ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2025

Assignment 4 - Due date 02/11/25

Rosie Wu

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github. And to do so you will need to fork our repository and link it to your RStudio.

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_A04_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).

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

R packages needed for this assignment: "xlsx" or "readxl", "ggplot2", "forecast","tseries", and "Kendall". Install these packages, if you haven't done yet. Do not forget to load them before running your script, since they are NOT default packages.\

```r
#Load/install required package here
library(lubridate)
library(ggplot2)
library(forecast)
library(Kendall)
library(tseries)
library(dplyr)
library(here)
library(readxl)
library(openxlsx)
library(cowplot)
library(trend)
```

Questions

Consider the same data you used for A3 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 January 2021 Monthly Energy Review. **For this assignment you will work only with the column "Total Renewable Energy Production"**.

```
#Importing data set - you may copy your code from A3
here()
```

```
## [1] "/home/guest/TSA_Sp25"
```

```
#Importing data set without change the original file using read.xlsx
energy_data1 <- read_excel(path= "./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sour
                           skip = 12, sheet="Monthly Data",col_names=FALSE)

#Now let's extract the column names from row 11
read_col_names <- read_excel(
  path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
  skip = 10,n_max = 1, sheet="Monthly Data",col_names=FALSE)
#Assign the column names to the data set
colnames(energy_data1) <- read_col_names

#Visualize the first rows of the data set
head(energy_data1)
```

```
## # A tibble: 6 x 14
##   Month               `Wood Energy Production` `Biofuels Production`
##   <dttm>                                 <dbl> <chr>
## 1 1973-01-01 00:00:00                     130. Not Available
## 2 1973-02-01 00:00:00                     117. Not Available
## 3 1973-03-01 00:00:00                     130. Not Available
## 4 1973-04-01 00:00:00                     125. Not Available
## 5 1973-05-01 00:00:00                     130. Not Available
## 6 1973-06-01 00:00:00                     125. Not Available
## # i 11 more variables: `Total Biomass Energy Production` <dbl>,
## #   `Total Renewable Energy Production` <dbl>,
## #   `Hydroelectric Power Consumption` <dbl>,
## #   `Geothermal Energy Consumption` <dbl>, `Solar Energy Consumption` <chr>,
## #   `Wind Energy Consumption` <chr>, `Wood Energy Consumption` <dbl>,
## #   `Waste Energy Consumption` <dbl>, `Biofuels Consumption` <chr>,
## #   `Total Biomass Energy Consumption` <dbl>, ...
```

```
# select columns
energy_df4_selected <- energy_data1 %>%
  select("Month",
         "Total Renewable Energy Production")
energy_df4_selected$Month <- as.Date(energy_df4_selected$Month)
head(energy_df4_selected)
```

```
## # A tibble: 6 x 2
##   Month      `Total Renewable Energy Production`
##   <date>                                   <dbl>
## 1 1973-01-01                                220.
## 2 1973-02-01                                197.
## 3 1973-03-01                                219.
## 4 1973-04-01                                209.
## 5 1973-05-01                                216.
## 6 1973-06-01                                208.
```

Stochastic Trend and Stationarity Tests

For this part you will work only with the column Total Renewable Energy Production.

