ENV 790.30 - Time Series Analysis for Energy Data | Spring 2023

Assignment 7 - Due date 03/20/23

Katherine Burley

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_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.

Set up

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

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

```
library(tseries)

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --
```

```
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(readxl)
library(Kendall)
library(lubridate)
```

```
##
```

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

Importing and processing the data set

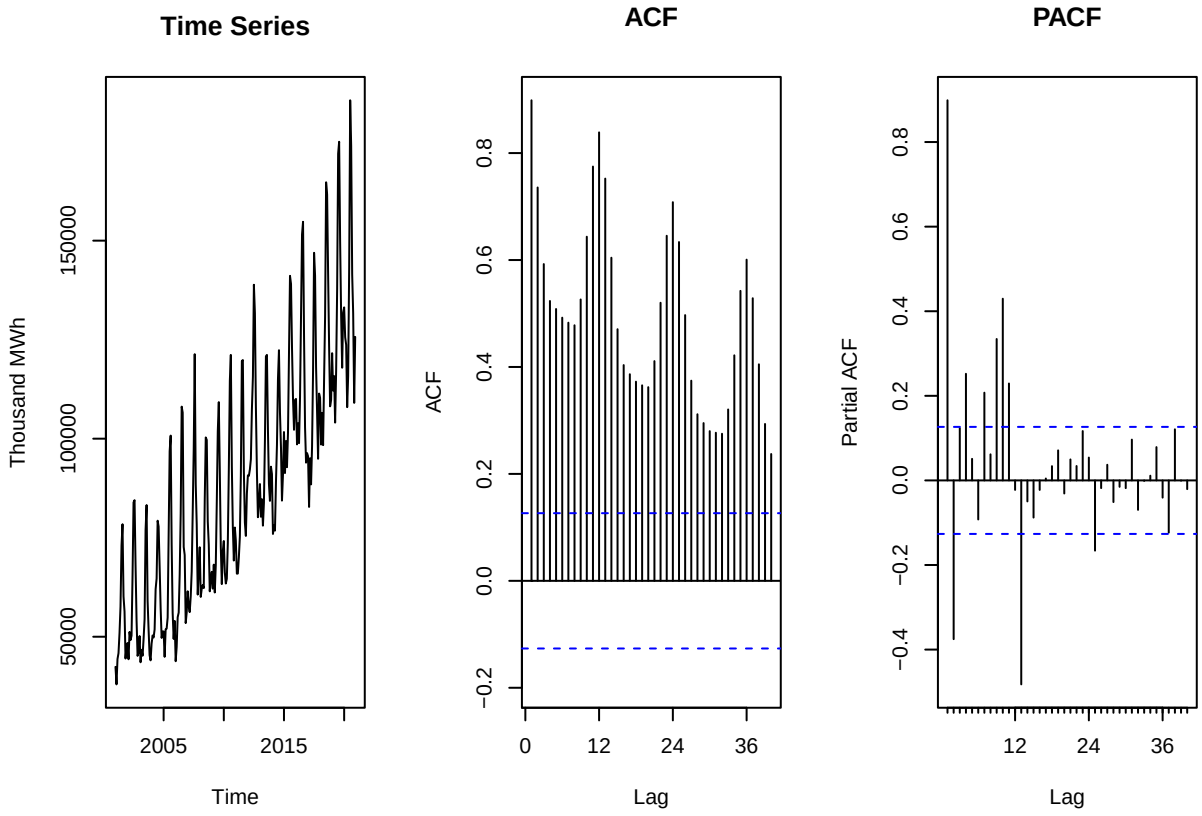
Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

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

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.

