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

Assignment 3 - Due date 05/01/24

Shubhangi Gupta

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_Consumpti 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 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.\

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

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

```
library(tseries)
library(Kendall)
library(stats)
library(ggplot2)
```

```r
library(cowplot)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:cowplot':
##
##     stamp

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

```r
#Importing dataset
Raw_REDataset <- read.csv("Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.csv",
                  stringsAsFactors = TRUE)

#Subsetting dataset to get the required columns
REDataset <- Raw_REDataset[,c(1, 5:6)]
head(REDataset)
```

```
##            Month Total.Renewable.Energy.Production
## 1  1973 January                            219.839
## 2 1973 February                            197.330
## 3    1973 March                            218.686
## 4    1973 April                            209.330
## 5      1973 May                            215.982
## 6     1973 June                            208.249
##   Hydroelectric.Power.Consumption
## 1                          89.562
## 2                          79.544
## 3                          88.284
## 4                          83.152
## 5                          85.643
## 6                          82.060
```

```r
#Renaming column headings
colnames(REDataset) <- c("Date", "RE", "Hydro")
head(REDataset)
```

```
##             Date      RE  Hydro
## 1  1973 January 219.839 89.562
## 2 1973 February 197.330 79.544
## 3    1973 March 218.686 88.284
## 4    1973 April 209.330 83.152
## 5      1973 May 215.982 85.643
## 6     1973 June 208.249 82.060
```

```r
#Converting the Date column to a Date object
REDataset_Date <- REDataset$Date
REDataset$Date <- ym(format(REDataset_Date, format = "%Y %B"))
str(REDataset)
```

```
## 'data.frame':    609 obs. of  3 variables:
##  $ Date : Date, format: "1973-01-01" "1973-02-01" ...
##  $ RE   : num  220 197 219 209 216 ...
##  $ Hydro: num  89.6 79.5 88.3 83.2 85.6 ...
```

```r
#NOTE FOR GRADERS: I have not included biomass even though the question mentions
#it based on Luana's message on slack to not include it.
```

##Trend Component

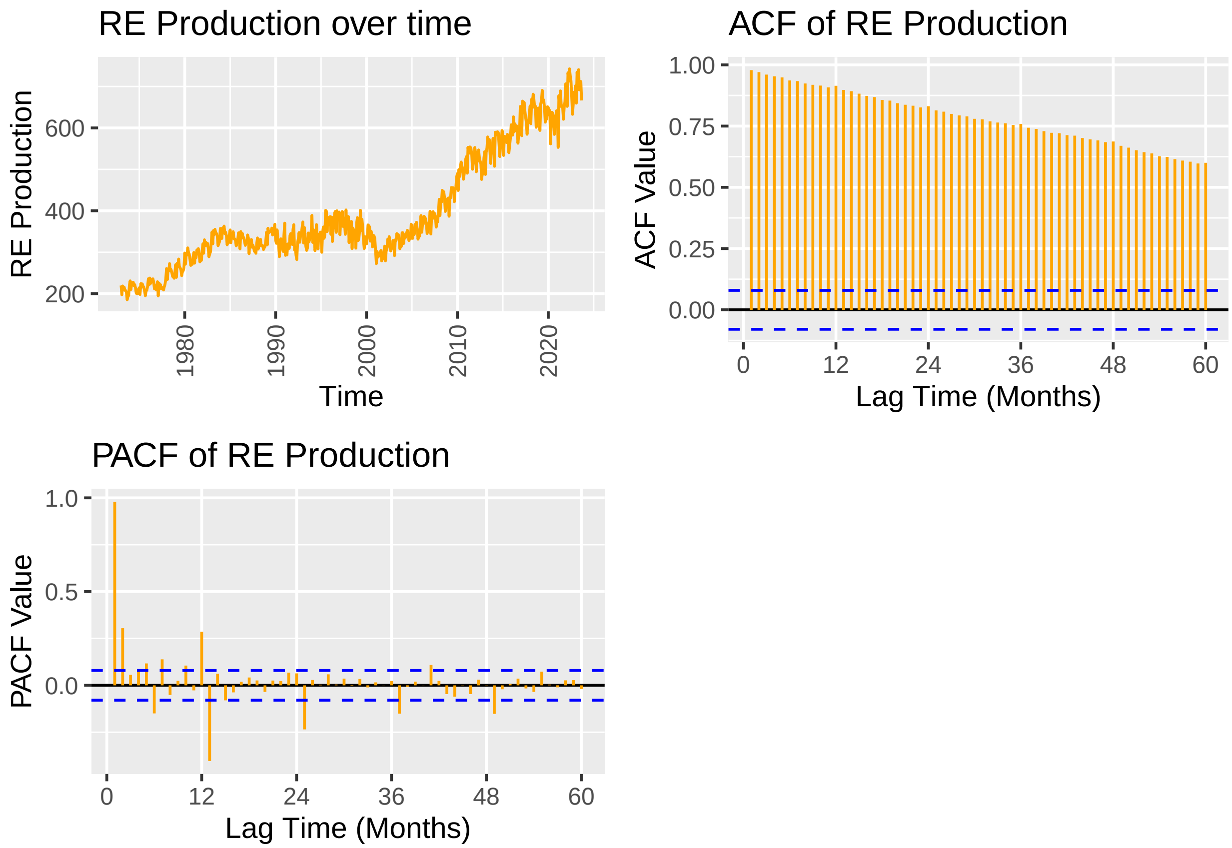
Q1

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)

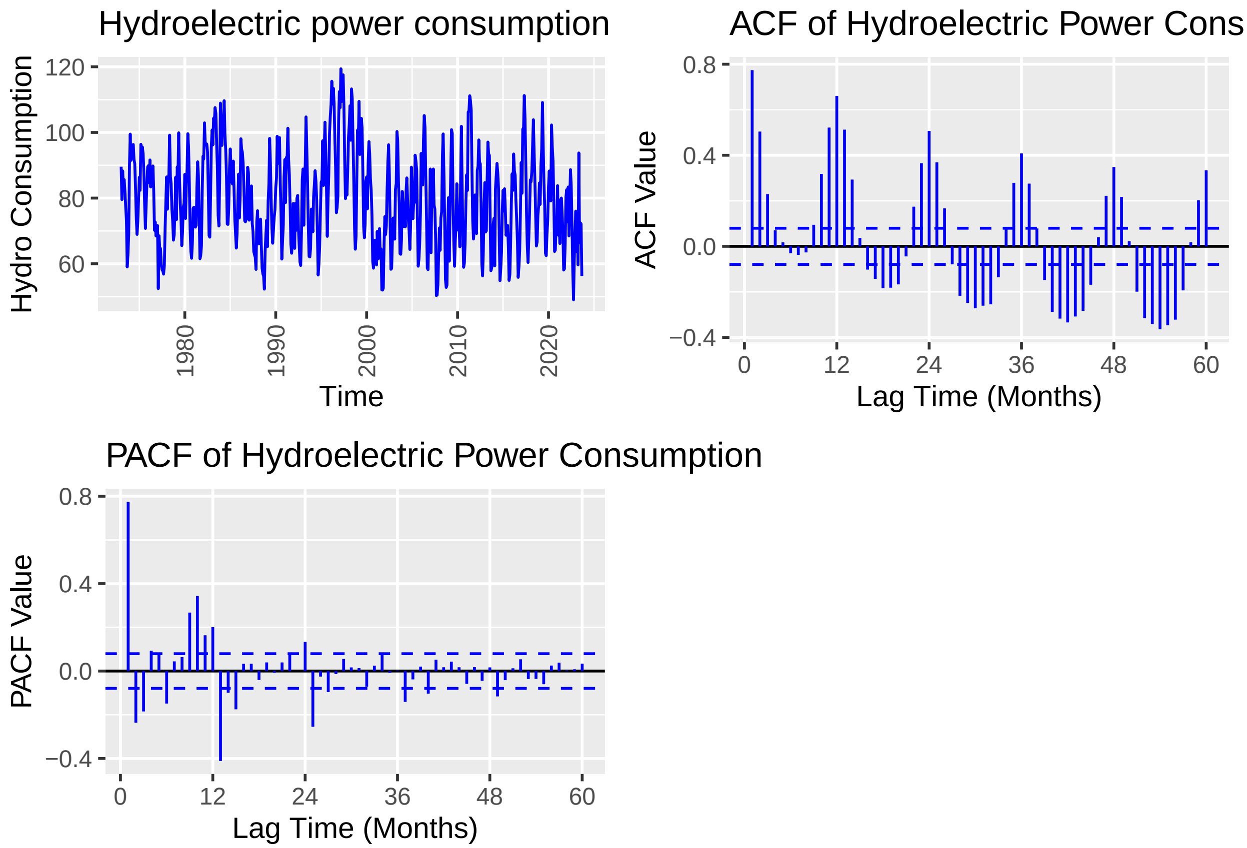
```r
#Converting dataframe into time series

ts_REDataset <- ts(REDataset[,2:3], start=c(1973, 01), frequency = 12)
head(ts_REDataset)
```

```
##               RE  Hydro
## Jan 1973 219.839 89.562
## Feb 1973 197.330 79.544
## Mar 1973 218.686 88.284
## Apr 1973 209.330 83.152
## May 1973 215.982 85.643
## Jun 1973 208.249 82.060
```

