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

Assignment 3 - Ayoung Kim

Ayoung Kim

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.\

```
#Load/install required package here
#install.packages("forecast")
#install.packages("tseries")
#install.packages("Kendall")
#install.packages("openxlsx")
#install.packages("readxl")
#install.packages("dplyr")
#install.packages("cowplot")
#install.packages("ggplot2")


library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.3
```

```r
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.3.3
```

```r
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.3.3
```

```r
library(dplyr)
library(cowplot)
library(ggplot2)
```

```r
library(openxlsx)
```

```
## Warning: package 'openxlsx' was built under R version 4.3.3
```

```r
library(readxl)

renewable_data2 <- read.xlsx(xlsxFile="./Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sour

read_col_names2  <- read.xlsx(xlsxFile="./Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sou

#Assign the column names to the data set
colnames(renewable_data2) <- read_col_names2

#Using "select" function, selected only Total Renewable Energy Production and Hydroelectric Power Consu
renewable_data2_filtered <- select(renewable_data2,`Total Renewable Energy Production`,`Hydroelectric Po

df_renewable_data2_filtered<-as.data.frame(renewable_data2_filtered)

head(df_renewable_data2_filtered)
```

```
##   Total Renewable Energy Production Hydroelectric Power Consumption
## 1                           219.839                          89.562
## 2                           197.330                          79.544
## 3                           218.686                          88.284
## 4                           209.330                          83.152
## 5                           215.982                          85.643
## 6                           208.249                          82.060
```

```r
#1 Time series of Total Renewable Energy Production
ts1_renewable_data2_filtered<-ts(df_renewable_data2_filtered$`Total Renewable Energy Production`, start

ts1_renewable_data2_filtered
```

```
##        Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 1  219.839 197.330 218.686 209.330 215.982 208.249 207.800 203.432 185.300
## 2  231.010 210.188 226.384 223.218 227.793 218.976 221.909 214.197 200.900
```

```
## 3  214.319 198.008 224.384 215.679 223.695 217.798 216.202 206.312 194.934
## 4  236.073 221.374 237.807 224.756 234.082 229.595 235.984 228.336 211.665
## 5  228.907 194.523 225.781 216.602 221.823 211.752 215.097 214.871 208.974
## 6  260.677 233.933 258.863 255.285 272.691 254.703 258.056 250.652 241.494
## 7  270.000 239.377 273.485 265.526 283.727 264.118 262.394 257.423 243.468
## 8  298.221 271.194 294.931 293.043 310.682 299.633 295.537 281.831 268.204
## 9  299.483 273.604 293.454 286.764 305.297 305.860 308.821 296.678 276.720
## 10 320.311 297.475 330.131 316.183 323.939 316.816 321.854 310.059 289.054
## 11 348.969 320.213 352.422 343.331 355.330 346.012 345.359 338.025 315.758
## 12 355.607 333.238 358.566 348.756 363.212 344.623 348.366 340.669 317.887
## 13 353.933 323.067 344.083 334.259 349.644 332.457 332.393 328.026 315.367
## 14 326.552 307.952 349.995 338.487 345.587 334.442 335.334 325.501 316.539
## 15 334.890 296.606 327.541 315.231 330.797 311.957 317.495 311.395 302.090
## 16 334.583 307.533 326.015 316.232 331.539 315.603 317.391 315.766 306.500
## 17 348.321 317.572 358.115 346.511 350.304 349.753 351.720 358.320 341.553
## 18 329.327 321.465 353.956 334.136 317.791 289.276 315.872 332.580 311.965
## 19 370.278 292.511 317.683 293.309 320.120 313.437 309.257 340.813 345.122
## 20 366.577 305.537 311.299 292.073 282.361 323.546 333.005 347.510 324.027
## 21 373.255 322.185 359.855 330.605 313.546 304.450 309.916 346.577 324.882
## 22 388.854 323.751 354.509 332.955 303.865 313.708 366.741 333.540 307.933
## 23 336.872 299.810 346.752 361.046 333.643 342.092 400.977 399.583 349.815
## 24 385.971 343.243 385.026 325.915 356.221 375.816 395.278 398.870 347.920
## 25 397.124 342.279 381.623 374.093 398.347 362.325 382.540 370.673 343.197
## 26 386.269 323.378 360.492 348.763 374.487 309.019 358.537 354.150 332.989
## 27 383.582 328.183 334.062 355.198 401.370 353.158 379.433 360.215 328.356
## 28 319.978 334.369 366.040 364.110 361.267 326.724 351.077 343.214 312.937
## 29 303.197 272.585 301.844 288.028 290.338 298.272 297.654 304.239 279.069
## 30 314.861 279.136 302.856 309.709 331.378 326.674 337.792 311.593 302.858
## 31 318.956 291.767 330.201 327.749 345.099 341.209 342.647 333.101 308.470
## 32 347.154 321.055 342.168 334.068 344.066 346.968 353.034 344.004 328.252
## 33 361.269 333.479 354.763 342.863 367.186 362.264 372.396 356.107 331.447
## 34 388.583 348.049 368.883 367.940 386.890 383.011 381.340 370.019 345.317
## 35 399.004 343.865 390.167 383.102 398.044 382.096 393.450 386.428 360.587
## 36 427.860 388.671 424.851 421.184 449.522 444.695 446.062 431.761 398.411
## 37 431.011 386.812 432.104 431.059 456.231 455.356 455.962 449.335 421.927
## 38 489.844 449.090 499.560 482.552 507.544 517.750 508.593 497.073 476.105
## 39 530.909 490.715 553.169 538.506 554.011 553.885 549.374 534.036 500.175
## 40 539.030 494.125 541.241 519.625 546.973 528.767 520.173 513.269 475.611
## 41 542.692 487.697 541.012 551.448 578.378 563.561 572.289 542.610 514.219
## 42 574.074 507.104 589.448 582.906 589.532 590.551 588.452 559.856 530.545
## 43 580.459 532.998 579.274 569.372 578.595 564.148 583.940 572.235 539.599
## 44 599.152 581.670 626.078 589.659 609.017 593.636 604.272 591.306 562.170
## 45 627.073 580.264 663.855 635.068 661.222 642.277 625.487 612.088 583.803
## 46 652.294 609.263 668.458 656.425 680.571 668.645 647.806 651.821 600.580
## 47 644.675 593.023 656.855 665.815 689.814 661.001 666.840 647.495 612.975
## 48 648.257 632.086 641.509 560.555 618.177 637.050 632.878 618.503 583.472
## 49 636.532 552.157 677.204 650.405 688.670 656.020 650.413 648.043 619.939
## 50 696.686 651.094 732.321 711.645 742.103 724.756 712.392 671.642 631.913
## 51 696.038 659.518 735.318 708.522 740.890 698.192 715.729 713.484 672.812
## 52 684.313 698.914 771.513 750.907 762.088 757.944 746.007 751.485 695.378
##        Oct     Nov     Dec
## 1  193.514 195.326 220.755
## 2  200.312 200.068 211.046
## 3  206.489 208.436 217.911
```

```
## 4  218.818 209.968 216.239
## 5  216.727 222.663 235.754
## 6  241.095 237.214 250.285
## 7  253.559 255.317 262.637
## 8  273.058 270.913 288.131
## 9  284.684 280.364 304.193
## 10 296.056 300.864 323.054
## 11 320.524 325.785 357.437
## 12 326.373 323.172 343.652
## 13 327.776 330.222 346.947
## 14 325.125 323.172 341.787
## 15 309.095 297.439 319.908
## 16 310.737 313.792 326.992
## 17 356.682 359.731 367.555
## 18 312.873 301.883 341.584
## 19 324.454 318.757 355.690
## 20 340.565 345.048 360.200
## 21 331.480 338.485 352.074
## 22 343.569 338.304 348.732
## 23 384.663 366.200 373.129
## 24 400.155 387.043 378.537
## 25 402.188 355.868 355.807
## 26 345.379 309.809 370.867
## 27 308.985 337.650 332.407
## 28 341.025 339.223 333.069
## 29 292.015 283.668 302.843
## 30 315.739 309.716 328.629
## 31 313.818 314.096 347.074
## 32 332.739 332.106 367.856
## 33 339.018 338.541 360.826
## 34 353.690 359.164 376.761
## 35 374.075 373.327 397.970
## 36 412.573 409.976 428.996
## 37 450.940 456.527 481.882
## 38 489.125 500.488 524.855
## 39 517.691 528.710 552.823
## 40 491.520 489.081 527.555
## 41 543.689 548.475 574.712
## 42 557.212 569.440 593.582
## 43 556.624 575.262 607.029
## 44 584.344 586.159 650.886
## 45 614.591 613.732 635.064
## 46 627.834 623.070 647.358
## 47 633.410 620.528 650.319
## 48 611.896 629.909 640.842
## 49 649.287 662.792 705.767
## 50 658.345 684.997 679.561
## 51 693.952 682.056 720.952
## 52
```

```r
#2 Time series of Hydroelectricc Power Consumption
ts2_renewable_data2_filtered <-ts(df_renewable_data2_filtered$`Hydroelectric Power Consumption`,start =

ts2_renewable_data2_filtered
```

```
##        Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 1   89.562  79.544  88.284  83.152  85.643  82.060  77.400  72.936  59.029
## 2   99.500  91.476  94.950  95.969  96.337  91.719  90.437  82.727  73.610
## 3   86.356  82.404  96.386  91.791  95.581  93.550  87.900  77.892  70.756
## 4   89.904  84.626  91.629  83.378  88.065  88.182  89.807  82.153  70.186
## 5   71.630  52.424  68.518  64.508  64.629  59.609  58.130  57.830  56.835
## 6   86.454  76.606  84.951  87.281  99.185  86.645  84.339  76.518  73.042
## 7   86.378  73.446  89.483  87.645  99.903  86.230  78.573  73.393  65.516
## 8   87.244  73.781  83.978  88.865  99.622  95.451  84.448  70.517  63.819
## 9   77.214  72.830  71.150  71.718  83.301  91.061  86.714  74.556  61.534
## 10  92.763  91.907 102.924  96.303  96.572  96.463  94.087  82.333  68.612
## 11 100.743  96.206 104.348 103.334 107.568 105.810  96.883  88.929  74.808
## 12 102.459  96.034 104.801 103.270 109.683  99.261  94.772  86.573  72.076
## 13  94.973  89.219  85.029  84.276  91.284  82.425  73.612  68.980  64.761
## 14  73.934  80.075  98.081  94.922  93.958  90.562  83.094  73.104  72.767
## 15  87.702  73.264  80.279  76.163  83.712  72.271  69.864  63.744  62.756
## 16  76.171  66.029  67.539  66.195  73.601  65.346  58.636  56.923  56.241
## 17  73.277  65.188  79.268  84.295  98.200  90.614  79.373  70.675  66.237
## 18  83.078  86.122  98.841  90.589  95.753  98.444  84.490  75.559  61.422
## 19  91.732  78.638  92.384  91.874 101.296  91.854  86.984  77.658  65.997
## 20  77.844  64.670  78.408  70.180  79.980  80.819  70.671  65.474  60.474
## 21  88.873  71.730  83.868  90.980 104.743  95.182  84.360  71.587  62.245
## 22  72.773  69.738  80.508  84.264  88.353  85.343  79.712  69.257  56.561
## 23  84.852  85.447  97.479  83.847  94.791 103.128  93.729  83.175  68.333
## 24 104.821 108.488 115.603 108.647 113.485 108.928  98.519  89.620  75.536
## 25 112.458 107.528 119.397 109.441 116.635 117.564 108.011  91.802  79.836
## 26  98.328 102.347 108.119  97.610 113.264 110.348  97.785  85.678  71.060
## 27 100.724  97.902 109.456  93.280  98.936 104.294 101.288  87.778  71.283
## 28  86.468  76.714  91.110  97.207  93.560  85.931  82.333  74.999  60.736
## 29  64.323  59.617  69.868  61.460  65.427  70.723  61.686  64.534  52.054
## 30  74.364  68.894  71.682  82.729  90.973  96.262  86.906  71.938  58.300
## 31  70.287  67.489  82.578  84.478 100.297  97.536  84.765  78.381  63.055
## 32  78.419  71.357  78.184  71.270  81.955  86.161  79.562  73.672  70.032
## 33  82.817  73.722  78.258  78.675  93.074  91.384  88.565  73.582  59.245
## 34  93.614  84.487  84.019  97.432 105.153 101.532  86.799  74.137  58.691
## 35  88.865  63.349  82.446  81.515  88.872  77.850  76.694  68.037  50.302
## 36  70.898  64.108  73.934  75.862  92.879  99.553  87.194  72.434  55.200
## 37  80.149  60.775  74.475  87.927 100.858  99.744  79.789  66.808  59.228
## 38  76.371  70.252  71.262  65.158  85.570 101.861  83.651  68.647  58.909
## 39  87.112  82.336 106.231 106.435 111.187 109.700 106.743  87.905  72.940
## 40  78.842  69.209  88.393  89.720  97.724  90.956  90.387  78.591  60.065
## 41  84.715  69.668  70.063  85.631  97.072  93.434  92.993  73.813  57.871
## 42  73.815  59.356  82.765  86.801  90.568  87.838  83.107  67.582  54.846
## 43  82.360  76.040  82.846  76.671  68.668  69.653  71.701  65.245  54.913
## 44  87.397  82.362  93.454  88.296  86.960  79.284  73.206  66.771  55.847
## 45  90.854  81.485 101.040 100.345 111.255 104.323  90.753  75.180  65.346
## 46  85.519  84.967  88.236  95.929 103.876  94.163  85.640  75.122  65.393
## 47  84.610  78.068  89.852  94.922 109.123  95.801  84.875  77.038  63.210
## 48  83.587  88.262  81.284  79.139 102.279  95.534  91.243  79.443  63.732
## 49  83.799  68.706  72.404  66.155  79.530  80.025  75.397  69.360  58.080
## 50  82.562  72.746  83.377  68.465  79.700  88.670  83.824  72.106  58.093
## 51  77.637  68.107  72.783  67.625  94.346  73.604  74.988  72.652  57.716
## 52  74.805  68.583  79.551  66.116  77.156  72.234  72.288  72.875  56.844
##        Oct     Nov     Dec
```

```
## 1   62.967  69.063  90.131
## 2   68.931  72.773  79.542
## 3   78.060  84.171  89.510
## 4   72.690  68.463  69.900
## 5   59.480  70.583  78.744
## 6   67.184  68.818  76.162
## 7   69.619  77.213  78.457
## 8   61.661  66.325  76.858
## 9   62.420  65.459  82.279
## 10  68.091  80.245  95.522
## 11  71.491  84.956 108.936
## 12  71.968  76.704  88.949
## 13  69.105  79.075  87.328
## 14  73.498  79.755  89.397
## 15  61.964  58.272  72.753
## 16  52.265  63.762  68.748
## 17  70.285  74.172  76.402
## 18  66.657  71.863  86.440
## 19  63.197  66.085  78.349
## 20  59.474  69.964  85.579
## 21  62.087  64.729  76.662
## 22  59.757  65.325  75.959
## 23  78.993  87.148  99.640
## 24  77.094  80.374 103.400
## 25  84.394  80.900  88.252
## 26  64.434  68.310  85.937
## 27  67.908  72.210  85.198
## 28  58.639  65.377  67.181
## 29  51.980  52.589  66.010
## 30  58.589  67.319  73.933
## 31  62.878  67.268  82.039
## 32  64.360  71.437  89.431
## 33  61.438  66.031  75.546
## 34  58.192  69.167  73.685
## 35  50.485  53.507  62.582
## 36  52.783  53.459  71.179
## 37  67.186  71.678  84.378
## 38  60.334  66.744  79.053
## 39  67.515  70.562  80.973
## 40  56.304  63.918  78.423
## 41  58.682  60.313  72.090
## 42  58.547  63.548  76.186
## 43  56.743  65.981  79.041
## 44  59.160  64.174  76.865
## 45  60.386  67.859  75.910
## 46  66.698  74.766  77.784
## 47  62.459  68.982  73.284
## 48  64.181  71.286  73.385
## 49  58.458  66.102  80.393
## 50  49.022  61.068  69.706
## 51  53.475  58.092  64.922
## 52
```

