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

Assignment 7 - Due date 03/06/25

Lucy Wang

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_A07_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.

Packages needed for this assignment: "forecast","tseries". Do not forget to load them before running your script, since they are NOT default packages.\

Set up

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

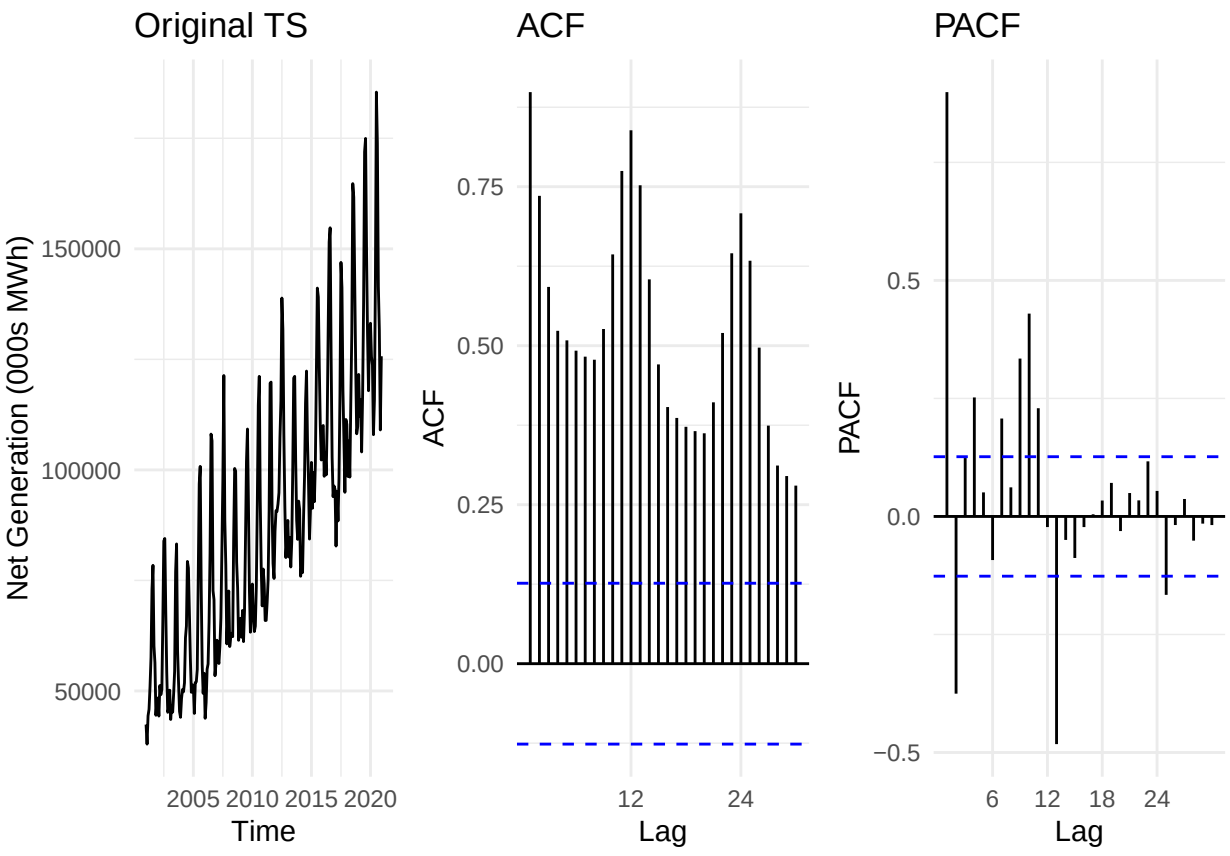
Importing and processing the data set

Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

Q1

Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

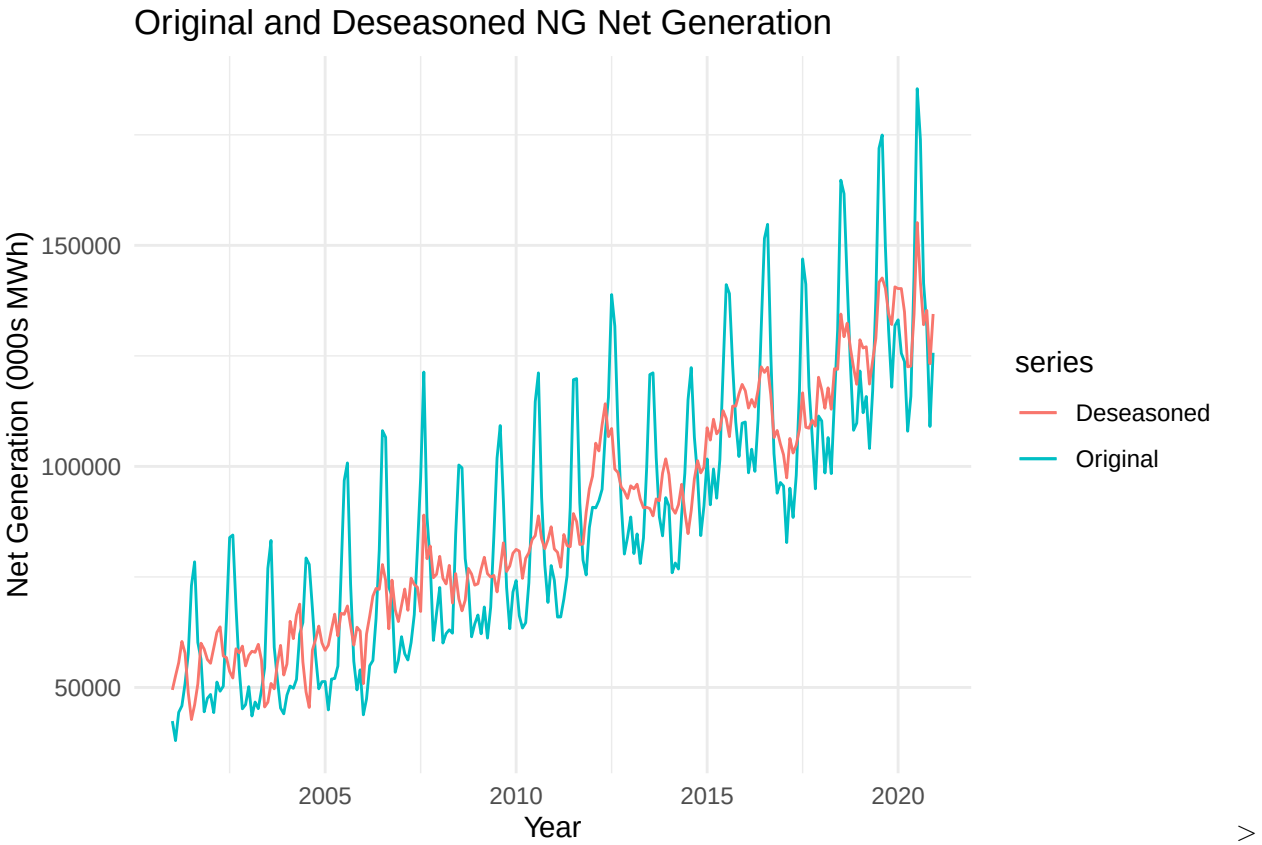
```r
# Read csv file
gen_data <- read.csv(file="./Data/Net_generation_United_States_all_sectors_monthly.csv", header=TRUE,sk
  mutate(Month = dmy(paste0("01 ", Month))) %>%
  arrange(Month)

# Create a time series for natural gas column
NG_ts <- ts(gen_data$natural.gas.thousand.megawatthours,start = c(2001,1), frequency = 12)

# Plot the time series over time, ACF and PACF
ts_plot <- autoplot(NG_ts) +
  xlab("Time") +
  ylab("Net Generation (000s MWh)") +
  ggtitle("Original TS") +
  theme_minimal()

acf_plot <- autoplot(Acf(NG_ts, lag = 30, plot=FALSE)) +
    ggtitle("ACF") +
    theme_minimal()

pacf_plot <- autoplot(Pacf(NG_ts, lag = 30, plot=FALSE)) +
    ggtitle("PACF") +
    theme_minimal()

# Arrange the plots side by side using plot_grid()
plot_grid(ts_plot, acf_plot, pacf_plot, ncol = 3, align = 'h')
```