```r
#RE Production Plots

ts_RE_plot <- autoplot(ts_REDataset[,1],  col="orange") +
  xlab("Time") + ylab("RE Production") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ggtitle("RE Production over time")

ts_RE_ACF <- Acf(ts_REDataset[, 1], lag.max = 60, type = "correlation", plot = FALSE)
ts_RE_ACF <- autoplot(ts_RE_ACF, col = "orange") +
  xlab ("Lag Time (Months)") + ylab ("ACF Value") +
  ggtitle("ACF of RE Production")

ts_RE_PACF <- Pacf(ts_REDataset[, 1], lag.max = 60, type = "correlation", plot = FALSE)
ts_RE_PACF <- autoplot(ts_RE_PACF, col = "orange") + xlab ("Lag Time (Months)")+
  ylab ("PACF Value") +
  ggtitle("PACF of RE Production")

plot_grid(ts_RE_plot, ts_RE_ACF, ts_RE_PACF )
```




```r
#Hydroelectric Power Consumption

ts_Hydro_plot <- autoplot(ts_REDataset[,2],  col="blue") +
  xlab("Time") + ylab("Hydro Consumption") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ggtitle("Hydroelectric power consumption over time")

ts_Hydro_ACF <- Acf(ts_REDataset[, 2], lag.max = 60, type = "correlation", plot = FALSE)
ts_Hydro_ACF <- autoplot(ts_Hydro_ACF, col = "blue") +
  xlab ("Lag Time (Months)") + ylab ("ACF Value") +
    ggtitle("ACF of Hydroelectric Power Consumption")

ts_Hydro_PACF <- Pacf(ts_REDataset[, 2], lag.max = 60, type = "correlation", plot = FALSE)
ts_Hydro_PACF <- autoplot(ts_Hydro_PACF, col = "blue") +
  xlab ("Lag Time (Months)") + ylab ("PACF Value") +
  ggtitle("PACF of Hydroelectric Power Consumption")

plot_grid(ts_Hydro_plot, ts_Hydro_ACF, ts_Hydro_PACF )
```



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?

RE Production has had an overall rising trend from 1973-2020 with steadily declining autocorrelation and almost insignificant PACF beyond the 12th lag mark (with the exception of mildly significant PACF values at the annual 12th, 24th and 36th lag marks). On the other hand, hydroelectric power consumption showcases a seasonal trend with autocorrelation and PACF decreasing over time but maintaining their 6-month positive and negative seasonal trend.

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.

```
nobs <- nrow(REDataset)
t <- c(1:nobs)

#Fitting the lm() function to the RE column
lm_RE <- lm(ts_REDataset[,1]~t)
print(summary(lm_RE))
```

```
##
```

```
## Call:
## lm(formula = ts_REDataset[, 1] ~ t)
##
## 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
```

```r
beta0_RE=as.numeric(lm_RE$coefficients[1])
beta0_RE
```

```
## [1] 180.9894
```

```r
beta1_RE=as.numeric(lm_RE$coefficients[2])
beta1_RE
```

```
## [1] 0.7040391
```

```r
#Fitting the lm () function to the hydro column
lm_hydro <- lm(ts_REDataset[,2]~t)
print(summary(lm_hydro))
```

```
##
## Call:
## lm(formula = ts_REDataset[, 2] ~ t)
##
## 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
```

```r
beta0_Hydro=as.numeric(lm_hydro$coefficients[1])
beta0_Hydro
```

```
## [1] 82.73475
```

```r
beta1_Hydro=as.numeric(lm_hydro$coefficients[2])
beta1_Hydro
```

```
## [1] -0.009849298
```

Interpretation of results: RE regression: Slope (B1) = 0.7 indicating a coherent upward rising trend Intercept (B0) = 180.98 P-value = 2x10^-16 indicating strong statistical significance (p-value < 0.05) Rsquared value of 80% signifying that 80% of the variability in the data is determined by time.

Hydro regression: Slope (B1) = -0.0098 indicating no significant upward or downward trend. This is verified by the plot where we can clearly see a seasonal cyclical trend in the data. Intercept (B0) - 82.73 P-value = 2x10^-16 indicating strong statistical significance (p-value < 0.05) R squared value = 1.3% indicating that the variability in the data is not really explained by 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?

```r
#Detrending the data
detrend_RE_vector <- REDataset[,2]-(beta0_RE+beta1_RE*t)
detrend_RE <- data.frame(detrend_RE_vector)
colnames(detrend_RE)<-"detrended RE"
head(detrend_RE)
```

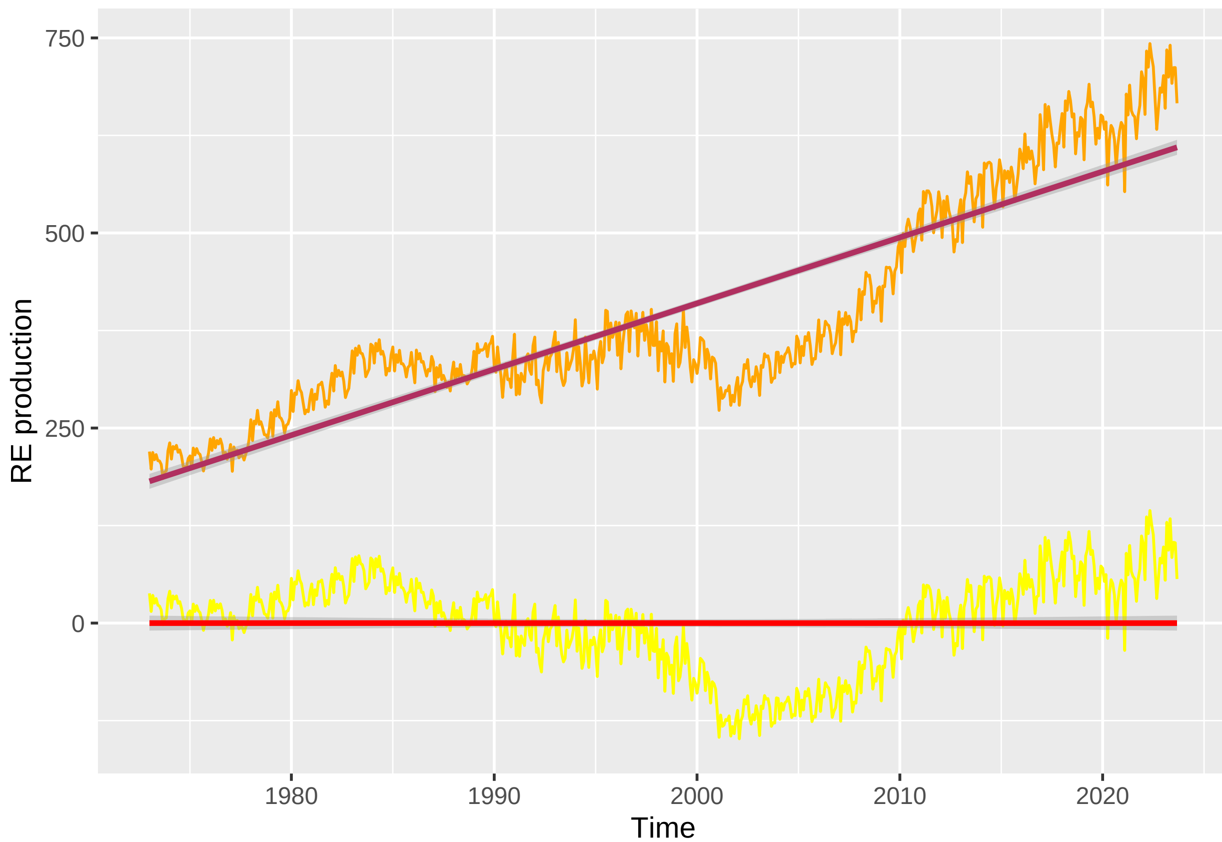
```
##   detrended RE
## 1     38.14556
## 2     14.93252
## 3     35.58448
## 4     25.52444
## 5     31.47240
## 6     23.03536
```

```r
detrend_hydro_vector <- REDataset[,3]-(beta0_Hydro + beta1_Hydro*t)
detrend_hydro <- data.frame(detrend_hydro_vector)
colnames(detrend_hydro)<- "detrended hydro"
head(detrend_hydro)
```

```
##   detrended hydro
## 1       6.8371025
## 2      -3.1710482
## 3       5.5788011
## 4       0.4566504
## 5       2.9574997
## 6      -0.6156510
```

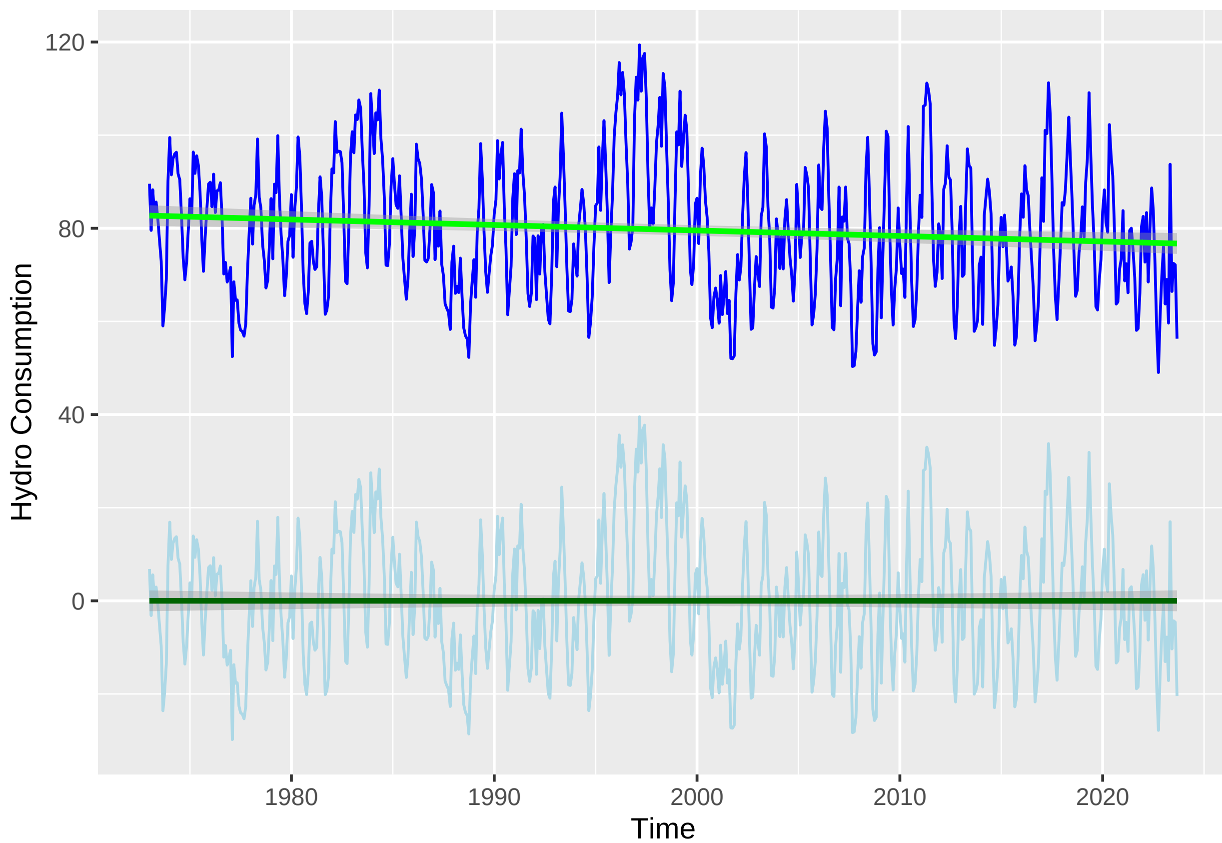
```r
#Plotting the original and detrended data for RE with their regression lines
ggplot(REDataset, aes(x=Date, y=RE))+
  geom_line(color="orange")+
  geom_smooth(color="maroon", method="lm")+
  geom_line(aes(y=detrend_RE$`detrended RE`), color = "yellow")+
  geom_smooth(aes(y=detrend_RE$`detrended RE`), color = "red", method="lm")+
  ylab("RE production")+
  xlab("Time")
```

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