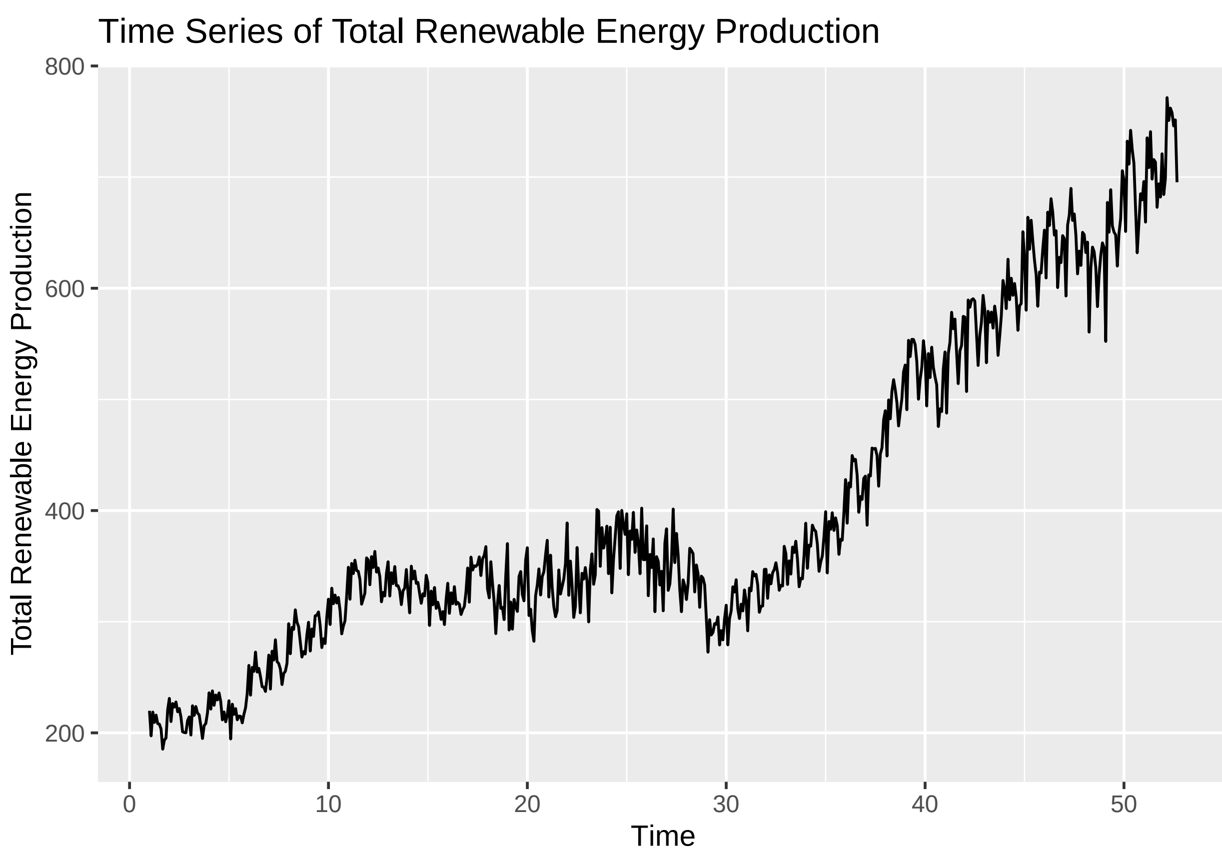
##Trend Component

Q1

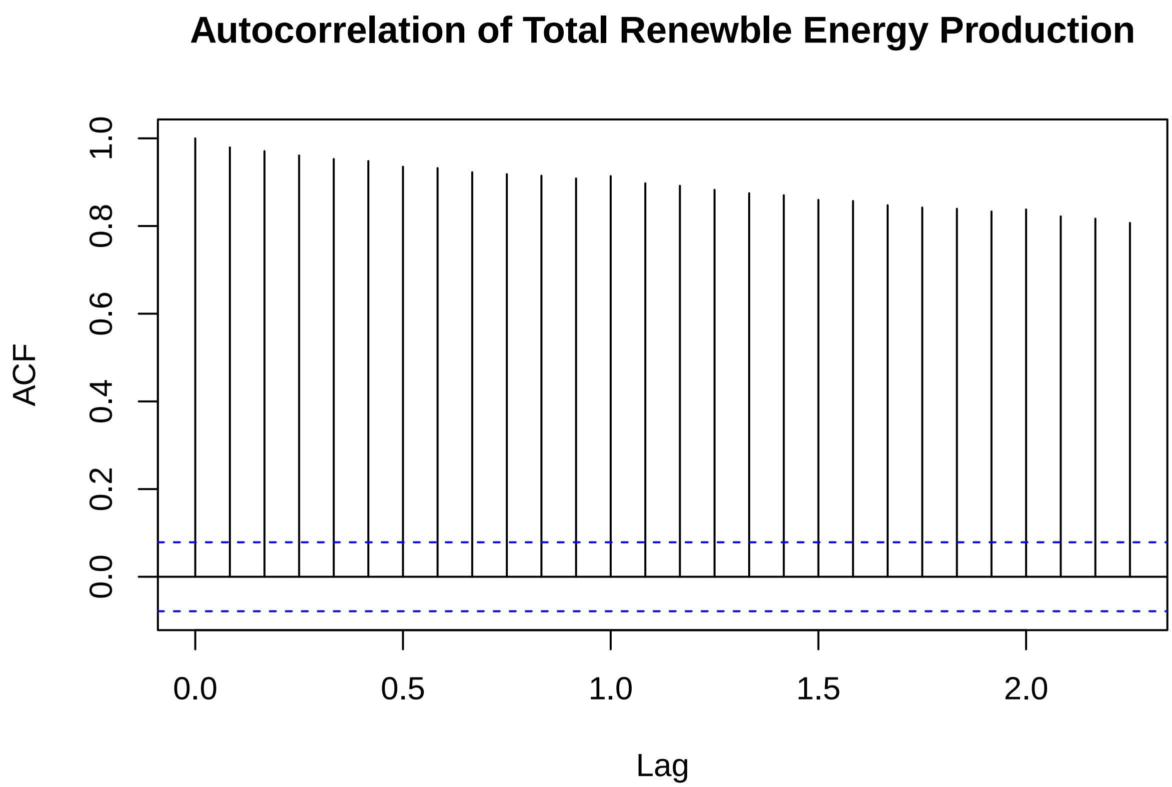
For each time series, i.e., Renewable Energy Production and Hydroelectric Consumption create three plots: one with time series, one with the ACF and with the PACF. You may use the some code form A2, but I want all the three plots side by side as in a grid. (Hint: use function `plot_grid()` from the `cowplot`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

#Checked my code with AI.

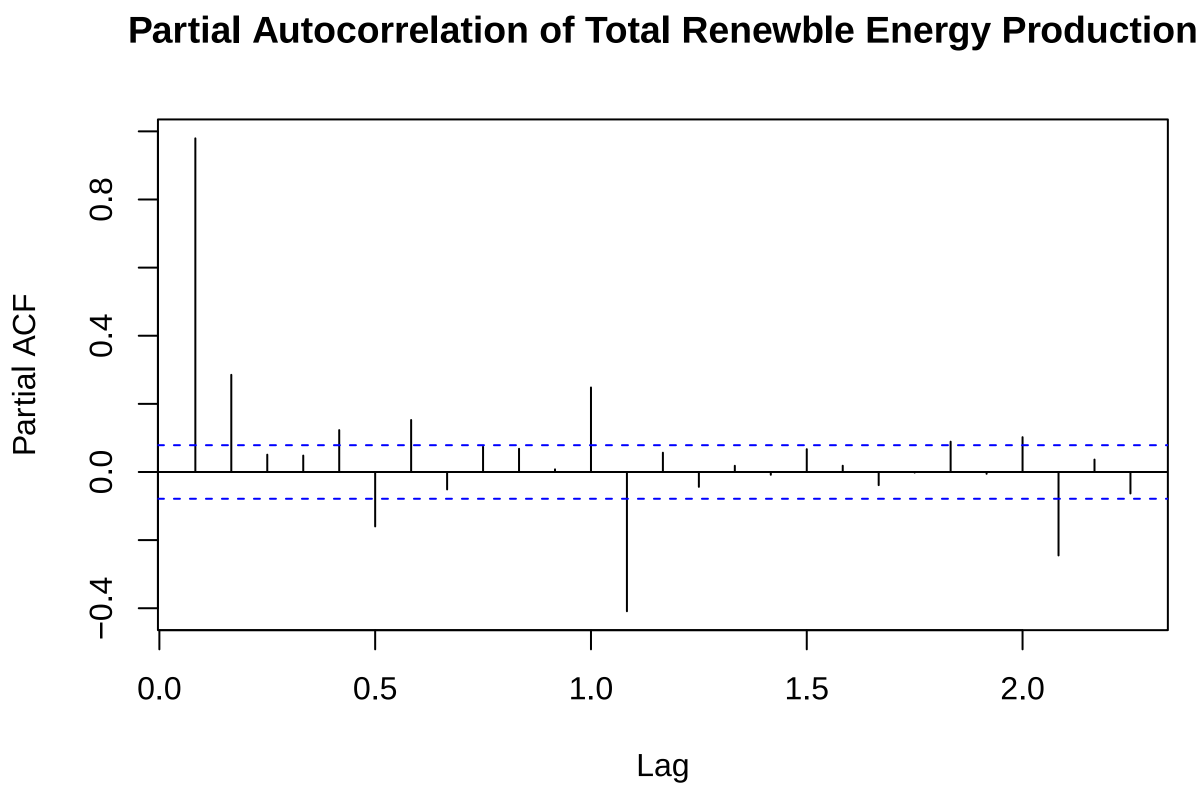
```
#Renewable Energy Production Plot1 - Time series
ts_plot1<-autoplot(ts1_renewable_data2_filtered) +
  labs(title = "Time Series of Total Renewable Energy Production",
       y = "Total Renewable Energy Production",
       x = "Time")
ts_plot1
```



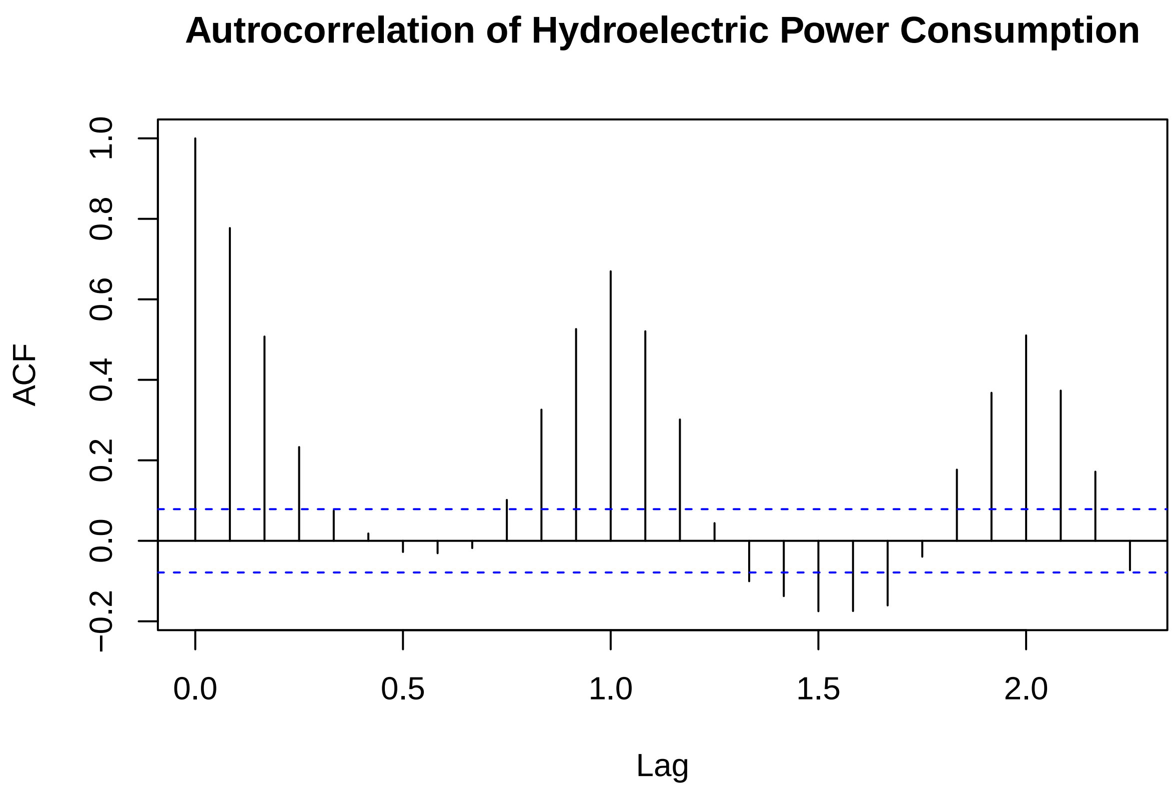

```
#Renewable Energy Production Plot2 - ACF
acf_ts1<-acf(ts1_renewable_data2_filtered,
             main="Autocorrelation of Total Renewble Energy Production",
             type="correlation",plot=TRUE)
```



