Time Series Analysis for Energy Data | Spring 2023

Kelsey Husted

```
#Load/install required package here
library(readxl)
library(openxlsx)
library(lubridate)
library(ggplot2)
library(forecast)
library(tseries)
library(Kendall)
library(tidyverse)
```

Part 1: Import data

The data comes from the US Energy Information and Administration and corresponds to the December 2022 Monthly Energy Review. Will focus only on the column "Total Renewable Energy Production".

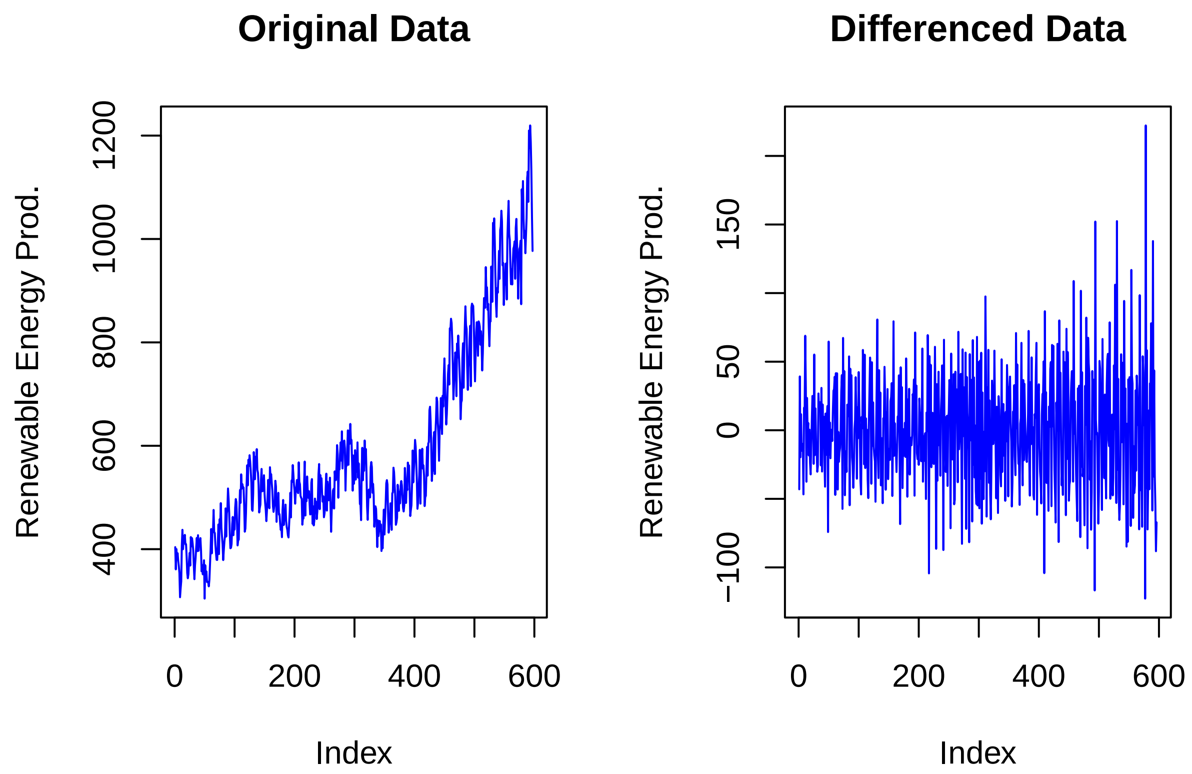
```
#Importing data set - using xlsx package
energy_data <- read.xlsx("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx"

#Selecting for Renewable Energy Production column
energy_data_processed <- energy_data %>%
  mutate(Date = convertToDate(energy_data$Month)) %>% #Month column imports incorrectly
  select(Date, Total.Renewable.Energy.Production)

#Remove the row with the units
energy_data_processed <- energy_data_processed[-c(1), ]

#Transform energy data column from character to numeric
energy_data_processed$Total.Renewable.Energy.Production = as.numeric(energy_data_processed$Total.Renewa
```