```r
#Plotting the original and detrended data for Hydro with their regression lines
ggplot(REDataset, aes(x=Date, y=Hydro))+
  geom_line(color = "blue")+
  geom_smooth(color="green", method="lm")+
  geom_line(aes(y=detrend_hydro$`detrended hydro`), colour = "lightblue")+
  geom_smooth(aes(y=detrend_hydro$`detrended hydro`), method="lm", color="darkgreen")+
  xlab("Time")+
  ylab("Hydro Consumption")
```

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



Analysis: The detrended data shifts the datapoints to around zero (because f the removal of the intercept). In the case of RE, the regression line went from being upward rising in the raw data to horizontal around zero in the detrended data. This indicates that the upward (linear) rising trend in the RE dataset was removed in the detrending process. On the other hand, in hydro, while the data shifted to around zero, the regression line of the raw data is almost parallel to that of the detrended data, and the shape of the time series remains perpetually the same, indicating that it was not detrended sufficiently (if at all).

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. nut mot mandatory. Did the plots change? How?

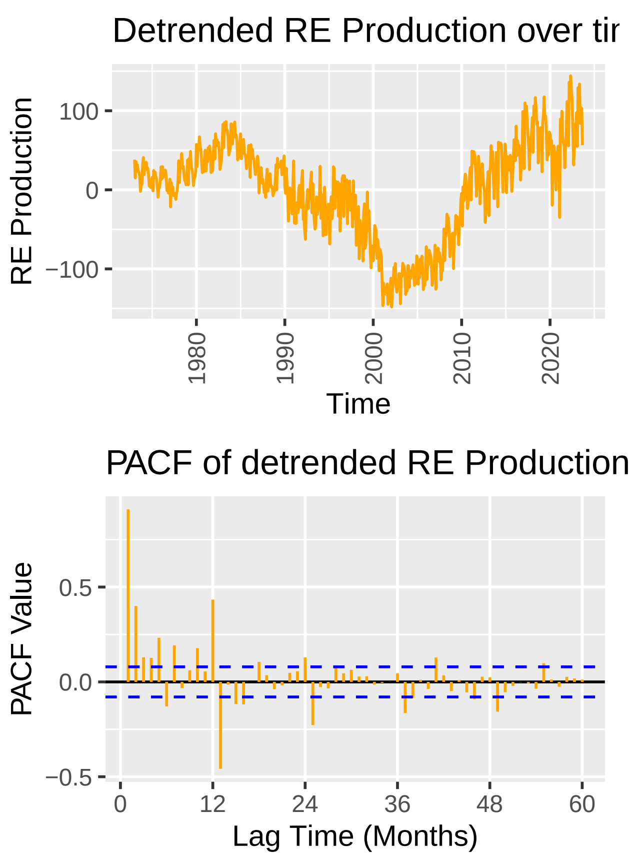
```
#Converting detrended datasets into time series

ts_detrend_RE <- ts(detrend_RE$`detrended RE`, start=c(1973, 01), frequency = 12)
head(ts_detrend_RE)
```

```
##           Jan      Feb      Mar      Apr      May      Jun
## 1973 38.14556 14.93252 35.58448 25.52444 31.47240 23.03536
```

```
ts_detrend_hydro <- ts(detrend_hydro$`detrended hydro`, start=c(1973, 01), frequency = 12 )
head(ts_detrend_hydro)
```

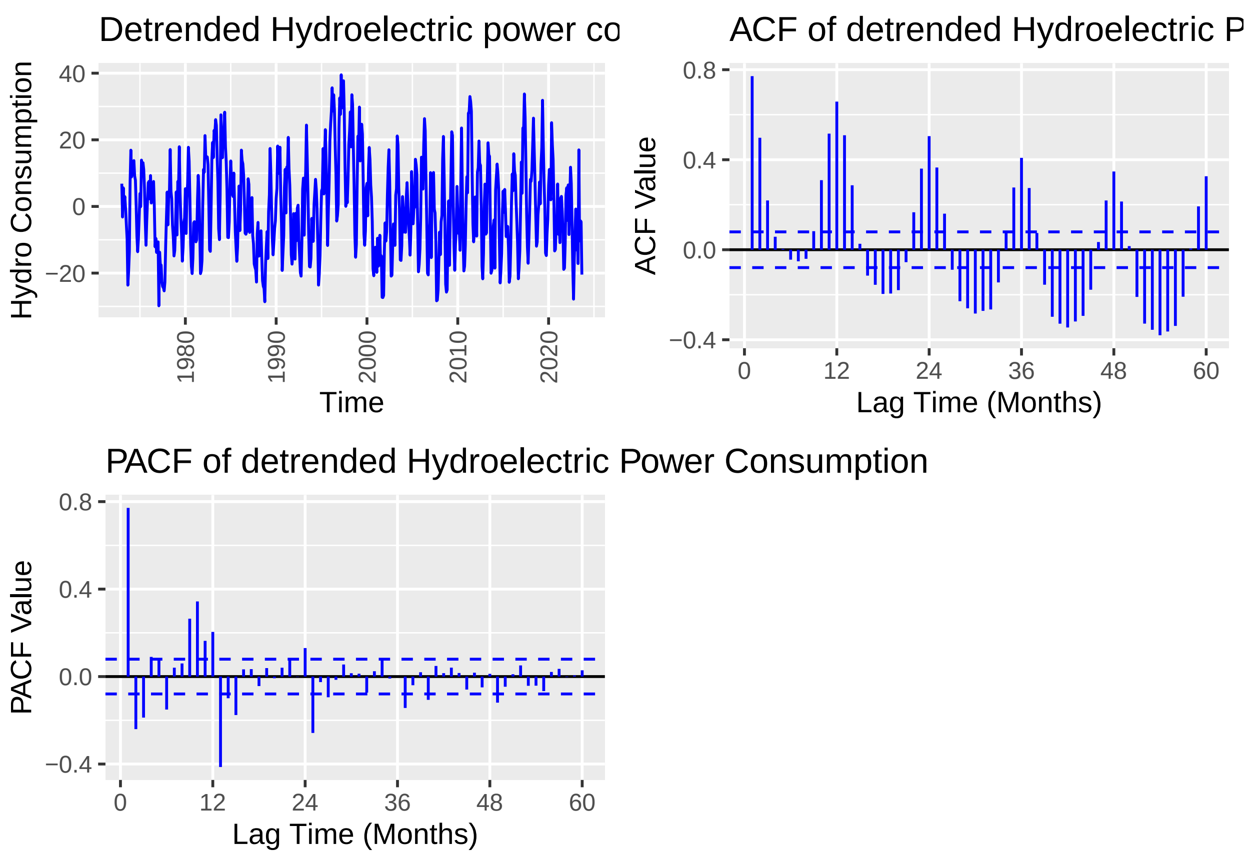
```
##             Jan        Feb        Mar        Apr        May        Jun
## 1973  6.8371025 -3.1710482  5.5788011  0.4566504  2.9574997 -0.6156510
```

```r
#Detrended RE Production Plots

ts_detrend_RE_plot <- autoplot(ts_detrend_RE,  col="orange") +
  xlab("Time") + ylab("RE Production") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ggtitle("Detrended RE Production over time")

ts_detrend_RE_ACF <- Acf(ts_detrend_RE, lag.max = 60, type = "correlation", plot = FALSE)
ts_detrend_RE_ACF <- autoplot(ts_detrend_RE_ACF, col = "orange") +
  xlab("Lag Time (Months)") + ylab("ACF Value") +
  ggtitle("ACF of detrended RE Production")

ts_detrend_RE_PACF <- Pacf(ts_detrend_RE, lag.max = 60, type = "correlation", plot = FALSE)
ts_detrend_RE_PACF <- autoplot(ts_detrend_RE_PACF, col = "orange") +
  xlab ("Lag Time (Months)") + ylab ("PACF Value") +
  ggtitle("PACF of detrended RE Production")