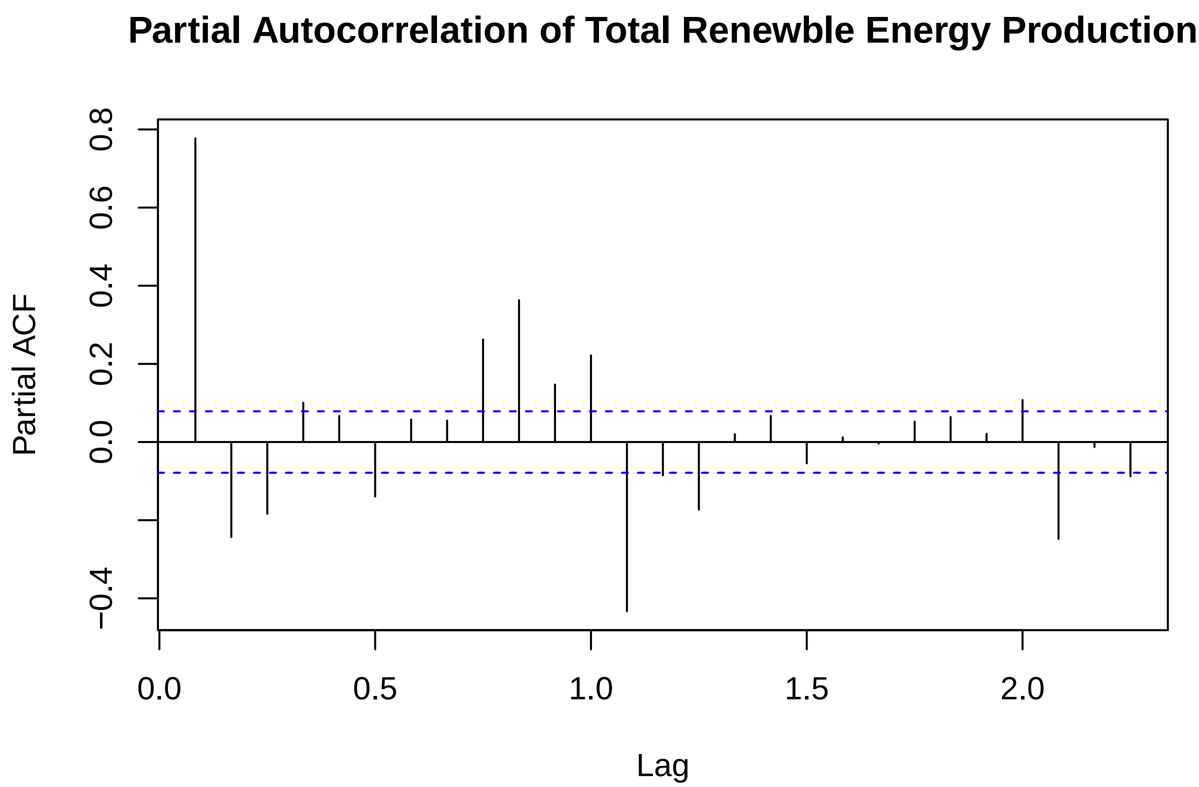

```r
acf_ts1_1 <- ggAcf(ts1_renewable_data2_filtered) +
  ggtitle("Autocorrelation of Total Renewable Energy Production")


#Renewable Energy Production Plot3 - PACF
pacf_ts1<-pacf(ts1_renewable_data2_filtered,
               main="Partial Autocorrelation of Total Renewble Energy Production",
               plot=TRUE)
```



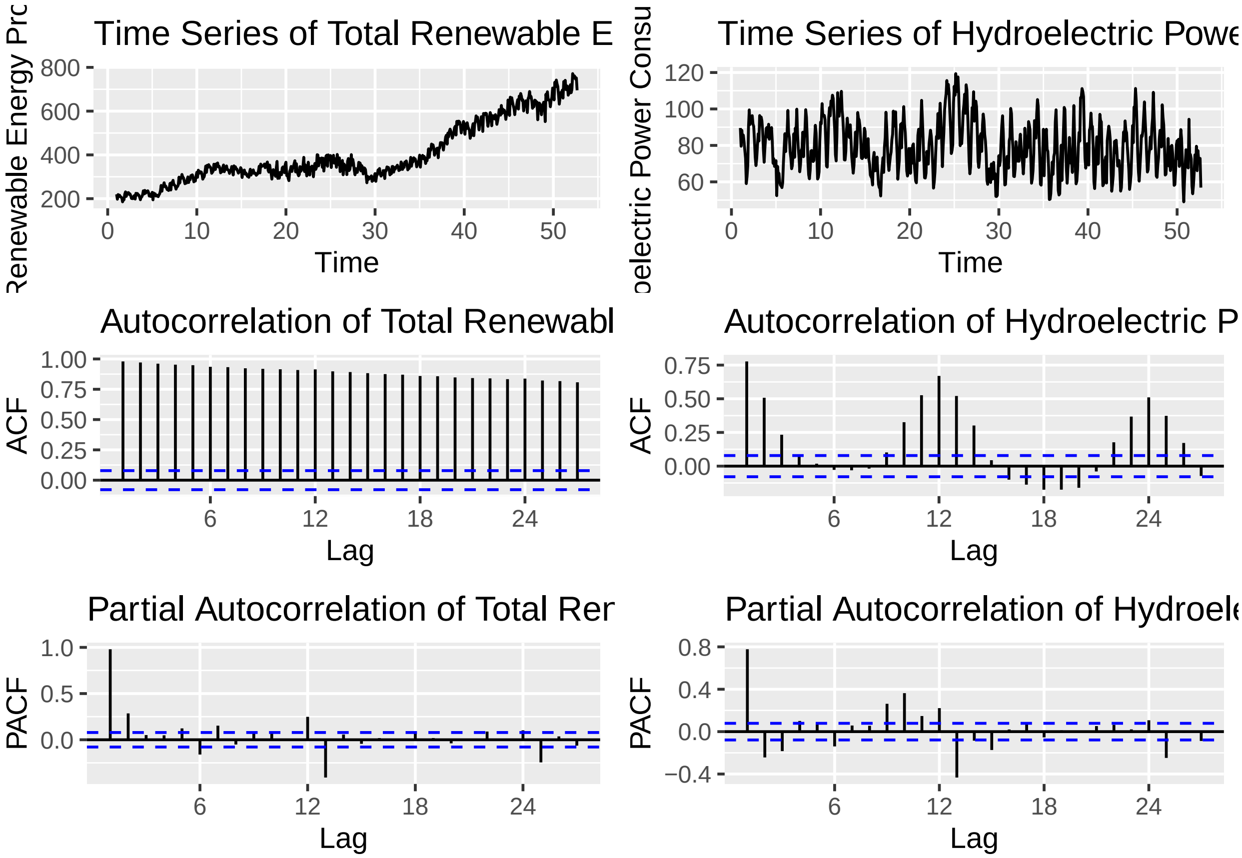

```
pacf_ts1_1 <- ggPacf(ts1_renewable_data2_filtered) +
  ggtitle("Partial Autocorrelation of Total Renewable Energy Production")

#Hydroelectric Consumption Plot 1- Time Series
ts_plot2<-autoplot(ts2_renewable_data2_filtered) +
  labs(title = "Time Series of Hydroelectric Power Consumption",
       y = "Hydroelectric Power Consumption",
       x = "Time")

#Hydroelectric Consumption Plot 2- ACF
acf_ts2<-acf(ts2_renewable_data2_filtered,
             main="Autrocorrelation of Hydroelectric Power Consumption",
             plot=TRUE)
```




```
acf_ts2_1<- ggAcf(ts2_renewable_data2_filtered) +
  ggtitle("Autocorrelation of Hydroelectric Power Consumption")

#Hydroelectric Consumption Plot 3- PACF
pacf_ts2<-pacf(ts2_renewable_data2_filtered,
               main="Partial Autocorrelation of Total Renewble Energy Production",
               plot=TRUE)
```




```
pacf_ts2_1 <- ggPacf(ts2_renewable_data2_filtered) +
  ggtitle("Partial Autocorrelation of Hydroelectric Power Consumption")

#Grid version of plots
plot_grid(ts_plot1,ts_plot2,acf_ts1_1,acf_ts2_1,pacf_ts1_1,pacf_ts2_1,ncol = 2, nrow = 3)
```



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?

Answer: The series of Total Renewable Energy Production seems to have an increasing trend while Hydroelectric power consumption appears to have a seasonal pattern (trend) and a slight downward trend.

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.

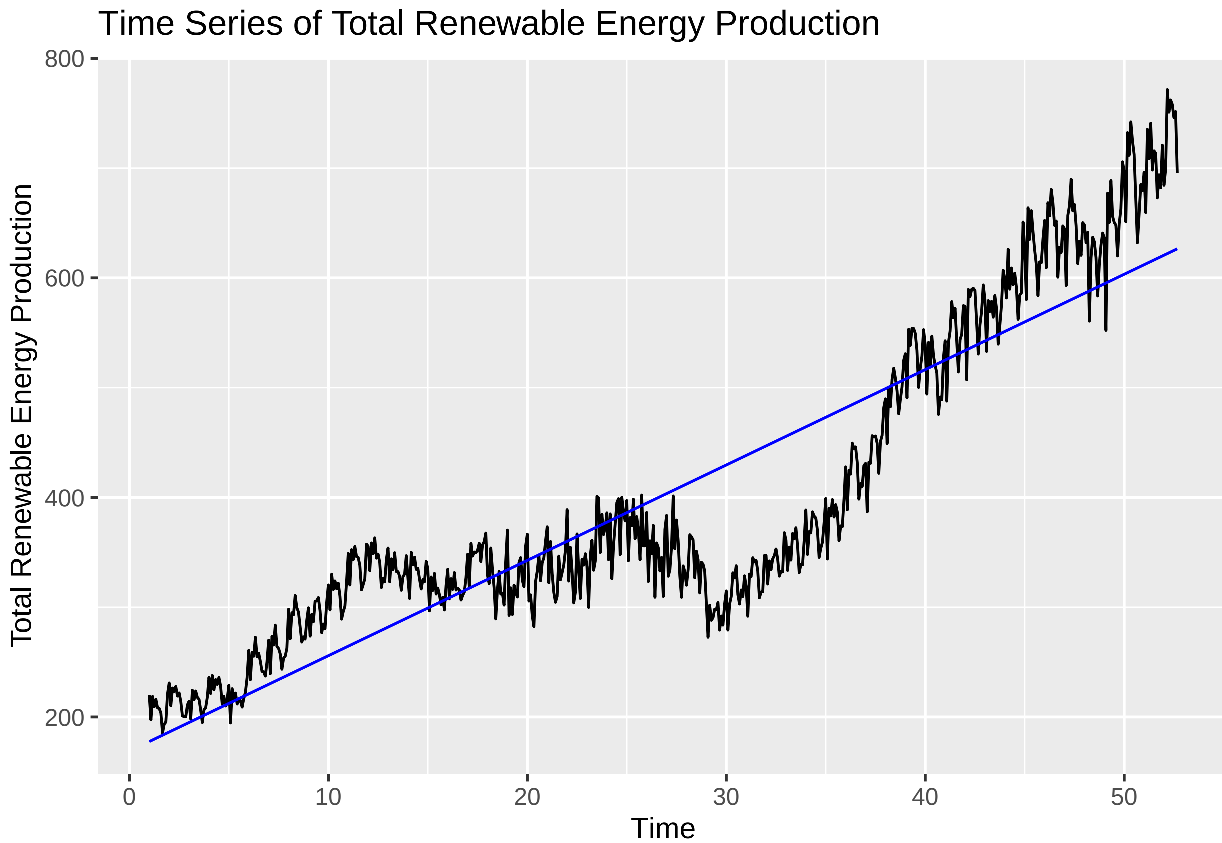
```
#Fit a linear trend to Total Renewable Energy Production
#nobs<-nrow(renewable_data2_filtered)
#t<-1:nobs

nobs<-nrow(renewable_data2_filtered)
t<-1:nobs
ts1_lm<-lm(ts1_renewable_data2_filtered~t)

#Print the summary of the regression
print(ts1_lm)
```

```
##
## Call:
## lm(formula = ts1_renewable_data2_filtered ~ t)
##
## Coefficients:
## (Intercept)            t
##    176.8729       0.7239
```

```r
summary(ts1_lm)
```

```
##
## Call:
## lm(formula = ts1_renewable_data2_filtered ~ 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
#linear trend in the plot (ts1-Renewable Energy Production)
plot_ts1_lm<-autoplot(ts1_renewable_data2_filtered) +
  geom_line(aes(y = fitted(ts1_lm)), color = "blue") +
  labs(title = "Time Series of Total Renewable Energy Production",
       y = "Total Renewable Energy Production",
       x = "Time")

plot_ts1_lm
```




```r
#Fit a linear trend to Hydroelectric Power consumption
ts2_lm<-lm(ts2_renewable_data2_filtered~t)

#Print the summary of the regression
print(ts2_lm)
```

