ENV 790.30 - Time Series Analysis for Energy Data | Spring 2022

Assignment 7 - Due date 03/25/22

Yared S. Asfaw

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 project open the first thing you will do is change "Student Name" on line 3 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. Rename the pdf file such that it includes your first and last name (e.g., "LuanaLima_TSA_A07_Sp22.Rmd"). Submit this pdf using Sakai.

Set up

```
#Loading/installing required package
library(forecast)
library(tseries)
library(lubridate)
library(tidyverse)
library(dplyr)
library(Kendall)
library(outliers)
```

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.

```r
# Importing the dataset

net_gen_all_sectors <- read.csv('./Data/Net_generation_United_States_all_sectors_monthly.csv',
skip= 4, header = TRUE, sep = ",", stringsAsFactors = TRUE)

# Formatting the date column to "Date"
net_gen_all_sectors$Month <- my(net_gen_all_sectors$Month)
# Selecting the natural gas column only
net_gen_natural_gas <- net_gen_all_sectors %>%
  select(Month, natural.gas.thousand.megawatthours)
head(net_gen_natural_gas)
```

```
##        Month natural.gas.thousand.megawatthours
## 1 2020-12-01                            127685.1
## 2 2020-11-01                            109657.9
## 3 2020-10-01                            131241.6
## 4 2020-09-01                            141163.7
## 5 2020-08-01                            173389.8
## 6 2020-07-01                            181259.7
```

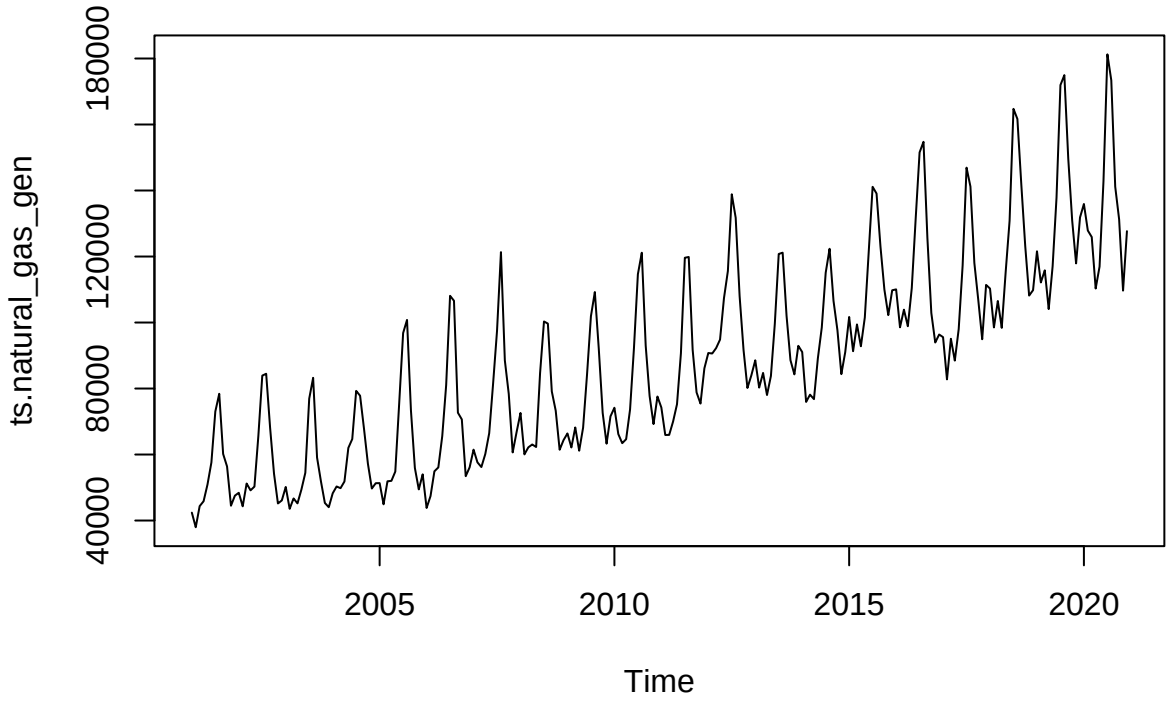
```r
# Re-ordering the date column in ascending order
net_gen_natural_gas <- net_gen_natural_gas[order(net_gen_natural_gas$Month),]
head(net_gen_natural_gas)
```

```
##          Month natural.gas.thousand.megawatthours
## 240 2001-01-01                            42388.66
## 239 2001-02-01                            37966.93
## 238 2001-03-01                            44364.41
## 237 2001-04-01                            45842.75
## 236 2001-05-01                            50934.21
## 235 2001-06-01                            57603.15
```

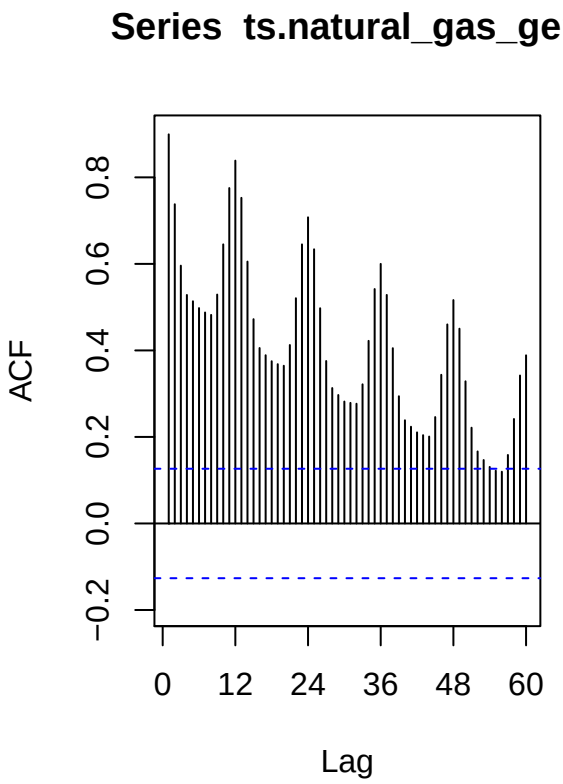
```r
# Renaming column names
colnames(net_gen_natural_gas) <- c("Date", "Natural_gas_gen_MWH")
class(net_gen_natural_gas$Date)
```

