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

Assignment 4 - Due date 02/12/24

Zhenghao Lin

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_Sp23.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.\

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

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

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

```
library(ggplot2)
library(forecast)
```

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

```
library(Kendall)
library(tseries)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(readxl)
```

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 - using readxl package
getwd()
```

```
## [1] "/Users/lzh/Desktop/TSA_Sp24"
```

```
energy_data <- read.table(file="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source

#Date Conversion
ym_date <- ym(energy_data$Month)

#New data frame
energy_data <- cbind(ym_date, energy_data[2:14])
rnw_data <- cbind(ym_date, energy_data[5])

#Verification
head(rnw_data)
```

```
##      ym_date Total.Renewable.Energy.Production
## 1 1973-01-01                           219.839
## 2 1973-02-01                           197.330
## 3 1973-03-01                           218.686
## 4 1973-04-01                           209.330
## 5 1973-05-01                           215.982
## 6 1973-06-01                           208.249
```

```r
#Convert to a time series data frame
ts_rnw <- ts(rnw_data[2], start = c(1973, 1), frequency = 12)
```

Stochastic Trend and Stationarity Tests

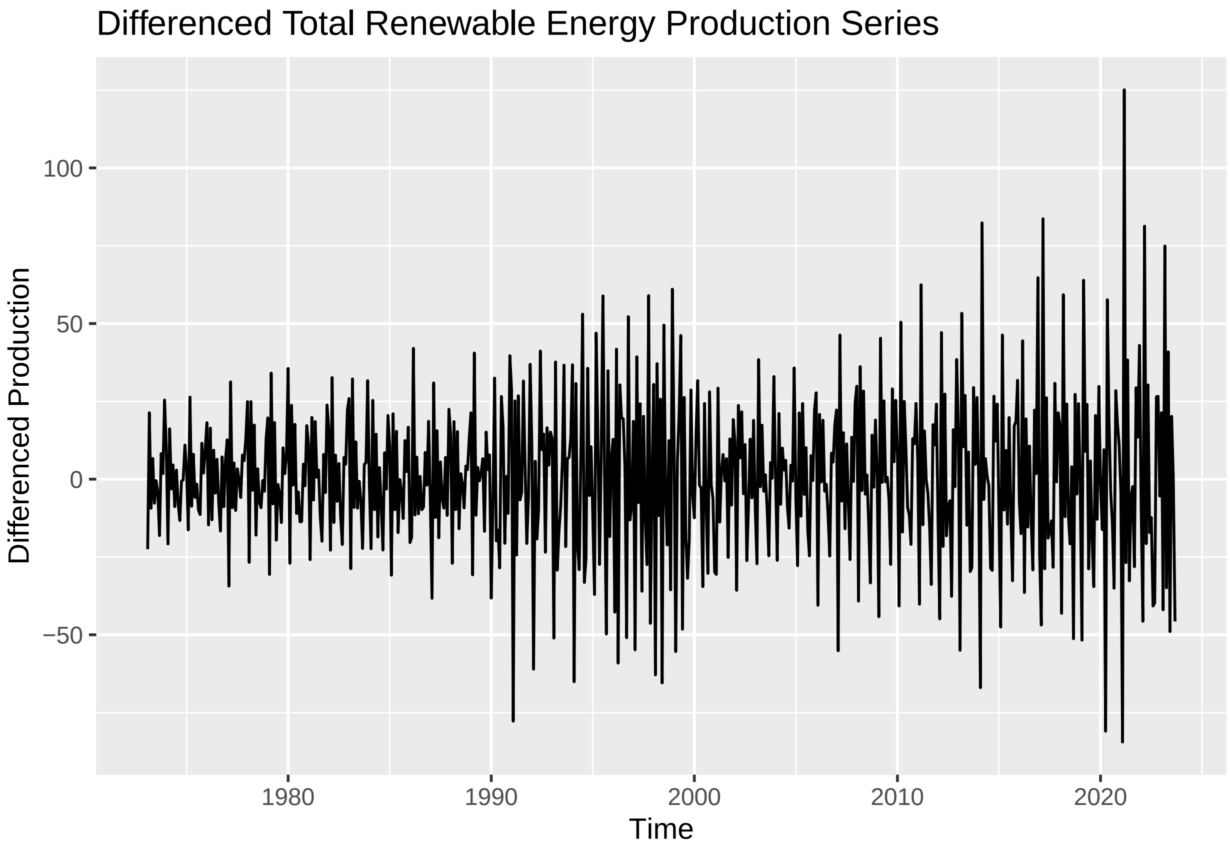
Q1

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
#difference total renewable energy production series once at lag 1
diff_rnw <- diff(x = ts_rnw, lag = 1, differences = 1)

#plot the differenced data
autoplot(diff_rnw) +
  ggtitle("Differenced Total Renewable Energy Production Series") +
  xlab("Time") +
  ylab("Differenced Production")
```