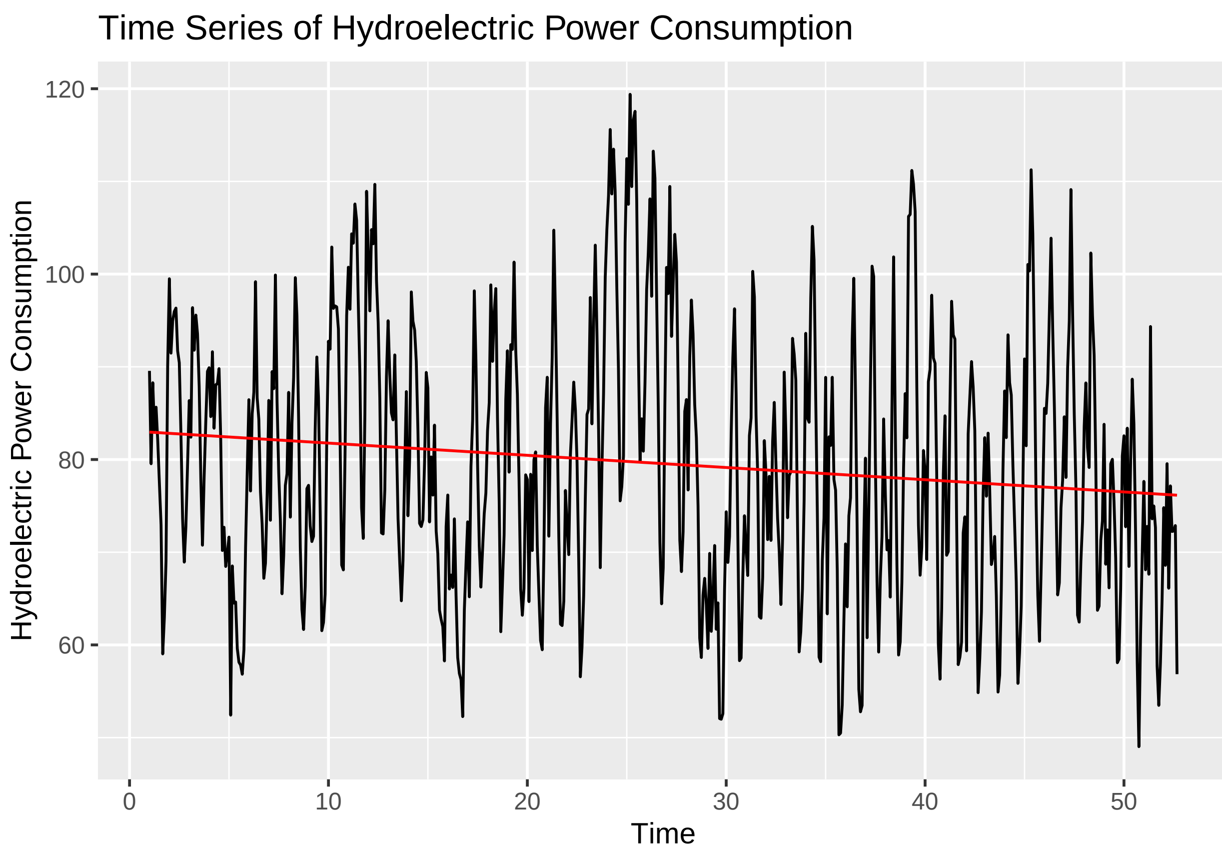
```
##
## Call:
## lm(formula = ts2_renewable_data2_filtered ~ t)
##
## Coefficients:
## (Intercept)            t
##    82.96767     -0.01098
```

```r
summary(ts2_lm)
```

```
##
## Call:
## lm(formula = ts2_renewable_data2_filtered ~ 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
#linear trend in the plot (ts2-Hydroelectric Power Consumption)
plot_ts2_lm<-autoplot(ts2_renewable_data2_filtered) +
  geom_line(aes(y = fitted(ts2_lm)), color = "red") +
  labs(title = "Time Series of Hydroelectric Power Consumption",
       y = "Hydroelectric Power Consumption",
       x = "Time")

plot_ts2_lm
```



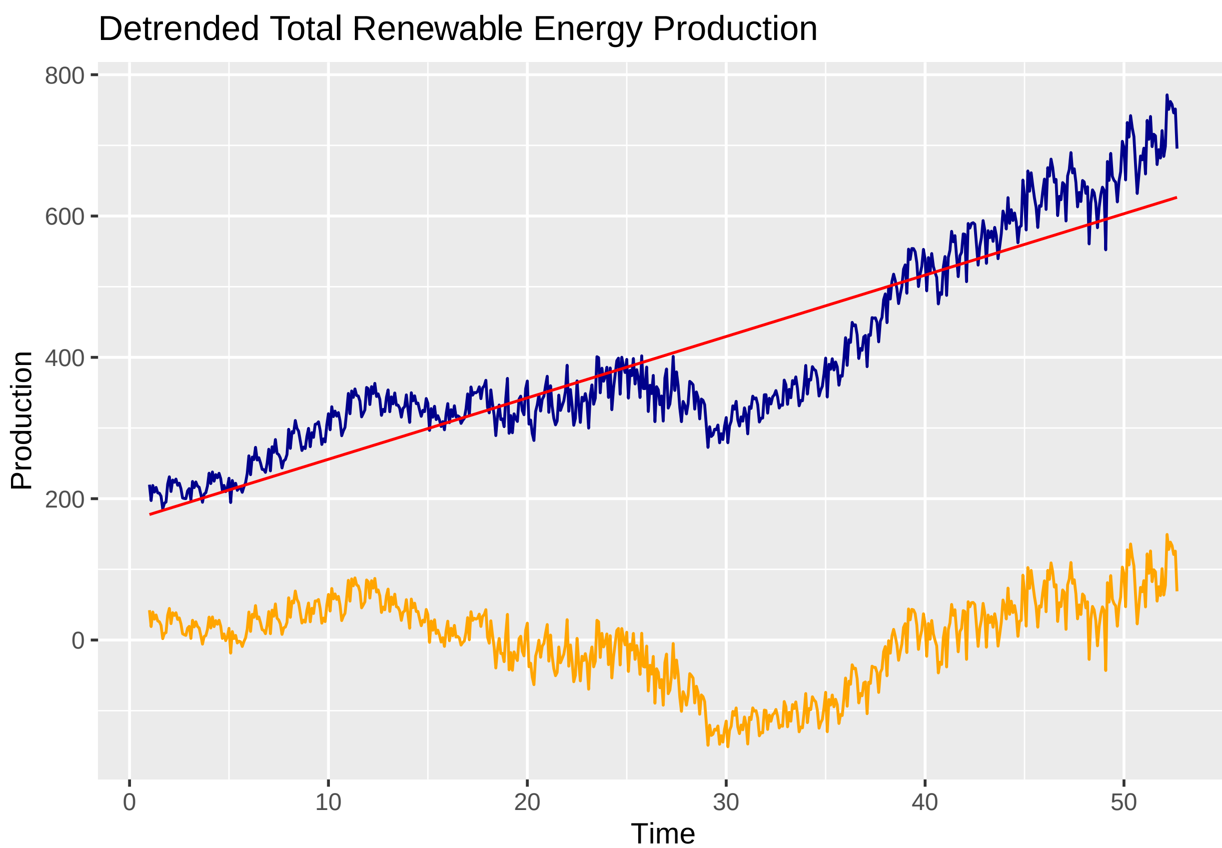
#Answer (Interpretation of the regression output): According to the summary of ts1_lm, the slope of ts1_lm, which is the linear regression of Total Renewable Energy Production, is around 0.72 and the its intercept is 176.87. It means that t=0 starts with 176.87 and it the production increases 0.72 unit as time increases. The slope of ts2_lm is around -0.11 with the intercept of 82.97. It means that it is 82.97 at t=0 and it decreases 0.11 unit as time goes by.

Q4

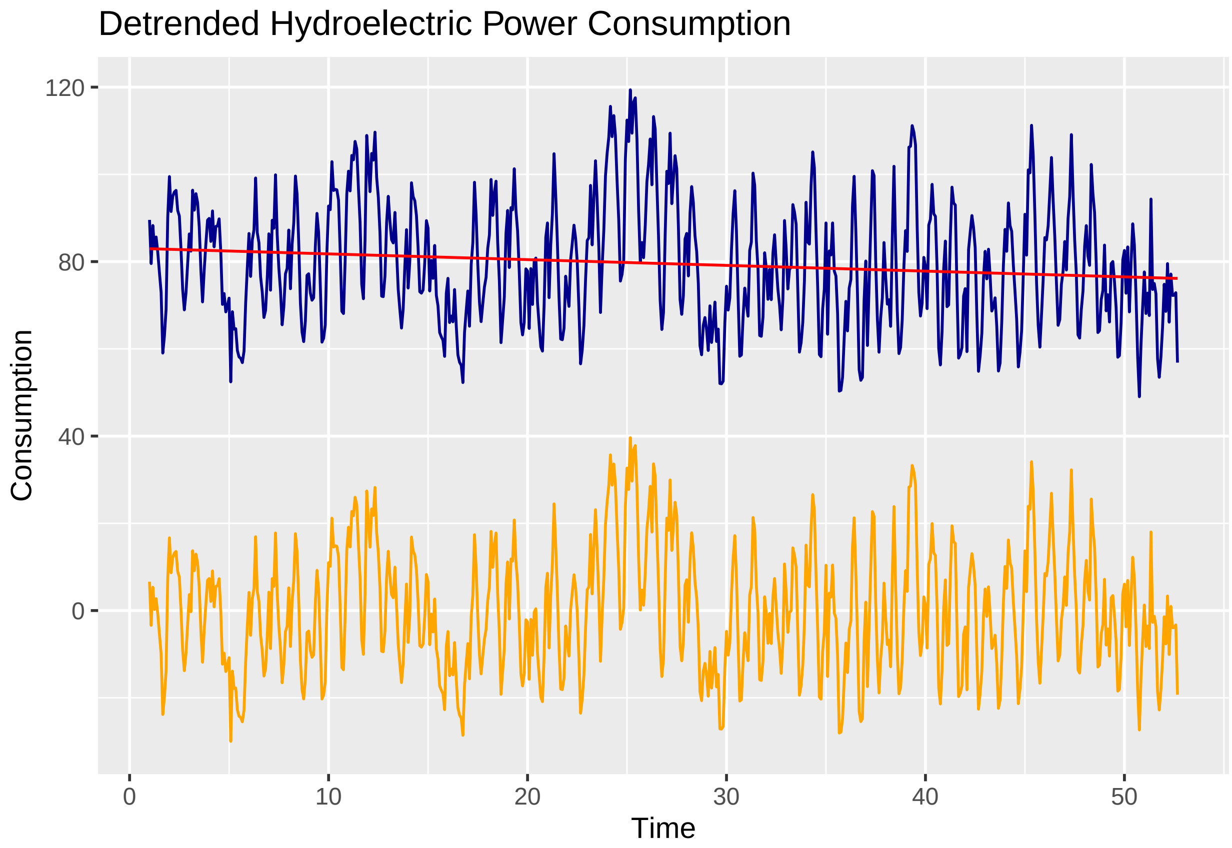
Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

Answer: Unlike the plots in Q1 and Q2 with only one series, the plots in Q4 have 2 different series. The dark blue or navy line is the original series, red line is the trend, and the orange line is the detrended series in each plot.

```r
#Detrend series
beta0_ts1 <- as.numeric(ts1_lm$coefficients[1])
beta1_ts1 <- as.numeric(ts1_lm$coefficients[2])

beta0_ts2 <- as.numeric(ts2_lm$coefficients[1])
beta1_ts2 <- as.numeric(ts2_lm$coefficients[2])

linear_trend_ts1 <- beta0_ts1 + beta1_ts1 * t
linear_trend_ts2 <- beta0_ts2 + beta1_ts2 * t

ts1_linear <- ts(linear_trend_ts1,start = c(1,1), frequency=12)
ts2_linear <- ts(linear_trend_ts2,start = c(1,1), frequency=12)

detrend_renewable_ts1 <- ts1_renewable_data2_filtered - linear_trend_ts1
detrend_renewable_ts2 <- ts2_renewable_data2_filtered - linear_trend_ts2

ts1_detrend_renewable <- ts(detrend_renewable_ts1, start = c(1,1), frequency=12)
ts2_detrend_renewable <- ts(detrend_renewable_ts2, start = c(1,1), frequency=12)

#Plot 1 - Detrended Total Renewable Energy Production
ts1_detrended_plot<-autoplot(ts1_renewable_data2_filtered, color = "darkblue") +
  autolayer(ts1_detrend_renewable, series = "Detrended", color = "orange") +
  autolayer(ts1_linear, series = "Linear Component", color = "red") +
  labs(title = "Detrended Total Renewable Energy Production", x = "Time", y = "Production")

ts1_detrended_plot
```