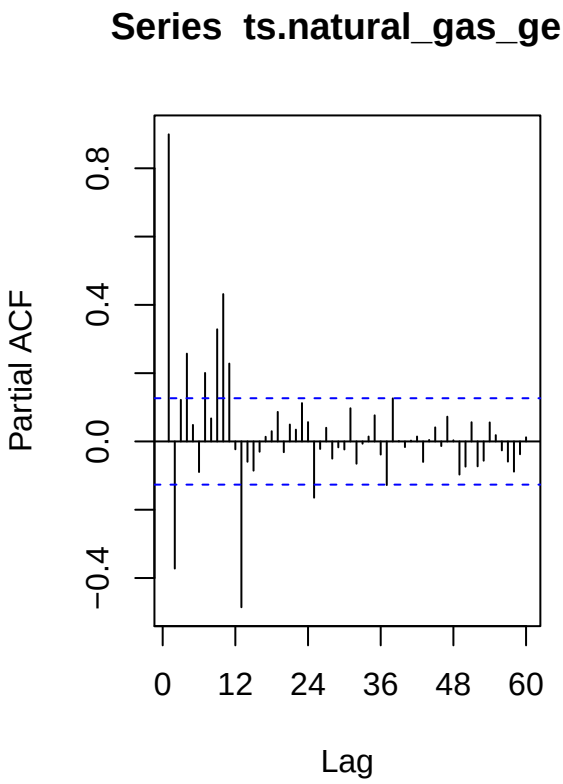
```
## [1] "Date"
```

```r
# Creating the time series object
ts.natural_gas_gen <- ts(net_gen_natural_gas$Natural_gas_gen_MWH, start = c(2001,01), frequency = 12)
# Plotting the time series object and the acf and pacf plots
plot(ts.natural_gas_gen)
```

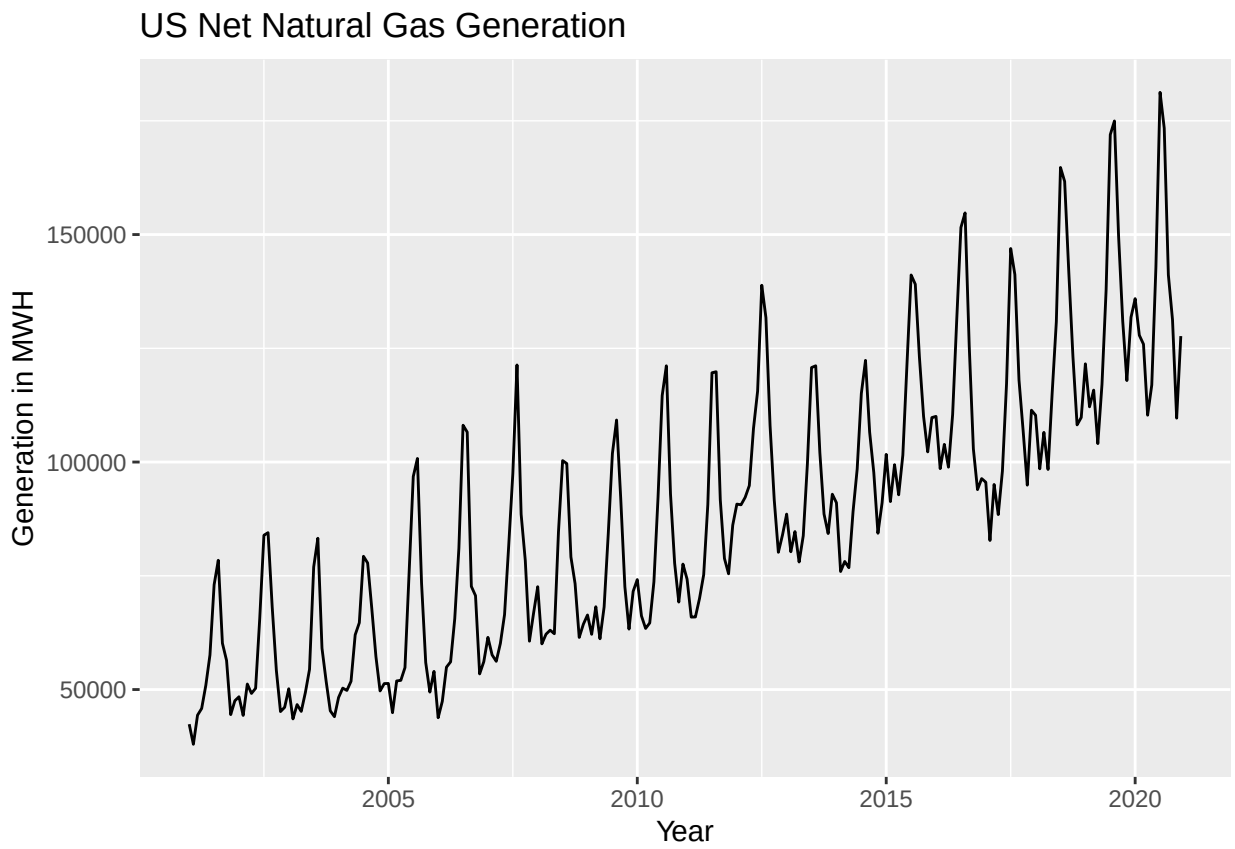



```
par(mfrow=c(1,2))
Acf(ts.natural_gas_gen, lag.max = 60)
Pacf(ts.natural_gas_gen, lag.max = 60)
```





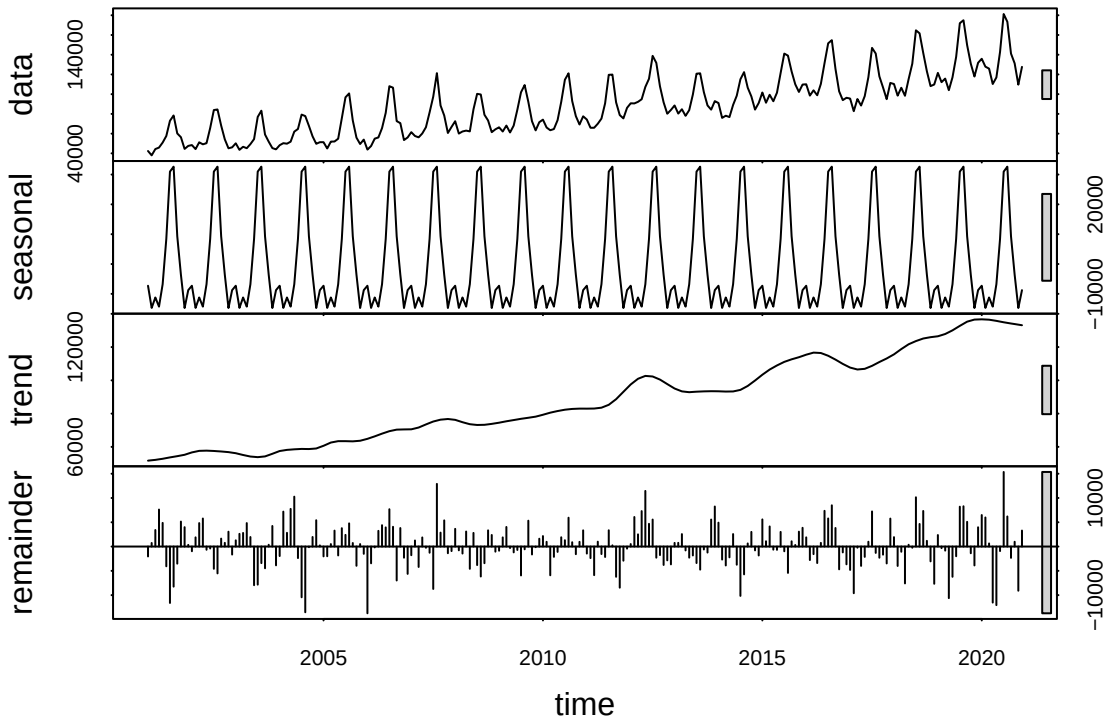

```
# Plotting using autoplot

autoplot(ts.natural_gas_gen) +
    ylab("Generation in MWH") +
    xlab("Year") +
    labs(title = "US Net Natural Gas Generation")
```