Part 2: Stochastic Trend and Stationary Tests

Differencing the data to remove the trend

Note: The trend was removed after differencing the time series as displayed in the plots below.

```
#Differencing the data
diff_data <- diff(energy_data_processed[,2], lag = 1, differences = 1)

#Plot original data and the differenced data
par(mfrow = c(1,2))
plot(energy_data_processed[,2], type="l", col = "blue", ylab = "Renewable Energy Prod.", main = "Origina
plot(diff_data, type = "l", col="blue", ylab = "Renewable Energy Prod.", main = "Differenced Data")
```



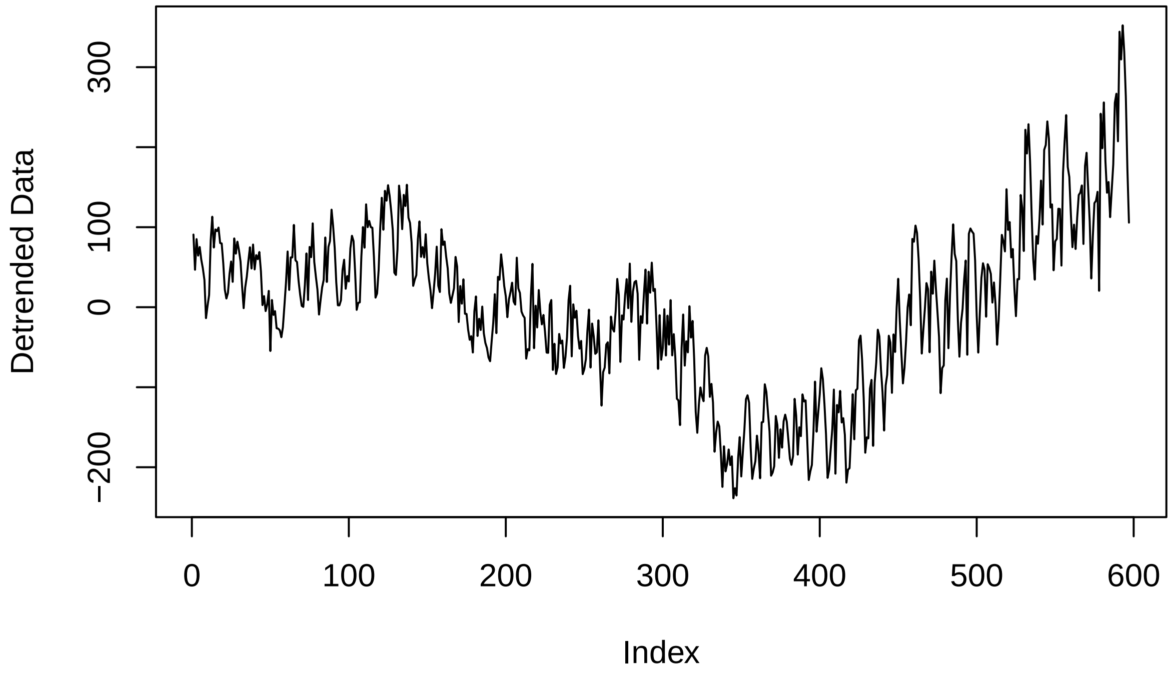
Part 3: Compare Differenced and Detrended Series

Comparing the differenced series with the detrended series that I calculated in the previous section.

```
num.obs <- nrow(energy_data_processed)
observ <- 1:num.obs

#Run linear model
linear_model<- lm(energy_data_processed[,2] ~ observ)
summary(linear_model)
```