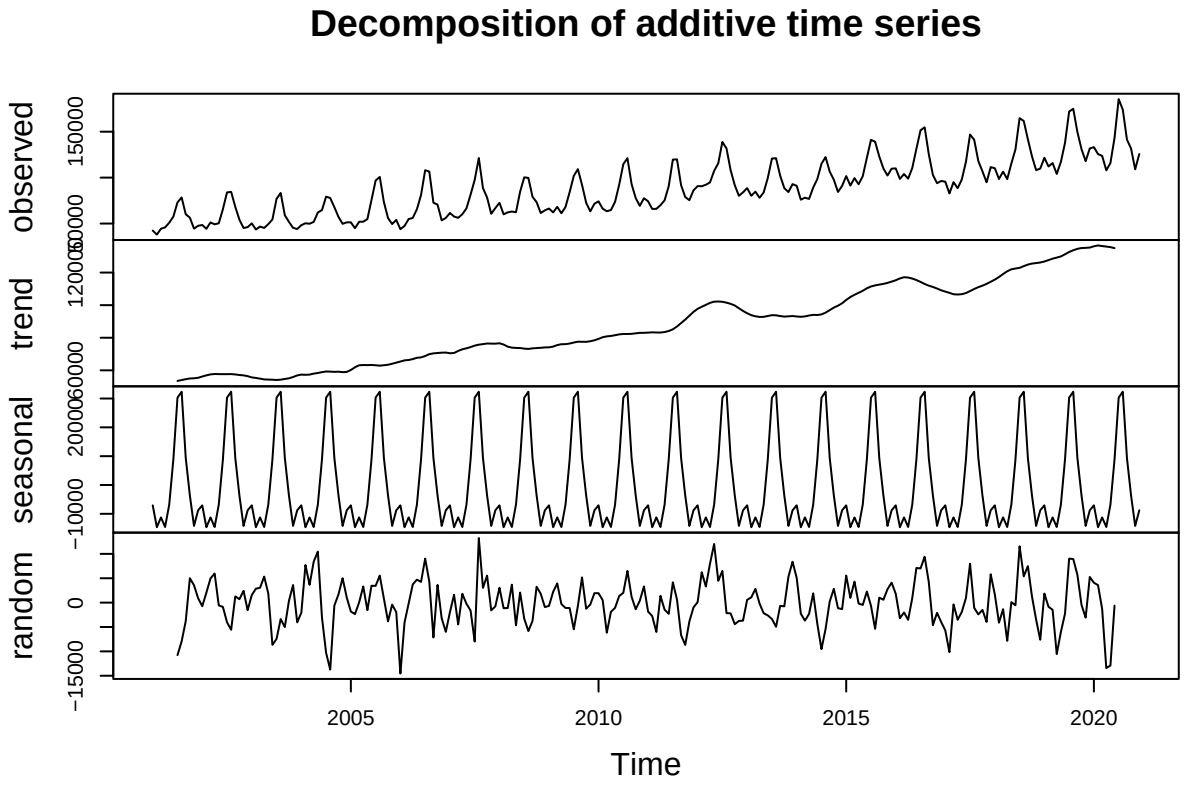
```
#Importing data set - using xlsx package
energy_data <- read.csv("../Data/Net_generation_United_States_all_sectors_monthly.csv",
                        header=TRUE, skip=4)
gas_data <- energy_data %>%
  select(Month, natural.gas.thousand.megawatthours) %>%
  mutate(Date = my(Month)) %>%
  arrange(Date) # Sort by month

ts_gas <- ts(gas_data$natural.gas.thousand.megawatthours, start=2001, frequency=12)

# Plot
par(mfrow=c(1,3))
plot(ts_gas, main="Time Series", ylab = "Thousand MWh")
Acf(ts_gas,lag.max=40, main="ACF")
Pacf(ts_gas,lag.max=40, main="PACF")
```