Based on the plots, there is no clear long-term upward or down-ward trend. ### 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.

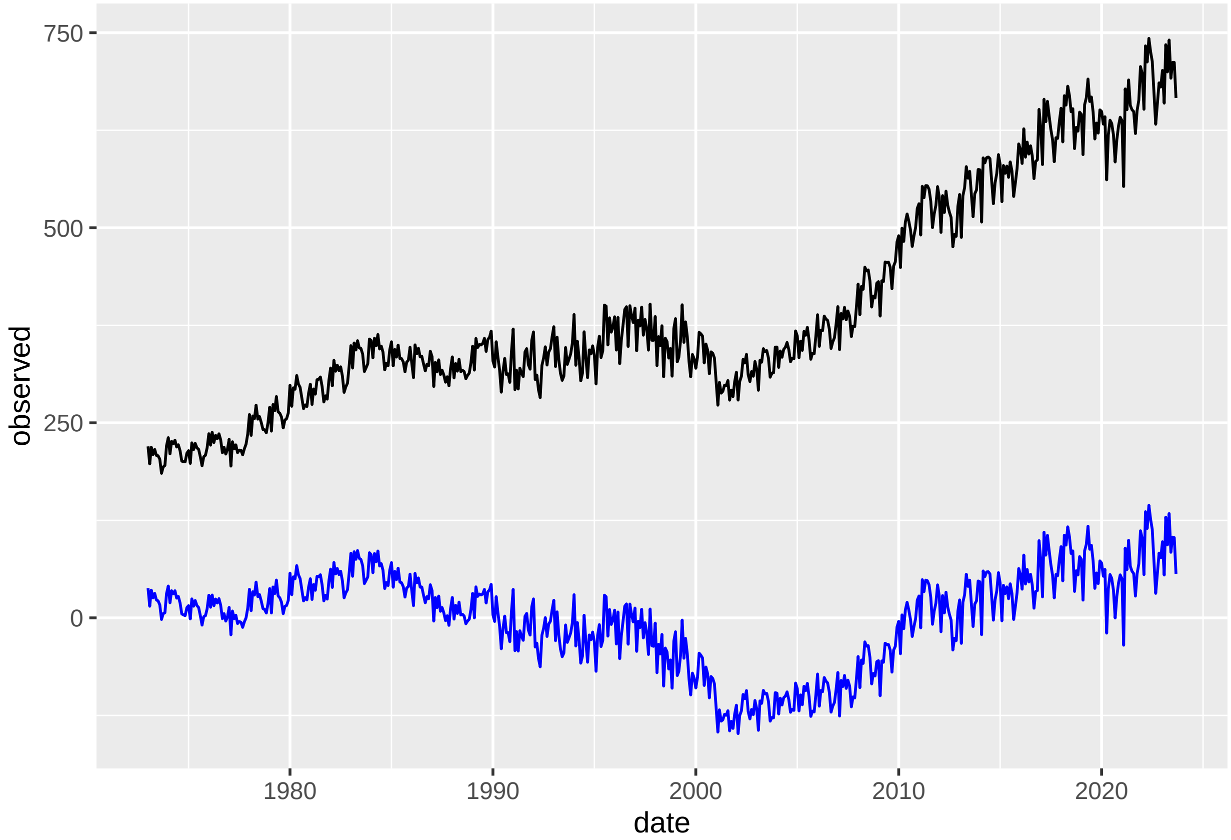
```
#Set up necessary parameters
n <- nrow(energy_data)
t <- 1:n

#Linear trend of Total Renewable Energy Production
trend_rnw <- lm(energy_data[,5]~t)
summary(trend_rnw)
```

```
##
## Call:
## lm(formula = energy_data[, 5] ~ 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
```

```
#Coefficients of the linear trend of Total Renewable Energy Production
beta0_rnw <- trend_rnw$coefficients[1]
beta1_rnw <- trend_rnw$coefficients[2]

#detrend renewable
rnw_detrend <- energy_data[,5] - (beta0_rnw + beta1_rnw*t)
df_detrend_rnw <- data.frame("date" = energy_data$ym_date,
                             "observed" = energy_data[,5],
                             "detrend" = rnw_detrend
                             )
ggplot(df_detrend_rnw, aes(x=date)) +
  geom_line(aes(y=observed), color = "black") +
  geom_line(aes(y=detrend), color = "blue")
```