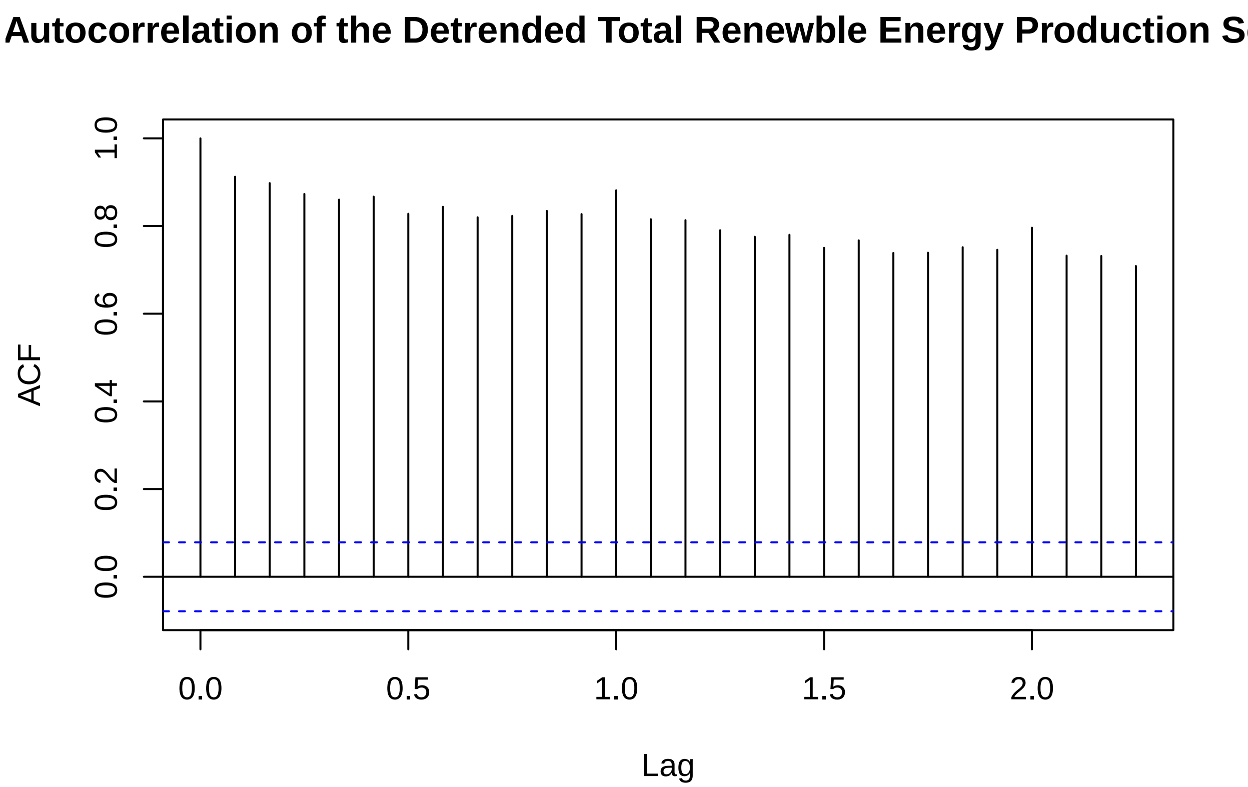

```r
#Plot 2 - Detrended Hydroelectric Power Consumption
ts2_detrended_plot<-autoplot(ts2_renewable_data2_filtered, color = "darkblue") +
  autolayer(ts2_detrend_renewable, series = "Detrended", color = "orange") +
  autolayer(ts2_linear, series = "Linear Component", color = "red") +
  labs(title = "Detrended Hydroelectric Power Consumption", x = "Time", y = "Consumption")

ts2_detrended_plot
```



Q5

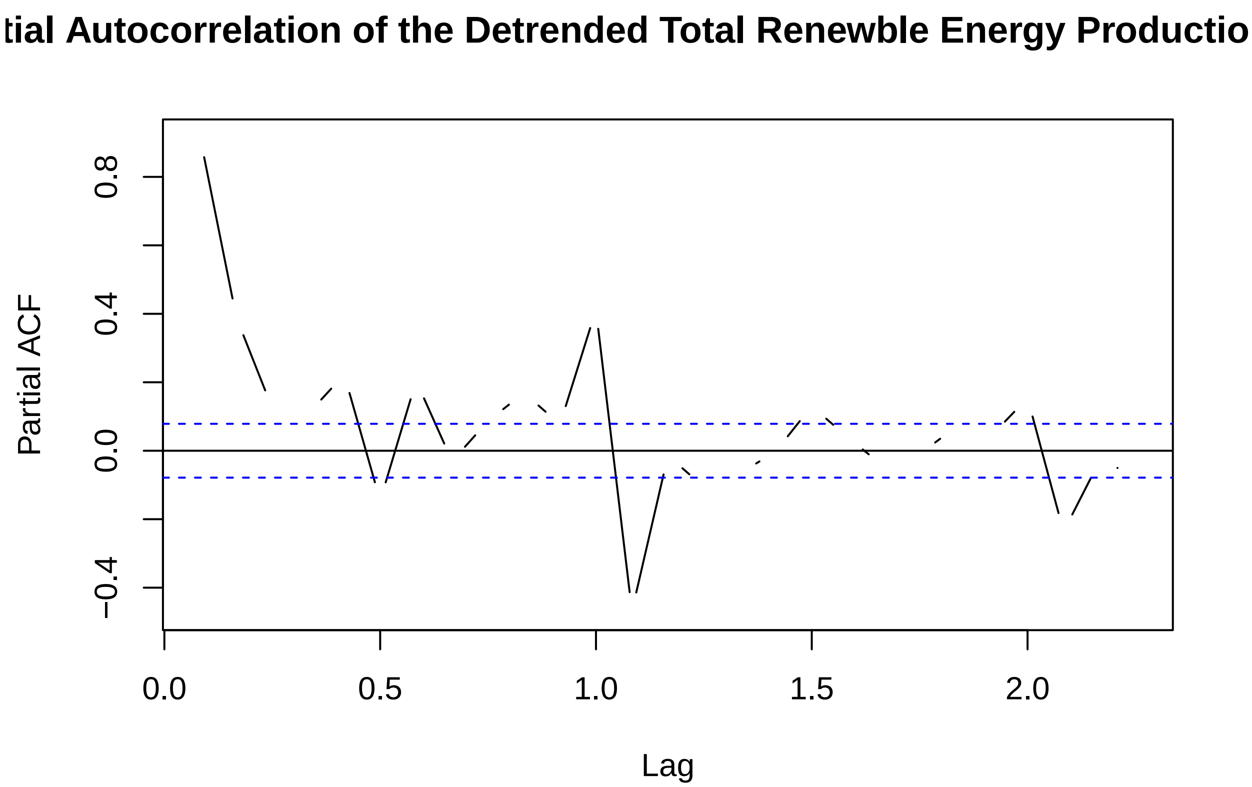
Plot ACF and PACF for the detrended series and compare with the plots from Q1. You may use plot_grid() again to get them side by side, but not mandatory. Did the plots change? How?

```
#Detrended series - Total Renewable Energy Production: acf
acf_ts1_detrend<-acf(ts1_detrend_renewable, main="Autocorrelation of the Detrended Total Renewble Energy
```

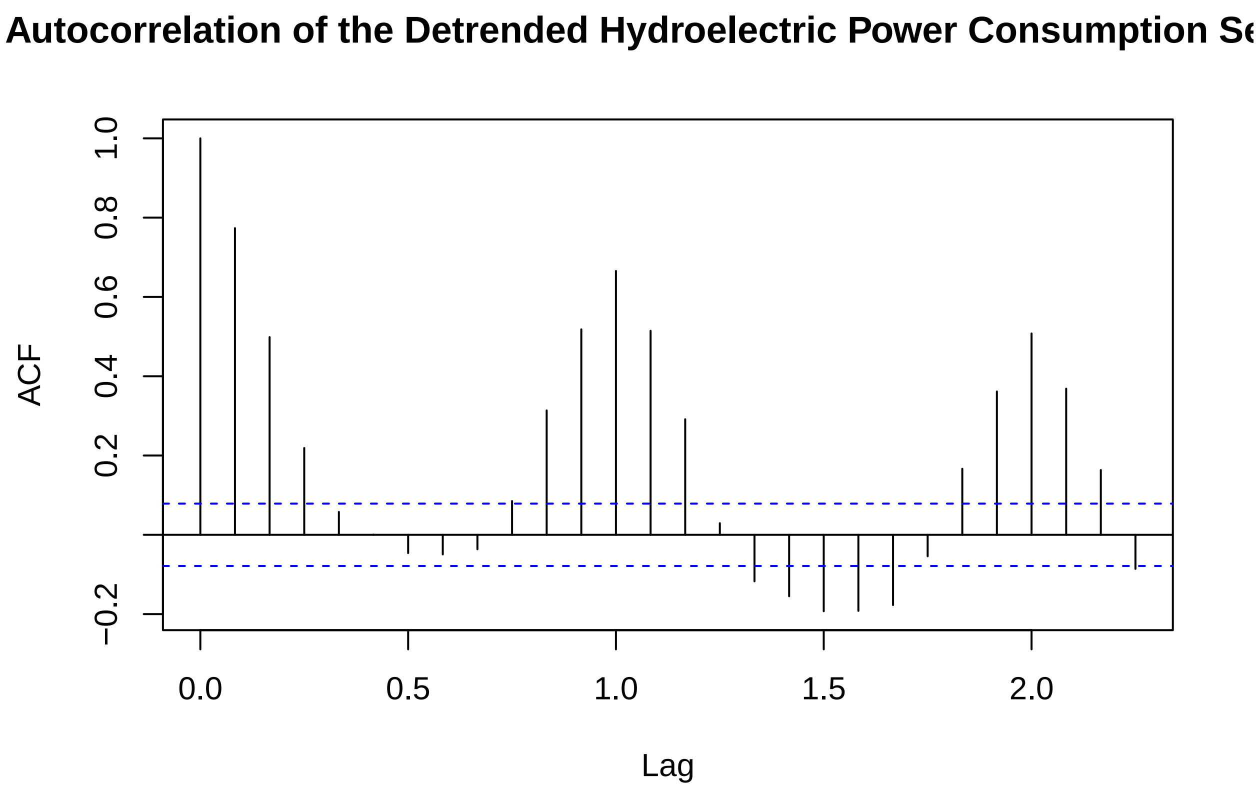



```
acf_ts1_detrend_plot<- ggAcf(ts1_detrend_renewable) + ggtitle("Autocorrelation of the Detrended Total Re

#Detrended series - Total Renewable Energy Production: pacf
pacf_ts1_detrend<-pacf(ts1_detrend_renewable, main="Partial Autocorrelation of the Detrended Total Renew
```

```
## Warning in plot.xy(xy, type, ...): plot type 'correlation' will be truncated to
## first character
```



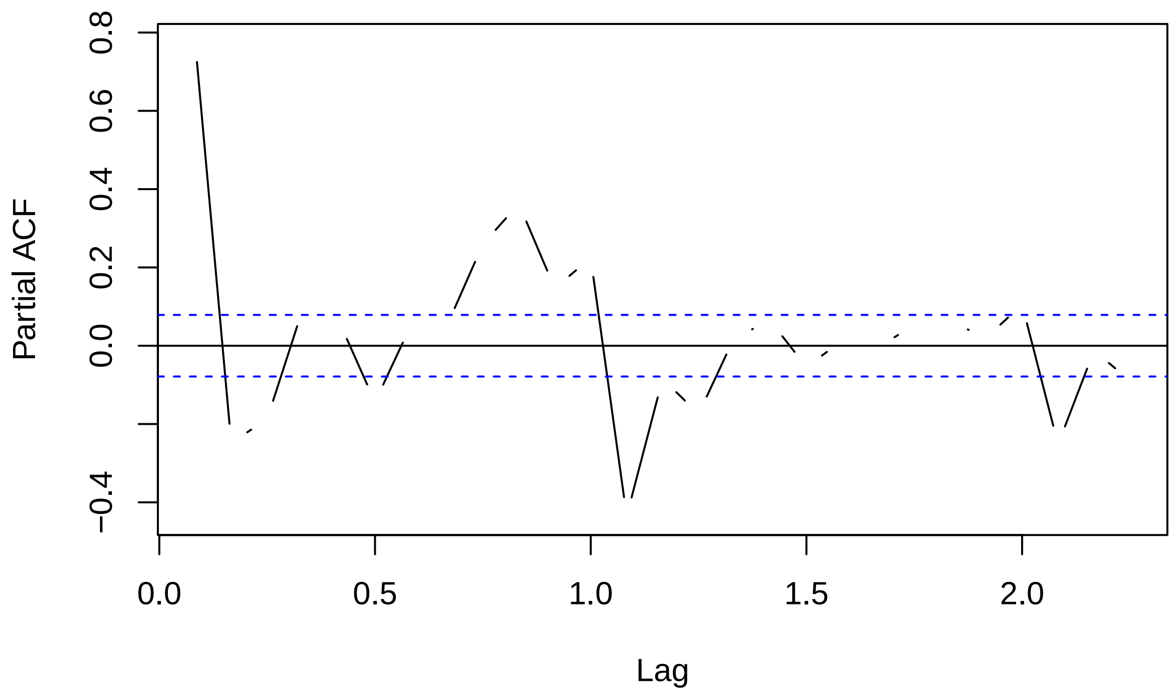

```
pacf_ts1_detrend_plot<- ggPacf(ts1_detrend_renewable) + ggtitle("Partial Autocorrelation of the Detrend

#Detrended Series - Hydroelectric Power Consumption : acf
acf_ts2_detrend<-acf(ts2_detrend_renewable, main="Autocorrelation of the Detrended Hydroelectric Power C
```