Q2

Using the *decompose*() and the *seasadj*() functions create a series without the seasonal component, i.e., a deseasonalized natural gas series. Plot the deseasonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

```
# Decompose time series
decompose_ts <- decompose(NG_ts,"additive")

# Non-seasonal time series
deseasonal_ts <- seasadj(decompose_ts)

#Plotting original and deseasoned series
autoplot(NG_ts, series="Original") +
  autolayer(deseasonal_ts, series="Deseasoned") +
  xlab("Year") + ylab("Net Generation (000s MWh)") +
  ggtitle("Original and Deseasoned NG Net Generation")+
  theme_minimal()
```

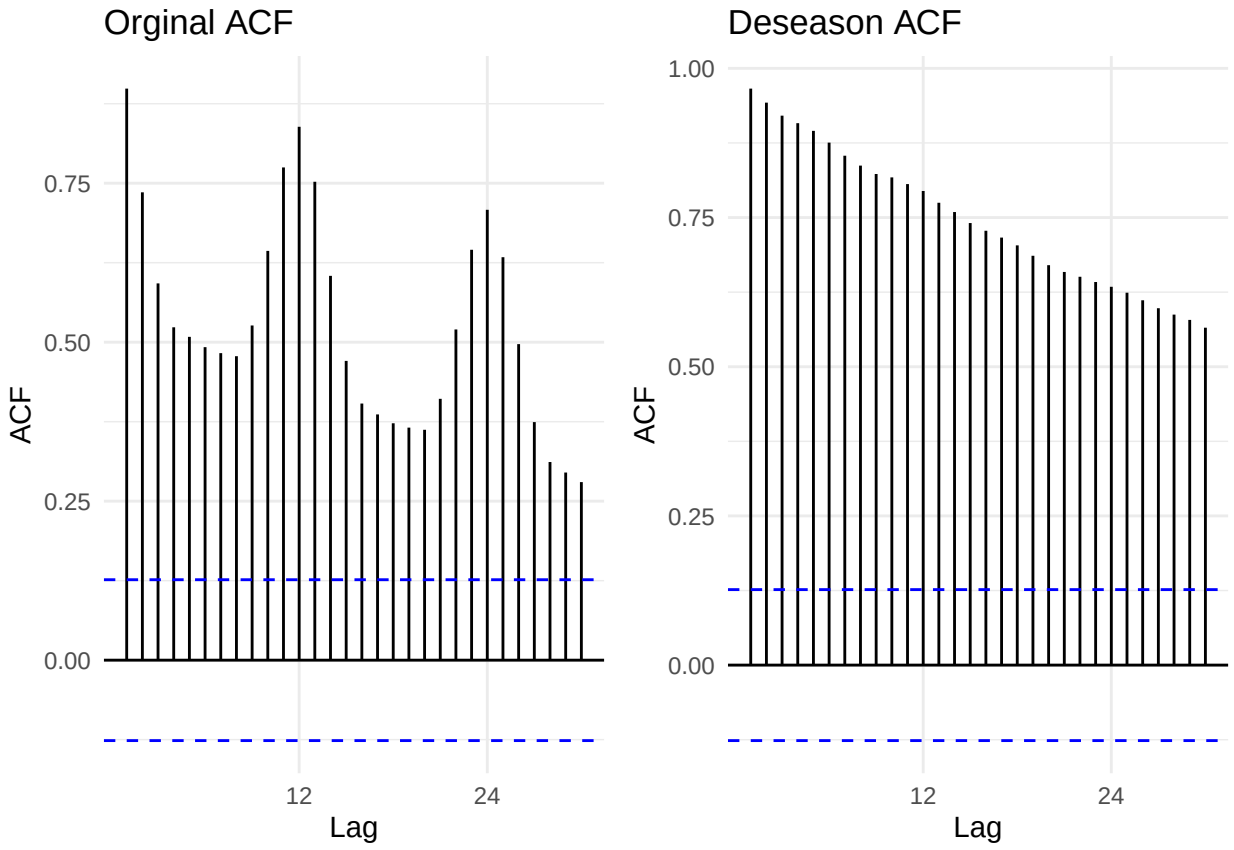


Time series plots

The original time series shows greater periodic fluctuations over time. The deseasoned series is smoothed, and the periodic spikes are less observable. The long-term upward trend becomes more clear.

```
#Comparing ACFs
plot_grid(
  autoplot(Acf(NG_ts, lag = 30, plot=FALSE),
                main = "Orginal ACF")+
                theme_minimal(),
```

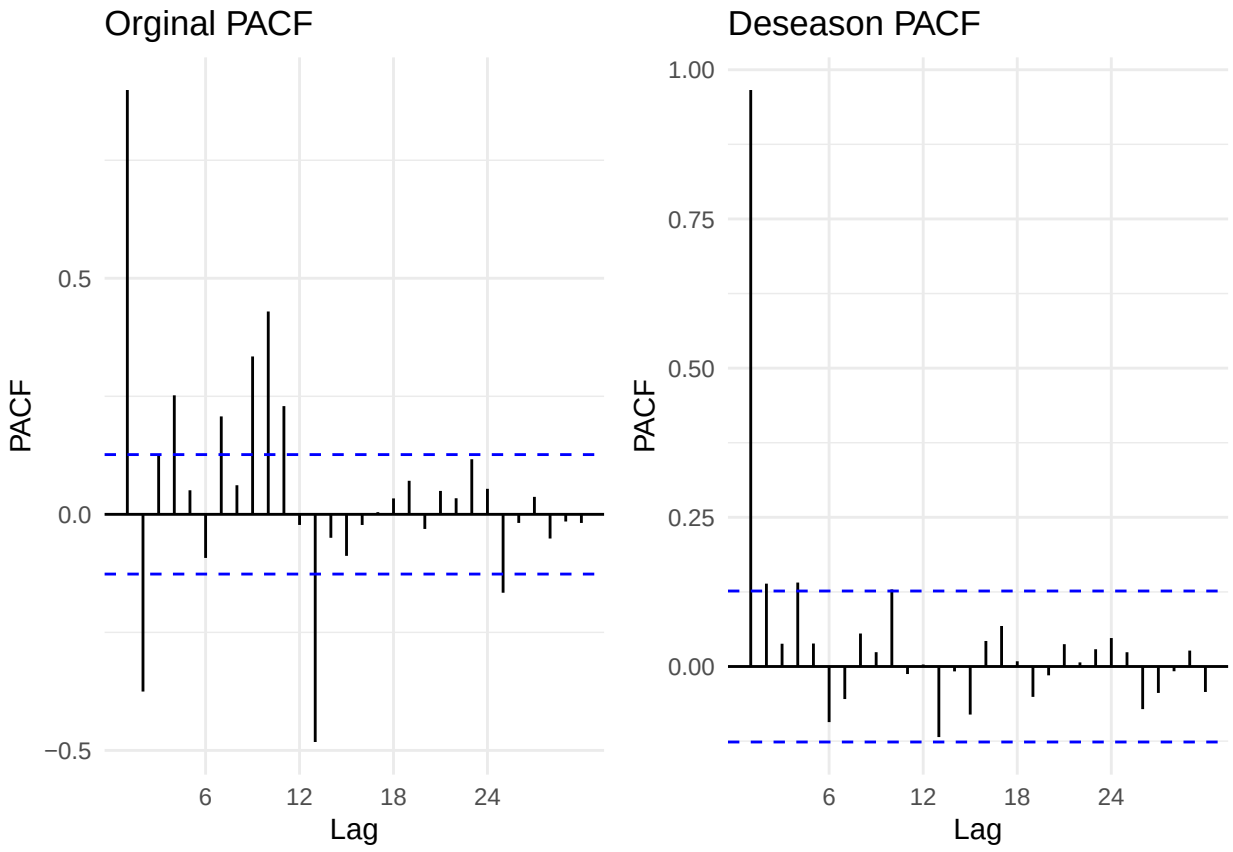
```
  autoplot(Acf(deseasonal_ts, lag = 30, plot=FALSE),
                  main = "Deseason ACF")+
                theme_minimal())
```