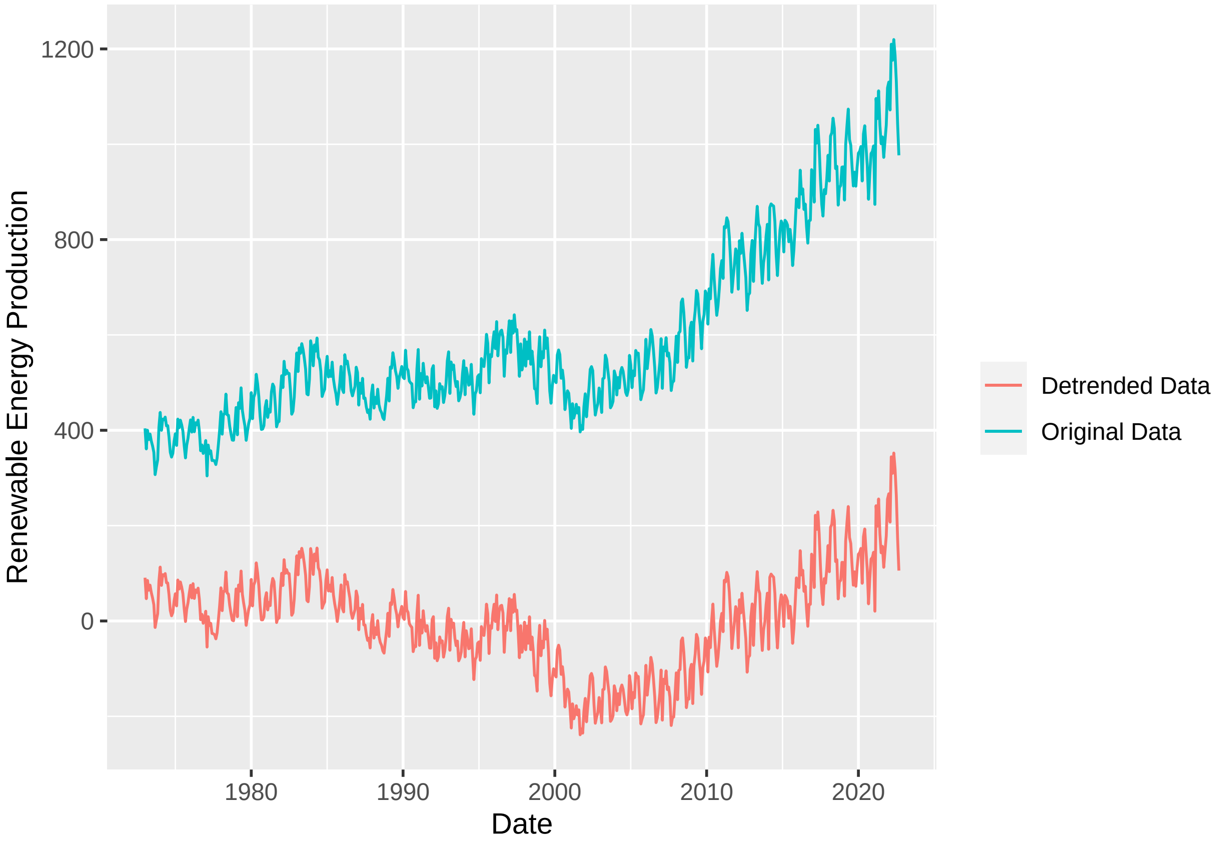
```
## 
## Call:
## lm(formula = energy_data_processed[, 2] ~ observ)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -238.75  -61.85    8.59   64.48  352.27 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 312.2475     8.4902   36.78   <2e-16 ***
## observ        0.9362     0.0246   38.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 103.6 on 595 degrees of freedom
## Multiple R-squared:  0.7088, Adjusted R-squared:  0.7083 
## F-statistic:  1448 on 1 and 595 DF,  p-value: < 2.2e-16
```

```
#Save regression coefficients as a variable
beta0_renewable=as.numeric(linear_model$coefficients[1])
#first coefficient is the intercept term or beta0
beta1_renewable=as.numeric(linear_model$coefficients[2])

#Detrend data by using the regression coefficients saved above
RenewableEnergy_detrend <- energy_data_processed[,2]-(beta0_renewable+beta1_renewable*observ)
plot(RenewableEnergy_detrend, type = "l", ylab = "Detrended Data")
```



```
#Plot original data versus detrended data
ggplot(energy_data_processed, aes(x = Date, y = energy_data_processed[,2], colour = "Original Data", col
  geom_line() +
  ylab(paste0("Detrended Renewable Energy")) +
  geom_line(aes(y=RenewableEnergy_detrend, colour = "Detrended Data"))+
  labs(x = "Date", y = "Renewable Energy Production", color = ' ')
```

```
## Warning: Duplicated aesthetics after name standardisation: colour
```



Part 4: Create New Dataframe

Creating a new data frame with 4 columns: date, original series, detrended by regression series and differenced series.