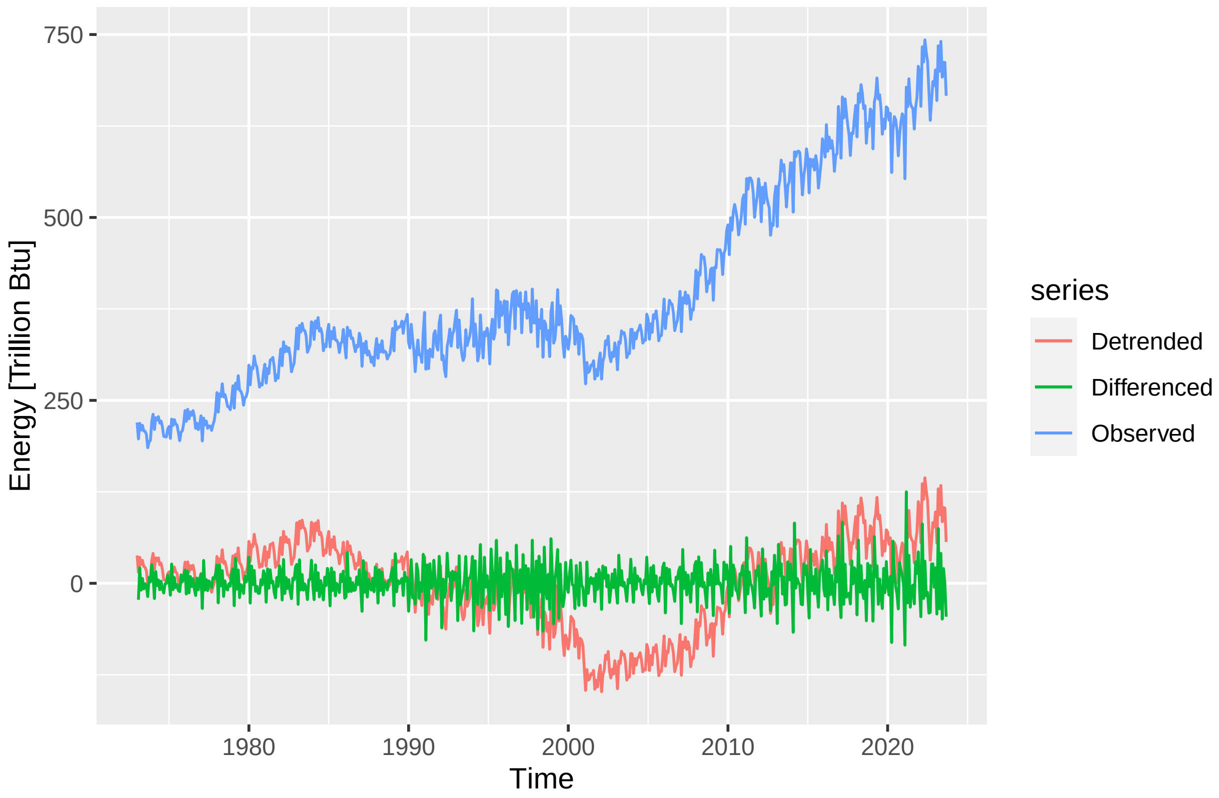
Q3

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.

```
ts_df_rnw_detrend <- ts(df_detrend_rnw[,2:3], frequency=12,start=c(1973,1))

autoplot(ts_df_rnw_detrend[,1], series = "Observed") +
  autolayer(ts_df_rnw_detrend[,2], series = "Detrended") +
  autolayer(diff_rnw, series = "Differenced") +
  ylab("Energy [Trillion Btu]")
```



```r
  ggtitle("Total Renewable Energy Production")
```