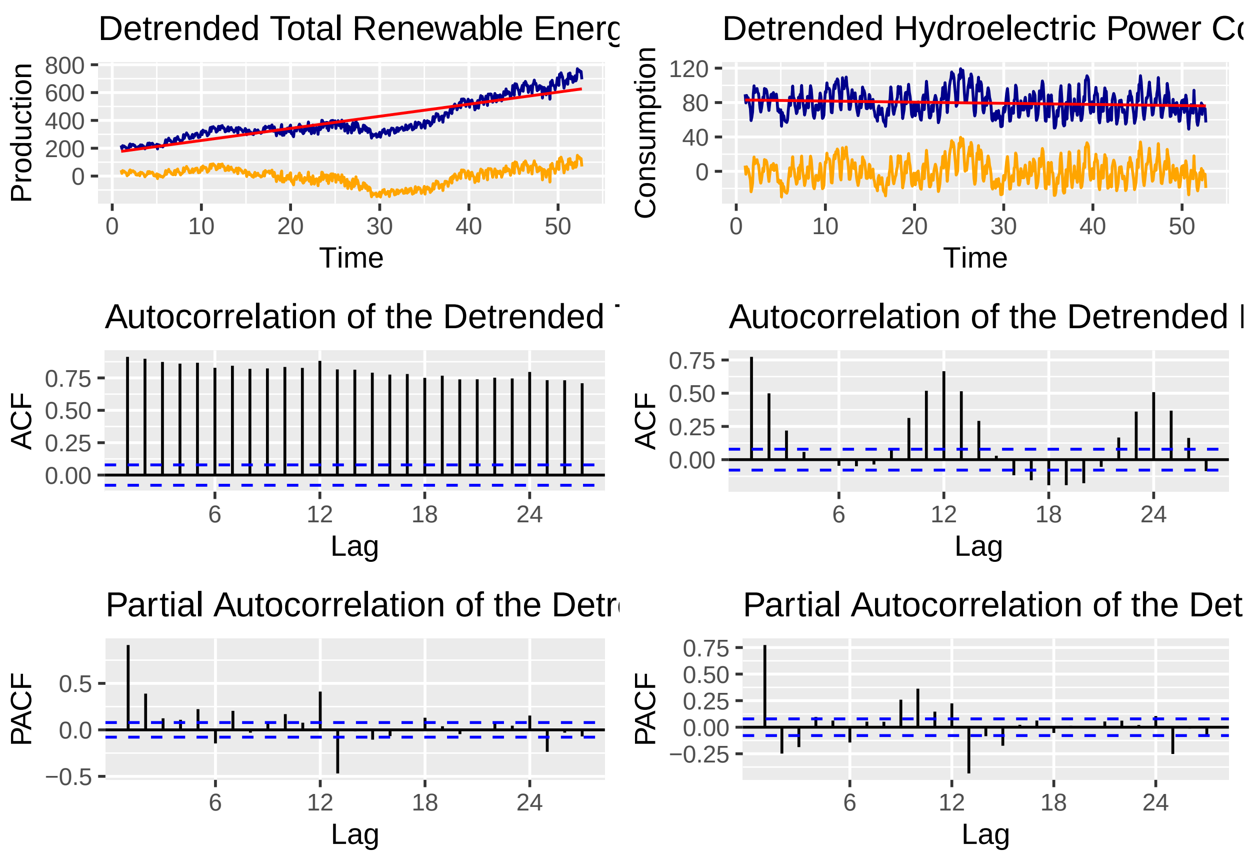

```
acf_ts2_detrend_plot<- ggAcf(ts2_detrend_renewable) + ggtitle("Autocorrelation of the Detrended Hydroele

#Detrended series - Hydroelectric Power Consumption: pacf
pacf_ts2_detrend<-pacf(ts2_detrend_renewable, main="Partial Autocorrelation of the Detrended Hydroelect
```

```
## Warning in plot.xy(xy, type, ...): plot type 'correlation' will be truncated to
## first character
```

tial Autocorrelation of the Detrended Hydroelectric Power Consumptio




```
pacf_ts2_detrend_plot<- ggPacf(ts2_detrend_renewable) + ggtitle("Partial Autocorrelation of the Detrend

plot_grid(ts1_detrended_plot,ts2_detrended_plot,acf_ts1_detrend_plot,acf_ts2_detrend_plot,pacf_ts1_detr
```



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.

#Answer for Q6 Yes. The acf plot and the time series of detrended Hydroelectric Power Consumption seem to have a seasonal trend, while the time series and acf plot of detrended Total Renewable Energy Produciton seem to have downwarding (decreasing) 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?

#Answer: Based on the result, hydroelectric power consumption seems to have a seasonal trend, which is similar to the results in Q6. Based on the summary of the Total Renewable Energy Production's seasonal means model, the p-value is less than 0.05 and R-squared value is 0.0312. Also, all regression's coefficients p-value is >0.05, which means that relationships in the model are statistically not significant.

On the other hand, according to the summary of the Hydroelectric Power Consumption, the overall p-value is less than 0.05 and R-squared value is 0.468, which means that relationships in the model are statistically significant than the former one.

```r
#Fit a seasonal means model (using seasonal dummies)
dummies_1 <- seasonaldummy(ts1_detrend_renewable)
dummies_2 <- seasonaldummy(ts2_detrend_renewable)

ts1_seasonal_means_model <- lm(detrend_renewable_ts1 ~ dummies_1)
ts2_seasonal_means_model <- lm(detrend_renewable_ts2 ~ dummies_2)

#Print the summary of the regression - Total Renewable Energy Production
summary(ts1_seasonal_means_model)
```

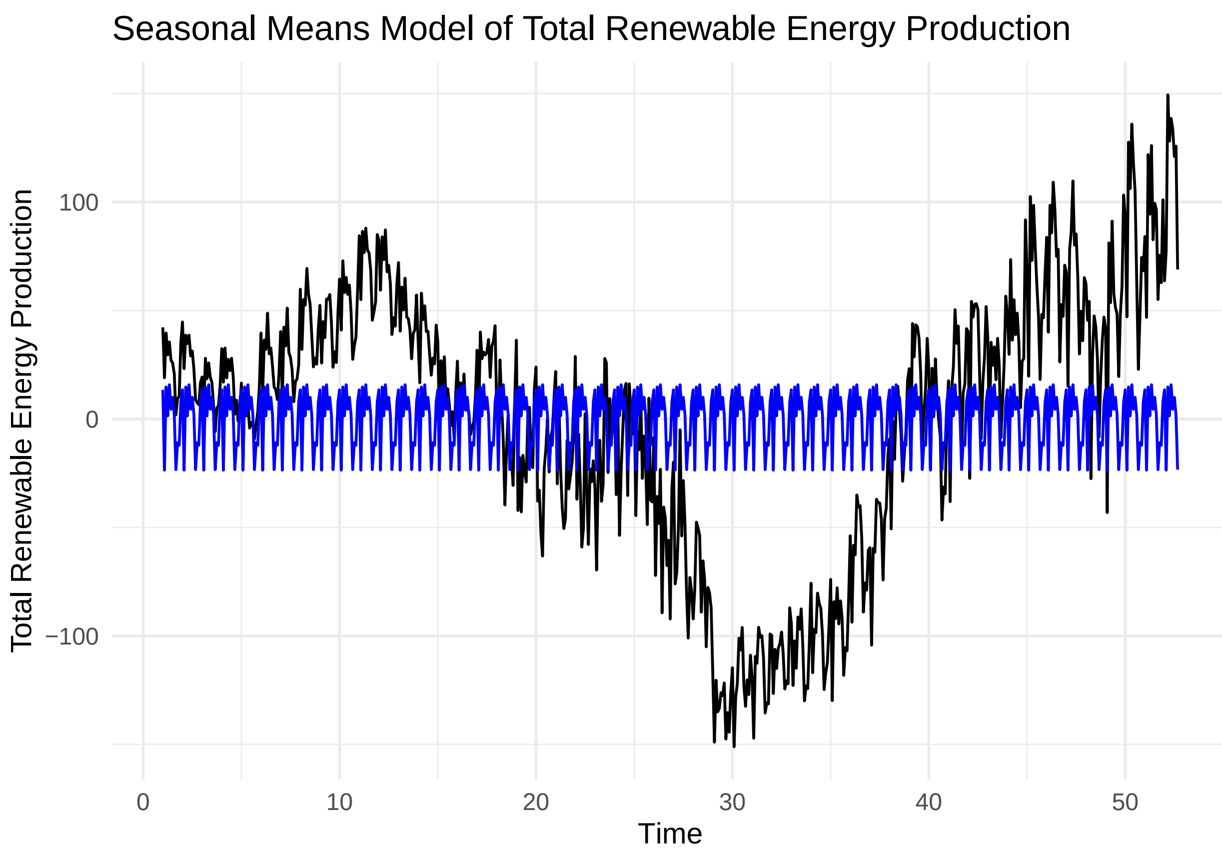
```
##
## Call:
## lm(formula = detrend_renewable_ts1 ~ dummies_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -149.18  -38.16   14.42   41.50  134.67
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)     7.858      8.504   0.924  0.35584
## dummies_1Jan    5.592     11.968   0.467  0.64048
## dummies_1Feb  -31.452     11.968  -2.628  0.00881 **
## dummies_1Mar    6.892     11.968   0.576  0.56491
## dummies_1Apr   -6.449     11.968  -0.539  0.59023
## dummies_1May    7.923     11.968   0.662  0.50822
## dummies_1Jun   -3.394     11.968  -0.284  0.77682
## dummies_1Jul    2.126     11.968   0.178  0.85906
## dummies_1Aug   -5.878     11.968  -0.491  0.62351
## dummies_1Sep  -31.209     11.968  -2.608  0.00934 **
## dummies_1Oct  -18.757     12.026  -1.560  0.11937
## dummies_1Nov  -19.982     12.026  -1.661  0.09713 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.73 on 609 degrees of freedom
## Multiple R-squared:  0.04839,    Adjusted R-squared:  0.0312
## F-statistic: 2.815 on 11 and 609 DF,  p-value: 0.001358
```

```r
#Print the summary of the regression - Hydroelectric Power Consumption
summary(ts2_seasonal_means_model)
```

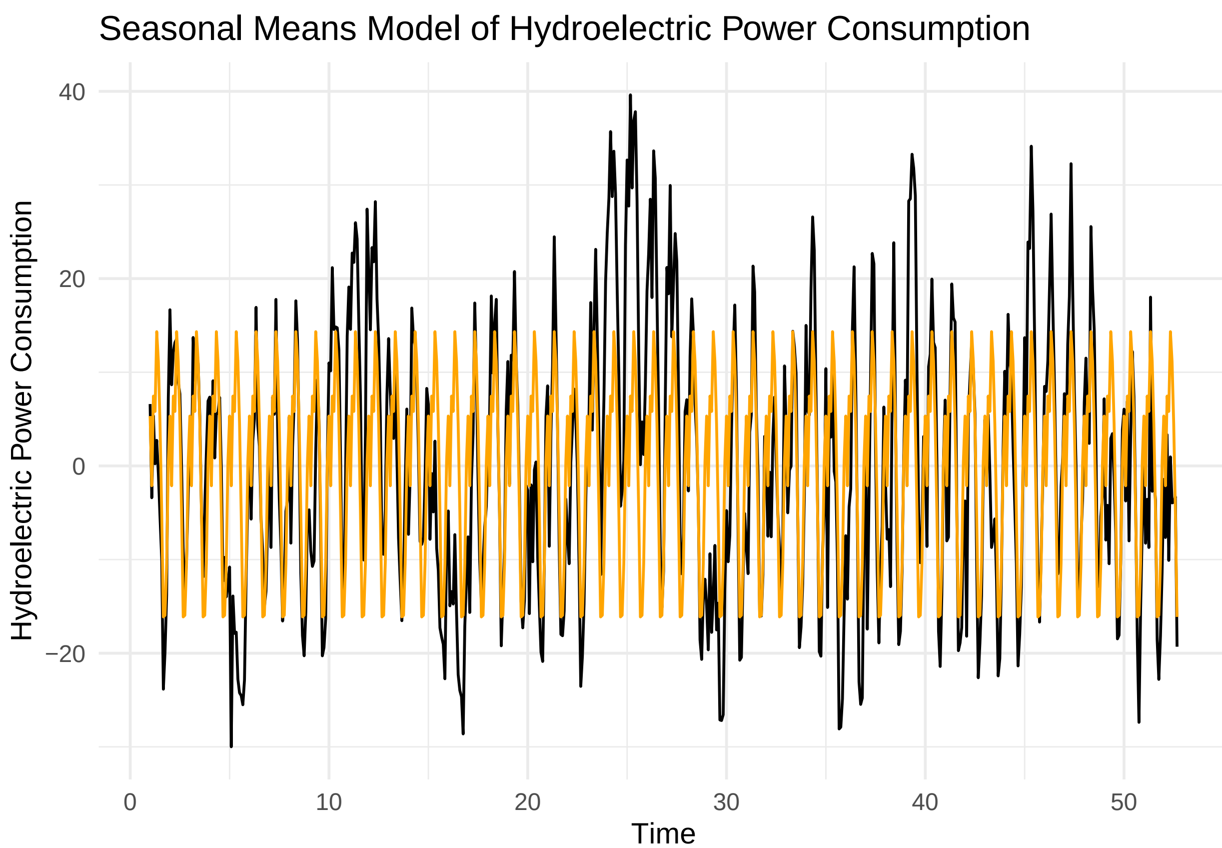
```
##
## Call:
## lm(formula = detrend_renewable_ts2 ~ dummies_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.933  -5.798  -0.531   5.721  32.166
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.4379     1.4258   0.307 0.758849
## dummies_2Jan   4.8863     2.0067   2.435 0.015177 *
## dummies_2Feb  -2.5567     2.0067  -1.274 0.203116
## dummies_2Mar   7.0202     2.0067   3.498 0.000502 ***
## dummies_2Apr   5.3770     2.0067   2.680 0.007572 **
## dummies_2May  13.8957     2.0067   6.925 1.11e-11 ***
## dummies_2Jun  10.7293     2.0067   5.347 1.27e-07 ***
## dummies_2Jul   4.0439     2.0067   2.015 0.044320 *
## dummies_2Aug  -5.3775     2.0067  -2.680 0.007566 **
## dummies_2Sep -16.5635     2.0067  -8.254 9.51e-16 ***
## dummies_2Oct -16.3915     2.0164  -8.129 2.43e-15 ***
## dummies_2Nov -10.8163     2.0164  -5.364 1.16e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.18 on 609 degrees of freedom
## Multiple R-squared:  0.4781, Adjusted R-squared:  0.4687
## F-statistic: 50.72 on 11 and 609 DF,  p-value: < 2.2e-16
```

```
#Plot 1 - Total Renewable Energy Production
plot_ts1_seasonal_means_model <- autoplot(ts1_detrend_renewable) +
  geom_line(aes(x = time(ts1_detrend_renewable), y = fitted(ts1_seasonal_means_model)), color = "blue")
  labs(title = "Seasonal Means Model of Total Renewable Energy Production",
       y = "Total Renewable Energy Production",
       x = "Time") +
  theme_minimal()

plot_ts1_seasonal_means_model
```




```
#Plot 2 - Hydroelectric Power Consumption
plot_ts2_seasonal_means_model <- autoplot(ts2_detrend_renewable) +
  geom_line(aes(x = time(ts2_detrend_renewable), y = fitted(ts2_seasonal_means_model)), color = "orange
  labs(title = "Seasonal Means Model of Hydroelectric Power Consumption",
       y = "Hydroelectric Power Consumption",
       x = "Time") +
  theme_minimal()

plot_ts2_seasonal_means_model
```