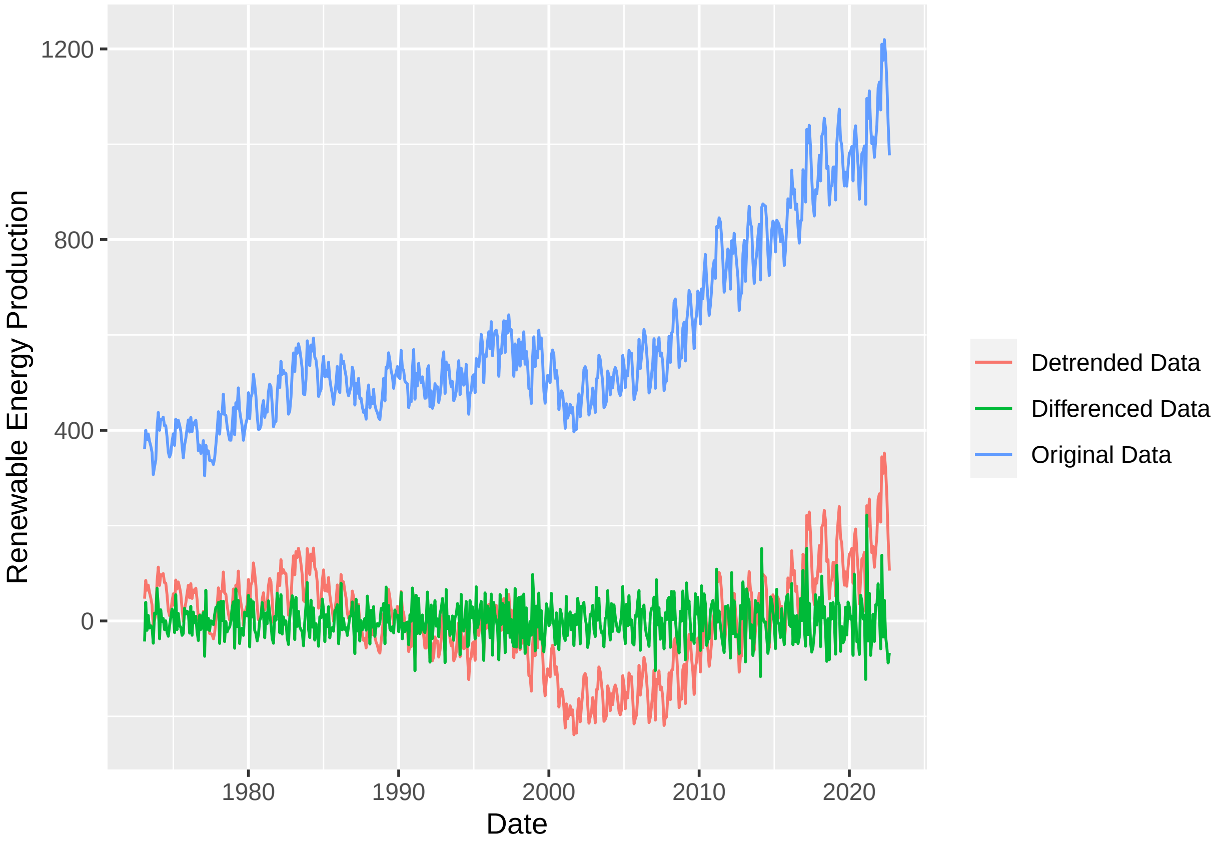
```
#Remove first row (i.e., January 1973) because differenced series will have less rows
energy_data_processed<- energy_data_processed[-c(1), ]
Detrend <- RenewableEnergy_detrend[2:597]

#Create data frame - remember to not include January 1973
date <- energy_data_processed[,1]
RenewableEnergyProduction <- energy_data_processed[,2]
energy_df <- data.frame(date ,RenewableEnergyProduction, Detrend, diff_data)
```