plot_grid(ts_detrend_RE_plot, ts_detrend_RE_ACF, ts_detrend_RE_PACF )
```

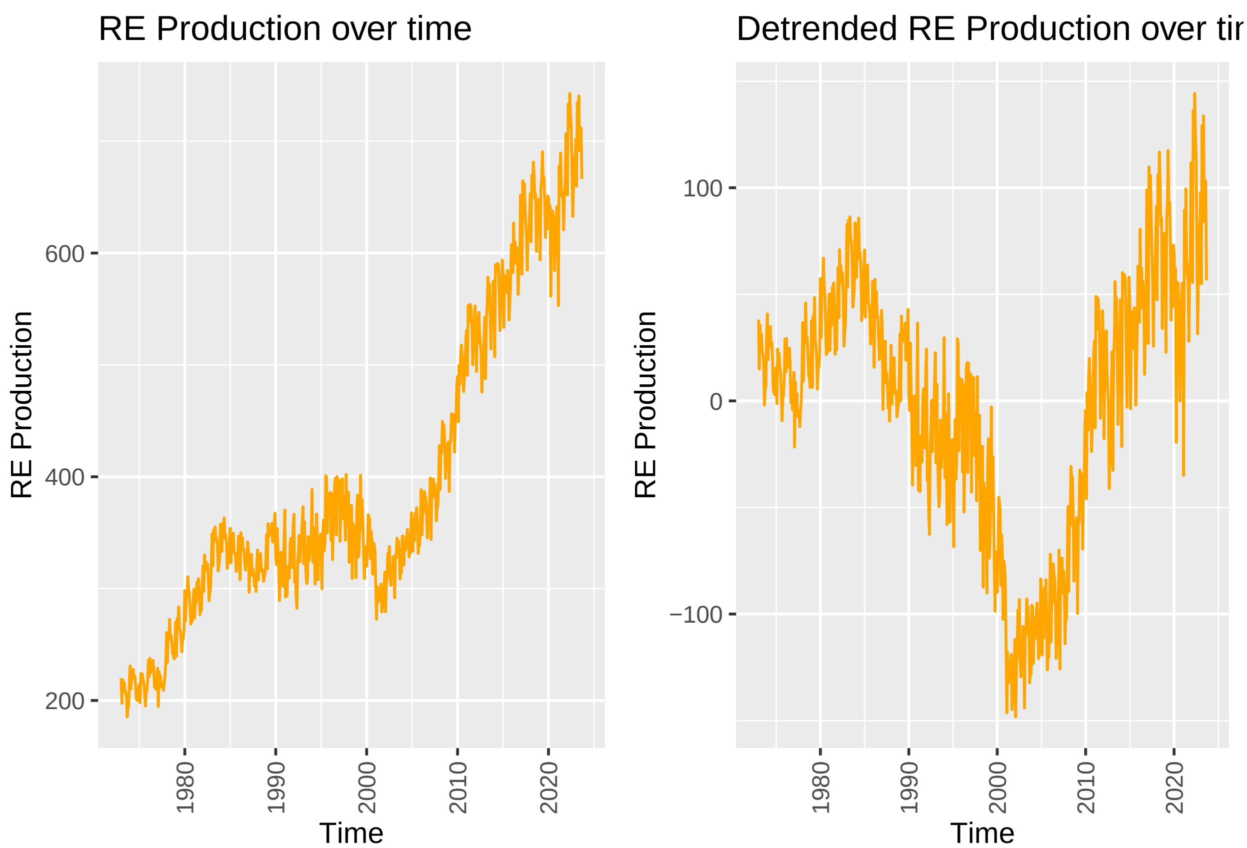



```r
#Detrended Hydroelectric Power Consumption

ts_detrend_Hydro_plot <- autoplot(ts_detrend_hydro,  col="blue") +
  xlab("Time") + ylab("Hydro Consumption") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ggtitle("Detrended Hydroelectric power consumption over time")
```

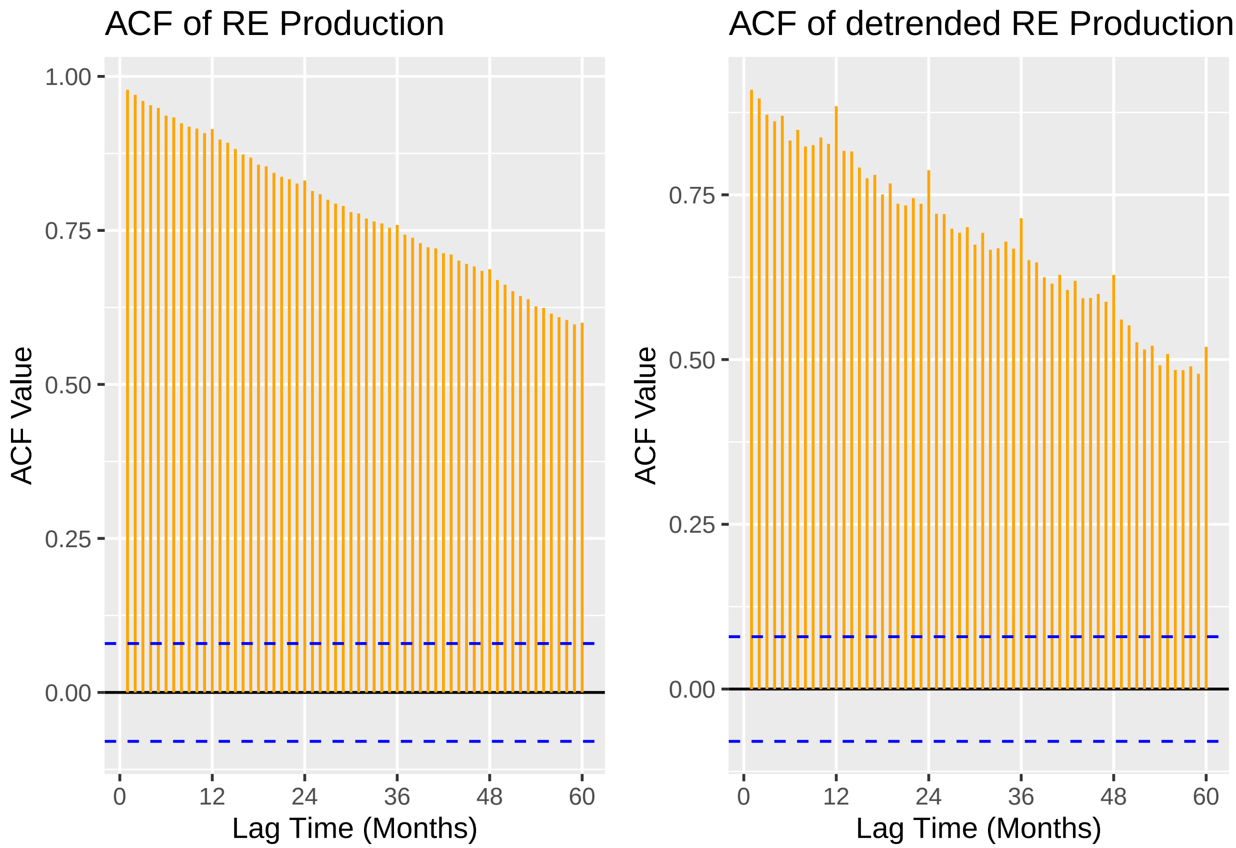
```r
ts_detrend_Hydro_ACF <- Acf(ts_detrend_hydro, lag.max = 60, type = "correlation", plot = FALSE)
ts_detrend_Hydro_ACF <- autoplot(ts_detrend_Hydro_ACF, col = "blue") +
  xlab ("Lag Time (Months)") + ylab ("ACF Value") +
  ggtitle("ACF of detrended Hydroelectric Power Consumption")

ts_detrend_Hydro_PACF <- Pacf(ts_detrend_hydro,  lag.max = 60, type = "correlation", plot = FALSE)
ts_detrend_Hydro_PACF <- autoplot(ts_detrend_Hydro_PACF, col = "blue") +
  xlab ("Lag Time (Months)") + ylab ("PACF Value") +
  ggtitle("PACF of detrended Hydroelectric Power Consumption")

plot_grid(ts_detrend_Hydro_plot, ts_detrend_Hydro_ACF, ts_detrend_Hydro_PACF )
```



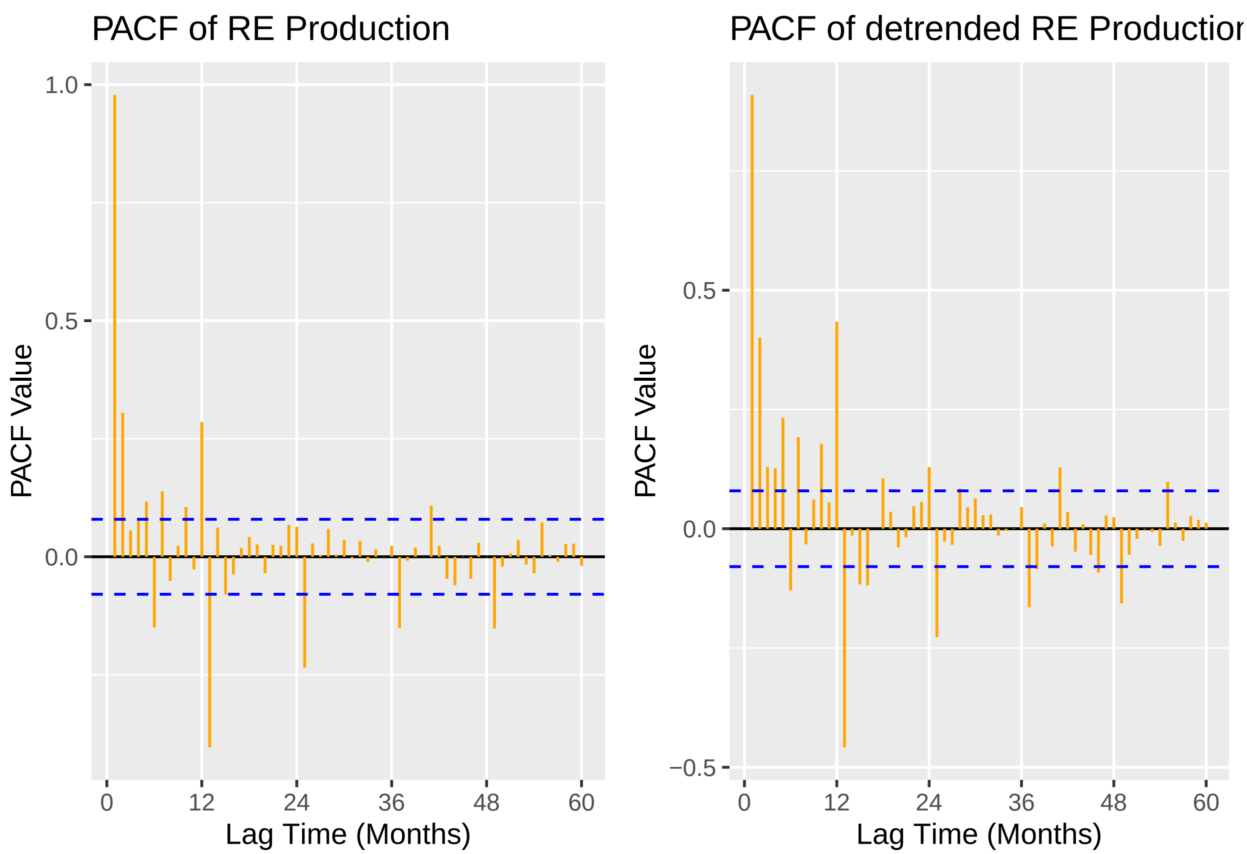