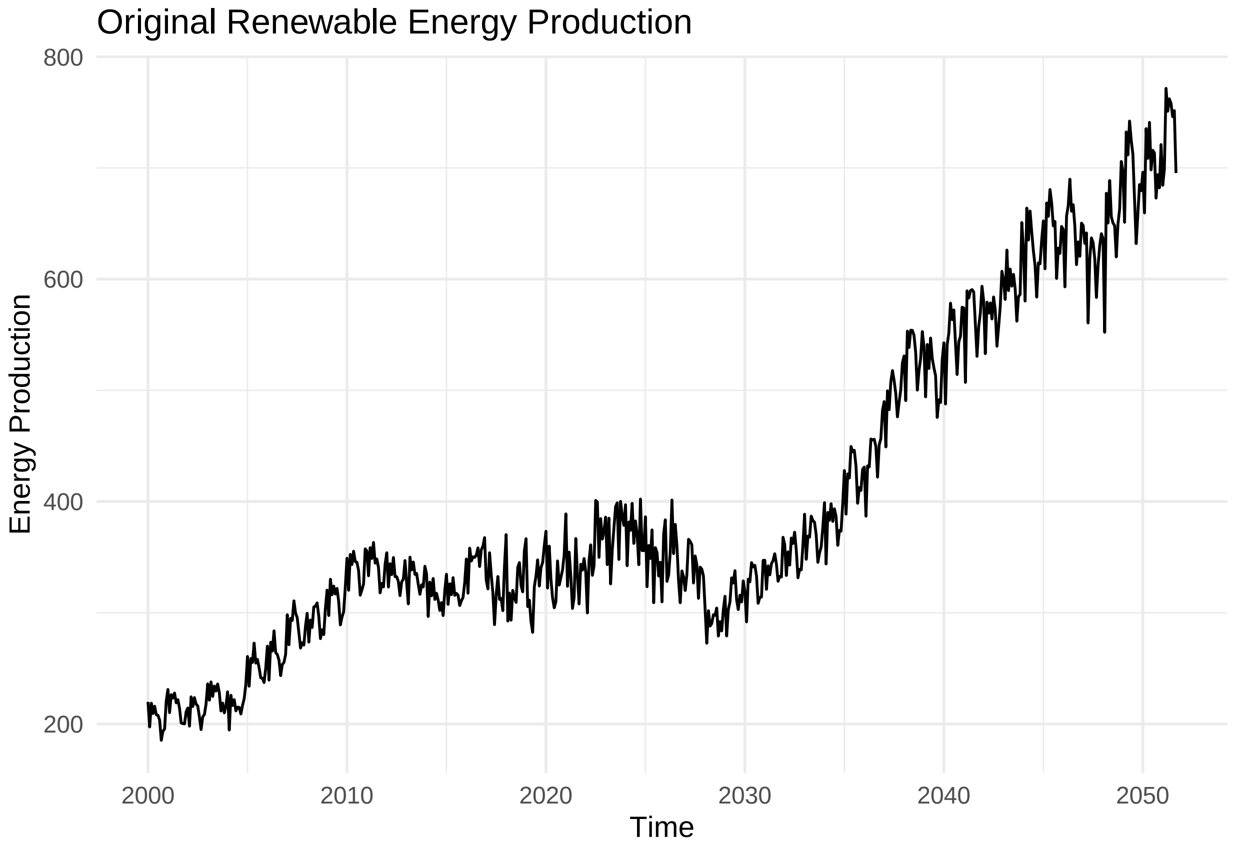
Q1

Difference the "Total Renewable Energy Production" series using function diff(). Function diff() is from package base and take three main arguments: * x vector containing values to be differenced; * *lag* integer indicating with lag to use; * *differences* integer indicating how many times series should be differenced.

Try differencing at lag 1 only once, i.e., make `lag=1` and `differences=1`. Plot the differenced series. Do the series still seem to have trend?

```r
# print the original plot first
renewable_ts <- ts(energy_df4_selected[,2],start=c(2000,1),frequency=12)
# Plot the original renewable energy time series
original_plot <- autoplot(renewable_ts) +
  ggtitle("Original Renewable Energy Production") +
  xlab("Time") + ylab("Energy Production") +
  theme_minimal() +
  scale_color_manual(values = c("Original" = "black"))
print(original_plot)
```

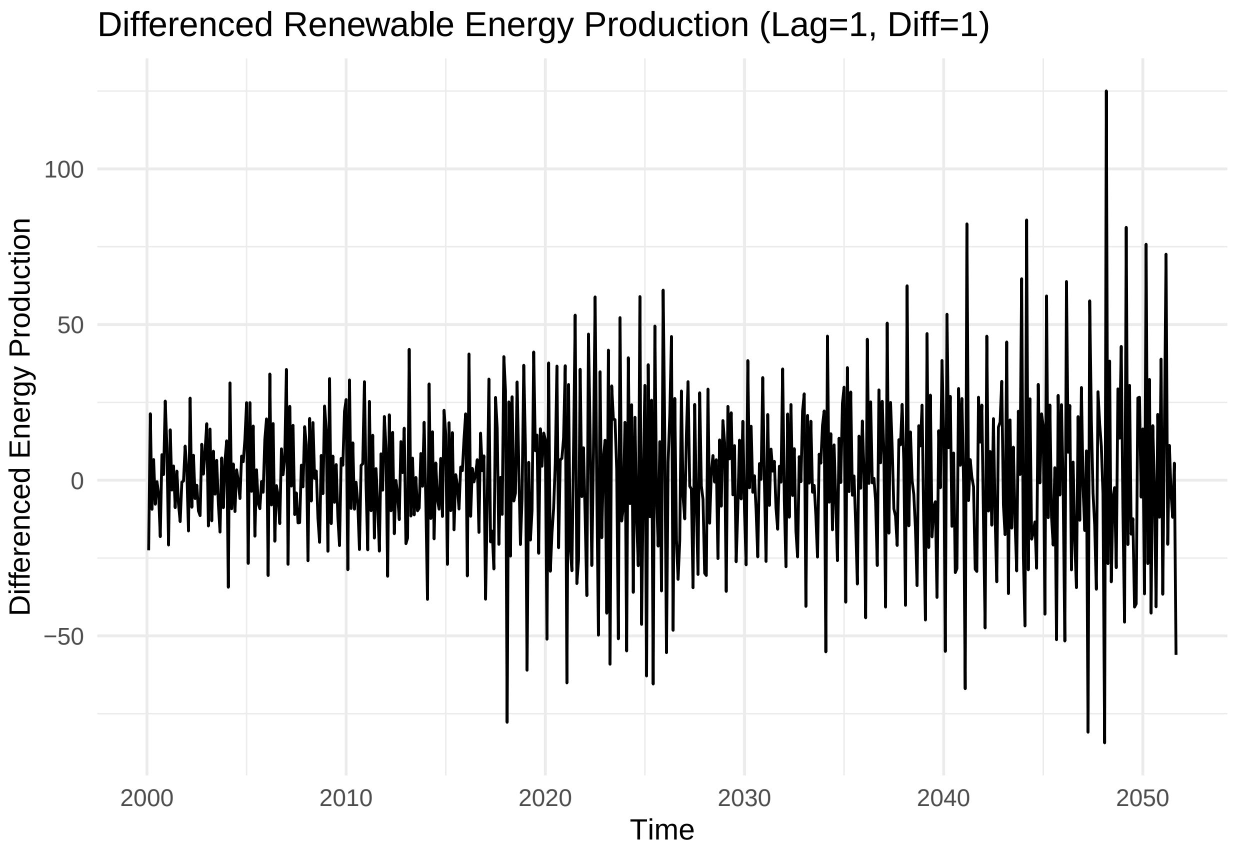
```
## Warning: No shared levels found between ‘names(values)‘ of the manual scale and the
## data’s colour values.
```



```
# Perform first-order differencing with lag = 1
diff_renewable_ts <- diff(renewable_ts, lag = 1, differences = 1)

# Plot the differenced renewable energy time series
diff_plot <- autoplot(diff_renewable_ts) +
  ggtitle("Differenced Renewable Energy Production (Lag=1, Diff=1)") +
  xlab("Time") + ylab("Differenced Energy Production") +
  theme_minimal()

print(diff_plot)
```