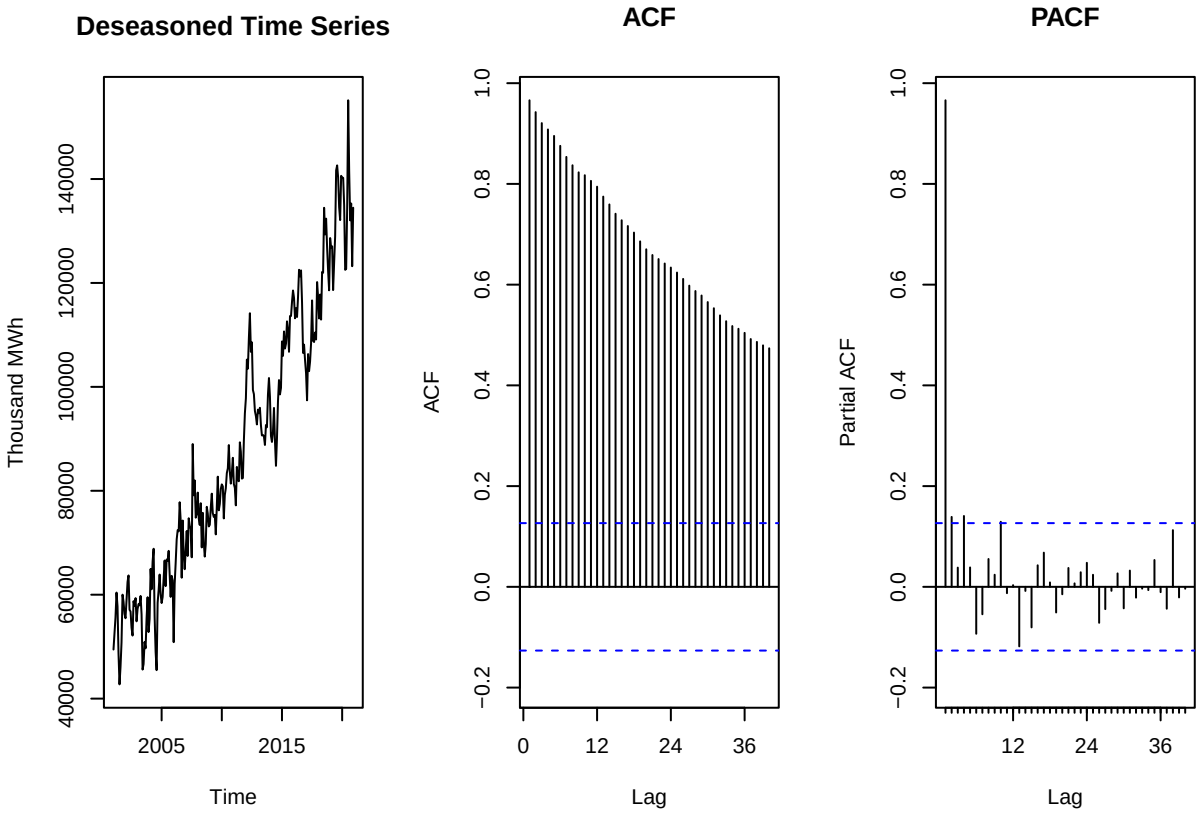
Q2

Using the *decompose*() or *stl*() 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.

```
# Deseason Data
decompose_gas <- decompose(ts_gas, type="additive")
plot(decompose_gas)
```




```r
deseasonal_gas <- seasadj(decompose_gas)

# Plot
# par(mar=c(3,3,3,0));par(mfrow=c(1,3))
par(mfrow=c(1,3))
plot(deseasonal_gas, main="Deseasoned Time Series", ylab = "Thousand MWh")
Acf(deseasonal_gas,lag.max=40, main="ACF")
Pacf(deseasonal_gas,lag.max=40, main="PACF")
```



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.

```
# ADF Test
print("Results for ADF Test on Annual Data")
```

```
## [1] "Results for ADF Test on Annual Data"
```

```
print(adf.test(deseasonal_gas, alternative = "stationary"))
```

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

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

Reject the null hypothesis (at 95% confidence level) that model has a unit root, suggests that series is stationary and does not have a stochastic trend. Although note that with this p-value, we would not reject the null hypothesis at the 99% confidence level, so this test is not very conclusive.

```r
# Mann Kendall
MKtest <- MannKendall(deseasonal_gas)
print("Results for Mann Kendall on Annual Data")
```