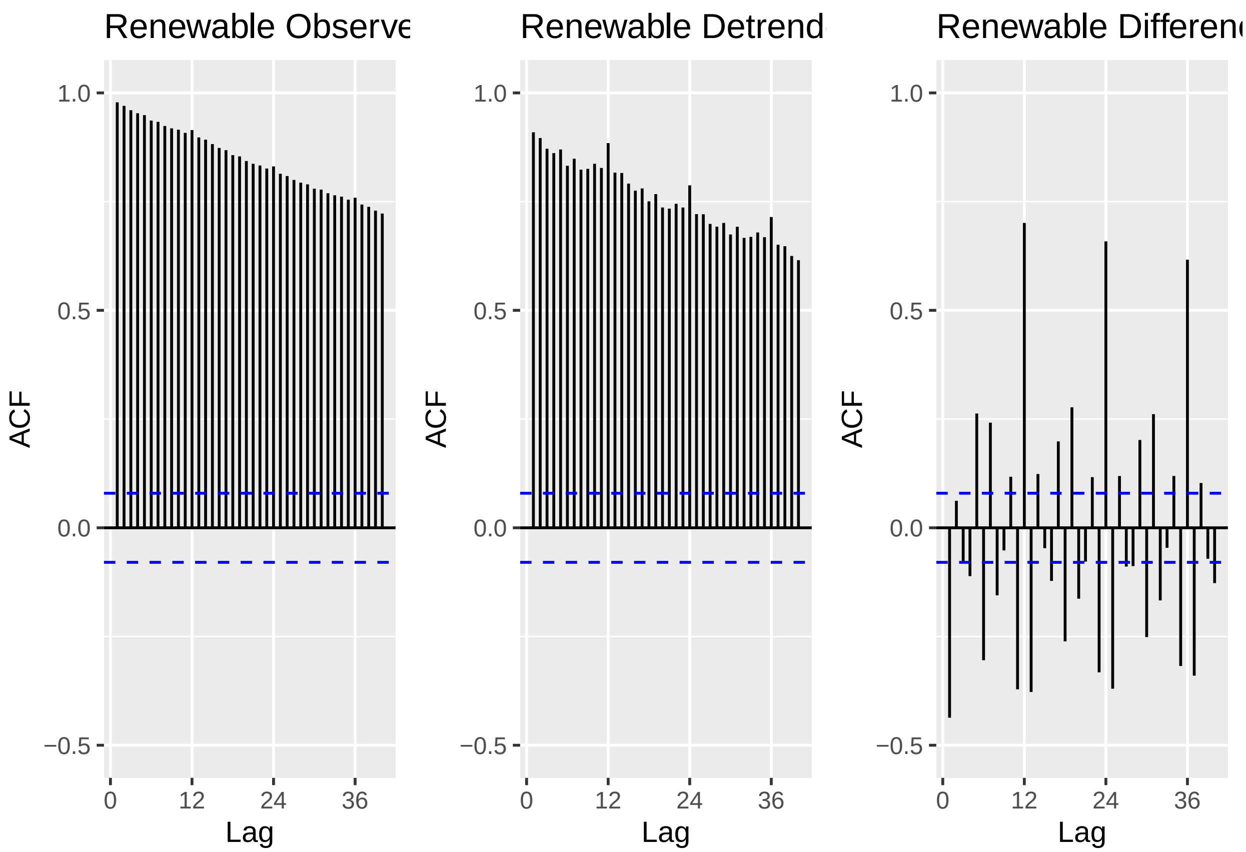
```
## $title
## [1] "Total Renewable Energy Production"
##
## attr(,"class")
## [1] "labels"
```