Answer: The original plot of the Total Renewables Production shows a significant slope-up trend. However, after differencing the series by lag = 1, difference = 1, the trend from the plot becomes less consistent/ visible. It becomes difficult to define a trend for the differenced time series for the renwables production.

Q2

Copy and paste part of your code for A3 where you run the regression for Total Renewable Energy Production and subtract that from the original series. This should be the code for Q3 and Q4. make sure you use the same name for you time series object that you had in A3, otherwise the code will not work.

```
# Create time index
time_index <- as.numeric(time(renewable_ts))
```

```r
# Fit regression model
renewable_lm <- lm(as.numeric(renewable_ts) ~ time_index)

print(summary(renewable_lm))
```

```
##
## Call:
## lm(formula = as.numeric(renewable_ts) ~ time_index)
##
## 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) -1.720e+04  3.360e+02  -51.17   <2e-16 ***
## time_index   8.687e+00  1.659e-01   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
```

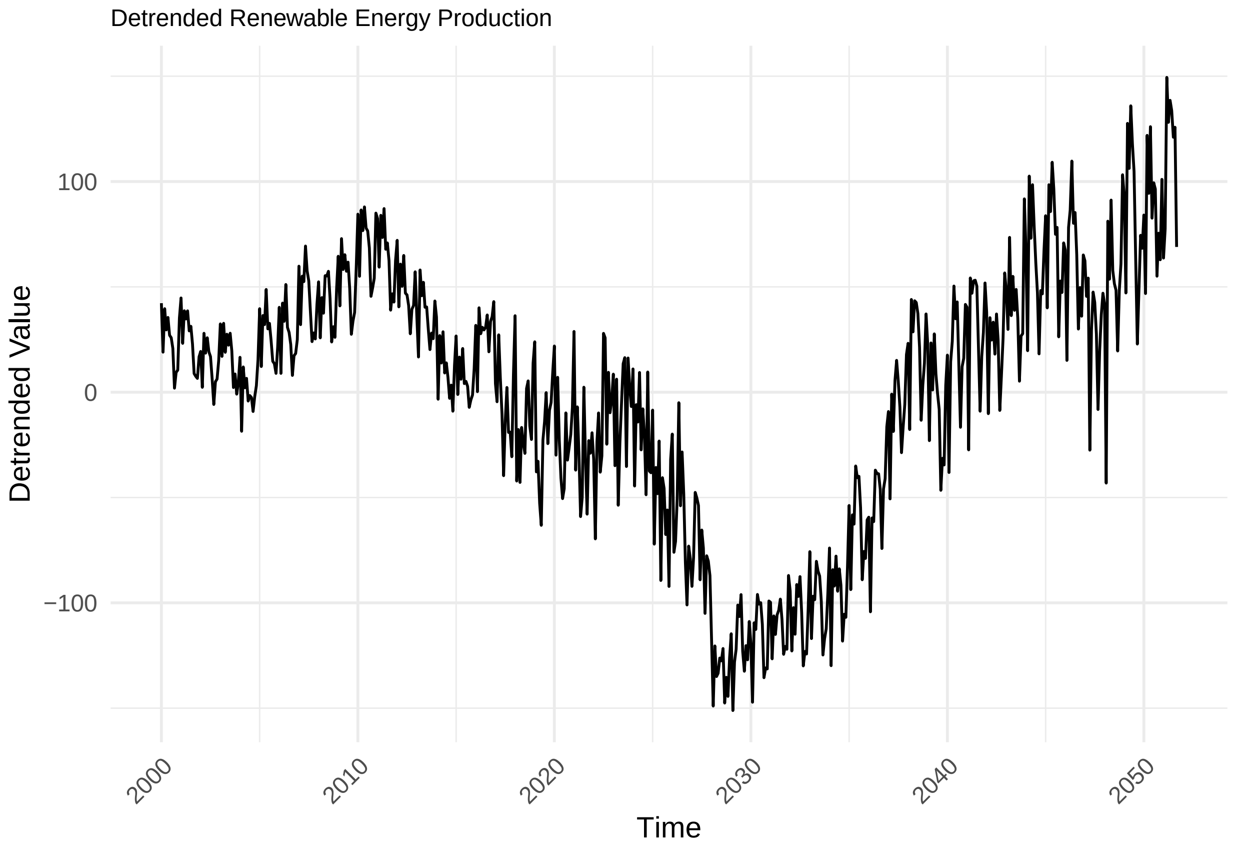
```r
# Save regression coefficients
renewable_coef <- coef(renewable_lm)
print(renewable_coef)
```

```
##   (Intercept)    time_index
## -17196.840061      8.687218
```

```r
# Compute fitted trend values
renewable_trend <- renewable_coef[1] + renewable_coef[2] * time_index

# Detrend the series (orignal ts subtract trend)
renewable_detrended <- renewable_ts - renewable_trend
# Convert to data frames for plotting
#renewable_detrended_df <- data.frame(Date = time(renewable_ts), Value = as.numeric(renewable_detrended

# plot detrended
renewable_detrended_plot <- autoplot(renewable_detrended) +
  ggtitle("Detrended Renewable Energy Production") +
  xlab("Time") + ylab("Detrended Value") +
  theme_minimal() +
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))
print(renewable_detrended_plot)
```



