          autoplot(HydroConsump_pacf))
```



For renewables production, the ACF and the PACF look very similar to Q1, but we can see a few minor changes in the graphs. The ACF coeficients still show high correlation, indicating the continued presence of trend. Detrending had some effect as evidenced by the spikes at lags 12,24,36 etc, whcih suggests some seasonality in the data. The detrended PACF looks similar to observed data, but is a bit more pronounced.This suggests the significance of certrain trends is higher in the deseasoned data. For hydroelectric consumption, the ACF and the PACF also look similar. It's hard to state that detrending had any impact.

Seasonal Component

Set aside the de-trended 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. > For renewables, the time series plot in Q1 indicates potentional seasonality based on the fluctuationing line. However, the ACF does not appear to indicate a seasonal pattern (no sine waves). > For hydro, there appears to be a seasonal component from observing the original time series plot as well as the ACF, which is sinusoidal.

Q7

Use function *lm()* to fit a seasonal means model (i.e. using the seasonal dummies) to the two time series. Ask R to print the summary of the regression. Interpret the regression output. From the results which series have a seasonal trend? Do the results match you answer to Q6?

```r
# Renewable Energy Seasonal Means Model

# creating seasonal dummy
REP_seasonal_dummy <- seasonaldummy(renewable_energy_ts[,2])

# fitting linear model to the seasonal dummy
REP_seasonal_means_model <- lm(renewable_energy_sub[,2]~REP_seasonal_dummy)
summary(REP_seasonal_means_model)
```

```
##
## Call:
## lm(formula = renewable_energy_sub[, 2] ~ REP_seasonal_dummy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -199.19  -86.35  -48.84  113.18  331.58
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              404.526     19.574  20.666   <2e-16 ***
## REP_seasonal_dummyJan      2.962     27.546   0.108    0.914
## REP_seasonal_dummyFeb    -34.476     27.546  -1.252    0.211
## REP_seasonal_dummyMar      3.929     27.546   0.143    0.887
## REP_seasonal_dummyApr     -8.695     27.546  -0.316    0.752
## REP_seasonal_dummyMay      6.645     27.546   0.241    0.809
## REP_seasonal_dummyJun     -4.198     27.546  -0.152    0.879
## REP_seasonal_dummyJul      2.460     27.546   0.089    0.929
## REP_seasonal_dummyAug     -5.026     27.546  -0.182    0.855
## REP_seasonal_dummySep    -29.119     27.546  -1.057    0.291
## REP_seasonal_dummyOct    -20.068     27.682  -0.725    0.469
## REP_seasonal_dummyNov    -20.346     27.682  -0.735    0.463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 138.4 on 597 degrees of freedom
## Multiple R-squared:  0.009296,   Adjusted R-squared:  -0.008958
## F-statistic: 0.5093 on 11 and 597 DF,  p-value: 0.8976
```

Only the intercept Beta0 appears to be significant. The R-squared value tells us that only 9.26% of variability in the data is explained by seasonality. This tells us that the seasonal means model may not be a great fit for removing seasonality from the data.

```r
# Hydroelectric Power Seasonal Means Model

# creating seasonal dummy
HPC_seasonal_dummy <- seasonaldummy(renewable_energy_ts[,3])

# fitting linear model to the seasonal dummy
HPC_seasonal_means_model <- lm(renewable_energy_sub[,3]~REP_seasonal_dummy)
summary(HPC_seasonal_means_model)
```