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. Which method do you think was more efficient in eliminating the trend? The linear regression or differencing?

```r
plot_grid(
autoplot(Acf(ts_df_rnw_detrend[,1],lag.max=40,plot=FALSE),main="Renewable Observed", ylim=c(-0.5,1)),
autoplot(Acf(ts_df_rnw_detrend[,2],lag.max=40,plot=FALSE),main="Renewable Detrended",ylim=c(-0.5,1)),
autoplot(Acf(diff_rnw,lag.max=40,plot=FALSE),main="Renewable Differenced",ylim=c(-0.5,1)),
nrow=1,ncol=3
)
```

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



Based on the plots, differencing appears to be more efficient in eliminating the trend in the time series data, as the autocorrelations become insignificant much faster than in the detrended series. In contrast, the detrending method still leaves some autocorrelation in the data, which may indicate that some aspects of the trend are still present. ### 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 Q2? Recall that having a unit root means the series has a stochastic trend. And when a series has stochastic trend we need to use a different procedure to remove the trend.

```r
smk_test <- SeasonalMannKendall(ts_rnw)
print(smk_test)
```

```
## tau = 0.783, 2-sided pvalue =< 2.22e-16
```

```r
# Augmented Dickey-Fuller (ADF) Test
adf_test <- adf.test(ts_rnw, alternative = "stationary")
print(adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_rnw
## Dickey-Fuller = -1.24, Lag order = 8, p-value = 0.9
## alternative hypothesis: stationary
```

SMK: The Seasonal Mann-Kendall test is used to detect trends in time series data, considering seasonal variations. Here, tau = 0.783 suggests a strong positive trend within the time series. The p-value of less than 2.22e-16 indicates that this trend is statistically significant. Therefore, we can conclude there's a strong and statistically significant upward trend in the time series data.

AD-Fuller: The ADF test is used to test for the presence of a unit root in a time series, which is an indication of non-stationarity. A series is said to be stationary if its statistical properties do not change over time. The Dickey-Fuller statistic of -1.24 and a high p-value of 0.9 suggests that we cannot reject the null hypothesis of the presence of a unit root, implying the time series is likely non-stationary and has a stochastic trend.