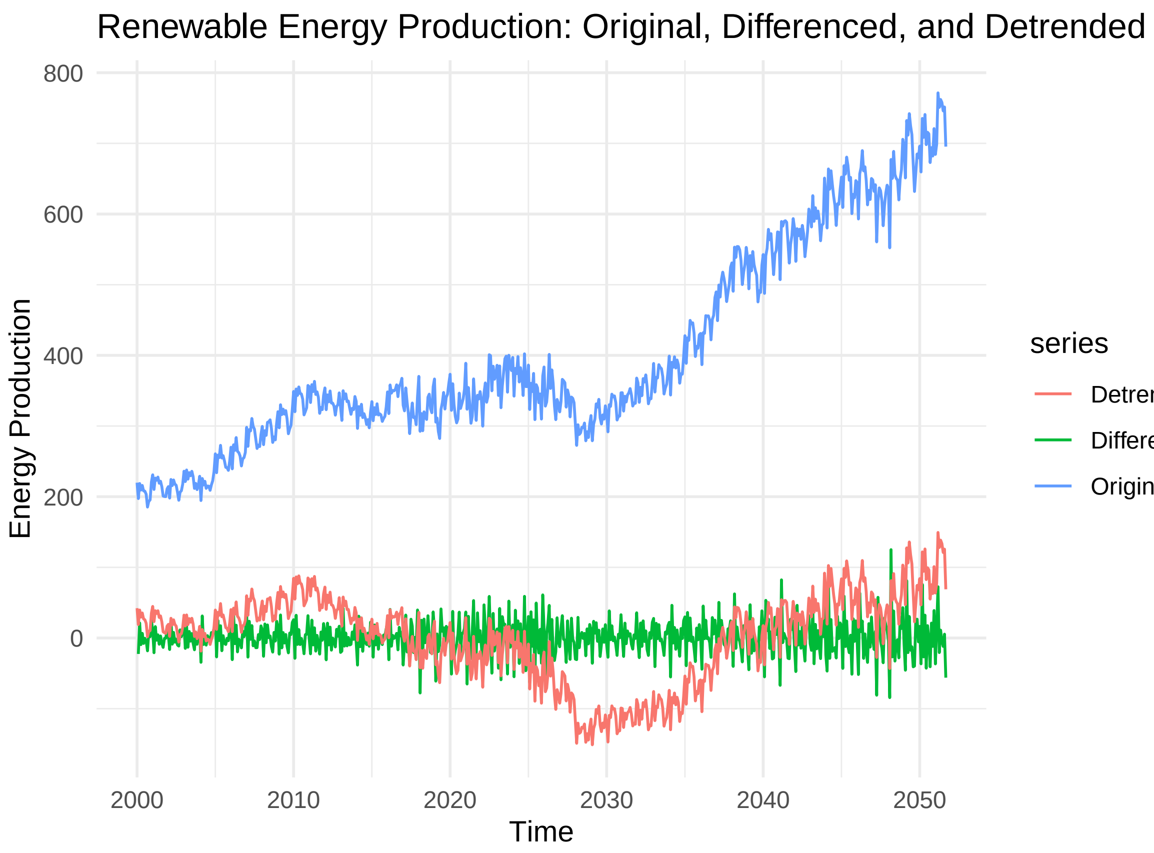
Q3

Now let's compare the differenced series with the detrended series you calculated on A3. In other words, for the "Total Renewable Energy Production" compare the differenced series from Q1 with the series you detrended in Q2 using linear regression.

Using autoplot() + autolayer() create a plot that shows the three series together. Make sure your plot has a legend. The easiest way to do it is by adding the `series=` argument to each autoplot and autolayer function. Look at the key for A03 for an example on how to use autoplot() and autolayer().