```
##
## Call:
## lm(formula = renewable_energy_sub[, 3] ~ REP_seasonal_dummy)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.323  -5.849  -0.468   6.243  32.290
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               80.282      1.470  54.601  < 2e-16 ***
## REP_seasonal_dummyJan      4.807      2.069   2.323  0.02050 *
## REP_seasonal_dummyFeb     -2.725      2.069  -1.317  0.18831
## REP_seasonal_dummyMar      6.825      2.069   3.298  0.00103 **
## REP_seasonal_dummyApr      5.319      2.069   2.571  0.01039 *
## REP_seasonal_dummyMay     13.922      2.069   6.729 4.02e-11 ***
## REP_seasonal_dummyJun     10.650      2.069   5.147 3.60e-07 ***
## REP_seasonal_dummyJul      3.912      2.069   1.891  0.05914 .
## REP_seasonal_dummyAug     -5.677      2.069  -2.744  0.00626 **
## REP_seasonal_dummySep    -16.797      2.069  -8.118 2.72e-15 ***
## REP_seasonal_dummyOct    -16.468      2.079  -7.920 1.17e-14 ***
## REP_seasonal_dummyNov    -10.885      2.079  -5.235 2.29e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.4 on 597 degrees of freedom
## Multiple R-squared:  0.4697, Adjusted R-squared:  0.4599
## F-statistic: 48.07 on 11 and 597 DF,  p-value: < 2.2e-16
```

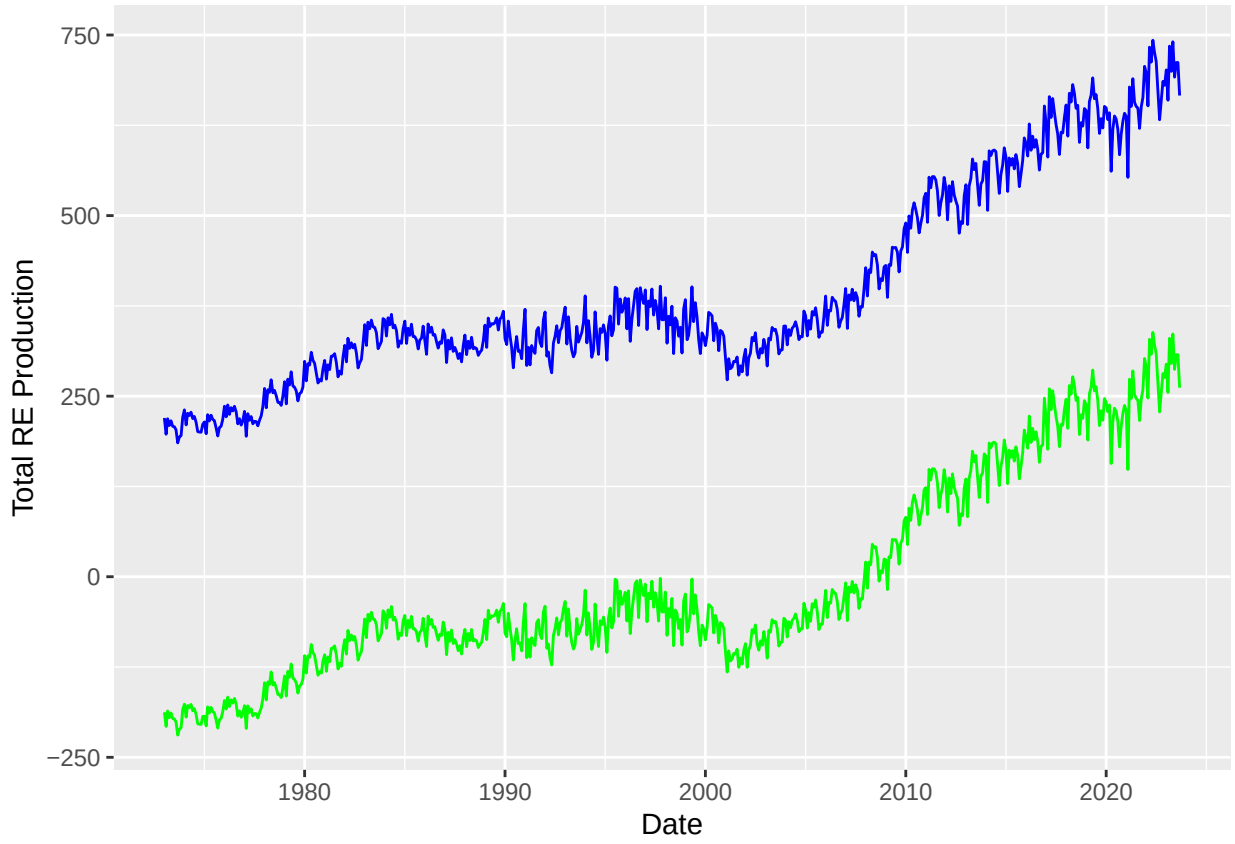
Beta0 (Dec), Beta5 (May), Beta6 (June),Beta9 (Sept), Beta10 (Oct), and beta11 (Nov) appear to be highly significant with p values less than 0.001. The R-squared value tells us that 46.97% of variability in the hydroelectric power consumption can be explained by seasonaility. The low p-value and higher R-squared show that the SMM may be a good fit for this data.

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?