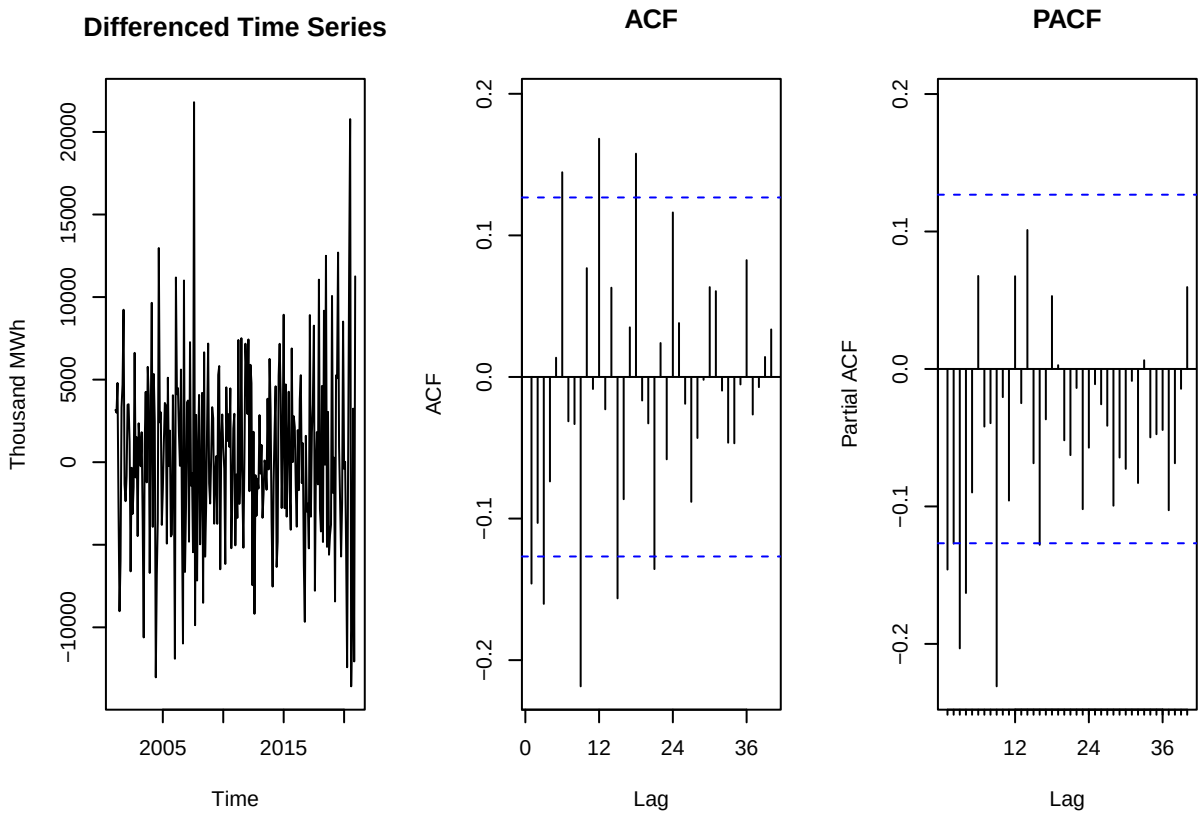
```
## [1] "Results for Mann Kendall on Annual Data"
```

```r
print(summary(MKtest))
```

```
## Score =  24186 , Var(Score) = 1545533
## denominator =  28680
## tau = 0.843, 2-sided pvalue =< 2.22e-16
## NULL
```

We reject the null hypothesis that series is stationary, suggests that the series has a trend. The ACF plot also has a slow decay which provides further evidence for non-stationarity.

```r
# Perform Additional Checks on Differenced Series
deseasonal_gas_diff <- diff(deseasonal_gas)

par(mfrow=c(1,3))
plot(deseasonal_gas_diff, main="Differenced Time Series", ylab = "Thousand MWh")
Acf(deseasonal_gas_diff,lag.max=40, main="ACF")
Pacf(deseasonal_gas_diff,lag.max=40, main="PACF")
```




```r
print(adf.test(deseasonal_gas_diff, alternative = "stationary")) # Reject null hypothesis
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  deseasonal_gas_diff
## Dickey-Fuller = -6.9137, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
# MKtest <- MannKendall(deseasonal_gas)
print(summary(MannKendall(deseasonal_gas_diff))) # Do not reject null that series is stationary. Indica
```

```
## Score =  -299 , Var(Score) = 1526334
## denominator =  28441
## tau = -0.0105, 2-sided pvalue =0.80939
## NULL
```

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 can read the plots and interpret the test results.