```r
plot_grid(ts_RE_plot, ts_detrend_RE_plot)
```



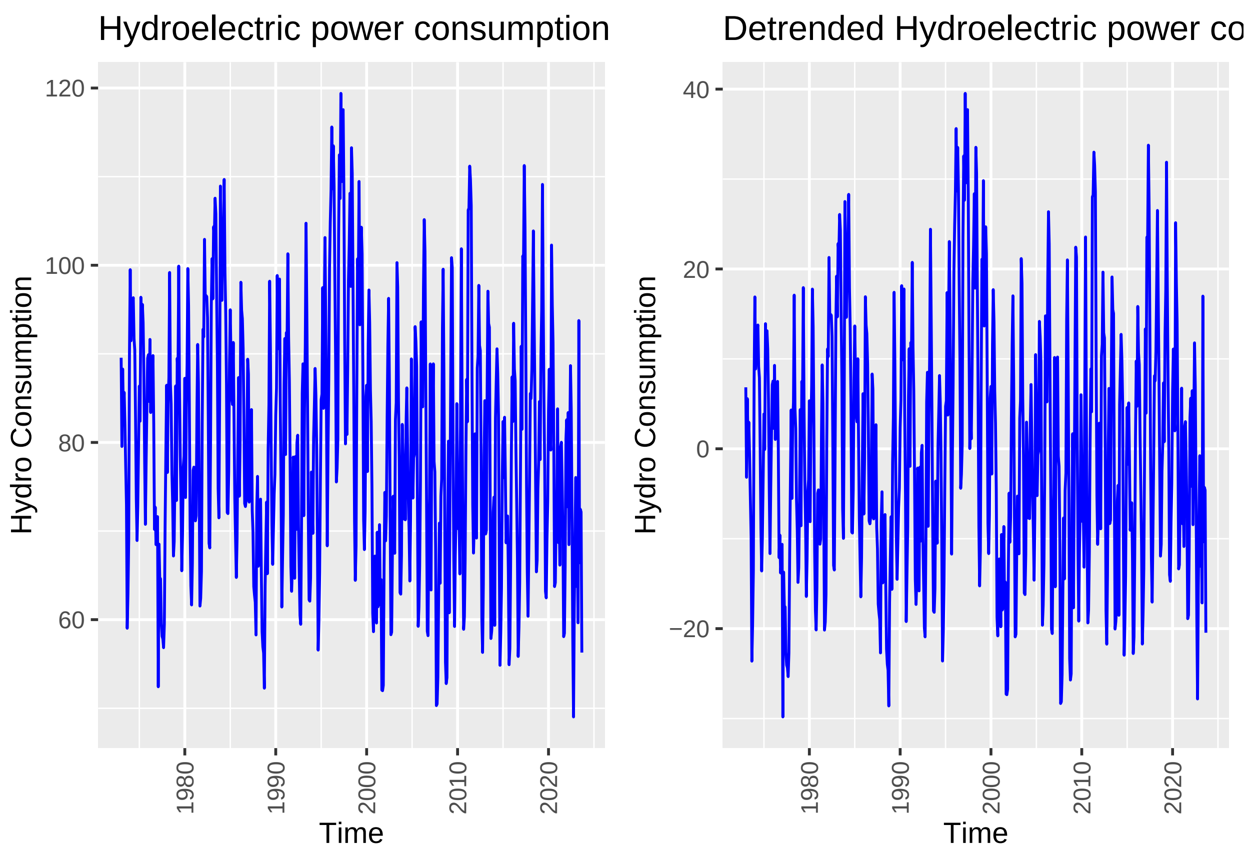

```
plot_grid(ts_RE_ACF, ts_detrend_RE_ACF)
```



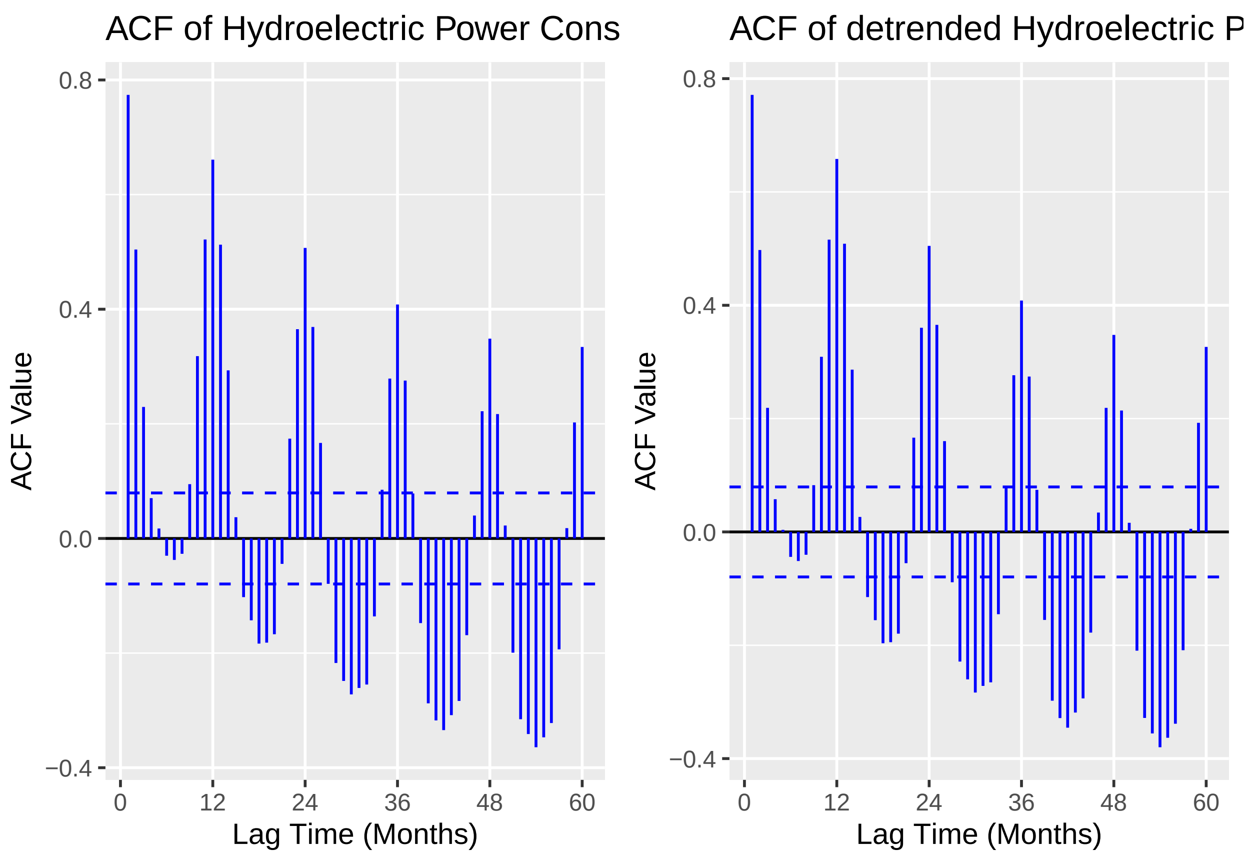

```
plot_grid(ts_RE_PACF, ts_detrend_RE_PACF)
```



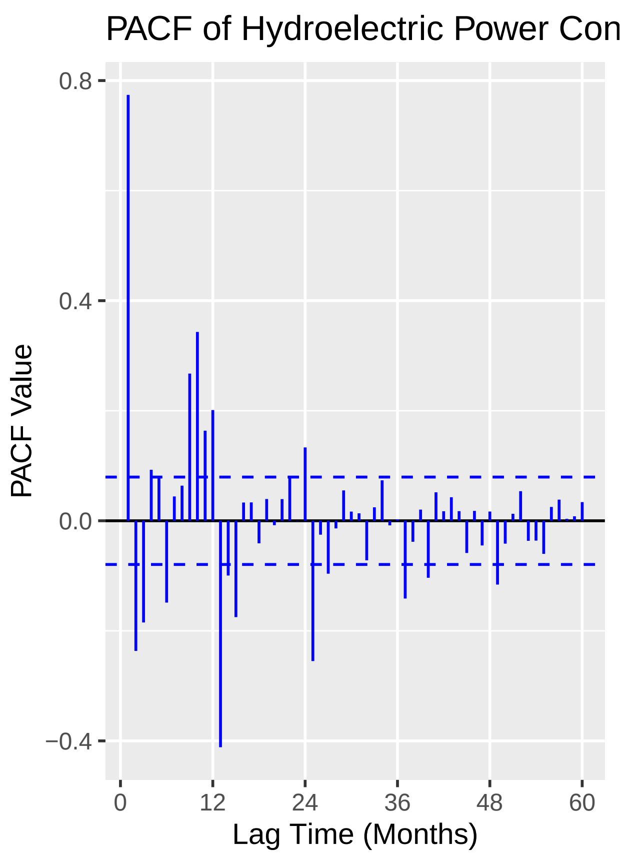