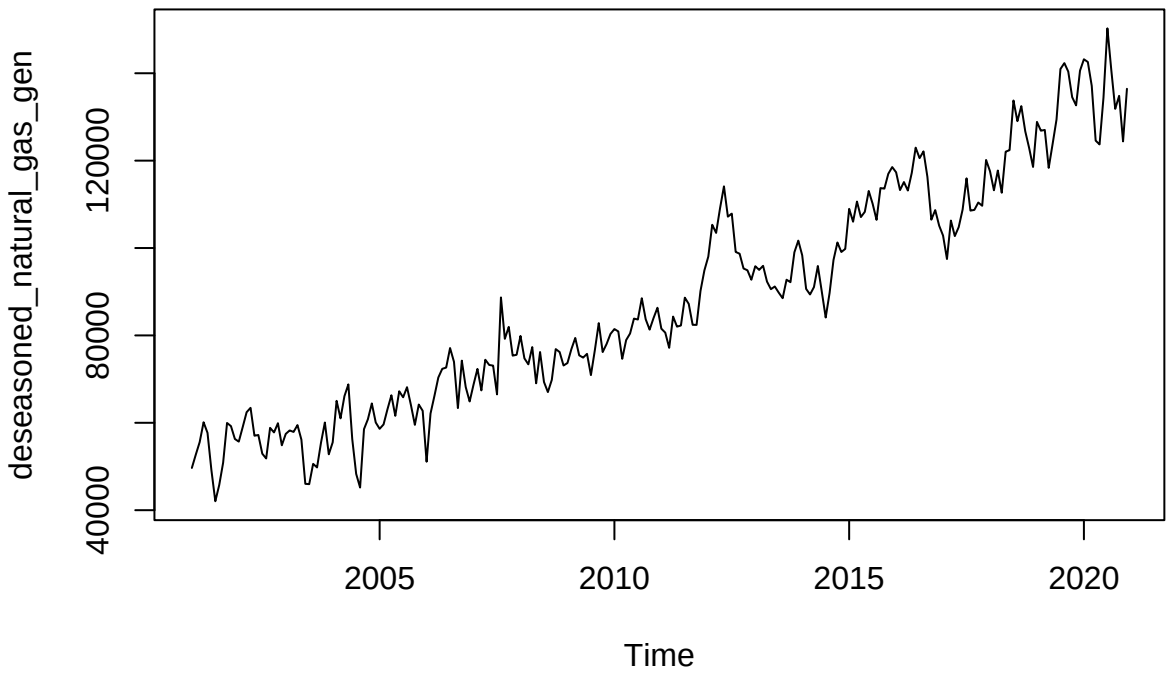
Q2

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.

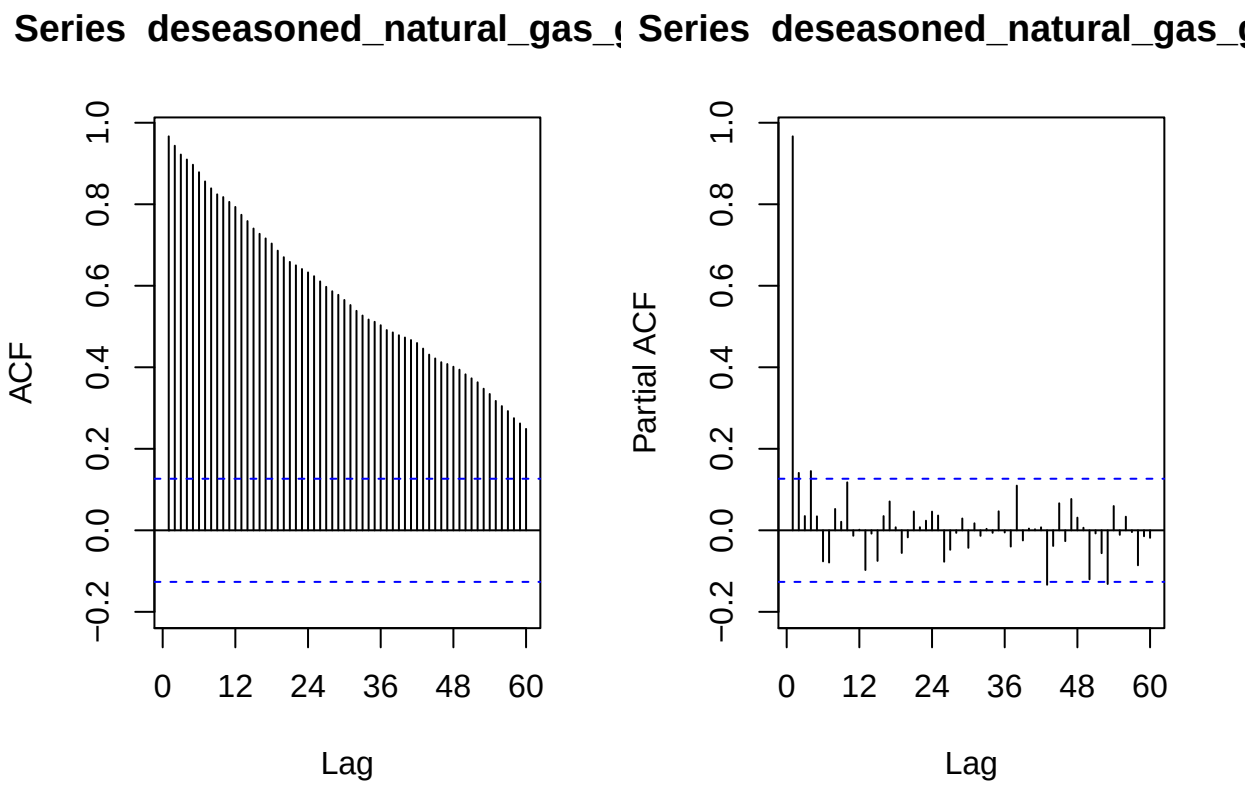
```
# Decomposing the time series object into its components
ts.natural_gas_gen_dec <- stl(ts.natural_gas_gen, s.window = "periodic")
plot(ts.natural_gas_gen_dec)
```




```
# Removing the seasonal component from the decomposed components using seasadj function

deseasoned_natural_gas_gen <- seasadj(ts.natural_gas_gen_dec, "Multiplicative")
plot(deseasoned_natural_gas_gen)
```




```
par(mfrow=c(1,2))
Acf(deseasoned_natural_gas_gen, lag.max = 60)
Pacf(deseasoned_natural_gas_gen, lag.max = 60)
```