> Answer: ARIMA(1,1,1) p=1: In the differened and deseasoned plot, the PACF function has a significant value at lag 1, but not significant at lag 2, indicating that we should include 1 AR component. d=1: the Mann Kendall test and slow decay of the ACF function for the deseasoned model indicate that there is a remaining trend and the series should be differenced. Even though the ADF test did not support this conclusion, the p-value was close to the border of significance. q=1: In the differened and deseasoned plot, the ACF function has a significant value at lag 1, but not significant at lag 2, indicating that we should include 1 MA component.

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*() function to print.

```
# Run the model
model = Arima(deseasonal_gas, order=c(1,1,1), include.drift=TRUE, include.mean=TRUE)
print(model)
```

```
## Series: deseasonal_gas
## 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
```

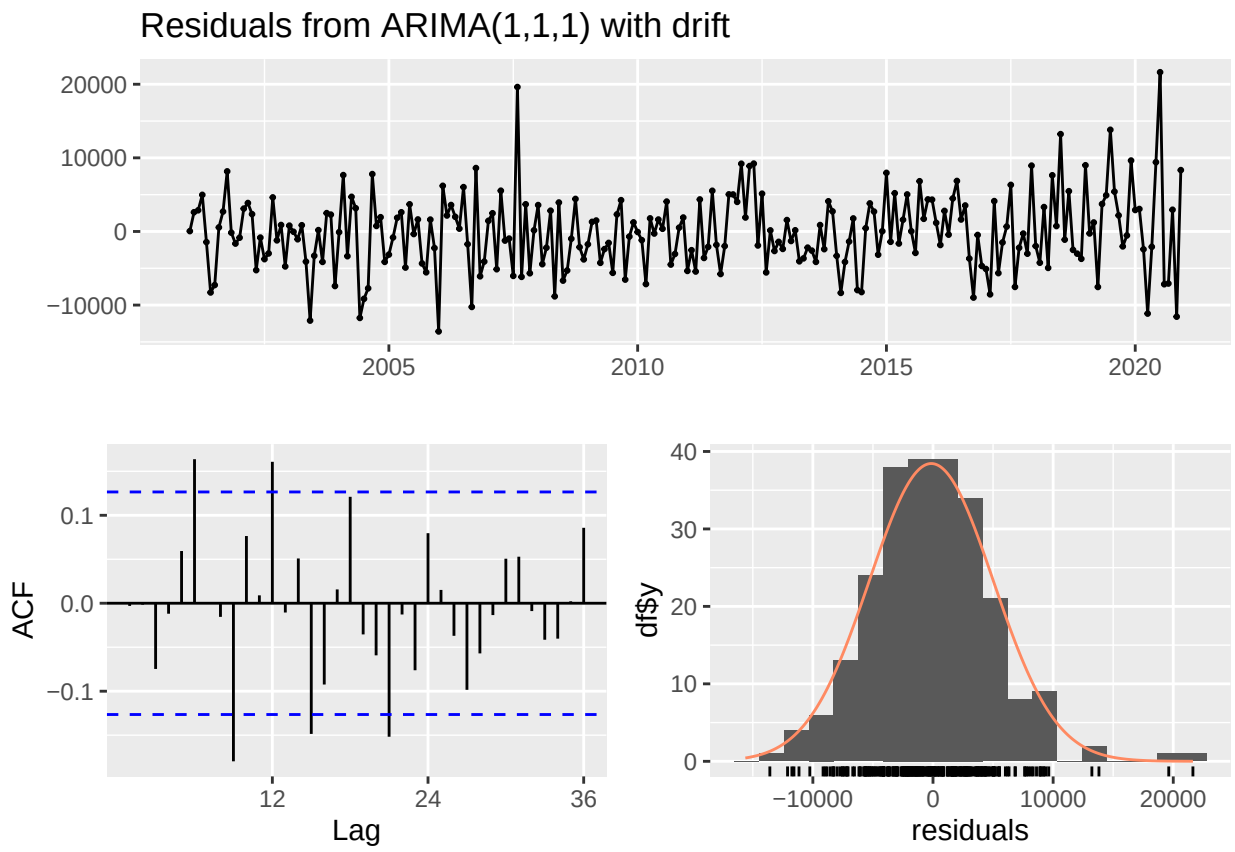
```
# Print the coefficients
cat(model$coef)
```

```
## 0.7065237 -0.9794655 359.5052
```