```r
plot_grid(ts_Hydro_plot, ts_detrend_Hydro_plot)
```

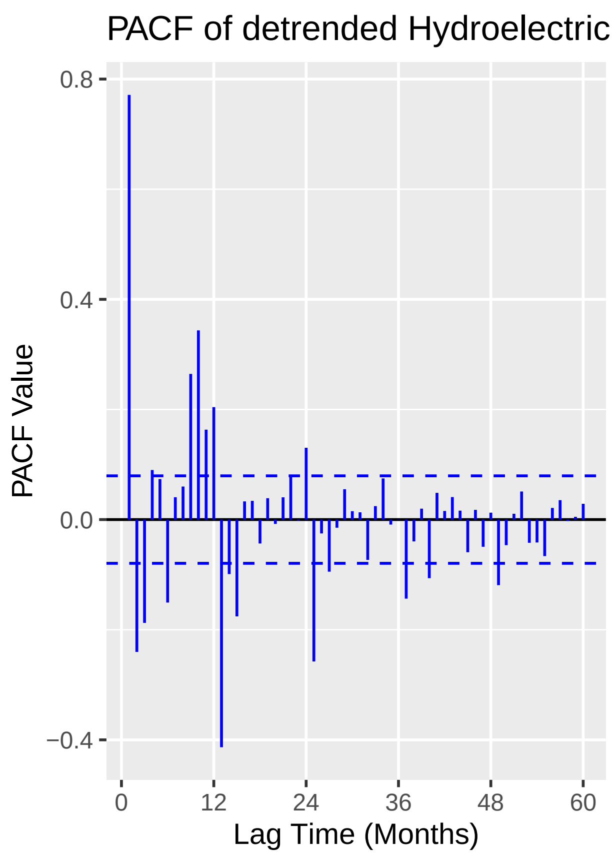



```
plot_grid(ts_Hydro_ACF, ts_detrend_Hydro_ACF)
```




```
plot_grid(ts_Hydro_PACF, ts_detrend_Hydro_PACF)
```





Analysis: In RE, the almost coherent upward trend changes significantly to cycle around zero although with a one large dip to below -100 in 2000-2015 and then a rise to above 100 from 2015-2020. The ACF also declines by about 0.25 in the initial lags but the shape of the curve remains similar, while PACF remains similar with only a slight decline in the valye in lag 0. In hydro, the trend of the data remains similar, although it now hovers around 0 instead of the intercept of 82. The ACF and PACF both remain the same indicating no impact of the detrending on the data.

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.

The hydro consumption dataset has a clear seasonal trend, which is most evident from the ACF plot. However, RE production does not seem to have a seasonal trend, and instead seems to possibly have a linear trend.

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?

```r
#Creating the seasonal dummies
dummies_RE <- seasonaldummy(ts_REDataset[,1])
dummies_Hydro <- seasonaldummy(ts_REDataset[,2])

#Fitting a linear model to the seasonal dummies
seas_means_model_RE <- lm(REDataset[,2]~dummies_RE)
summary(seas_means_model_RE)
```

```
##
## Call:
## lm(formula = REDataset[, 2] ~ dummies_RE)
##
## 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 ***
## dummies_REJan    2.962     27.546   0.108    0.914
## dummies_REFeb  -34.476     27.546  -1.252    0.211
## dummies_REMar    3.929     27.546   0.143    0.887
## dummies_REApr   -8.695     27.546  -0.316    0.752
## dummies_REMay    6.645     27.546   0.241    0.809
## dummies_REJun   -4.198     27.546  -0.152    0.879
## dummies_REJul    2.460     27.546   0.089    0.929
## dummies_REAug   -5.026     27.546  -0.182    0.855
## dummies_RESep  -29.119     27.546  -1.057    0.291
## dummies_REOct  -20.068     27.682  -0.725    0.469
## dummies_RENov  -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
```

```r
seas_means_model_Hydro <- lm(REDataset[,3]~dummies_Hydro)
summary(seas_means_model_Hydro)
```

```
##
## Call:
## lm(formula = REDataset[, 3] ~ dummies_Hydro)
##
## 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 ***
## dummies_HydroJan    4.807      2.069   2.323  0.02050 *
```

```
## dummies_HydroFeb   -2.725      2.069  -1.317  0.18831
## dummies_HydroMar    6.825      2.069   3.298  0.00103 **
## dummies_HydroApr    5.319      2.069   2.571  0.01039 *
## dummies_HydroMay   13.922      2.069   6.729 4.02e-11 ***
## dummies_HydroJun   10.650      2.069   5.147 3.60e-07 ***
## dummies_HydroJul    3.912      2.069   1.891  0.05914 .
## dummies_HydroAug   -5.677      2.069  -2.744  0.00626 **
## dummies_HydroSep  -16.797      2.069  -8.118 2.72e-15 ***
## dummies_HydroOct  -16.468      2.079  -7.920 1.17e-14 ***
## dummies_HydroNov  -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
```

```r
#Storing regression coefficients
B0_RE_seas <- seas_means_model_RE$coefficients[1]
B0_RE_seas
```

```
## (Intercept)
##    404.5261
```

```r
B1_RE_seas <- seas_means_model_RE$coefficients[2:12]
B1_RE_seas
```

```
## dummies_REJan dummies_REFeb dummies_REMar dummies_REApr dummies_REMay
##      2.962096    -34.476080      3.928880     -8.695335      6.645116
## dummies_REJun dummies_REJul dummies_REAug dummies_RESep dummies_REOct
##     -4.197727      2.459508     -5.025904    -29.118512    -20.068140
## dummies_RENov
##    -20.345560
```

```r
B0_Hydro_seas <- seas_means_model_Hydro$coefficients[1]
B0_Hydro_seas
```

```
## (Intercept)
##    80.28176
```

```r
B1_Hydro_seas <- seas_means_model_Hydro$coefficients[2:12]
B1_Hydro_seas
```

```
## dummies_HydroJan dummies_HydroFeb dummies_HydroMar dummies_HydroApr
##         4.807299        -2.725270         6.825024         5.319044
## dummies_HydroMay dummies_HydroJun dummies_HydroJul dummies_HydroAug
##        13.922220        10.649985         3.912260        -5.676917
## dummies_HydroSep dummies_HydroOct dummies_HydroNov
##       -16.797387       -16.467980       -10.884780
```

Analysis: The RE seasonal regression model does not give a significant output as the overall p-value is 0.8976 which is >0.05. Further the p-value of all beta values are also > 0.05, indicating that RE does not have a seasonal trend. On the other hand, the overall p=value of hydro seasonal regression model gives a significant output as the overall p-value is 2.2 x 10^-16 which is much < 0.05. Further, all beta values except for that of february are <0.05 indicating that hydro does have a seasonal trend. This matches my results from Qn 6.

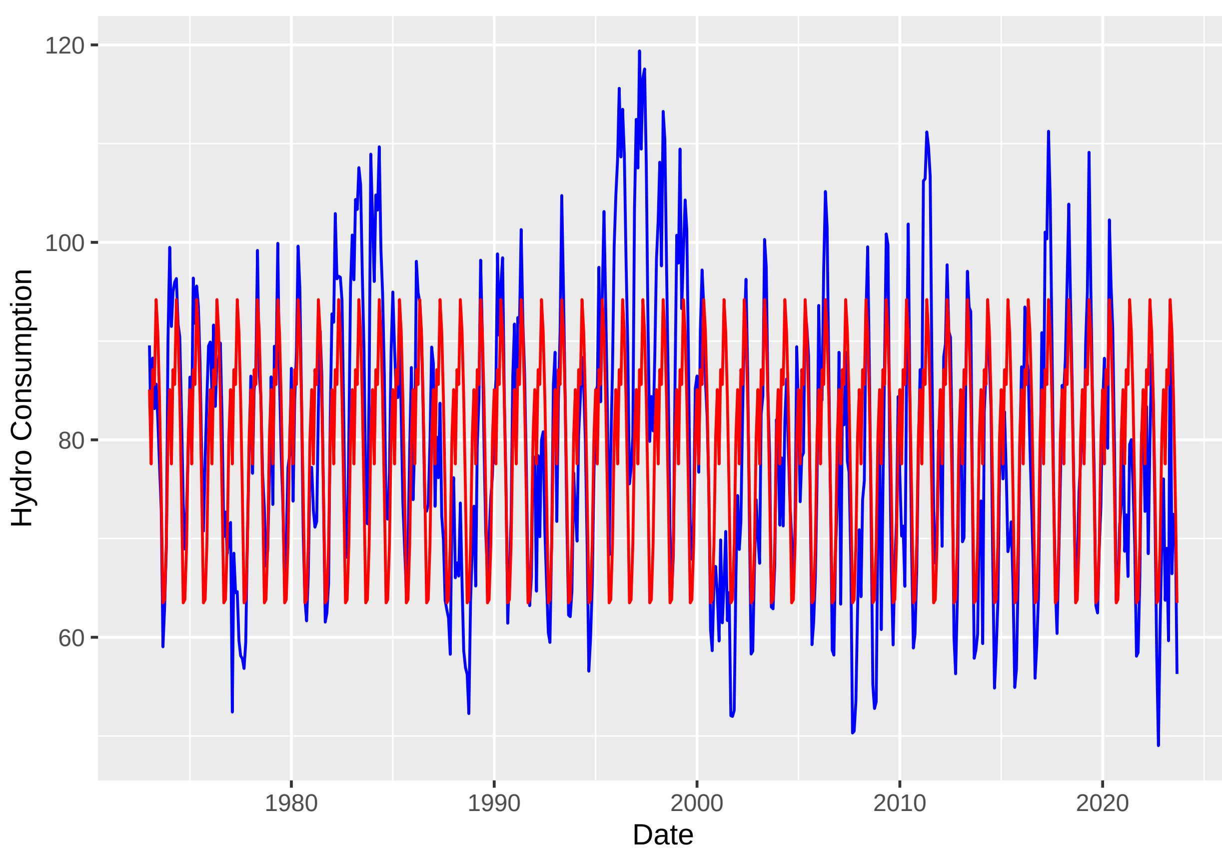
Q8

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
#computing seasonal component
Hydro_seas_comp <- array(0,nobs)
for(i in 1:nobs){
  Hydro_seas_comp[i] <- (B0_Hydro_seas+B1_Hydro_seas %*% dummies_Hydro[i,])
}
head(Hydro_seas_comp)
```

```
## [1] 85.08906 77.55649 87.10678 85.60080 94.20398 90.93175
```