```
# defining number of observations for ease of use in for loops
nobs <- nrow(renewable_energy_sub)


# Renewable Energy Production
## storing regression coefficients in order to de-season data
Renew_SMM_beta0 <- REP_seasonal_means_model$coefficients[1]
Renew_SMM_beta1 <- REP_seasonal_means_model$coefficients[2]

## computing seasonal component
REP_seas_comp <- array(0,nobs)
for(i in 1:nobs){
  REP_seas_comp[i] <- (Renew_SMM_beta0+Renew_SMM_beta1%*%REP_seasonal_dummy[i,])
}

## removing seasonal component
deseason_REP <- renewable_energy_sub[,2]-REP_seas_comp

## plotting deseasoned data
REP_deseason_plot <-
```

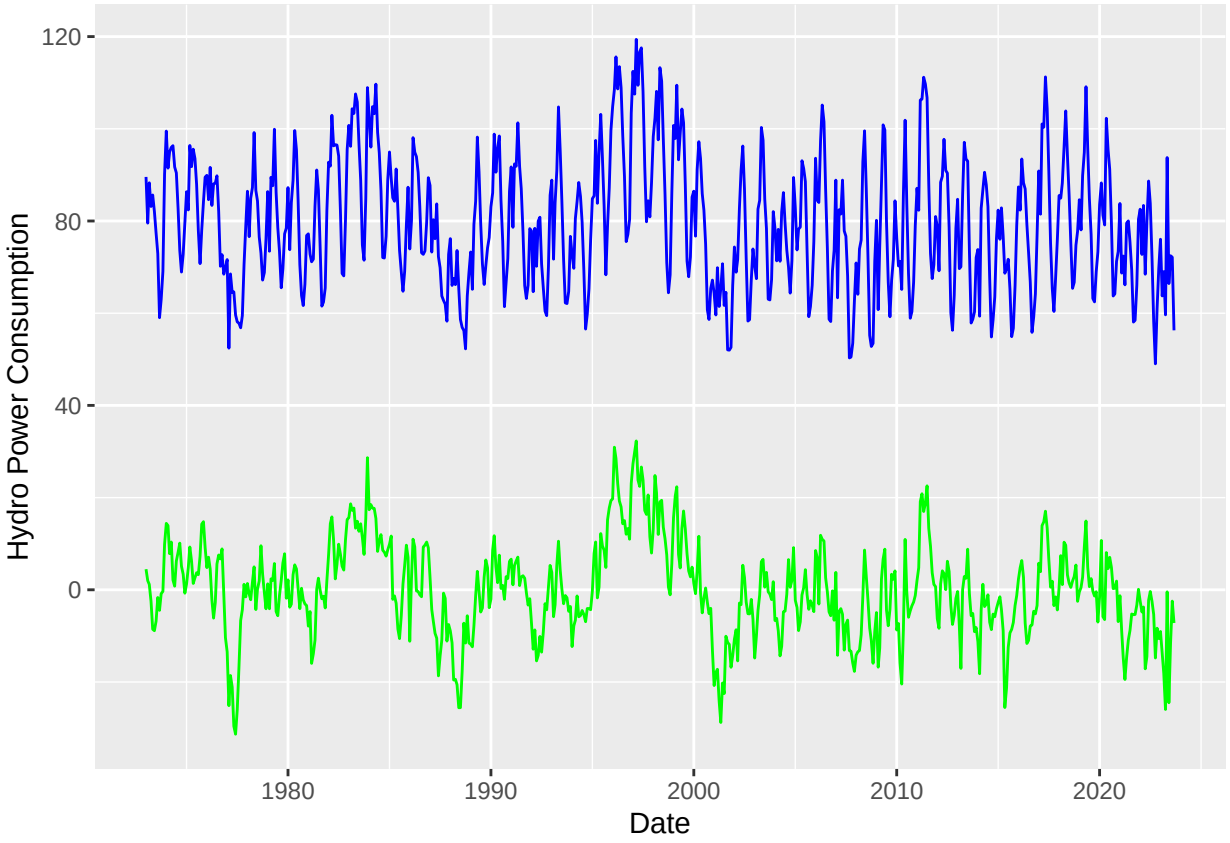
```
  ggplot(renewable_energy_sub,aes(x=Date,y=Total.Renewable.Energy.Production))+
  geom_line(color="blue")+
  labs(y="Total RE Production")+
  geom_line(aes(y=deseason_REP),color="green")

REP_deseason_plot
```



> Plotting the original data set against the deseasoned data set. There does not appear to be an significant difference since the model does show a seasonal trend.