In Q1, the plot show an increasing trend and a clear seasonal pattern whereas in Q2 where the seasonal component of the series is removed, the increasing trend still exists but the seasonal pattern is no more availble. Similarly, in the ACF plot of Q1, the seasonal pattern is clearly indicated with a slight decreasing in significance of spikes over time but in ACF plot of the deseasoned series, there is no seasonal pattern but only decreasing trend is observed. In the PACF plot of Q1, there are some significant spikes around lag 1, the most significant spike being at lag 1. On the other hand, for the deseasoned series in Q2, there is only one significant spike at lag 1.

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.

```
# Conducting the ADF and Mann Kendall tests on the deseasoned series

MKtest <- MannKendall(deseasoned_natural_gas_gen)
print(MKtest)
```

```
## tau = 0.844, 2-sided pvalue =< 2.22e-16
```

In the Mann Kendall test, the p-value is smaller than the significance level, thus, we reject the null hypothesis that says the series is stationary. Therefore, the series has a deterministic monotonic increasing trend (tau = 0.844) over time.

```
ADFtest <- adf.test(deseasoned_natural_gas_gen)
print(ADFtest)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  deseasoned_natural_gas_gen
## Dickey-Fuller = -4.0116, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

In the ADF test, the p-value is smaller than the significance level, thus we reject the null hypothesis and say that the series has no stochastic trend or it has no unit root.

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.

> From the ADF and Mann Kendall tests, we found out that the series follow a deterministic trend, thus, at least one round differencing is needed to make the series stationary. For this reason, the number of differences for the non-seasonal component,d, will have a value of 1, i.e. d=1. From the ACF and PACF plots of the deseasonalized series, it can be seen that the ACF plot show a decreasing trend that fades over time, and in the PACF plot, there is one significant spike at lag 1 that gives a cut off point. These features are the characterisitcs of an AR process. Therefore, the values for the parameters p and q are 1 and 0 respectively, i.e. p=1 and q=0.
>
> Hence, the ARIMA model parameters are p=1, d=1, and q=0

Q5

Use *Arima*() from package "forecast" to fit an ARIMA model to your series considering the order estimated in Q4. Should you allow for 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.

```
# Fitting an ARIMA model to the series with order (4,1,0)
# Since the series has a trend it needs to be differenced.
# Checking for the number of differences needed
n_diff <- ndiffs(deseasoned_natural_gas_gen, alpha = 0.1, test = c("kpss", "adf", "pp"), max.d = 2)
cat("Number of differencing needed: ",n_diff)
```

```
## Number of differencing needed:  1
```

```r
# Differencing the series once, at lag 1 to remove the trend.
diff_deseasoned_natural_gas_gen <- diff(deseasoned_natural_gas_gen, differences=1,lag=1)
# Fiting an arima model to the differenced series
arima_model <- Arima(diff_deseasoned_natural_gas_gen, order = c(1,1,0),
include.mean = FALSE, include.drift = TRUE)
print(arima_model)
```

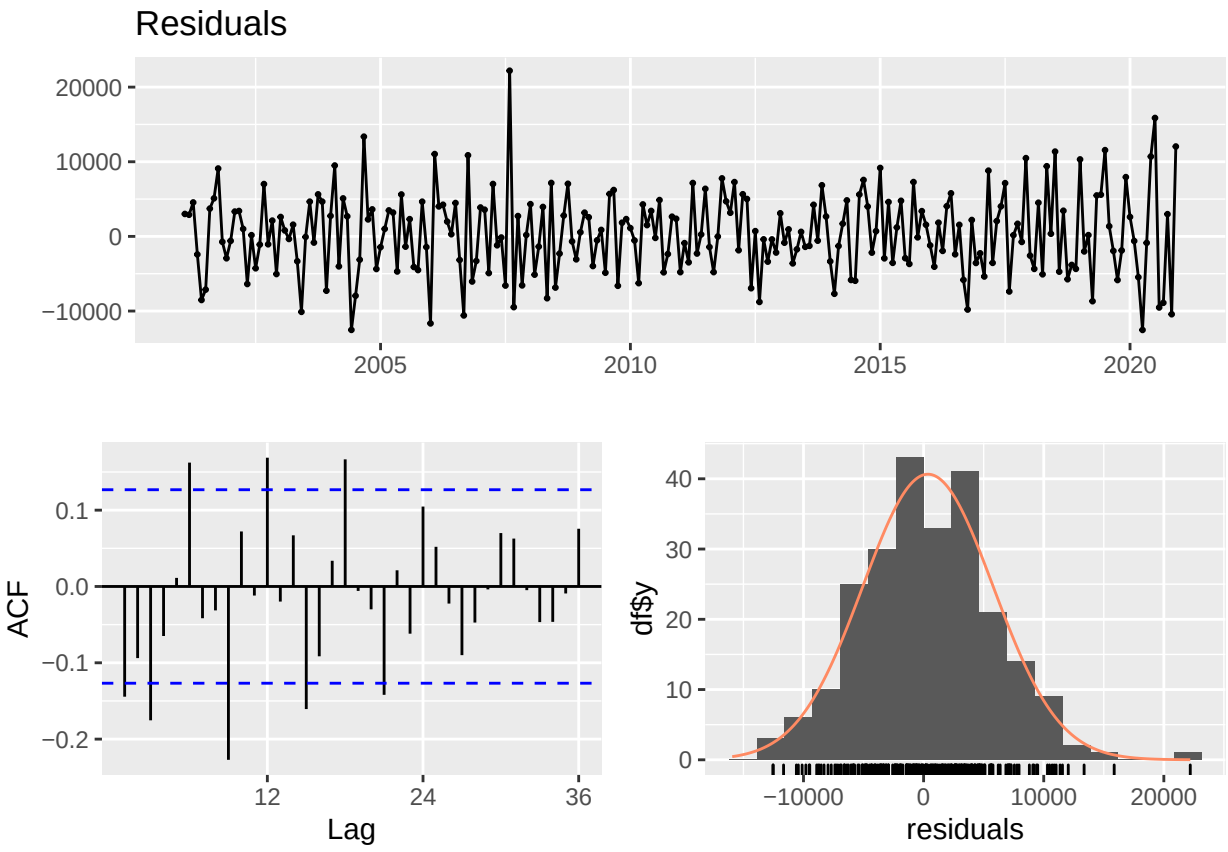
```
## Series: diff_deseasoned_natural_gas_gen
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1     drift
##       -0.5247    5.6157
## s.e.   0.0561  297.7968
##
## sigma^2 = 49336085:  log likelihood = -2444.85
## AIC=4895.7   AICc=4895.8   BIC=4906.12
```

In fitting the series to the arima model, the constant "include.mean" should be false as we have differenced the seires, and thus, the series will have a mean of zero. On the other hand, the "include.drift" should be set true to allow and see small deviation from zero.

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
# Plotting the residual series of the ARIMA fit
checkresiduals(diff_deseasoned_natural_gas_gen, lag=12)
```