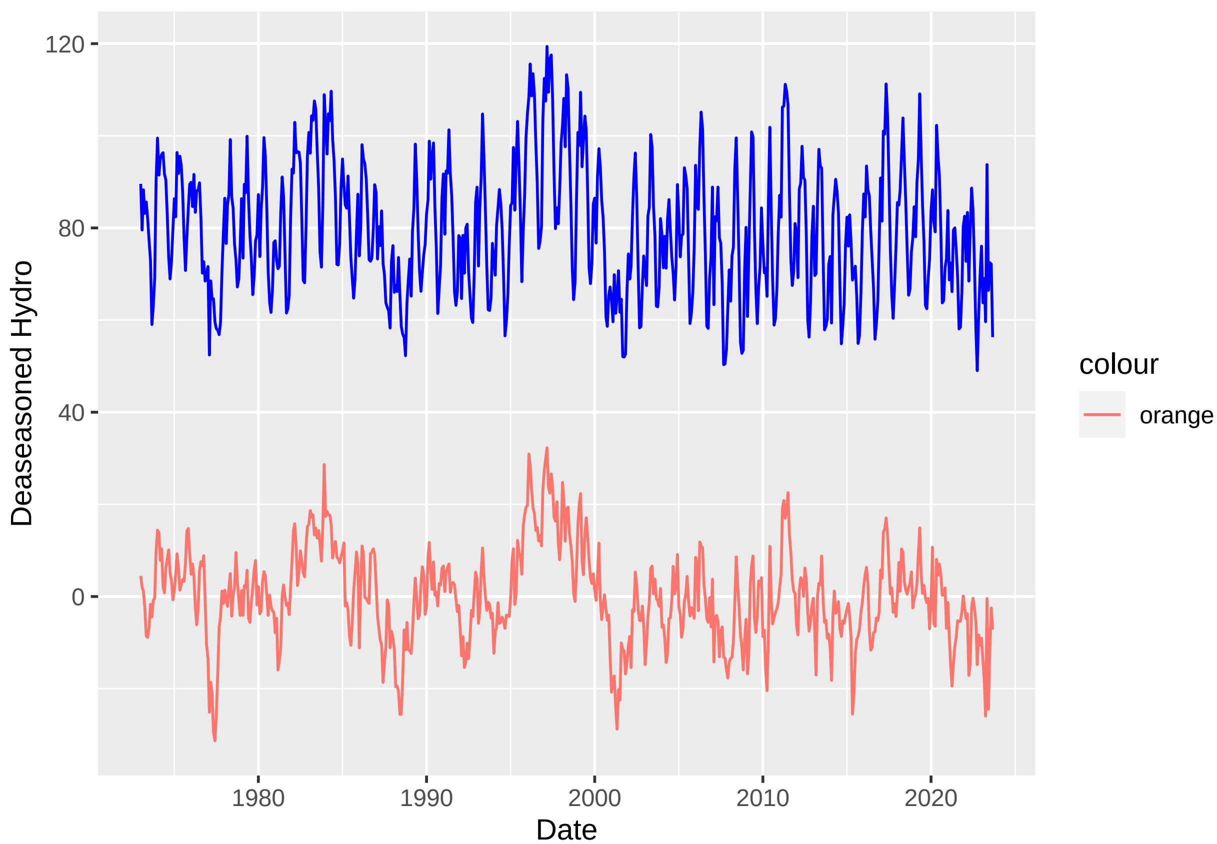
```r
#Plotting the seasonal component
ggplot(REDataset, aes(x=Date, y=Hydro)) +
            geom_line(color="blue") +
            ylab("Hydro Consumption")+
            geom_line(aes(y=Hydro_seas_comp), color="red")
```



```r
#Removing seasonal component
deseason_Hydro_data <- REDataset[,3]-Hydro_seas_comp
head(deseason_Hydro_data)
```

```
## [1]  4.472941  1.987510  1.177216 -2.448804 -8.560980 -8.871745
```

```r
#Understanding what we did
ggplot(REDataset, aes(x=Date, y=Hydro))+
  geom_line(color="blue")+
  ylab("Deaseasoned Hydro")+
  geom_line(aes(y=deseason_Hydro_data, color="orange"))
```



Analysis: Deseasoning the Hydro dataset did reduce some of its seasonality and shifted it to around zero. However, an element of seasonality does still exist in the plot and it requires additional exploration of the ACF and PACF.

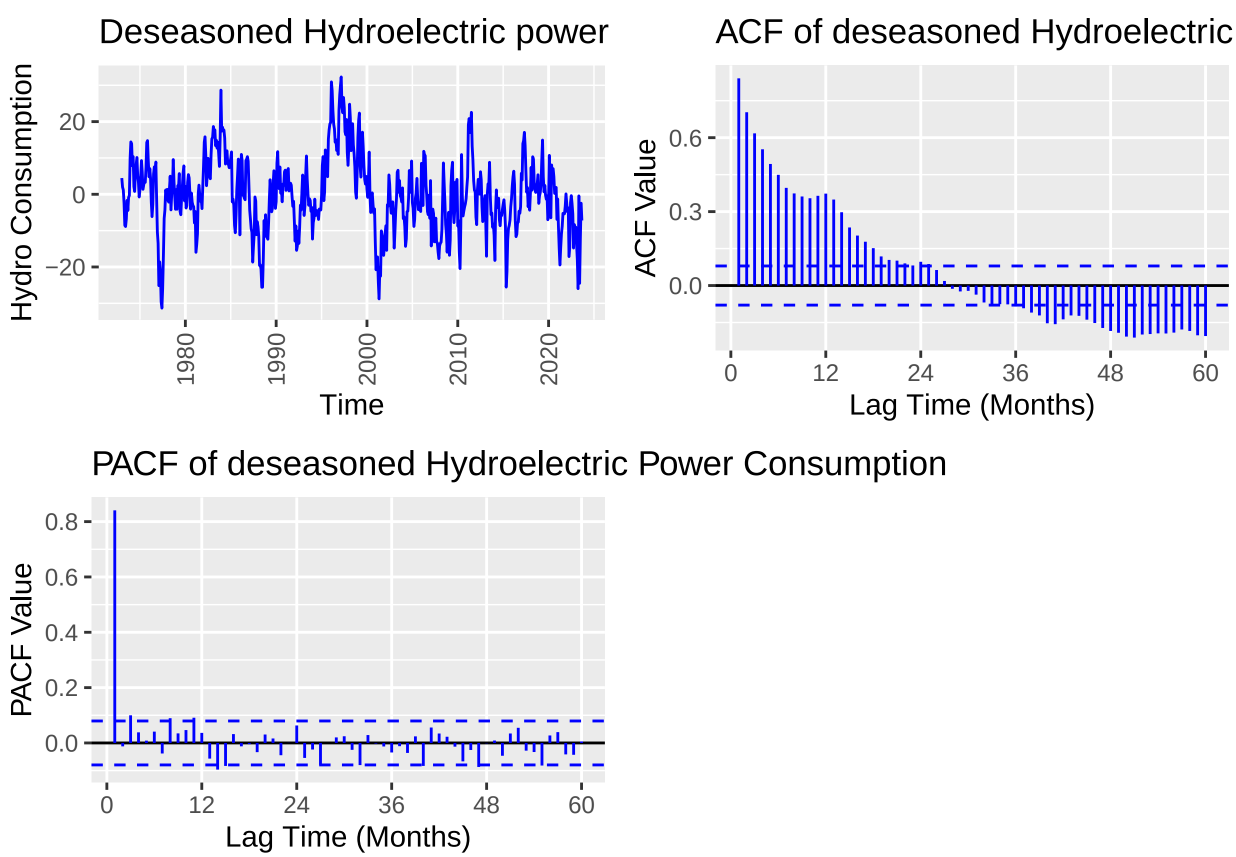
Q9

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?

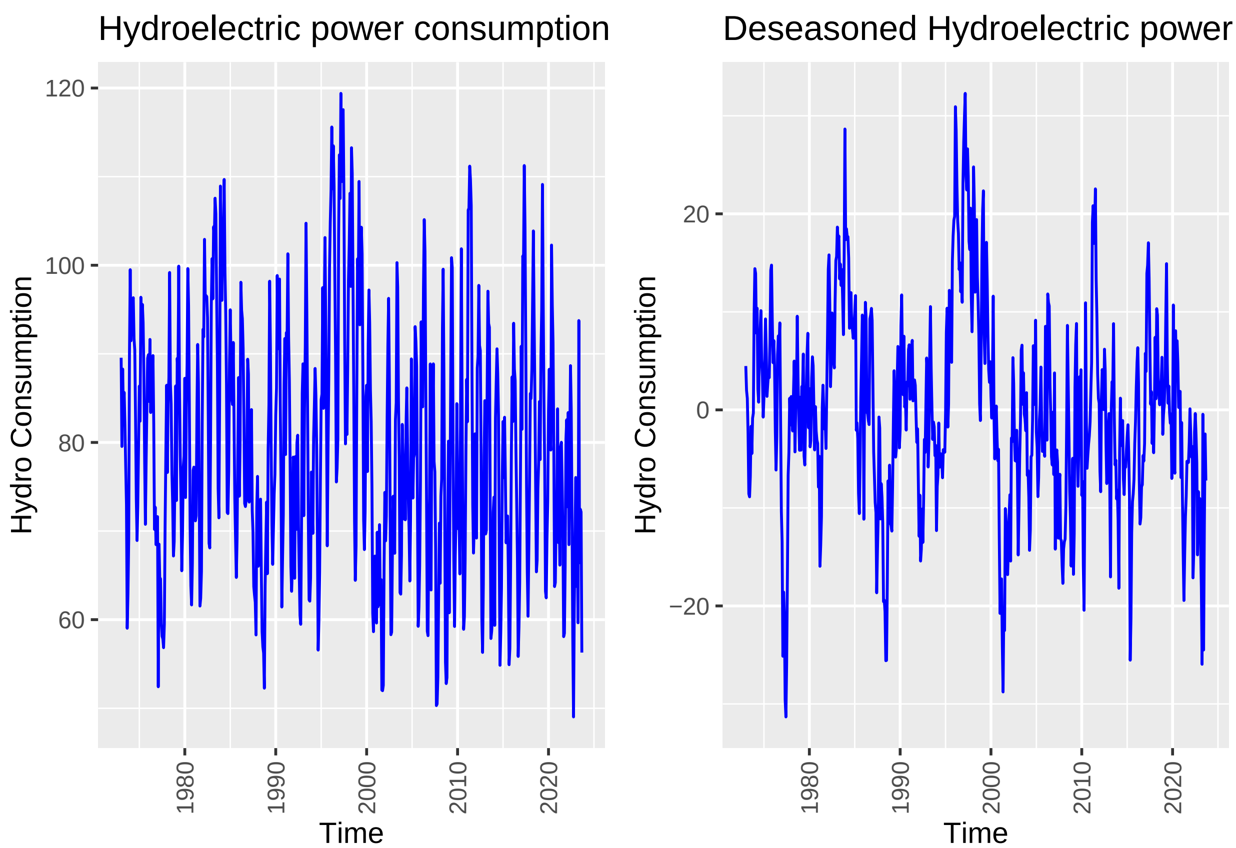
```r
#Converting to a time series
ts_deseason_hydro <- ts(deseason_Hydro_data, start=c(1973, 01), frequency = 12 )
head(ts_detrend_hydro)
```

```
##            Jan        Feb        Mar        Apr        May        Jun
## 1973  6.8371025 -3.1710482  5.5788011  0.4566504  2.9574997 -0.6156510
```