Q8

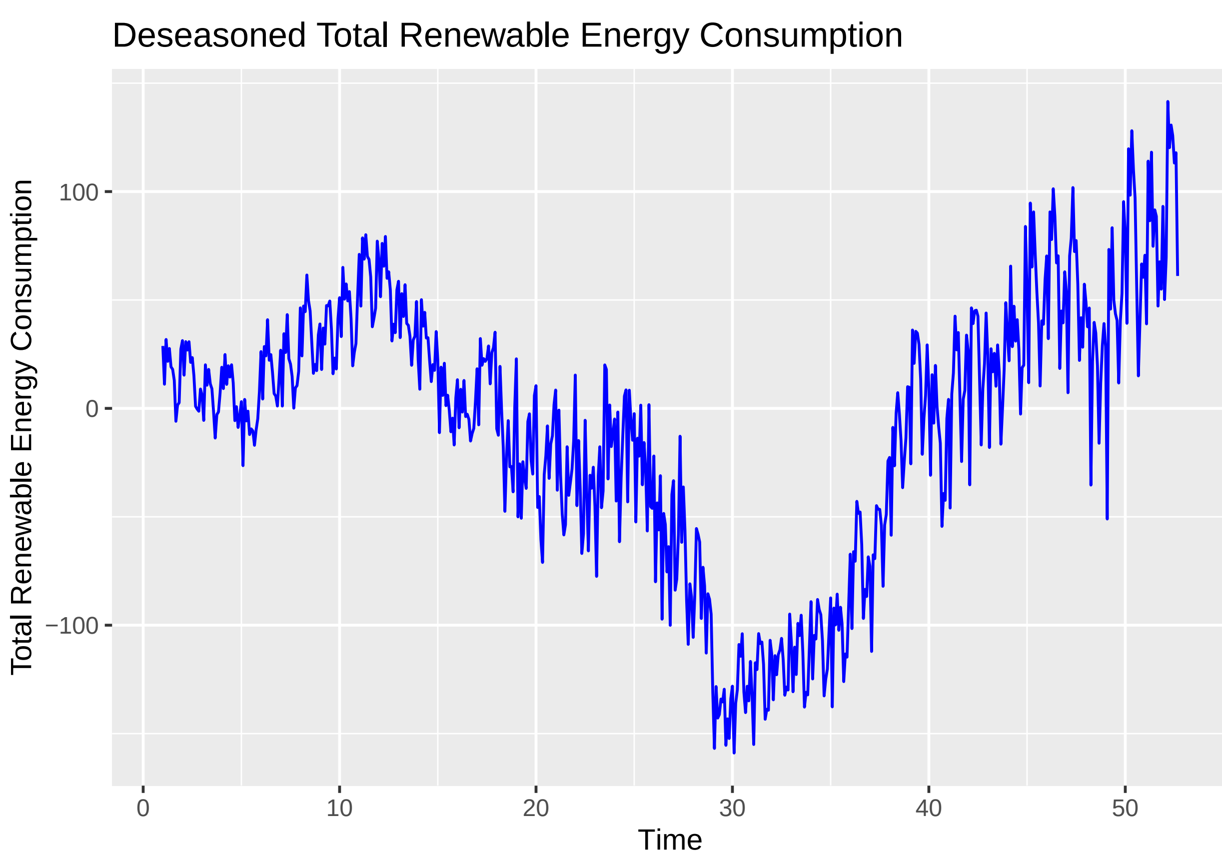
Use the regression coefficients from Q7 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

#Answer: Compared to the plots from Q1, the Hydroelectric Power Consumption plot is pretty similar to the one in Q1. However, the plot for Total Renewable Energy Consumption has a different trend compared to Q1, as it shows the downward trend at first but it is changed to upward (increasing) trend after t=30.

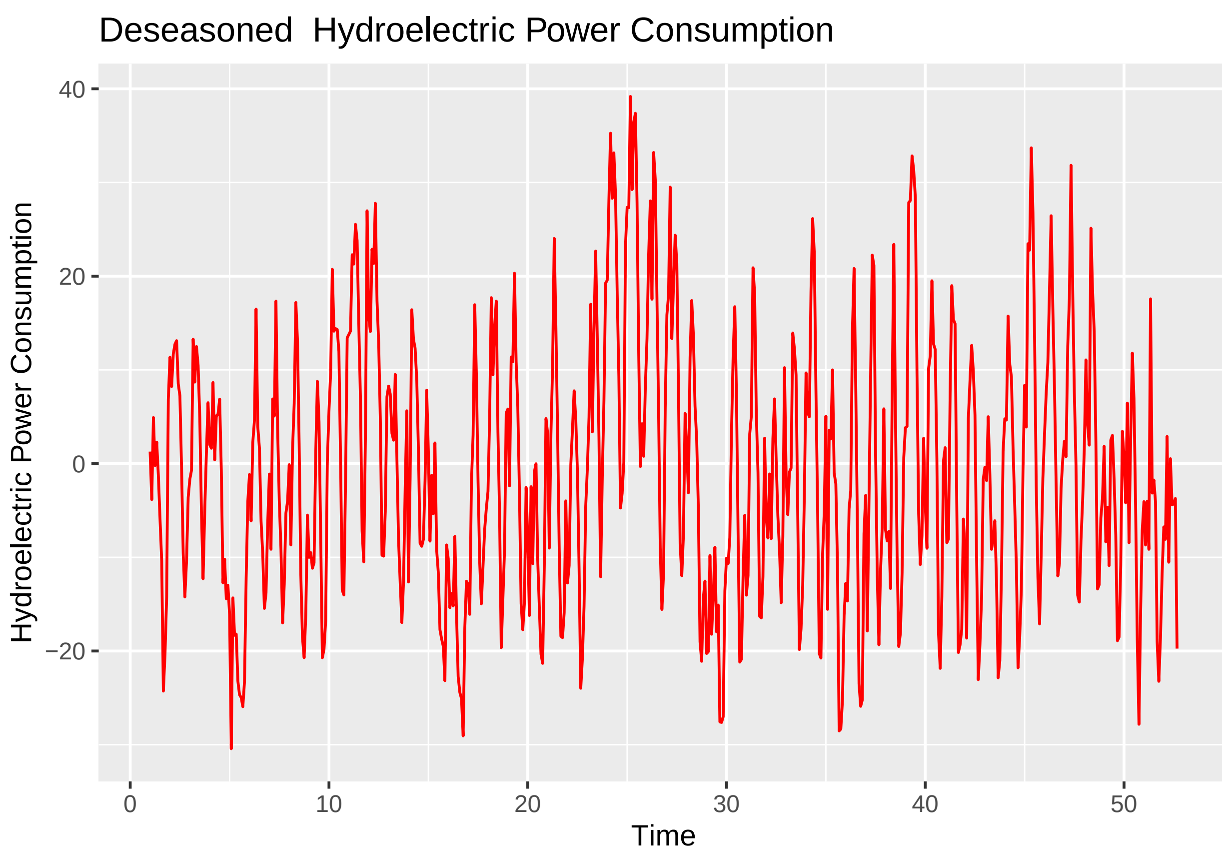
#Checked my code with AI because I was in trouble with knitting the document.

```
nobs_1 <- length(detrend_renewable_ts1)
nobs_2 <- length(detrend_renewable_ts2)

beta_intercept_1 <-ts1_seasonal_means_model$coefficients[1]
beta_intercept_2 <-ts2_seasonal_means_model$coefficients[1]

beta_coeff_1<-ts1_seasonal_means_model$coefficients[2:13]
beta_coeff_2<-ts2_seasonal_means_model$coefficients[2:13]

#Total Renewable Energy consumption
renewable_seasonal_comp_1<-array(0,nobs_1)

for (i in 1:nobs_1) {
  renewable_seasonal_comp_1[i] <- beta_intercept_1 + beta_coeff_1 * dummies_1[i,]
}
```

```r
deseason_renewable_ts_data_1<-detrend_renewable_ts1 - renewable_seasonal_comp_1

ts_deseason_renewable_ts_data_1 <-ts(deseason_renewable_ts_data_1,start = c(1,1), frequency=12)

#Hydroelectric Power Consumption
renewable_seasonal_comp_2<-array(0,nobs_2)

for (i in 1:nobs_2) {
  renewable_seasonal_comp_2[i] <- beta_intercept_2 + beta_coeff_2 * dummies_2[i,]
}

deseason_renewable_ts_data_2<-detrend_renewable_ts2 - renewable_seasonal_comp_2

ts_deseason_renewable_ts_data_2 <-ts(deseason_renewable_ts_data_2,start = c(1,1), frequency=12)

#Plot - Renewable Energy Production
plot_deseason_seasonal_model_renewable_1 <- autoplot(ts_deseason_renewable_ts_data_1, color = "blue") +
  labs(title = "Deseasoned Total Renewable Energy Consumption",
       y = "Total Renewable Energy Consumption",
       x = "Time")

plot_deseason_seasonal_model_renewable_1
```



```r
#Plot 2 - Hydroelectric Power Consumption
plot_deseason_seasonal_model_renewable_2 <- autoplot(ts_deseason_renewable_ts_data_2, color = "red") +
  labs(title = "Deseasoned  Hydroelectric Power Consumption",
       y = "Hydroelectric Power Consumption",
       x = "Time")

plot_deseason_seasonal_model_renewable_2
```

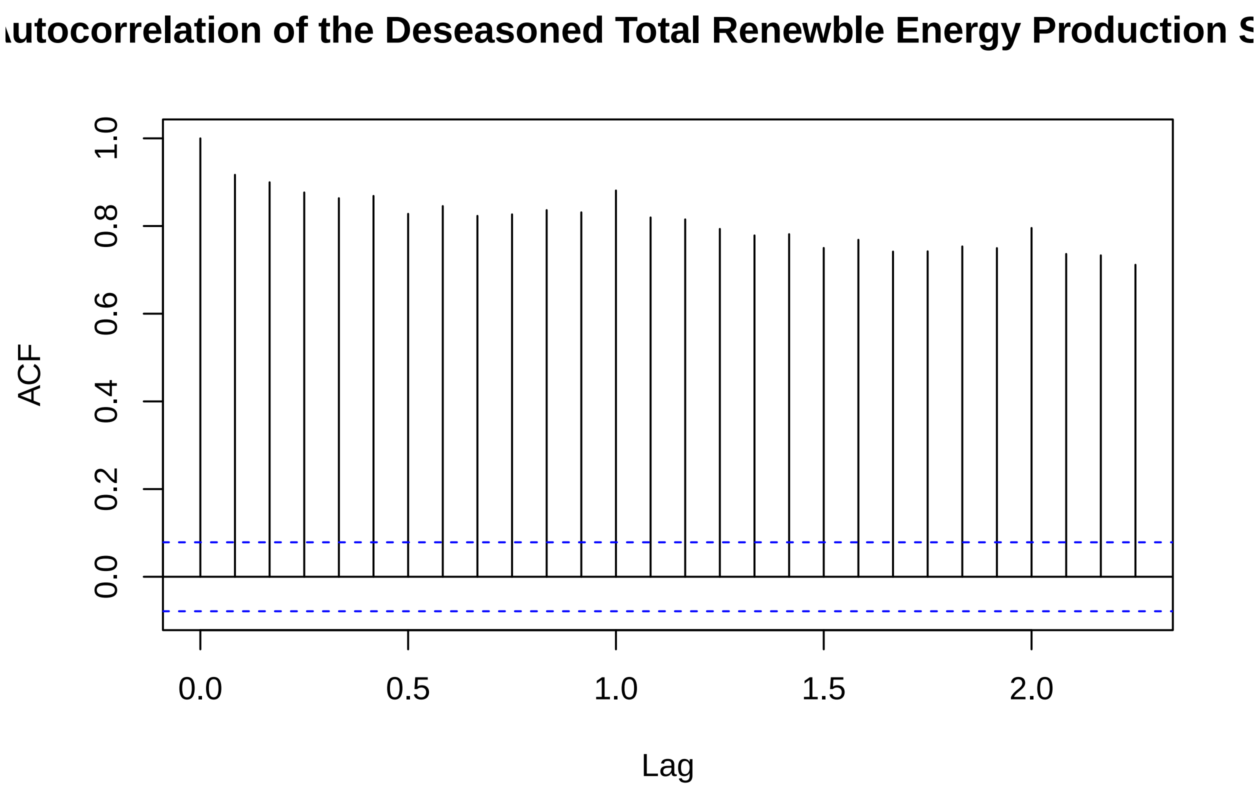


Q9

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?

#Answer: Although there is no big difference in the Hydroelectric Power Consumption plots, the Total Renewable Energy one has slight changes in acf and pacf. Though the acf plot in Q1 shows the contant decrease, the acf of deceasoned plot has the downward trend in overall with some exceptions. Also, unlike the time series of Q1 has contant downward trend, deasoned Total Renewable Energy Production has down and upward trend.