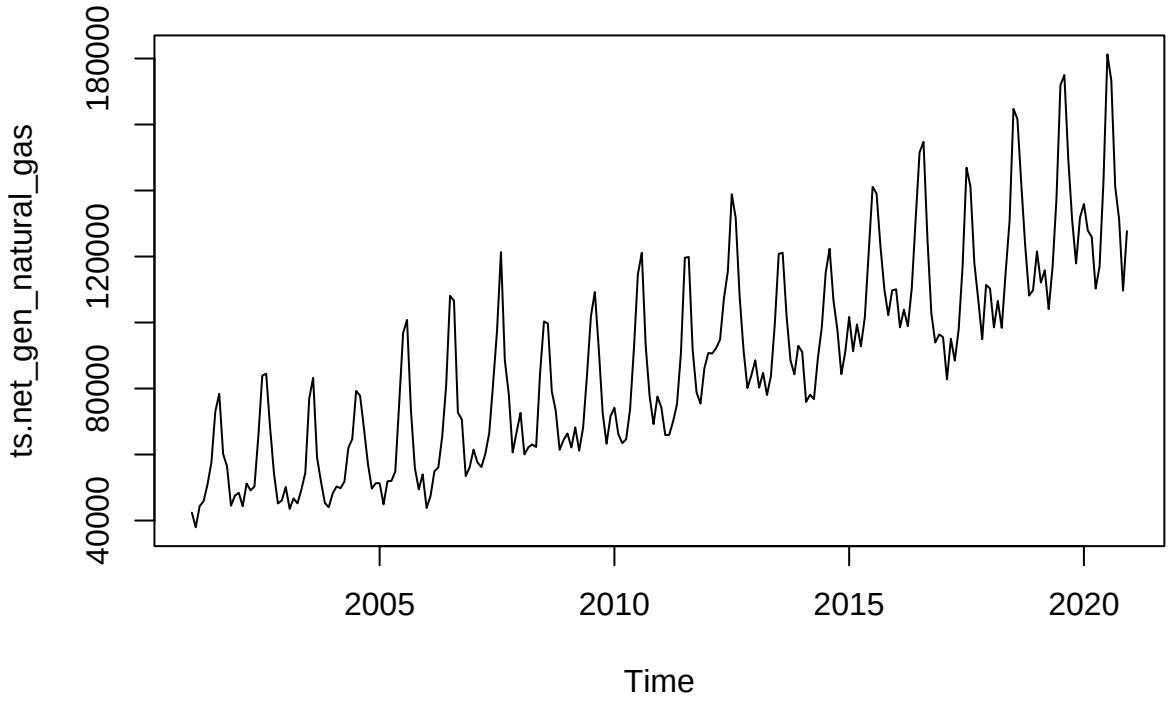
Yes, the residual series looks like a white noise series becuase it doesn't show a trend as well as a seasonal pattern, as shown in the histogram, the residuals follow a normal distribution, and in the ACF plot as well trend and seasonality are not seen.

Modeling the original series (with seasonality)

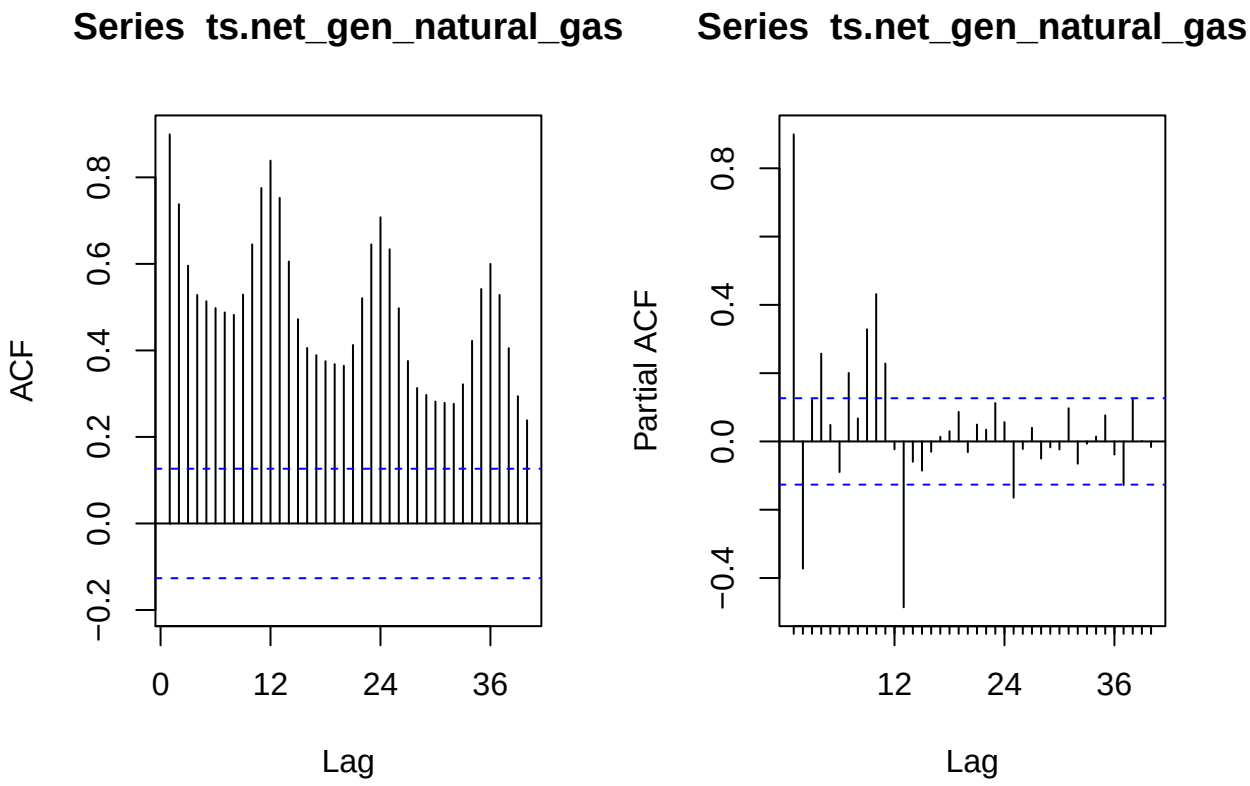
Q7

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.

```
#The original series is net_gen_natural_gas
# Creating the time series object for the original series
ts.net_gen_natural_gas <- ts(net_gen_natural_gas$Natural_gas_gen_MWH, start = c(2001,01),
frequency = 12)

# Plotting the time series object and the acf and pacf plots
plot(ts.net_gen_natural_gas)
```