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?

```
checkresiduals(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
```

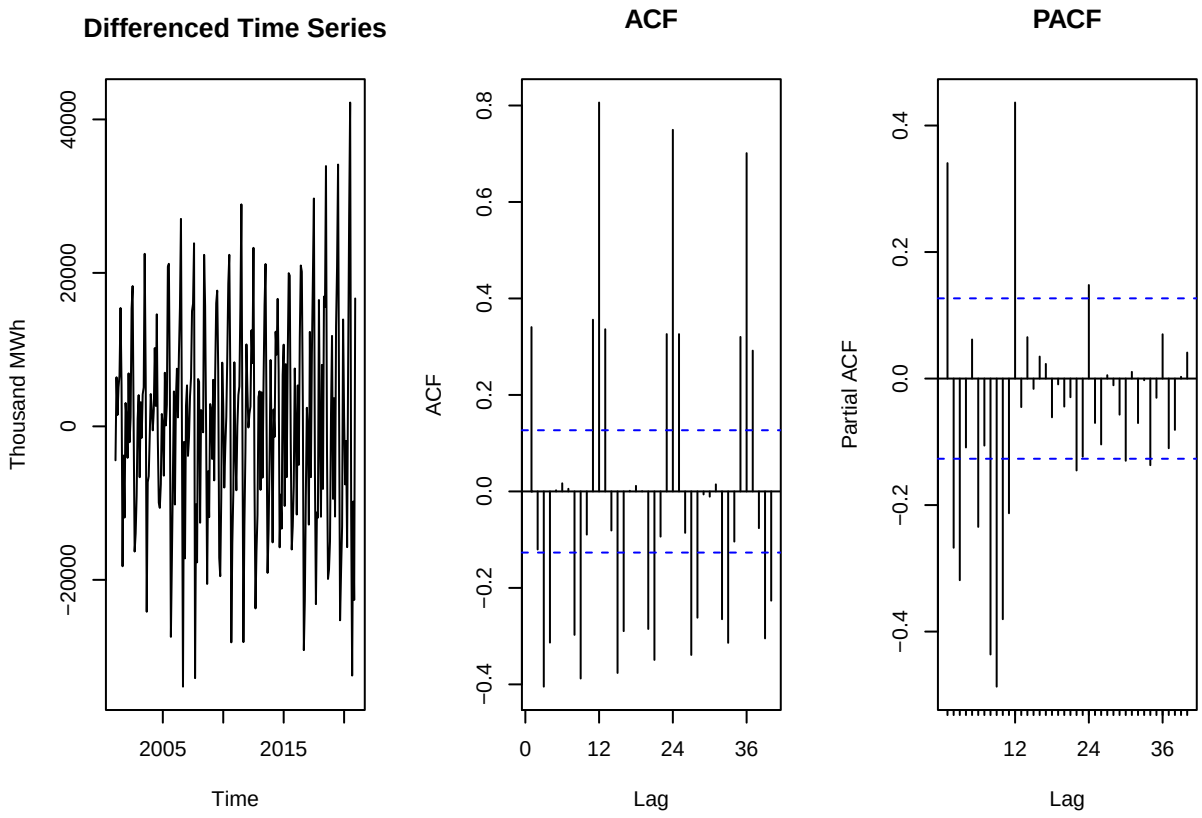
The series looks close to a white noise, because it is mostly fluctuating around 0, but there are a couple high value spikes in the mid-2000s and after 2020 that make it deviate from a white noise series. The ACF also has a few lags with significant values that appear to occur at regular intervals and fluctuate from positive significant values to negative significant values.