Part 5: Create ggplot

Using ggplot(), creating a line plot that shows the three series together.

```
#Use ggplot
ggplot(energy_df, aes(x = date, y = RenewableEnergyProduction, colour = "Original Data", col="blue")) +
  geom_line() +
  geom_line(aes(y=Detrend, colour = "Detrended Data", col = "green")) +
  geom_line(aes(y=diff_data, colour = "Differenced Data", col = "red")) +
  labs(x = "Date", y = "Renewable Energy Production", color = ' ')
```

```
## Warning: Duplicated aesthetics after name standardisation: colour
## Duplicated aesthetics after name standardisation: colour
## Duplicated aesthetics after name standardisation: colour
```

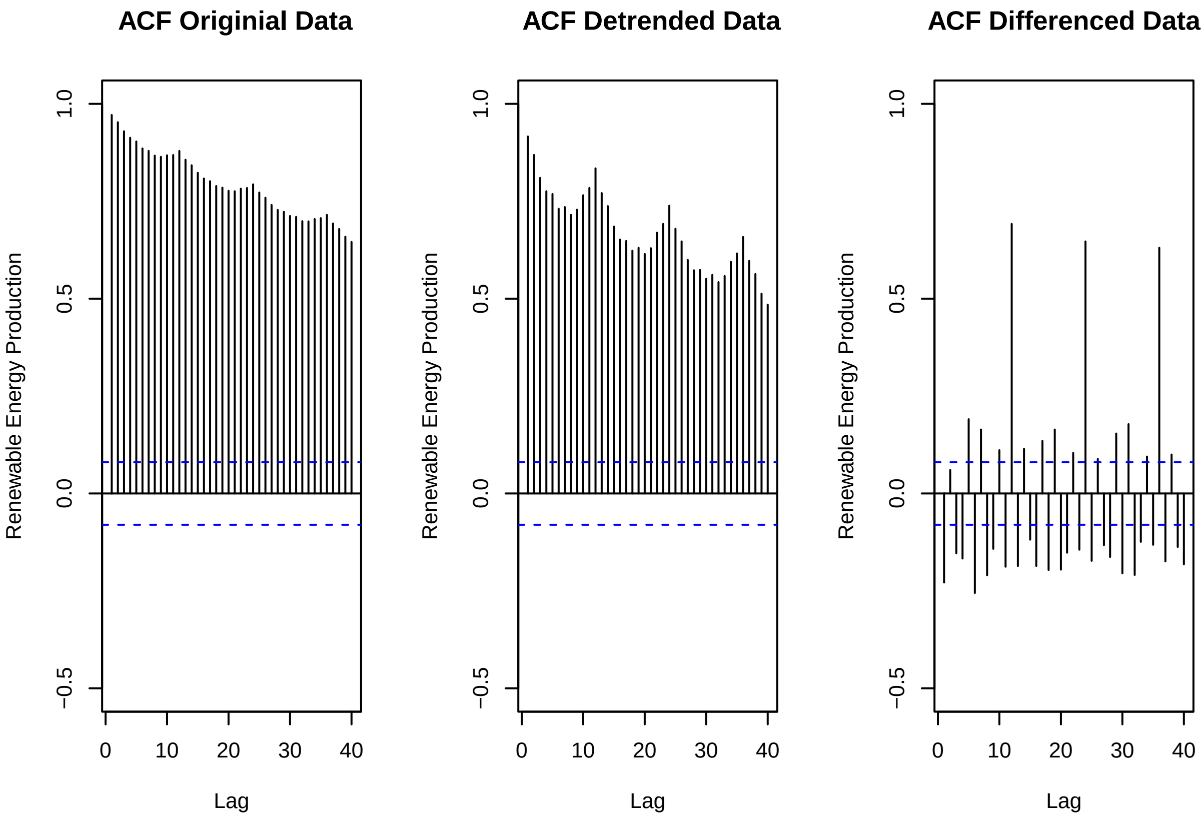


Part 6: Plot ACF (Autocorrelation function)

Plotting the ACF for the three series and comparing the plots.