```
# Hydroelectric Power Consumption
## storing regression coefficients in order to de-season data
Hydro_SMM_beta0 <- HPC_seasonal_means_model$coefficients[1]
Hydro_SMM_beta1 <- HPC_seasonal_means_model$coefficients[2:12]

## computing seasonal component
HPC_seas_comp <- array(0,nobs)
for(i in 1:nobs){
  HPC_seas_comp[i] <- (Hydro_SMM_beta0+Hydro_SMM_beta1%*%HPC_seasonal_dummy[i,])
}

## removing seasonal component
deseason_HPC <- renewable_energy_sub[,3]-HPC_seas_comp

## plotting deseasoned data
HPC_deseason_plot <-
  ggplot(renewable_energy_sub,aes(x=Date,y=Hydroelectric.Power.Consumption))+
```

```
  geom_line(color="blue")+
  labs(y="Hydro Power Consumption")+
  geom_line(aes(y=deseason_HPC),color="green")

HPC_deseason_plot
```

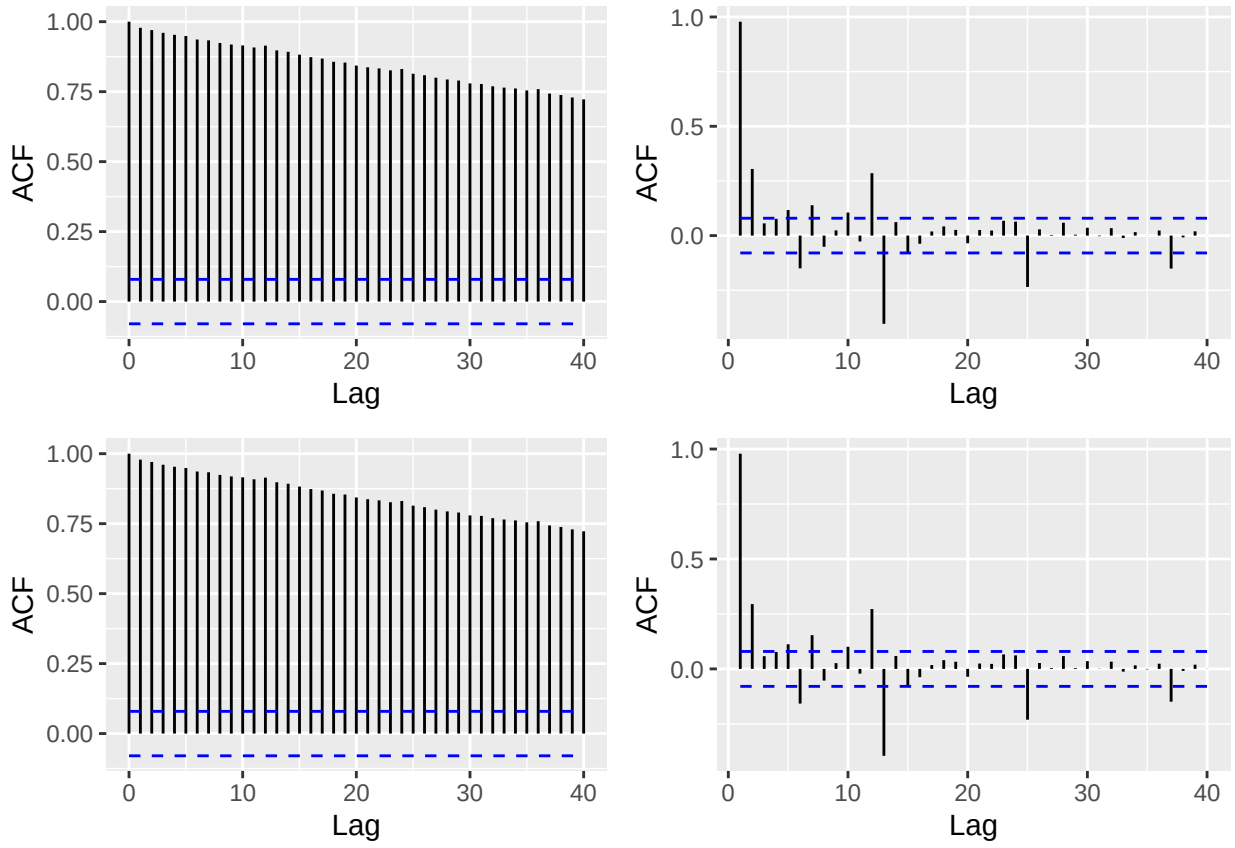


> Plotting the original data set against the deseasoned data set, the deseasoned data has a mean of 0 and the wave pattern has become less pronounced.

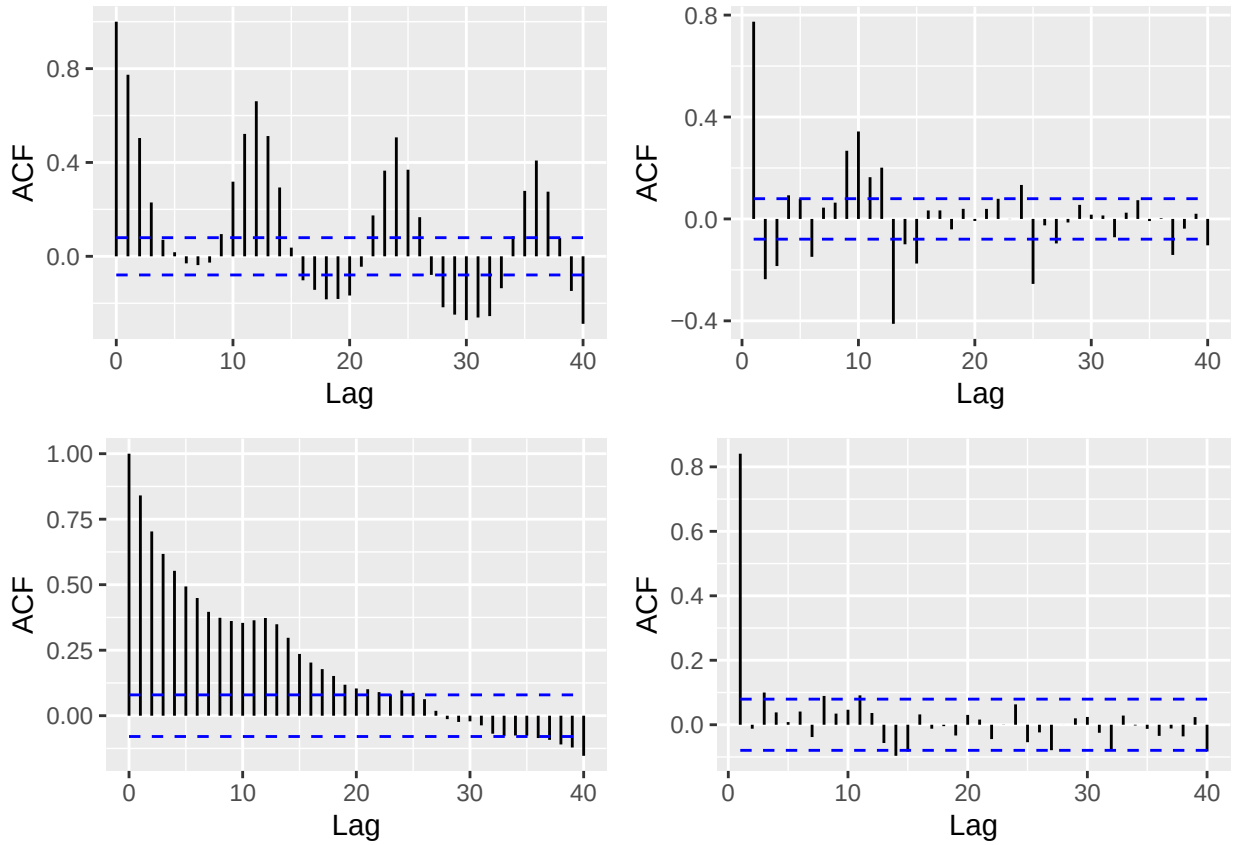
Q9

Plot ACF and PACF for the deseason series and compare with the plots from Q1. You may use plot_grid() again to get them side by side. nut mot mandatory. Did the plots change? How?