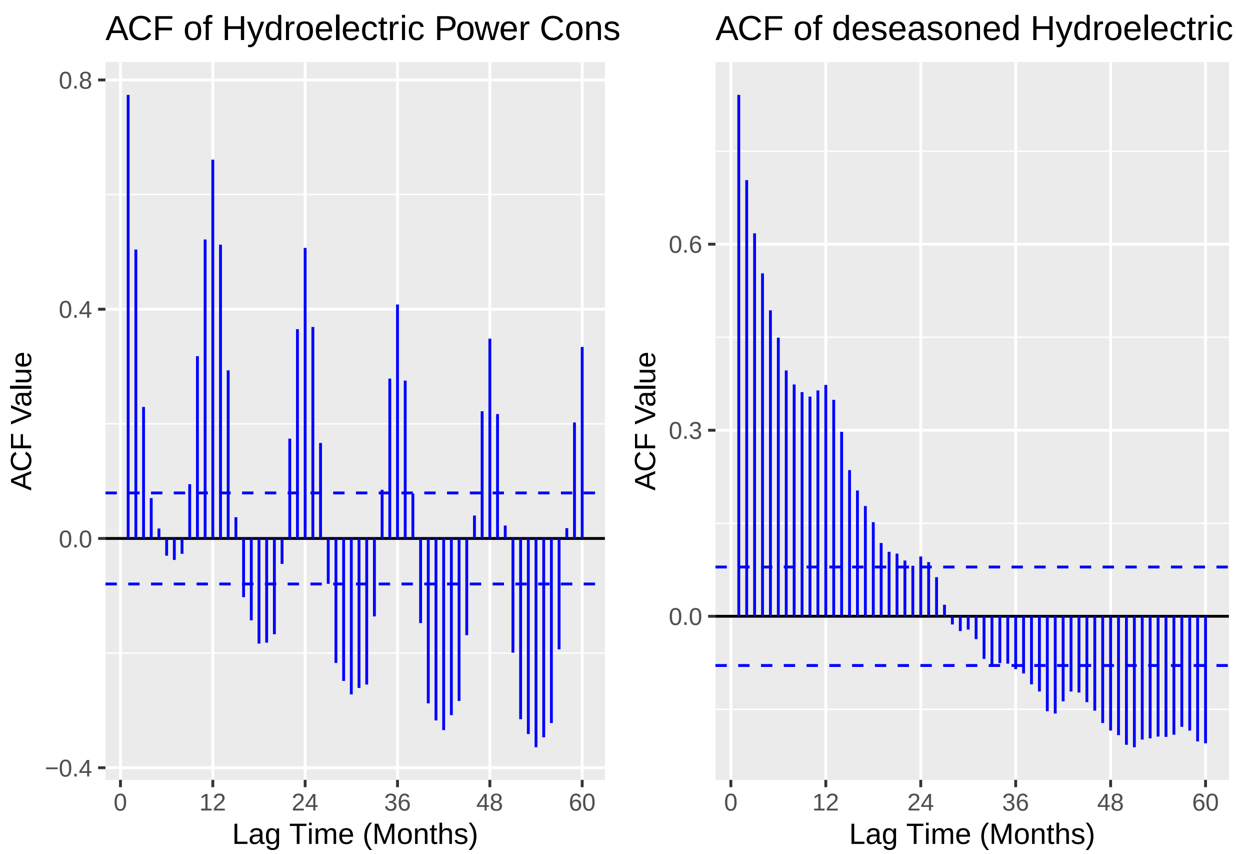
```r
#Creating plots of the deseasoned data and its ACF and PACF
ts_deseason_Hydro_plot <- autoplot(ts_deseason_hydro,  col="blue") +
  xlab("Time") + ylab("Hydro Consumption") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ggtitle("Deseasoned Hydroelectric power consumption over time")

ts_deseason_Hydro_ACF <- Acf(ts_deseason_hydro, lag.max = 60, type = "correlation", plot = FALSE)
ts_deseason_Hydro_ACF <- autoplot(ts_deseason_Hydro_ACF, col = "blue") +
  xlab ("Lag Time (Months)") + ylab ("ACF Value") +
  ggtitle("ACF of deseasoned Hydroelectric Power Consumption")

ts_deseason_Hydro_PACF <- Pacf(ts_deseason_hydro,  lag.max = 60, type = "correlation", plot = FALSE)
ts_deseason_Hydro_PACF <- autoplot(ts_deseason_Hydro_PACF, col = "blue") +
  xlab ("Lag Time (Months)") + ylab ("PACF Value") +
  ggtitle("PACF of deseasoned Hydroelectric Power Consumption")

plot_grid(ts_deseason_Hydro_plot, ts_deseason_Hydro_ACF, ts_deseason_Hydro_PACF )
```



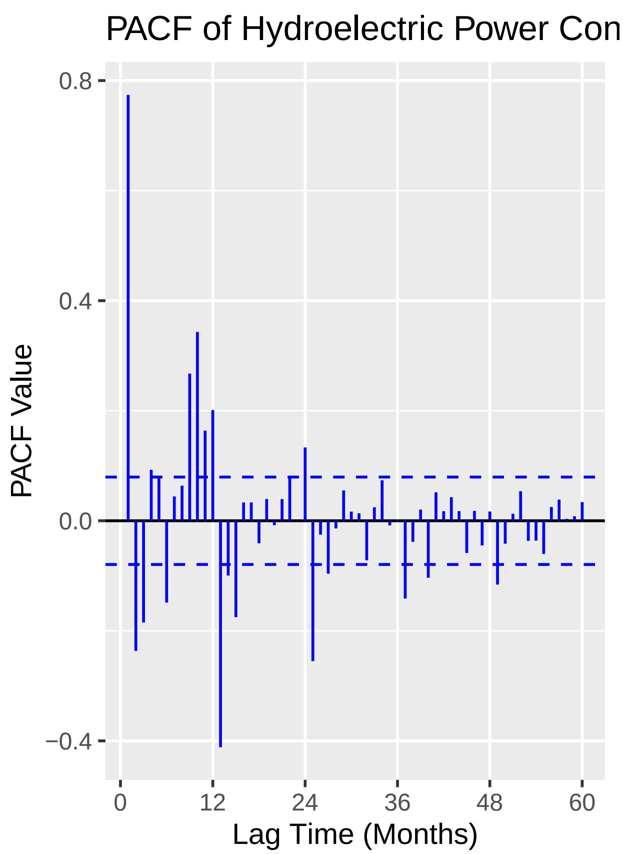

```r
plot_grid(ts_Hydro_plot, ts_deseason_Hydro_plot)
```

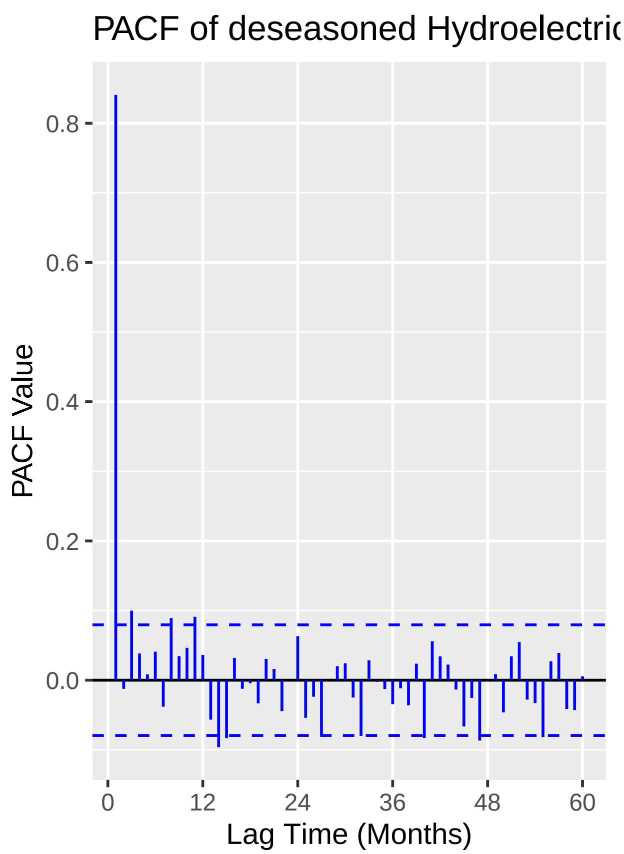



```
plot_grid(ts_Hydro_ACF, ts_deseason_Hydro_ACF)
```




```
plot_grid(ts_Hydro_PACF, ts_deseason_Hydro_PACF)
```





Analysis: The plots changed! This is most visible in the ACF graphs which went from being seasonal to linearly declining.