ACF plots

The original ACF shows periodic spikes and decays, showing a strong seasonal component. The deseasoned ACF shows a slow decay with no periodic spikes. The seasonality is lessened.

```
#Comparing PACFs
plot_grid(
  autoplot(Pacf(NG_ts, lag = 30, plot=FALSE),
                main = "Orginal PACF")+
                theme_minimal(),
  autoplot(Pacf(deseasonal_ts, lag = 30, plot=FALSE),
                  main = "Deseason PACF")+
                theme_minimal())
```



PACF plots

The original PACF shows some significant lags at regular intervals, suggesting a strong seasonal pattern. The deseason PACF shows less significant lags. indicating that the seasonality is greatly reduced.

Modeling the seasonally adjusted or deseasonalized series

Q3

Run the ADF test and Mann Kendall test on the deseasonalized data from Q2. Report and explain the results.

```
#Run ADF test
adf_result <- adf.test(deseasonal_ts)

# Run Mann-Kendall test
mk_result <- MannKendall(deseasonal_ts)

# Print results
print(adf_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  deseasonal_ts
## Dickey-Fuller = -4.0271, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
print(mk_result)
```

```
## tau = 0.843, 2-sided pvalue =< 2.22e-16
```

ADF test: The p-value is 0.01, less than 0.05. We can reject the null hypothesis and conclude that the series has a unit root. Therefore, the deseasoned series is stationary.

Mann Kendall test: The p-value is 2.22e-16, less than 0.05. We can reject the null hypothesis and conclude that there still has an upward trend and may need further detrending.

The two show contradictory results.

Q4

Using the plots from Q2 and test results from Q3 identify the ARIMA model parameters p, d and q. Note that in this case because you removed the seasonal component prior to identifying the model you don't need to worry about seasonal component. Clearly state your criteria and any additional function in R you might use. DO NOT use the *auto.arima*() function. You will be evaluated on ability to understand the ACF/PACF plots and interpret the test results.

```r
model_101 <- Arima(deseasonal_ts, order=c(1,0,1), include.drift=TRUE)
compare_aic <- data.frame(model_101$aic)
print(compare_aic)
```

```
##   model_101.aic
## 1      4792.202
```

```r
model_100 <- Arima(deseasonal_ts, order=c(1,0,0), include.drift=TRUE)
compare_aic <- data.frame(model_100$aic)
print(compare_aic)
```

```
##   model_100.aic
## 1      4790.217
```

```r
model_111 <- Arima(deseasonal_ts, order=c(1,1,1), include.drift=TRUE)
compare_aic <- data.frame(model_111$aic)
print(compare_aic)
```

```
##   model_111.aic
## 1      4774.213
```

p = 1. The PACF plot shows a significant spike at lag 1, followed by an immediate drop off, suggesting a pattern of AR(1) process.

d = 1. The previous tests shows cotradictory results which makes d a hard choice. If we only look at the ADF test, no differencing is needed. However, the MK test suggests the non-stationarity of the series.

q = 1. The ACF plot shows a gradual decay and the PACF cuts off sharply. This does not show a clear MA process. Therefore, I will assume the q value and test with AIC comparisons.

I used AIC to further assess the uncertain d and q values. ARMIA(1,1,1) shows the lowest AIC. Therefore, the overall model should be ARIMA(1,1,1).

Q5

Use `Arima()` from package "forecast" to fit an ARIMA model to your series considering the order estimated in Q4. You should allow constants in the model, i.e., `include.mean = TRUE` or `include.drift=TRUE`. **Print the coefficients** in your report. Hint: use the `cat()` or `print()` function to print.