What can you tell from this plot? Which method seems to have been more efficient in removing the trend?

```r
# differenced vs detrended vs original
# Create the plot with all three series
plot_q3 <- autoplot(renewable_ts, series = "Original")+
  autolayer(diff_renewable_ts, series = "Differenced")+
  autolayer(renewable_detrended, series = "Detrended")+
  ggtitle("Renewable Energy Production: Original, Differenced, and Detrended") +
  xlab("Time") +
  ylab("Energy Production")+
  theme_minimal()
print(plot_q3)
```



Answer: Comparing three plots, the differenced plot is most effective in removing the trend. The detrended plot series still shows a visible trend sloping down and up again over the timespan, whereas the differenced trend is relatively flat overtime. A flat trend in a differenced time series indicates that the original time series has no significant trend over time, meaning the data points are relatively stable and fluctuate around a constant level with no consistent increase or decrease.It also suggests the original series is likely stationary.

Q4

Plot the ACF for the three series and compare the plots. Add the argument `ylim=c(-0.5,1)` to the autoplot() or Acf() function - whichever you are using to generate the plots - to make sure all three y axis have the same limits. Looking at the ACF which method do you think was more efficient in eliminating the trend? The linear regression or differencing?

```
# Plot ACF for the 3 series
acf_original <- Acf(renewable_ts, plot = FALSE)
acf_diff <- Acf(diff_renewable_ts, plot = FALSE)
acf_detrended <- Acf(renewable_detrended, plot = FALSE)

# Create ACF plots with the same y-axis limits
plot_acf_original <- autoplot(acf_original, ylim = c(-0.5, 1), main = "ACF of Original Series")
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: ‘ylim‘ and ‘main‘
```

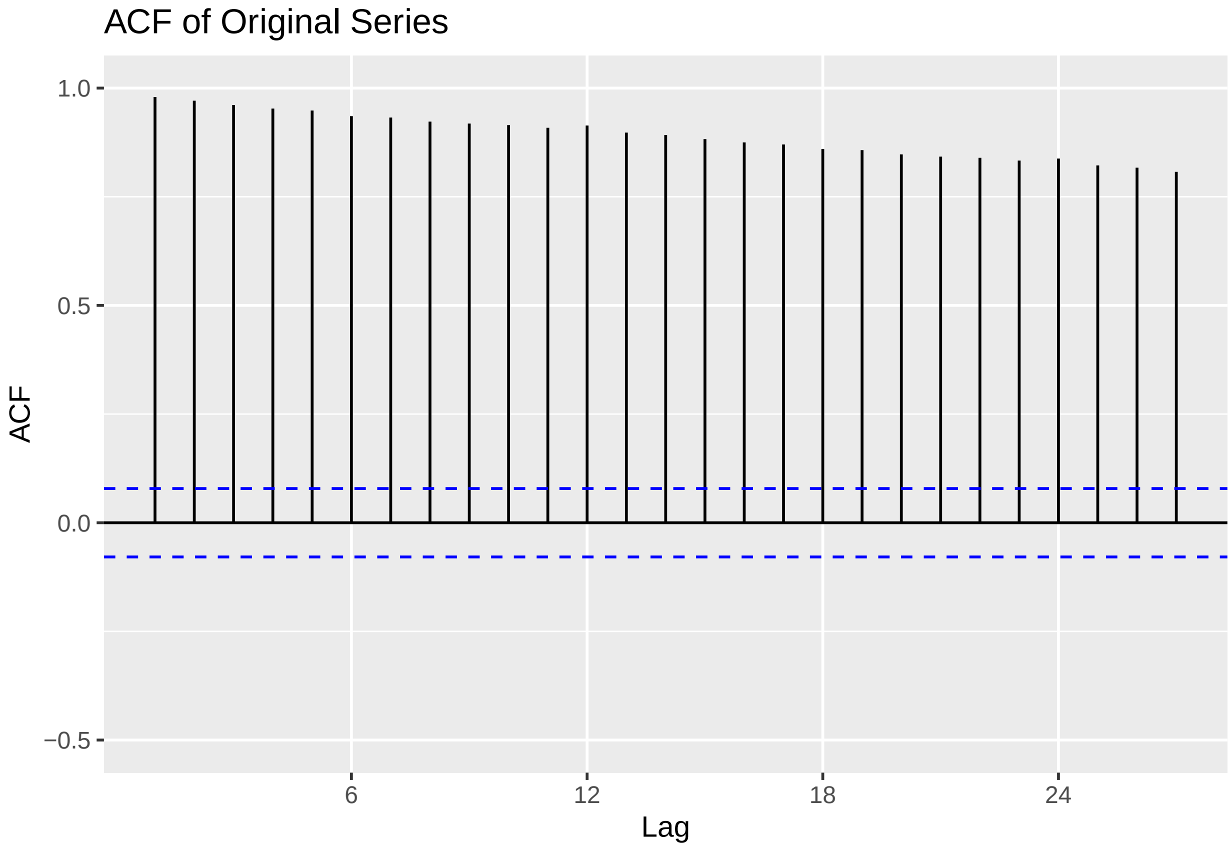
```
plot_acf_diff <- autoplot(acf_diff, ylim = c(-0.5, 1), main = "ACF of Differenced Series")
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: `ylim` and `main`
```