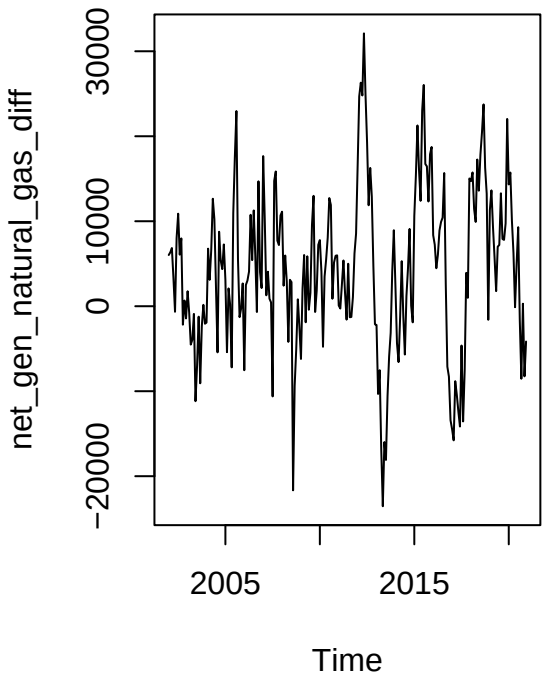

```
par(mfrow=c(1,2))
Acf(ts.net_gen_natural_gas, lag.max = 40)
Pacf(ts.net_gen_natural_gas, lag.max= 40)
```



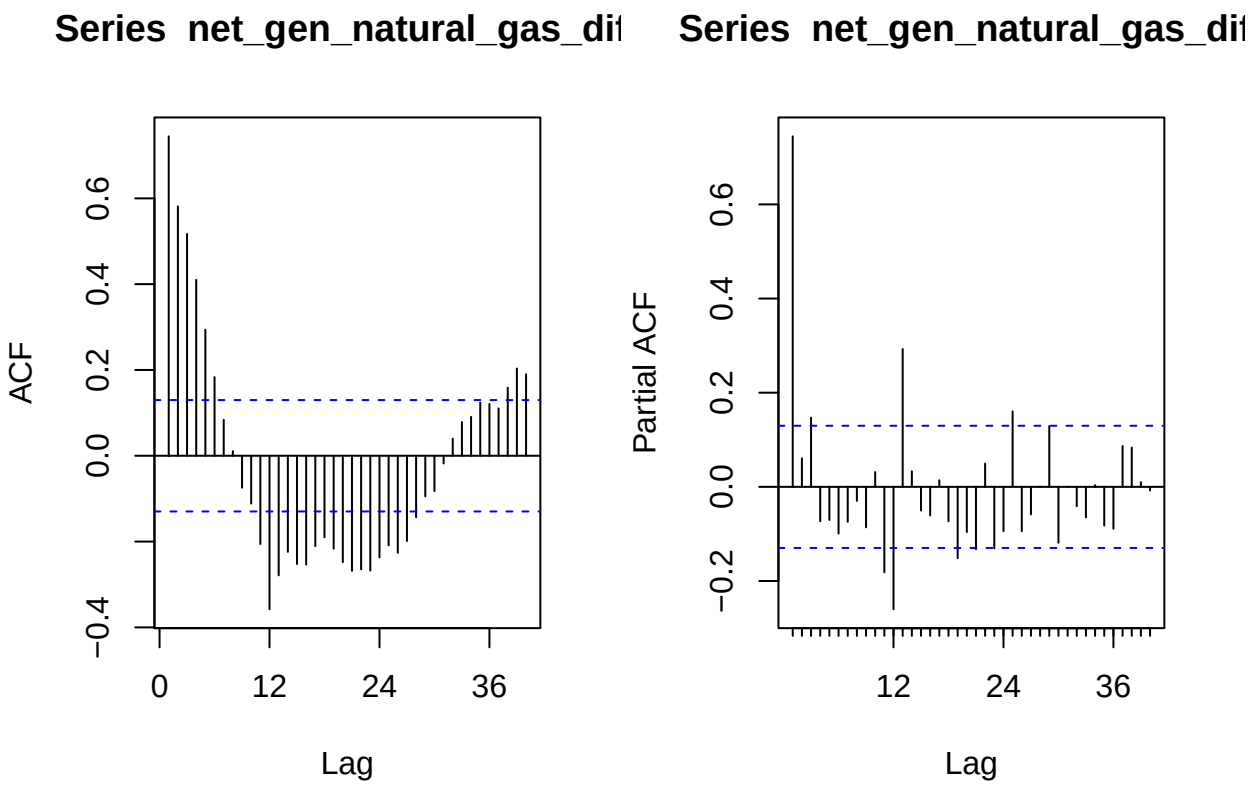

```
# The ACF plot clearly indicate the seasonality by spikes at equally spaced lags.
# Therefore, to achieve stationarity, the series has to be seasonally differenced.
# For this we need to find out the number of seasonal differences needed
ns_diff <- nsdiffs(ts.net_gen_natural_gas,test = c("seas", "ocsb", "hegy", "ch"),max.D = 1)
cat("Number of seasonal differences:", ns_diff)
```

```
## Number of seasonal differences: 1
```

```
# The number of seasonal differences needed to achieve statinarity is 1, i.e. D=1
# Differencing the series to remove the seasonal trend
net_gen_natural_gas_diff <- diff(ts.net_gen_natural_gas, lag = 12, differences = 1)
plot(net_gen_natural_gas_diff)
```




```
par(mfrow=c(1,2))
Acf(net_gen_natural_gas_diff, lag.max = 40)
Pacf(net_gen_natural_gas_diff, lag.max = 40)
```