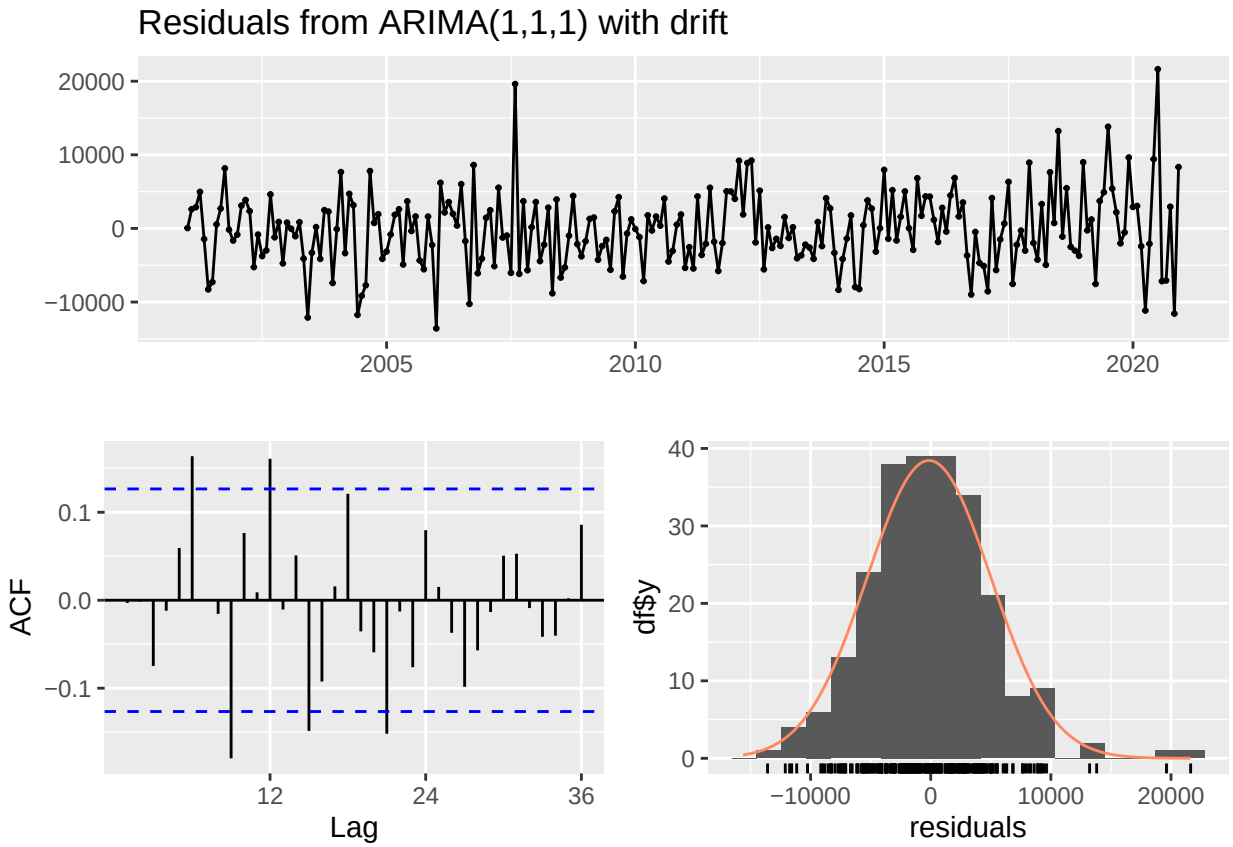
```r
arima_model <- Arima(deseasonal_ts, order=c(1,1,1), include.drift=TRUE)
print(arima_model[1])
```

```
## $coef
##         ar1         ma1       drift
##   0.7065237  -0.9794655 359.5051904
```

Q6

Now plot the residuals of the ARIMA fit from Q5 along with residuals ACF and PACF on the same window. You may use the *checkresiduals*() function to automatically generate the three plots. Do the residual series look like a white noise series? Why?

```r
# Check residuals
checkresiduals(arima_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1) with drift
## Q* = 48.356, df = 22, p-value = 0.0009736
##
## Model df: 2.   Total lags used: 24
```

> Yes, the residual series looks like a white noise series. The residual plot seems to fluctuate around zero with no clear pattern. In the ACF plot, most lags are within the confidence bands indicating that there is little autocorrelation despite a few spikes. The Q-Q plot generally shows a trend that follows normal distribution.