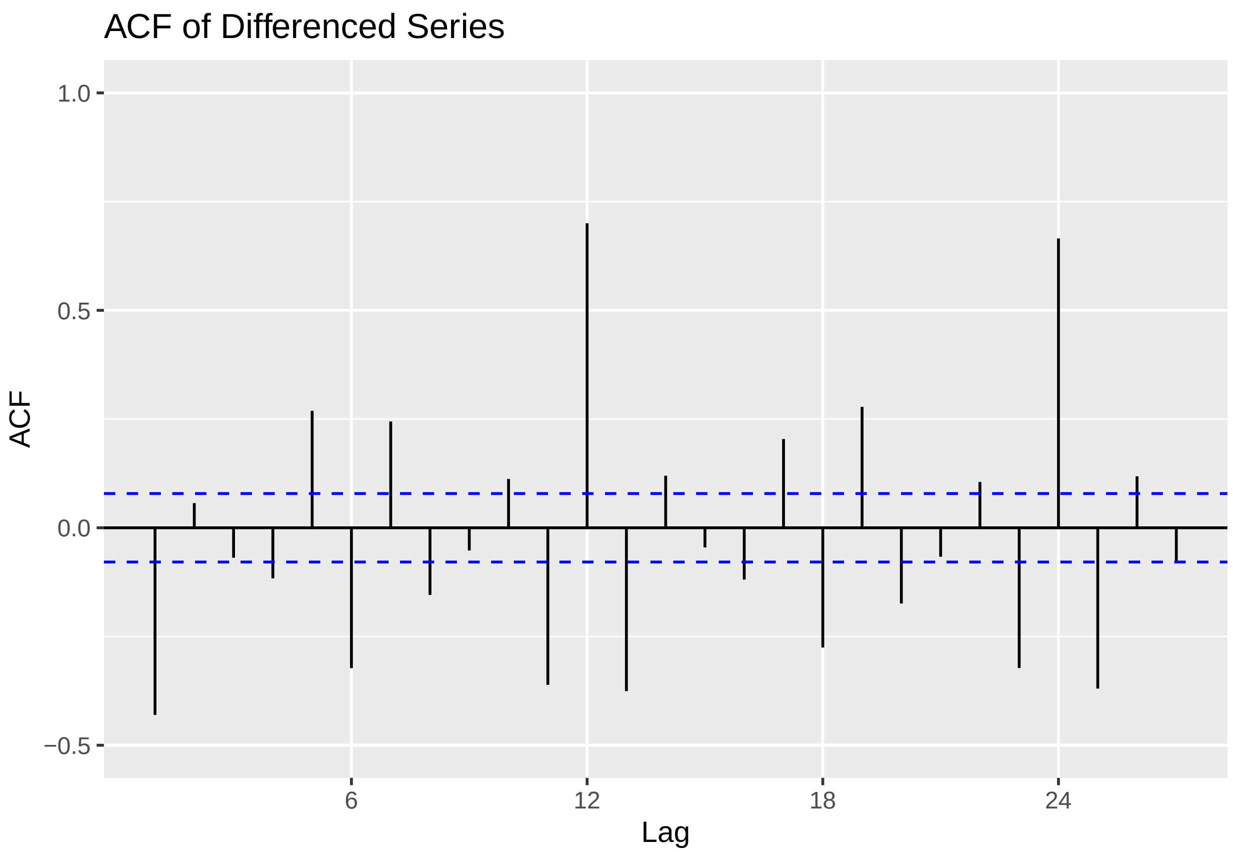
```
plot_acf_detrended <- autoplot(acf_detrended, ylim = c(-0.5, 1), main = "ACF of Detrended Series")
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: `ylim` and `main`
```