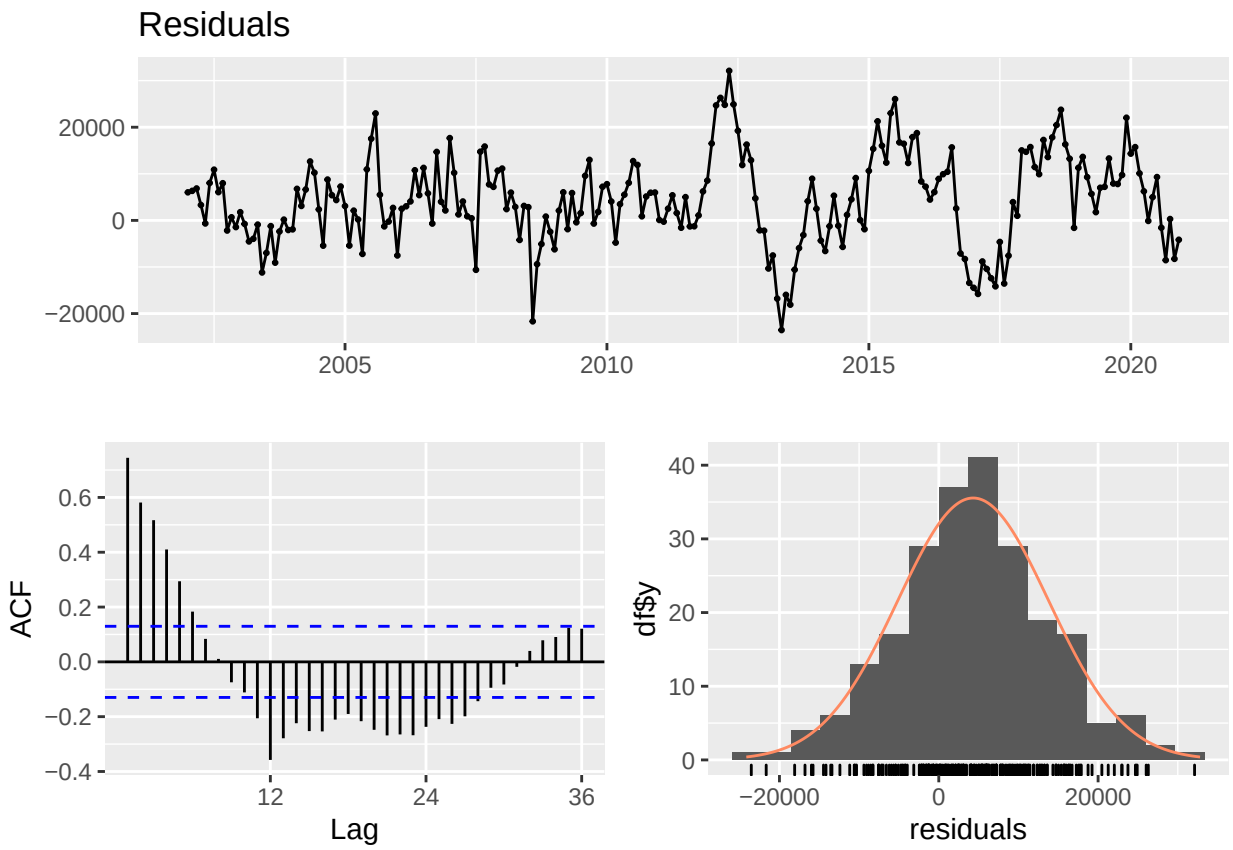
Looking at the ACF and PACF plots, the ACF plot clearly indicates the seasonality by multiple spikes at equally spaced lags (in the ACF plot there are multiple significant spikes at seasonal lags - in an interval of 12) indicating that the value of s is 12, i.e. s=12, and in the PACF plot, there is one significant spike at lag 1. These are the features of a SAR process giving the value of P to be equal to 1, i.e. P=1 and thus, Q=0. Therefore, the ARIMA model for the seasonal component of the series is in the order of (1,1,0). Hence, combining with the non-seasonal component with the seasonal component the ARIMA model becomes ARIMA(1,1,0)(1,1,0)[12].

```
# Fiting an arima model to the differenced seasonal series
arima_seasonal_model <- Arima(net_gen_natural_gas_diff, order = c(1,1,0), seasonal = c(1,1,0), include.m
print(arima_seasonal_model)
```

```
## Series: net_gen_natural_gas_diff
## ARIMA(1,1,0)(1,1,0)[12]
##
## Coefficients:
##           ar1     sar1
##       -0.1906  -0.6455
## s.e.   0.0670   0.0495
##
## sigma^2 = 75037241:  log likelihood = -2256.67
## AIC=4519.34   AICc=4519.45   BIC=4529.45
```

In fitting the seasonal series to the arima model, the constant "include.mean" should be false as we have differenced the seires, and the "include.drift" should be true to allow and see some deviation from zero mean.

```
# Plotting the residual of the seasonal series of the ARIMA fit
checkresiduals(net_gen_natural_gas_diff, lag=12)
```