Modeling the original series (with seasonality)

Q7

Repeat Q3-Q6 for the original series (the complete series that has the seasonal component). Note that when you model the seasonal series, you need to specify the seasonal part of the ARIMA model as well, i.e., P, D and Q.

```
#Run ADF test
adf_result_og <- adf.test(NG_ts)

# Run Mann-Kendall test
mk_result_og <- SeasonalMannKendall(NG_ts)

# Print results
print(adf_result_og)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  NG_ts
## Dickey-Fuller = -8.9602, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
print(mk_result_og)
```

```
## tau = 0.887, 2-sided pvalue =< 2.22e-16
```

ADF test: The p-value is 0.01, less than 0.05. We can reject the null hypothesis and conclude that the series has a unit root. Therefore, the deseasoned series is stationary.

Mann Kendall test: The p-value is 2.22e-16, less than 0.05. We can reject the null hypothesis and conclude that there still has an upward trend and may need further detrending.

D = 1. Though the ADF and MK tests show contradictory results, the small p-value in Mann Kendall test suggests that seasonal differencing may still be necessary.

s = 12. This shows monthly seasonality.

P = 1. The PACF suggests a seasonal autoregressive component at lag 12, meaning P = 1.

Q = 1. The PACF suggests a seasonal autoregressive component at lag 12, meaning Q = 1.

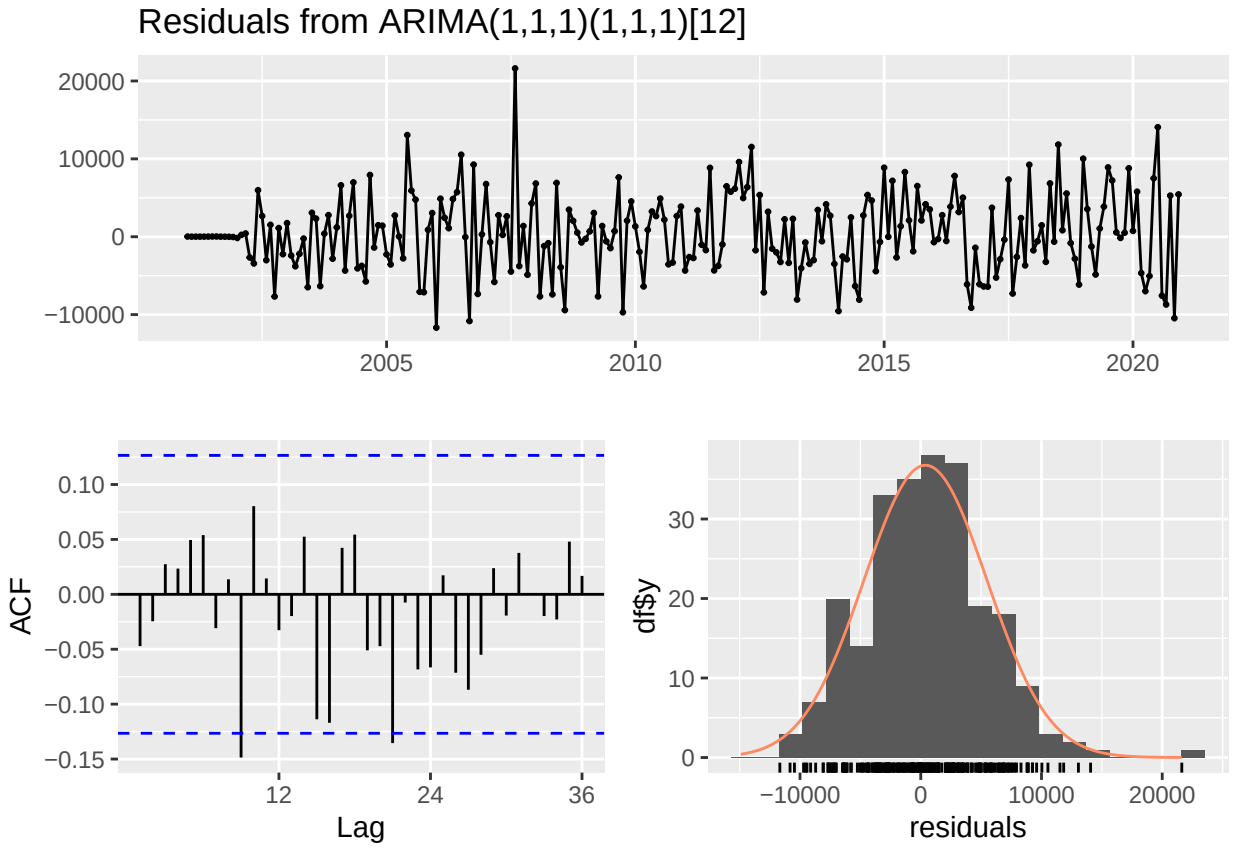
p,d,q remains the same as previouly stated due to explained reasons.