Modeling the original series (with seasonality)

Q7

Repeat Q4-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.

```r
# Perform Additional Checks on Differenced, Not Deseasoned Series
gas_diff <- diff(ts_gas)

par(mfrow=c(1,3))
plot(gas_diff, main="Differenced Time Series", ylab = "Thousand MWh")
Acf(gas_diff,lag.max=40, main="ACF")
Pacf(gas_diff,lag.max=40, main="PACF")
```




```r
print(adf.test(gas_diff, alternative = "stationary")) # Reject null hypothesis
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gas_diff
## Dickey-Fuller = -8.6642, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
# MKtest <- MannKendall(deseasonal_gas)
print(summary(MannKendall(gas_diff))) # Do not reject null that series is stationary. Indicates we shou
```

```
## Score =  -295 , Var(Score) = 1526334
## denominator =  28441
## tau = -0.0104, 2-sided pvalue =0.8119
## NULL
```

```r
nsdiffs(ts_gas) # suggests 1 difference
```

```
## [1] 1
```

Specify the model: > Answer: ARIMA(1,1,1)_(0,1,1)_12 p=1: In the differened and deseasoned plot, the PACF function has a significant value at lag 1, but not significant at lag 2, indicating that we should include 1 AR component. d=1: the Mann Kendall test and slow decay of the ACF function for the deseasoned model indicate that there is a remaining deterministic trend and the series should be differenced. q=1: In the differened and deseasoned plot, the ACF function has a significant value at lag 1, but not significant at lag 2, indicating that we should include 1 MA component. P=1: Because there are multiple spikes in the ACF at regular intervals and a single, positive spike in the PACF at lag 12, this indicates a SAR process. In the PACF, the seasonal spikes in the differenced series are not as clear. D=1: In the differenced original series, the seasonal pattern appears to be strong and stable over time. Q=0: Because there are multiple spikes in the ACF at regular intervals and a single, positive spike in the PACF at lag 12, this indicates a SAR process. In the PACF, the seasonal spikes in the differenced series are not as clear.Because P+Q<= 1, we know that Q=0 s=12: The positive spikes in the differenced original series occur every 12 lags.