```
# Deseasoned Renewable Energy Production - ACF, PACF

REP_deseasoned_acf <- Acf(deseason_REP,lag.max = 40,plot=FALSE)
REP_deseasoned_pacf <- Pacf(deseason_REP,lag.max = 40,plot=FALSE)


plot_grid(
  autoplot(Renewables_acf),
  autoplot(Renewables_pacf),
  autoplot(REP_deseasoned_acf),
  autoplot(REP_deseasoned_pacf),
  nrow=2, ncol=2
)
```




```
# Deseasoned Hydro Power Consumption - ACF, PACF

HPC_deseasoned_acf <- Acf(deseason_HPC,lag.max = 40,plot=FALSE)
HPC_deseasoned_pacf <- Pacf(deseason_HPC,lag.max = 40,plot=FALSE)

plot_grid(
  autoplot(HydroConsump_acf),
  autoplot(HydroConsump_pacf),
  autoplot(HPC_deseasoned_acf),
  autoplot(HPC_deseasoned_pacf),
  nrow=2, ncol=2)
```



> For renewables the acf and pacf do not appear changed. This is because the SMM is not a good fit for the data and therefore did not change the observed values. > For hydro, the acf and pacf are meaningfully different compared to the observed data. Notable, the ACF had become a downward-sloping curve (vs. sin curves), correlation in dataset had become less significant, and the PACF is only showing one coefficient as highly significant (t-1). Overall, the data has been meaningfully detrended.