Overall, the model is ARIMA(1,1,1)(1,1,1)[12]

```
arima_model_og <- Arima(NG_ts, order=c(1,1,1), seasonal = c(1,1,1), include.drift=TRUE)
print(arima_model_og[1])
```

```
## $coef
##        ar1        ma1       sar1       sma1
##  0.7328185 -0.9818820 -0.0202173 -0.6908948
```

```
# Check residuals
checkresiduals(arima_model_og)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(1,1,1)[12]
## Q* = 27.654, df = 20, p-value = 0.1179
##
## Model df: 4.   Total lags used: 24
```

> The residual series looks like a white noise series. The residual plot seems to fluctuate around zero with no clear pattern. In the ACF plot, most lags (except for two) are within the confidence bands indicating that there is little autocorrelation – less than Q6. The Q-Q plot generally showsa clearer trend that follows normal distribution.

Q8

Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response.

> The ARIMA model in Q7 is better representing the Natural Gas Series. The residual ACF shows lags that are closer to zero, and the Q-Q plot fits the normal distribution better. Q7's residuals appear to have less visible autocorrelation. Therefore, the residuals in Q7 are closer to white noise, and Q7 presents a better model. This is not a fair comparison because this dataset exhibits a clear seasonality, but we are comparing a non-seasonal model with seasonal model.

Checking your model with the auto.arima()

Please do not change your answers for Q4 and Q7 after you ran the *auto.arima*(). It is **ok** if you didn't get all orders correctly. You will not loose points for not having the same order as the *auto.arima*().

Q9

Use the *auto.arima*() command on the **deseasonalized series** to let R choose the model parameter for you. What's the order of the best ARIMA model? Does it match what you specified in Q4?

```
#Auto ARIMA
arima_autofit <- auto.arima(deseasonal_ts)
print(arima_autofit)
```

```
## Series: deseasonal_ts
## ARIMA(1,1,1) with drift
##
## Coefficients:
##          ar1      ma1     drift
##       0.7065  -0.9795  359.5052
## s.e.  0.0633   0.0326   29.5277
##
## sigma^2 = 26980609:  log likelihood = -2383.11
## AIC=4774.21   AICc=4774.38   BIC=4788.12
```

The order of the best ARIMA model is ARIMA(1,1,1). This matches with my previous answer.

Q10

Use the *auto.arima*() command on the **original series** to let R choose the model parameters for you. Does it match what you specified in Q7?

```
arima_autofit_og <- auto.arima(NG_ts)
print(arima_autofit_og)
```

```
## Series: NG_ts
## ARIMA(1,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     sma1     drift
##       0.7416  -0.7026  358.7988
## s.e.  0.0442   0.0557   37.5875
##
## sigma^2 = 27569124:  log likelihood = -2279.54
## AIC=4567.08   AICc=4567.26   BIC=4580.8
```

The best model chosen is ARIMA(1,0,0)(0,1,1)[12], which does not match with my answer in Q7.