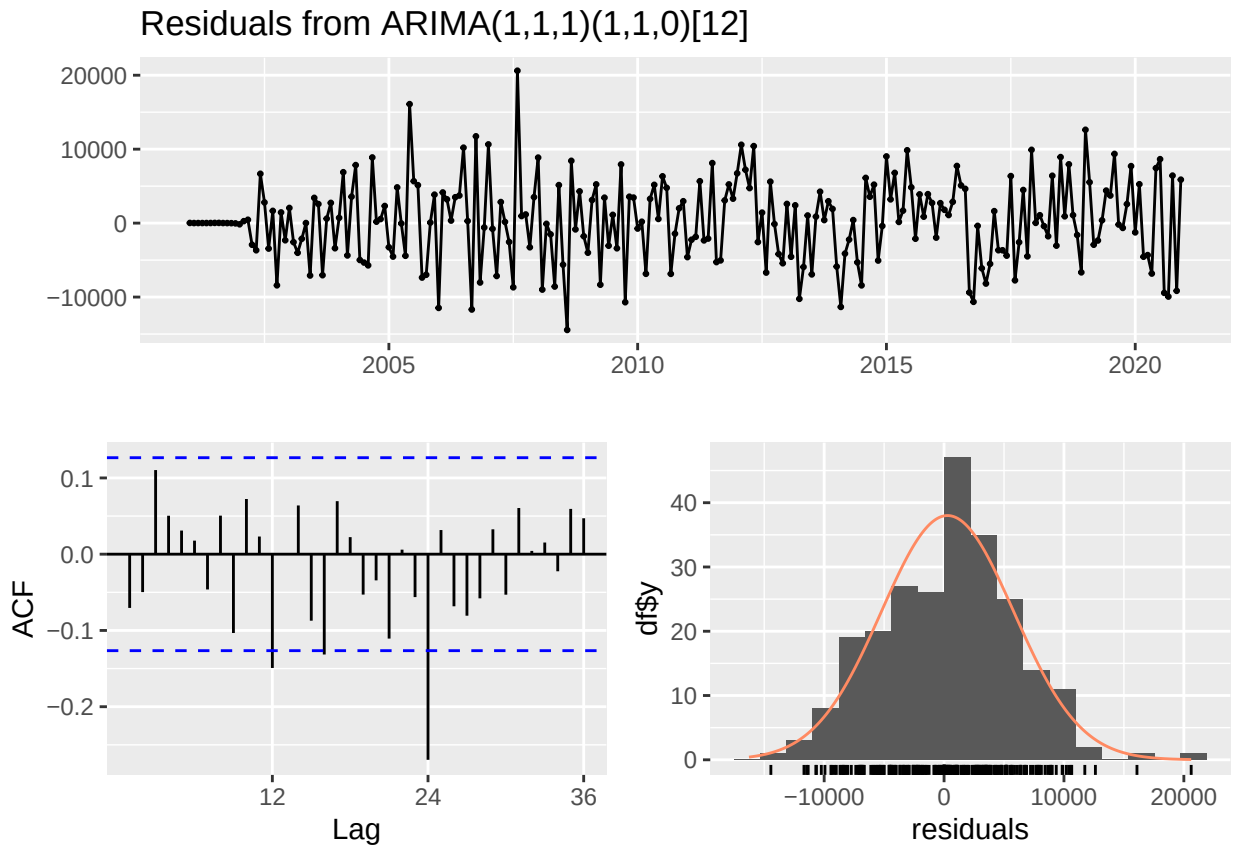
```r
# Run Model
model_orig <- Arima(ts_gas, order=c(1,1,1), seasonal=c(1,1,0), include.drift=FALSE)
print(model_orig)
```

```
## Series: ts_gas
## ARIMA(1,1,1)(1,1,0)[12]
##
## Coefficients:
##          ar1      ma1     sar1
##       0.7722  -1.0000  -0.4526
## s.e.  0.0432   0.0213   0.0595
##
## sigma^2 = 32606982:  log likelihood = -2287.56
## AIC=4583.12   AICc=4583.3   BIC=4596.82
```

```r
# Print the coefficients
cat(model_orig$coef)
```

```
## 0.7722449 -0.9999998 -0.4526265
```

```r
# Check Residuals
checkresiduals(model_orig)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(1,1,0)[12]
## Q* = 50.306, df = 21, p-value = 0.0003307
##
## Model df: 3.   Total lags used: 24
```

This series looks more like a white noise than the first model on the deseasoned data, although there is one significant spike in the ACF at lag 24.

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.

```
print(model)
```

```
## Series: deseasonal_gas
## 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
```

```
print(model_orig)
```

```
## Series: ts_gas
## ARIMA(1,1,1)(1,1,0)[12]
##
## Coefficients:
##          ar1      ma1     sar1
##       0.7722  -1.0000  -0.4526
## s.e.  0.0432   0.0213   0.0595
##
## sigma^2 = 32606982:  log likelihood = -2287.56
## AIC=4583.12   AICc=4583.3   BIC=4596.82
```

The SARIMA model on the original series has a lower AIC than the ARIMA model for the deseasoned series, however these are not exactly comparable since we are running the models on different series. We would be able to compare performance with AIC if we were running multiple models on the same series - for example, if we tried an ARIMA(1,2,1) on the deseasoned series to compare to the ARIMA(1,1,1) that we used here.

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(deseasonal_gas)
```

```
## Series: deseasonal_gas
## 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
```

Yes! It does match the model I specified exactly. The auto.arima tells us that the best order of the model for this series is ARIMA(1,1,1) with drift.

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?

```
auto.arima(ts_gas)
```

```
## Series: ts_gas
## 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
```

Auto arima shows that the best model for the original series is ARIMA(1,0,0)(0,1,1) with drift. This does not match my model, which was ARIMA(1,1,1)(1,1,0) without drift. I think part of the issue was that I assumed the non-seasonal part of the model would remain the same, even though I now see that I should have tested the parameters for the non-seasonal part of the model on the original series as well, rather than carrying over the parameters from the model on the deseasoned series. I also chose to set include.drift=FALSE because I had specified d=1 and D=1.