Q6

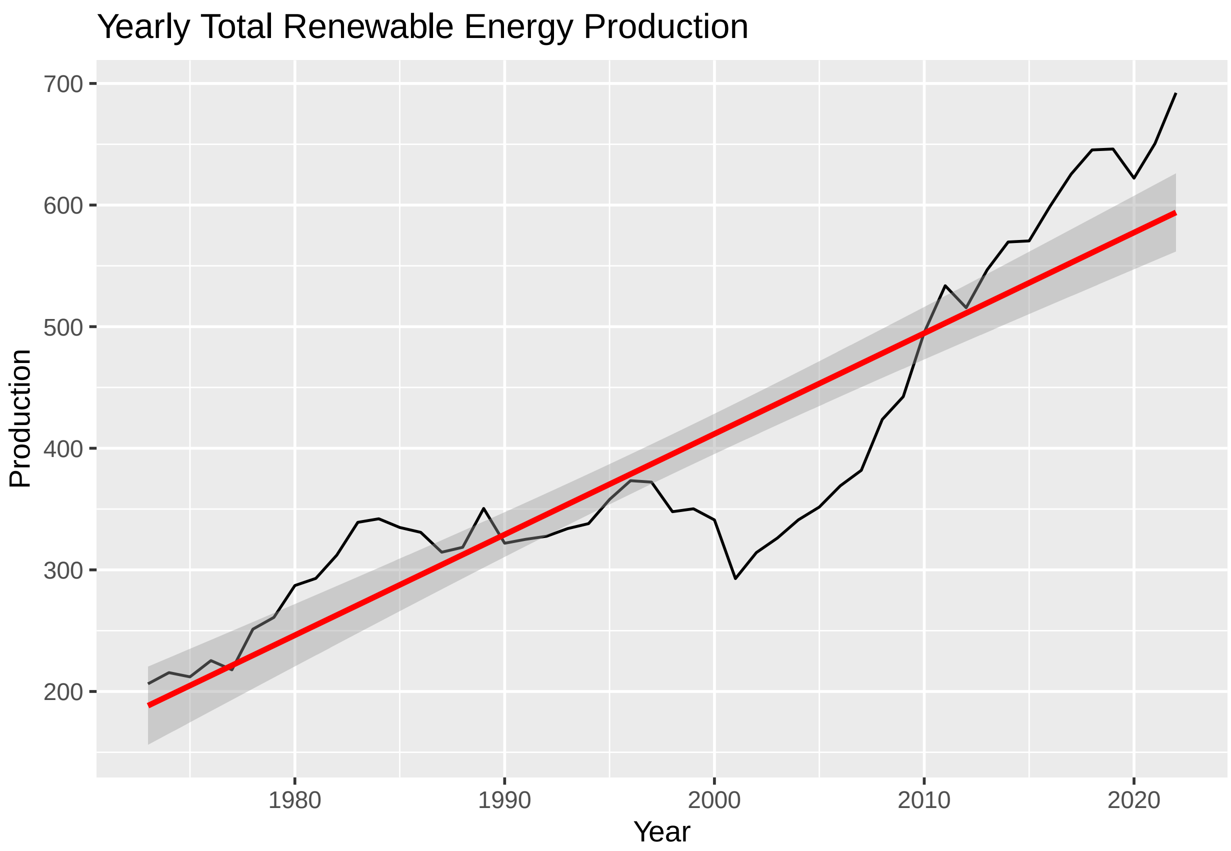
Aggregate the original "Total Renewable Energy Production" series by year. You can use the same procedure we used in class. Store series in a matrix where rows represent months and columns represent years. And then take the columns mean using function colMeans(). Recall the goal is the remove the seasonal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

```r
# Aggregate series by year
yearly_ts <- ts_rnw[1:(n-9),]
yearly_data <- matrix(yearly_ts, nrow = 12, byrow = FALSE)
yearly_means <- colMeans(yearly_data)

# Convert to time series object
ts_yearly_means <- ts(yearly_means, start = 1973, frequency = 1)

# Plot the series
autoplot(ts_yearly_means) +
  ggtitle("Yearly Total Renewable Energy Production") +
  xlab("Year") +
  ylab("Production") +
  geom_smooth(color="red",method="lm")
```

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



Q7

Apply the Mann Kendal, 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?

```
mk_test_yearly <- MannKendall(ts_yearly_means)
print(mk_test_yearly)
```

```
## tau = 0.802, 2-sided pvalue =< 2.22e-16
```

```
# Spearman Correlation Rank Test for yearly data
spearman_test_yearly <- cor.test(ts_yearly_means, 1:length(ts_yearly_means), method = "spearman")
print(spearman_test_yearly)
```

```
##
##  Spearman's rank correlation rho
##
## data:  ts_yearly_means and 1:length(ts_yearly_means)
## S = 1852, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9110684
```

```
# ADF Test for yearly data
adf_test_yearly <- adf.test(ts_yearly_means, alternative = "stationary")
print(adf_test_yearly)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_yearly_means
## Dickey-Fuller = -1.0881, Lag order = 3, p-value = 0.9156
## alternative hypothesis: stationary
```

SMK_yearly: Similar to the Seasonal Mann-Kendall test for the monthly series, the Mann-Kendall test for the yearly data shows a strong and statistically significant upward trend with tau = 0.802 and a p-value smaller than 2.22e-16. This indicates a consistent trend over time, confirming the presence of a significant trend in both the monthly and yearly series.

Spearman_yearly: The Spearman rank correlation test also indicates a very strong and statistically significant positive correlation between time and the yearly means (rho = 0.9110684 with a p-value smaller than 2.2e-16). This test further supports the presence of a strong upward trend in the data, in line with the Mann-Kendall test results.

ADF_yearly: Similar to the ADF test results for the monthly series, the ADF test for the yearly data suggests that the series is likely non-stationary (p-value = 0.9156), indicating the presence of a unit root. This means that the time series does not exhibit constant statistical properties over time, which matches the findings from the monthly series.