> Plot interpretations: According to the ACF plots displayed below, differencing the original data seems to be a more efficient approach to removing the trend. Small lags that are small and positive and diminish over time are typically associated with a trend which is not seen in the differenced ACF plot compared to the other plots.

```
#Compare ACFs

par(mfrow=c(1,3))
Acf(energy_df[,2], lag.max = 40, ylim=c(-0.5,1), ylab="Renewable Energy Production", main=" ACF Originia
Acf(energy_df[,3], lag.max = 40, ylim=c(-0.5,1), ylab="Renewable Energy Production", main=" ACF Detrende
Acf(energy_df[,4], lag.max = 40, ylim=c(-0.5,1),ylab="Renewable Energy Production", main="ACF Difference
```



Part 7: Seasonal Mann-Kendall & ADF Test

Computing the Seasonal Mann-Kendall and ADF Test for the original "Total Renewable Energy Production" series.

> Statistical Results Interpretation: A Seasonal Mann-Kendall Test is used to determine whether or not a trend exists in a time series data. Since the results of the Seasonal Mann-Kendall Test had a p-value less than 0.05, the null hypothesis is rejected which indicates a trend present in the data. The ADF test had a p-value almost equal to 1 which indicates a stoichastic trend and differencing (rather than using regression) removed the trend more efficiently as seen in the plots from Q4.

```
#Create a ts variable
ts_energy_data <- ts(energy_data_processed[,2], frequency = 12, start = c(1973,1))

#Mann-Kendall Test
SMKtest <- SeasonalMannKendall(ts_energy_data)
print(summary(SMKtest))
```

```
## Score =  10532 , Var(Score) = 168168
## denominator =  14504
## tau = 0.726, 2-sided pvalue =< 2.22e-16
## NULL
```

```
#p-value=<2.22e-16 so we reject the null hypothesis (i.e., no trend)

#ADF Test
print(adf.test(ts_energy_data,alternative = "stationary"))
```

```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  ts_energy_data
## Dickey-Fuller = -1.1948, Lag order = 8, p-value = 0.9073
## alternative hypothesis: stationary
```

```
#p-value = 0.9056 so we accept the null hypothesis (i.e., stochastic trend)
```

Part 8: Aggregate by Year

Aggregating the original "Total Renewable Energy Production" series by year.

```
#Create a year column
annual_energy_data <- energy_data_processed %>%
  mutate(Year = lubridate::year(energy_data_processed$Date))%>%
  select('Year', 'Total.Renewable.Energy.Production')

#Group by year for annual data
annual_energy_data <- annual_energy_data %>%
  group_by(Year) %>%
  summarise(RenewableEnergy = mean(Total.Renewable.Energy.Production))
```

Part 9: Mann-Kendall, Spearman Correlation Rank Test, & ADF Test

Applying the Mann-Kendall, Spearman Correlation Rank Test, and ADF Test.

```
#Create a ts variable
ts_annual_energy_data <- ts(annual_energy_data[,2], frequency = 1, start = c(1973))

#Mann-Kendall Test
MKtest <- MannKendall(ts_annual_energy_data)
print(summary(MKtest))#p-value=<2.22e-16 so we reject the null hypothesis (i.e., no trend)
```

```
## Score =  913 , Var(Score) = 14291.67
## denominator =  1225
## tau = 0.745, 2-sided pvalue =< 2.22e-16
## NULL
```

```
#Spearman Correlation Test
cor.test(annual_energy_data$RenewableEnergy,annual_energy_data$Year, method = "spearman")
```

```
##
##  Spearman's rank correlation rho
##
## data:  annual_energy_data$RenewableEnergy and annual_energy_data$Year
## S = 2548, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8776471
```

```
#ADF Test
print(adf.test(ts_annual_energy_data,alternative = "stationary"))
```

```
## Warning in adf.test(ts_annual_energy_data, alternative = "stationary"): p-value
## greater than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  ts_annual_energy_data
## Dickey-Fuller = 0.066004, Lag order = 3, p-value = 0.99
## alternative hypothesis: stationary
```

```
#p-value = 0.99 so we accept the null hypothesis (i.e., stochastic trend)
```