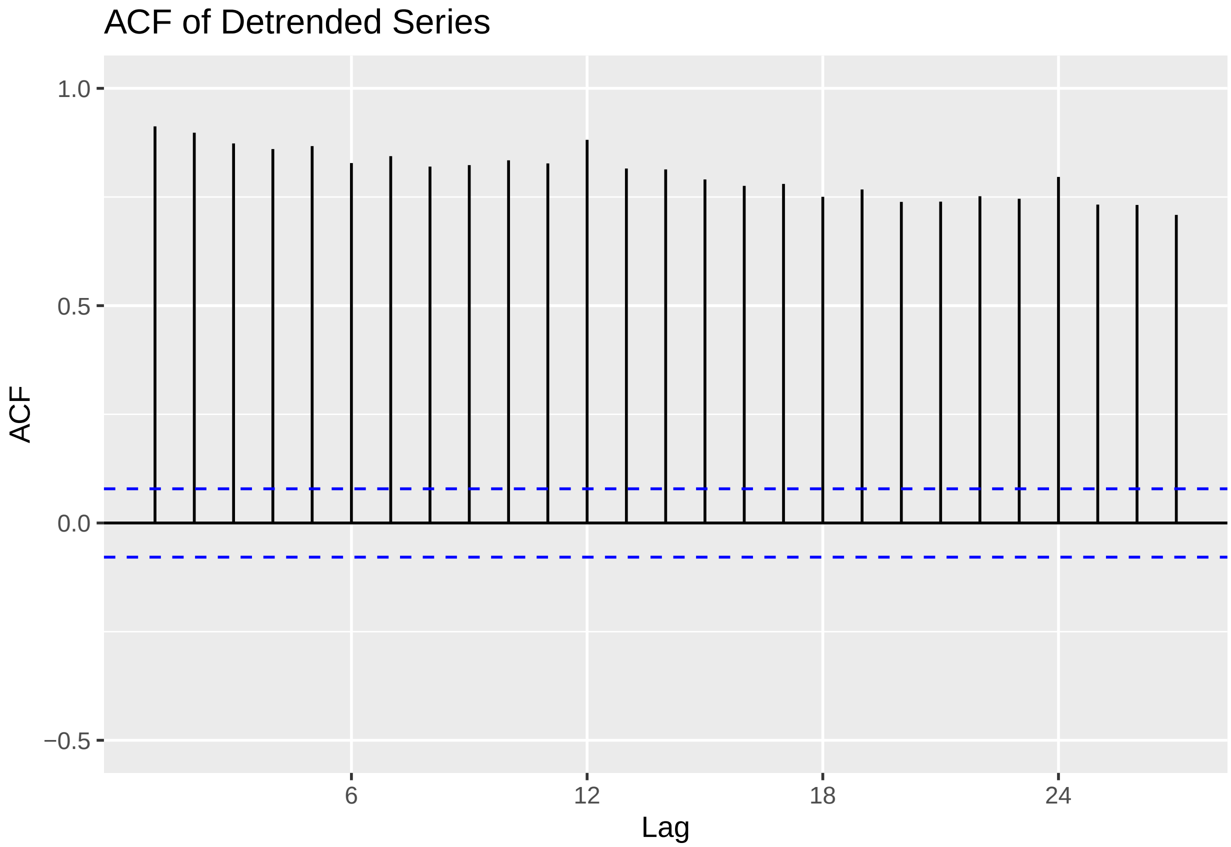
```
# Print the plots
print(plot_acf_original)
```



```
print(plot_acf_diff)
```



```
print(plot_acf_detrended)
```



Answer: Original Series: The ACF plot shows a slow decay, indicating the presence of a trend. Differenced Series: The ACF plot shows a significant reduction in autocorrelation at higher lags and showing a greater seasonal pattern on 12-mon cycle, and the ACF drops off quickly and stays close to zero after a few lags. Detrended Series: The ACF plot shows reduced autocorrelation. Although shows a seasonal pattern, It does not drop as much in later lags. This contrast indicates that differencing has effectively removed the trend. In addition, Linear Regression Detrending can be effective if the trend is deterministic and well-captured by the regression model, but it is not as effective for more complex or stochastic trends. Our series of Total Renewable Production may not be deterministic and can be more complex or stochastic, so linear regression detrending may not be as effective as differencing.

Q5

Compute the Seasonal Mann-Kendall and ADF Test for the original "Total Renewable Energy Production" series. Ask R to print the results. Interpret the results for both test. What is the conclusion from the Seasonal Mann Kendall test? What's the conclusion for the ADF test? Do they match what you observed in Q3 plot? Recall that having a unit root means the series has a stochastic trend. And when a series has stochastic trend we need to use differencing to remove the trend.

```
library(trend)
# Seasonal Mann-Kendall Test
smk_test <- smk.test(renewable_ts)
print(smk_test)
```

```
##
```

```
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  renewable_ts
## z = 28.601, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##      S   varS
##  12468 190008
```

```
# Augmented Dickey-Fuller (ADF) Test
adf_test5 <- adf.test(renewable_ts)
print(adf_test5)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  renewable_ts
## Dickey-Fuller = -1.0898, Lag order = 8, p-value = 0.9242
## alternative hypothesis: stationary
```

> Answer: The purpose of the Seasonal Mann-Kendall Test used to detect trends in seasonal time series. The null is no trend in the data. The Alternative Hypothesis is that there is trend in the data. The t-test statistic is about 28.6, which is much great than 2.64, and the p-value is very low, close to 0 and < 0.05 or even 0.01. Therefore, there is evidence to reject the null and state there is a trend in the seasonal time series. The p-value for the ADF test shows a p-value about 0.92, which is > 0.05 or 0.01, so there is no evidence from the ADF test to reject the null of non-stationary, so the series shows a stochastic trend. This matches the result from comparisons in question 3 as well.

Q6

Aggregate the original "Total Renewable Energy Production" series by year. You can use the same procedure we used in class. Store series in a matrix where rows represent months and columns represent years. And then take the columns mean using function colMeans(). Recall the goal is the remove the seasonal variation from the series to check for trend. Convert the accumulates yearly series into a time series object and plot the series using autoplot().