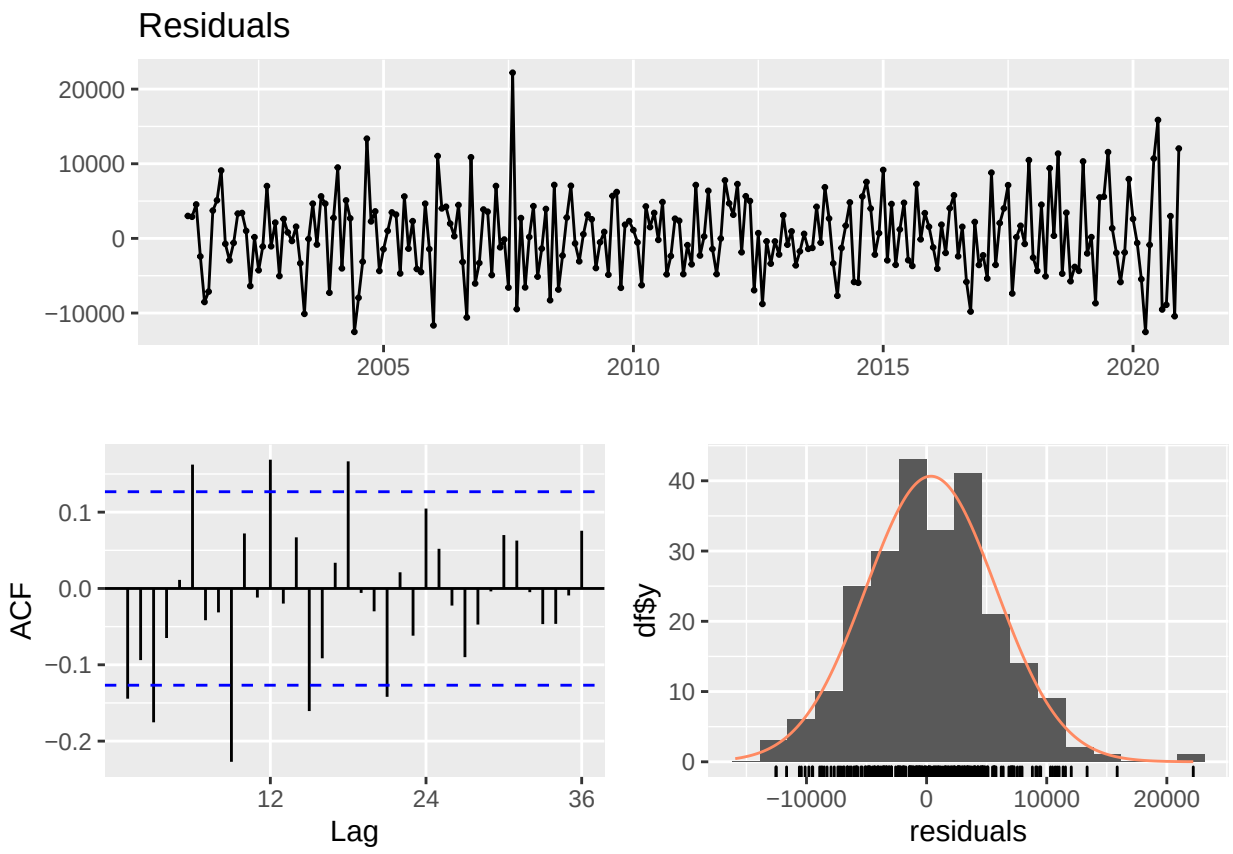


Here also, the residual series looks like a white noise series because it doesn't show a trend as well as a seasonal pattern, it moves on and around 0, as shown in the histogram, the residuals follow a normal distribution, and in the ACF plot there seem a trend and seasonality, but not long lasting or changes over time.

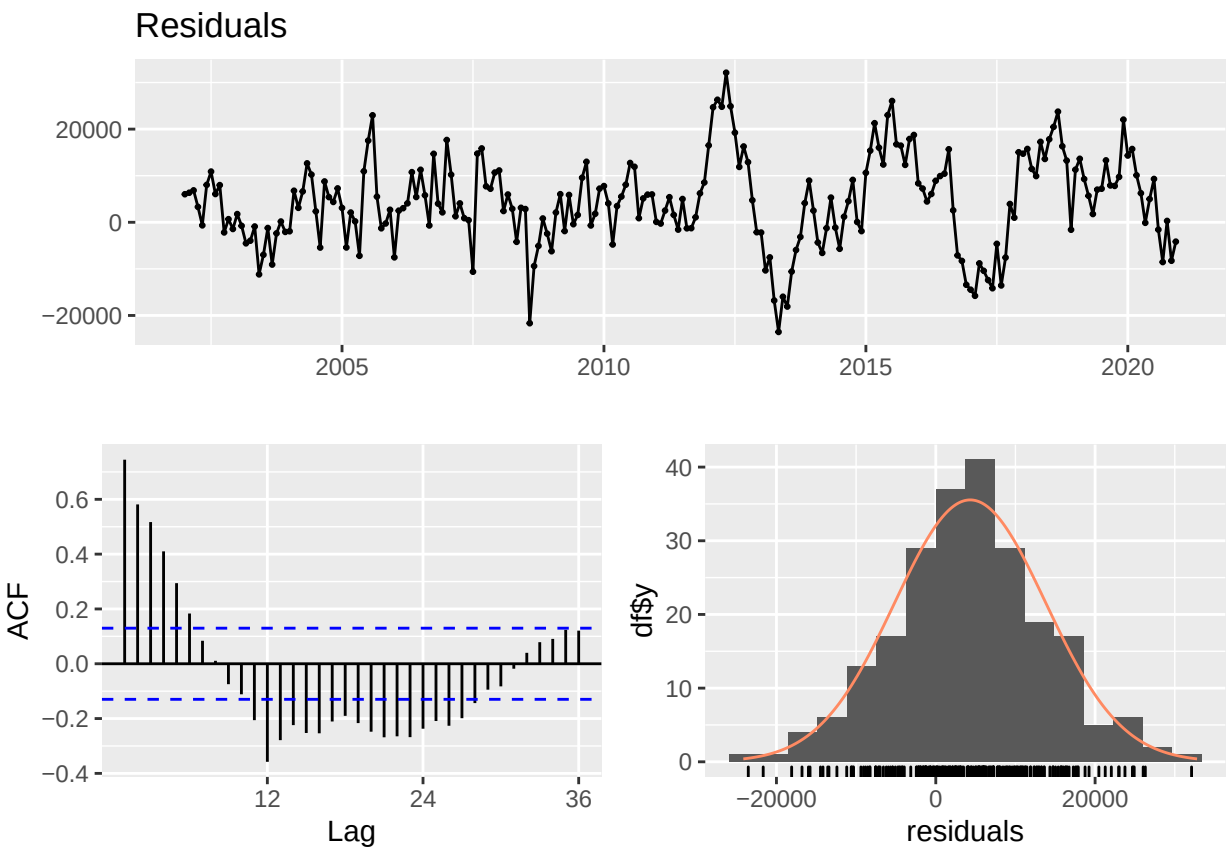
Q8

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.

```
# Comparing the residuals of the deseasoned and original series
checkresiduals(diff_deseasoned_natural_gas_gen, lag=12)
```




```
checkresiduals(net_gen_natural_gas_diff, lag=12)
```



The two ARIMA models are different as one has the seasonal component but the other dosen't have the seasonal component. Thus, the comparision between the two is not a logical or fair comparision. Otherwise, looking at the residuals of the two ARIMA models, the residual of the deseasoned series show that it is better positioned on and around zero, and has a much better normal distribution, when compared with the residuals of the original (that has seasonal component) series, thus the ARIMA model of the deseasoned series seem to better representing the Natural Gas Series.

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 correct orders. The intention of the assignment is to walk you to the process and help you figure out what you did wrong (if you did anything wrong!).

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?

```
# Checking the deseasoned series with auto.arima function to get the model parameter from R
arima_autofit <- auto.arima(diff_deseasoned_natural_gas_gen, max.D = 0, max.P = 0, max.Q = 0)
print(arima_autofit)
```

```
## Series: diff_deseasoned_natural_gas_gen
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ma1      mean
##       0.7132  -0.9773  361.0339
## s.e.  0.0609   0.0299   31.8932
##
## sigma^2 = 26009964:  log likelihood = -2378.67
## AIC=4765.34   AICc=4765.51   BIC=4779.25
```

The order of the best ARIMA model of the deseasoned series obtained from R is ARIMA(1,0,1), this doesn't match with the order that is specified under question number 4, i.e. ARIMA(1,1,0).

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_original <- auto.arima(ts.net_gen_natural_gas)
print(arima_autofit_original)
```

```
## Series: ts.net_gen_natural_gas
## ARIMA(1,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     sma1     drift
##       0.7490  -0.7104  360.5891
## s.e.  0.0436   0.0551   37.4462
##
## sigma^2 = 26922530:  log likelihood = -2276.98
## AIC=4561.96   AICc=4562.13   BIC=4575.67
```

The order of the best ARIMA model for the original series obtained from R is ARIMA(1,0,0)(0,1,1)[12], this as well doesn't match with the order that is specified under question number 7.