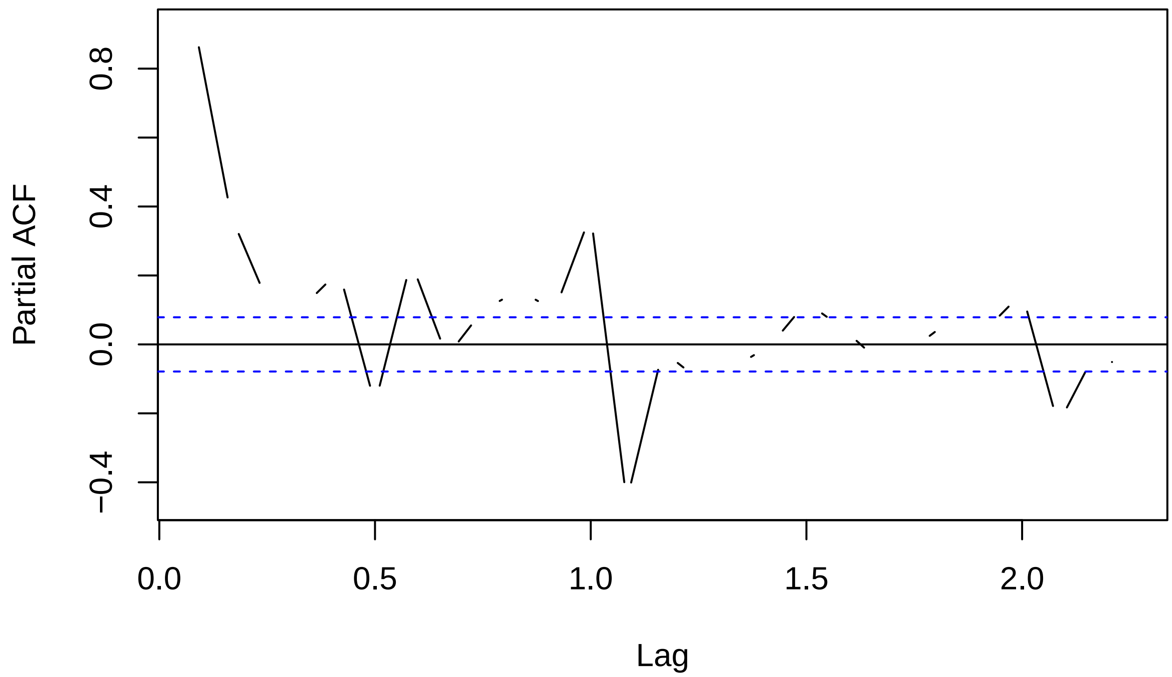
```r
#Deseason series - Total Renewable Energy Production: acf
acf_ts1_deseason<-acf(ts_deseason_renewable_ts_data_1,
                      main="Autocorrelation of the Deseasoned Total Renewble Energy Production Series",
                      type="correlation",
                      plot=TRUE)
```



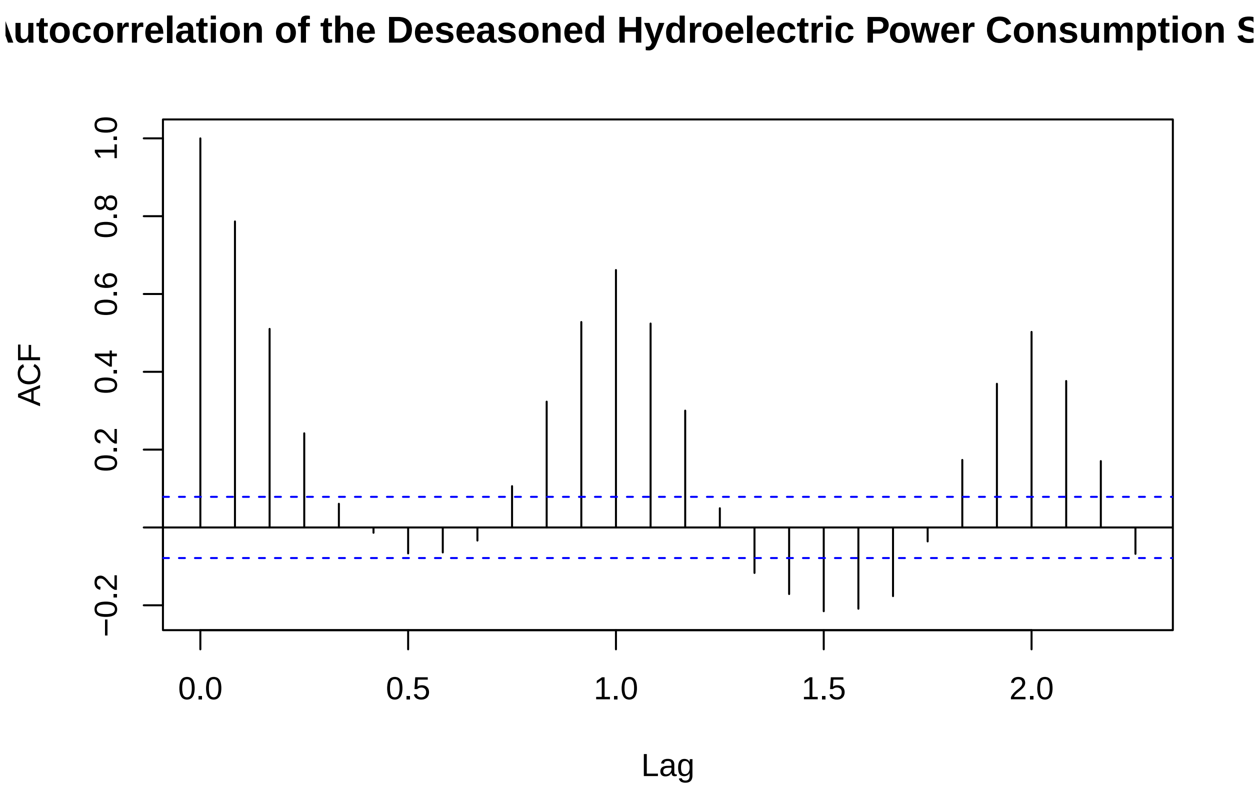

```
acf_ts1_deseason_plot<- ggAcf(ts_deseason_renewable_ts_data_1) +
  ggtitle("Autocorrelation of the Deseasoned Total Renewable Energy Production")

#Detrended series - Total Renewable Energy Production: pacf
pacf_ts1_deseason<-pacf(ts_deseason_renewable_ts_data_1,
                        main="Partial Autocorrelation of the Deseasoned Total Renewble Energy Productio
                        type="correlation",
                        plot=TRUE)
```

```
## Warning in plot.xy(xy, type, ...): plot type 'correlation' will be truncated to
## first character
```

al Autocorrelation of the Deseasoned Total Renewble Energy Productic

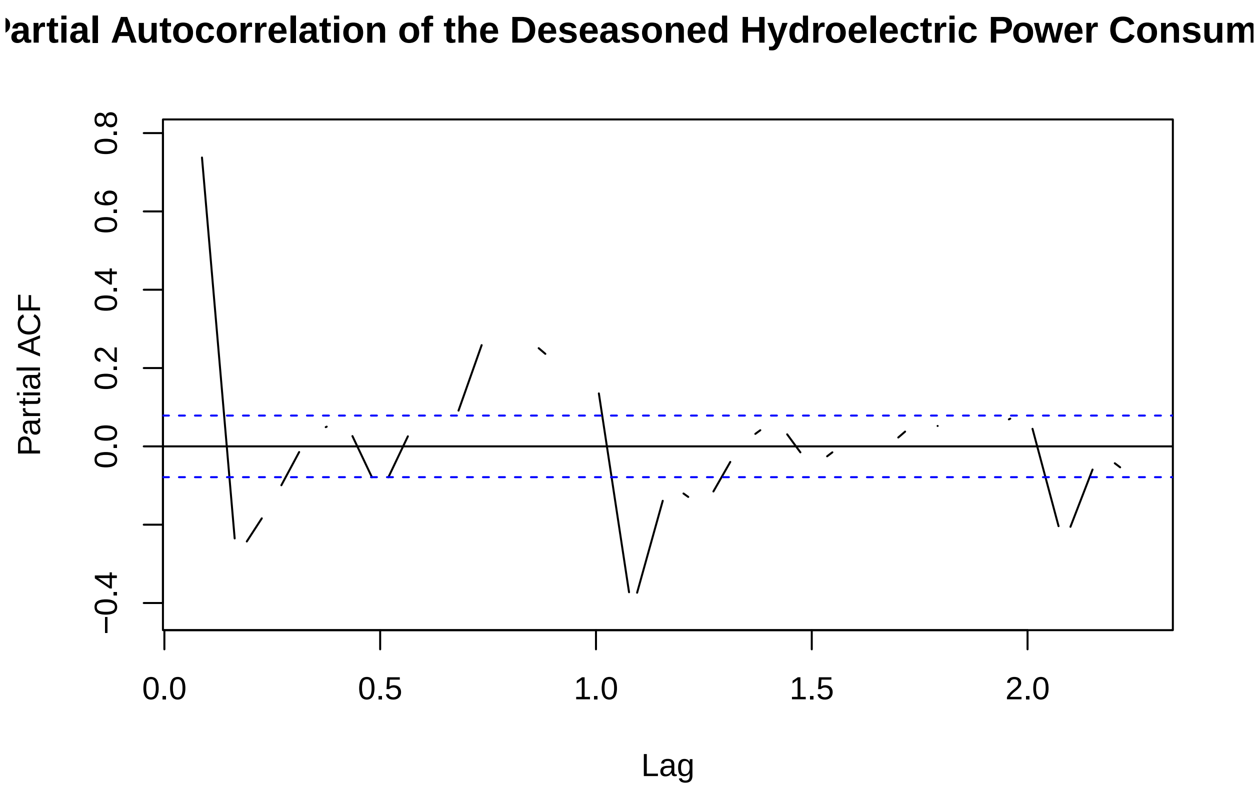



```
pacf_ts1_deseason_plot<- ggPacf(ts_deseason_renewable_ts_data_1) +
  ggtitle("Partial Autocorrelation of the Deseasoned Total Renewable Energy Production")

#Detrended Series - Hydroelectric Power Consumption : acf
acf_ts2_deseason<-acf(ts_deseason_renewable_ts_data_2,
                      main="Autocorrelation of the Deseasoned Hydroelectric Power Consumption Series",
                      type="correlation",plot=TRUE)
```

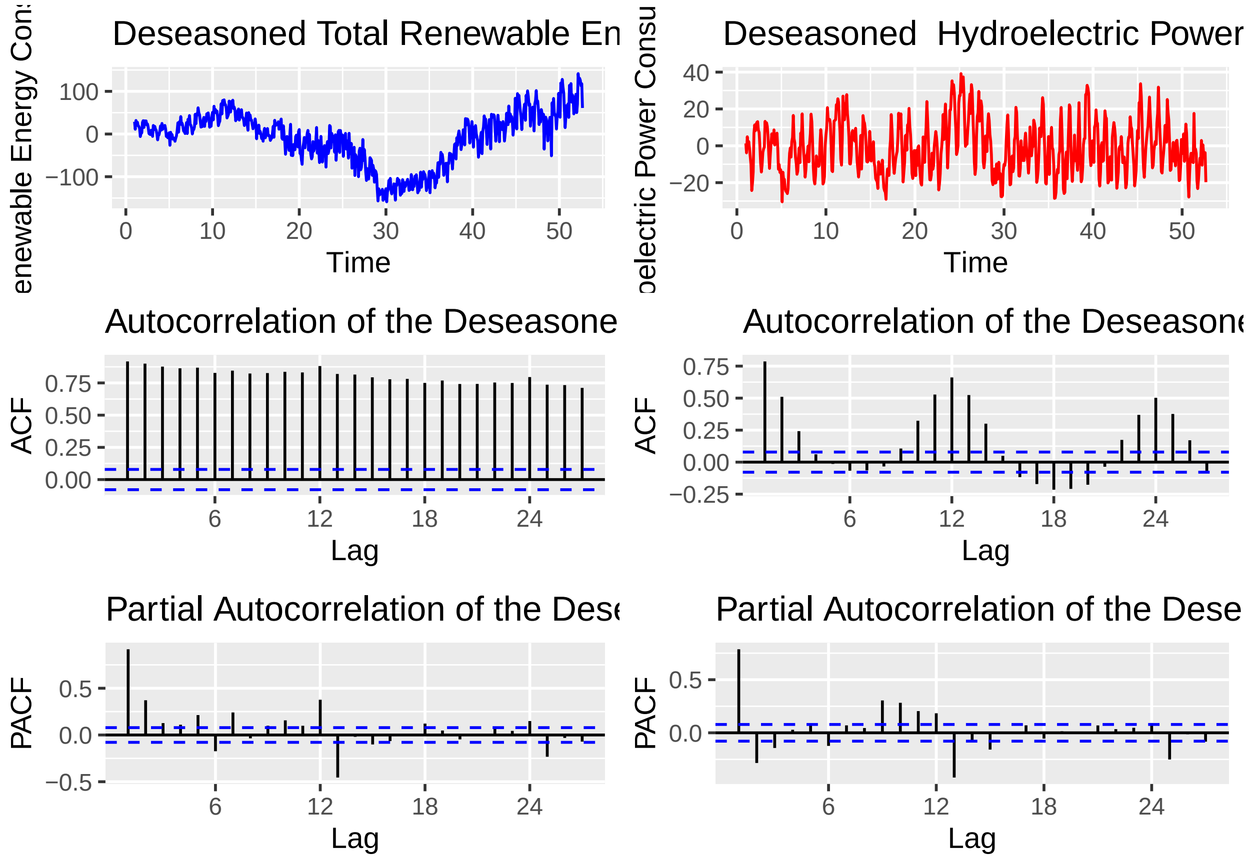



```r
acf_ts2_deseason_plot<- ggAcf(ts_deseason_renewable_ts_data_2) +
  ggtitle("Autocorrelation of the Deseasoned Hydroelectric Power Consumption")

#Detrended series - Hydroelectric Power Consumption: pacf
pacf_ts2_deseason<-pacf(ts_deseason_renewable_ts_data_2,
                         main="Partial Autocorrelation of the Deseasoned Hydroelectric Power Consumption
                       type="correlation",plot=TRUE)
```

```
## Warning in plot.xy(xy, type, ...): plot type 'correlation' will be truncated to
## first character
```




```
pacf_ts2_deseason_plot<- ggPacf(ts_deseason_renewable_ts_data_2) +
  ggtitle("Partial Autocorrelation of the Deseasoned Hydroelectric Power Consumption")

plot_grid(plot_deseason_seasonal_model_renewable_1,plot_deseason_seasonal_model_renewable_2, acf_ts1_des
```


Deseasoned Total Renewable En
enewable Energy Con:
100
0
−100
0
10
20
30
40
50
Time
Deseasoned Hydroelectric Power
oelectric Power Consu
40
20
0
−20
0
10
20
30
40
50
Time
Autocorrelation of the Deseasone
ACF
0.75
0.50
0.25
0.00
6
12
18
24
Lag
Autocorrelation of the Deseasone
ACF
0.75
0.50
0.25
0.00
−0.25
6
12
18
24
Lag
Partial Autocorrelation of the Dese
PACF
0.5
0.0
−0.5
6
12
18
24
Lag
Partial Autocorrelation of the Dese
PACF
0.5
0.0
6
12
18
24
Lag