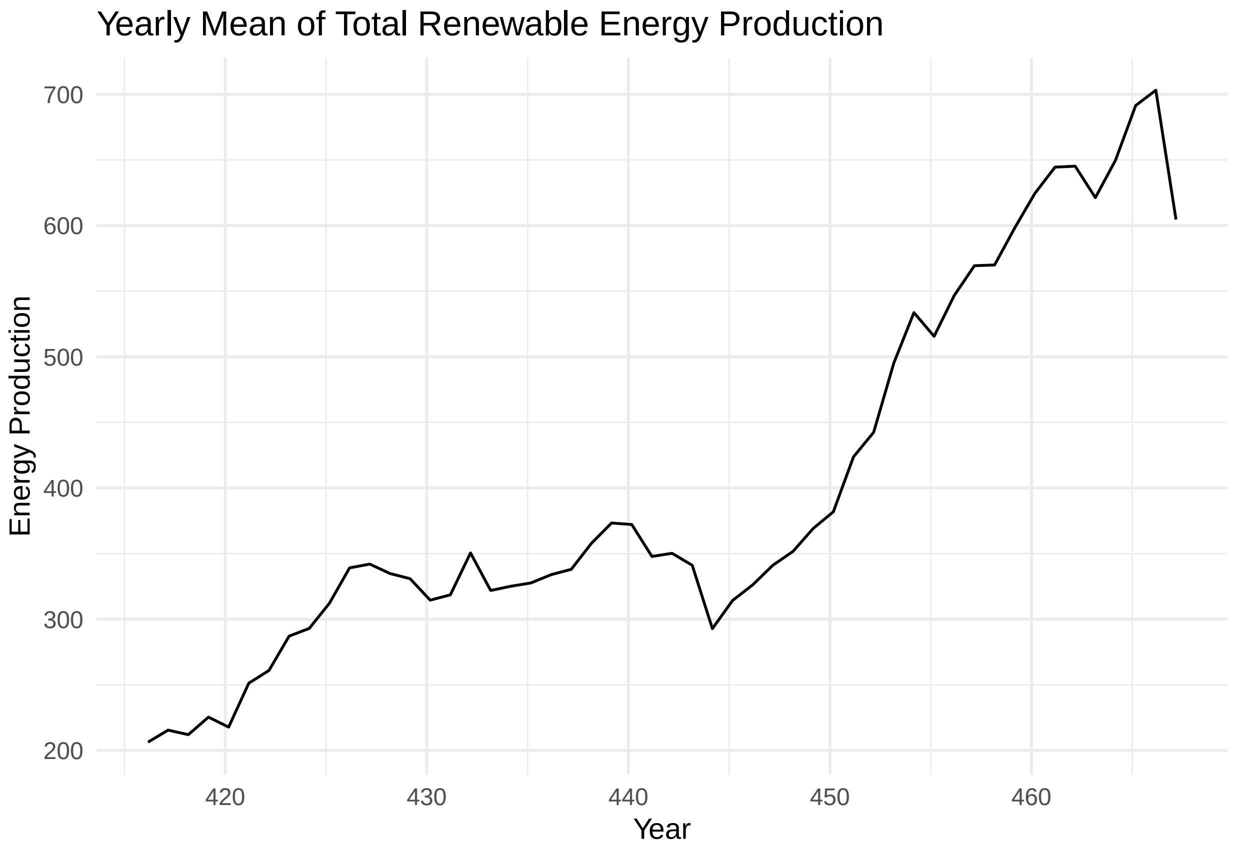
```
# Convert the time series to a matrix with months as rows and years as columns
renewable_matrix <- matrix(renewable_ts, nrow = 12)  # 12 months per year
```

```
## Warning in matrix(renewable_ts, nrow = 12): data length [621] is not a
## sub-multiple or multiple of the number of rows [12]
```

```
# Compute the column means to remove seasonal variation
yearly_means <- colMeans(renewable_matrix, na.rm = TRUE)

# Convert the yearly series into a time series object
# Assuming the series starts in January of the first year
start_year <- start(renewable_ts)[1]  # Extract the starting year
yearly_ts <- ts(yearly_means, start = renewable_ts, frequency = 1)
```

```r
# Step 4: Plot the yearly series using autoplot()
autoplot(yearly_ts) +
  ggtitle("Yearly Mean of Total Renewable Energy Production") +
  xlab("Year") +
  ylab("Energy Production") +
  theme_minimal()
```



Q7

Apply the Mann Kendall, Spearman correlation rank test and ADF. Are the results from the test in agreement with the test results for the monthly series, i.e., results for Q6?

```r
# Mann Kendall
mk_test7 <- mk.test(yearly_ts)
print(mk_test7)
```

```
##
##  Mann-Kendall trend test
##
## data:  yearly_ts
## z = 8.4356, n = 52, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 1.070000e+03 1.605933e+04 8.069382e-01
```

```r
# Spearman correlation
spearman_test <- cor.test(time(yearly_ts), yearly_ts, method = "spearman")
print(spearman_test)
```

```
##
##  Spearman's rank correlation rho
##
## data:  time(yearly_ts) and yearly_ts
## S = 1908, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.918552
```

```r
# ADF
adf_test7 <- adf.test(yearly_ts)
print(adf_test7)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  yearly_ts
## Dickey-Fuller = -1.6634, Lag order = 3, p-value = 0.7098
## alternative hypothesis: stationary
```

> Answer: The Mann Kendal test here shows p-value close to 0, which is <0.05, so we can reject the null and believe that there is trend in the data. The Spearman test results shows that the rho is 0.91, which close to 1, and the p-vale is also very close to 0, and so there is also evidence to reject the null of the spearman test and state that there is a monotonic trend in the Total Renewable Production data. The ADF test result here shows a p-value of roughly 0.71, which indicate that the there is no evidence to reject the null (non-stationary), and the series has a unit root (stochastic trend). This results align similarly to the SMK and ADF test results in Q5 for the original Total Renewable Production series.