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

Assignment 7 - Due date 03/06/25

Aye Nyein Thu

## 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)
library(dplyr)
library(trend)
```

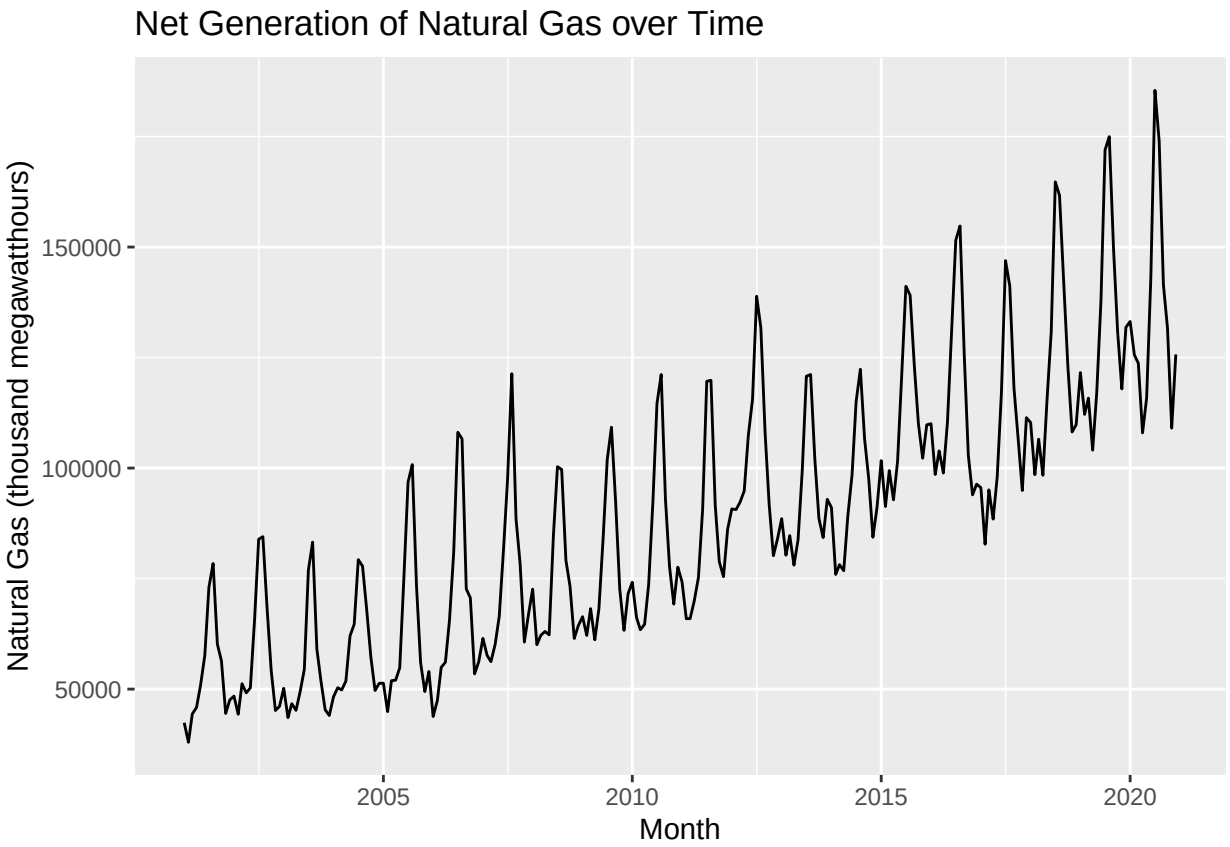
## Importing and processing the data set

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**.

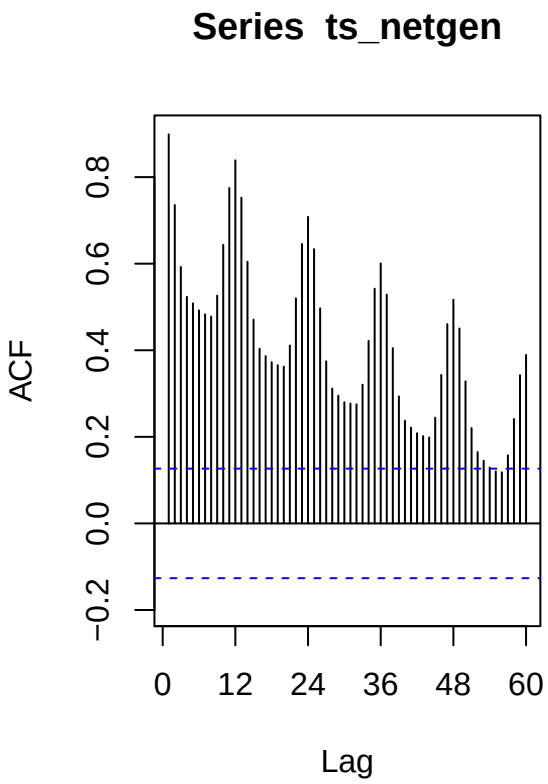
**Q1**

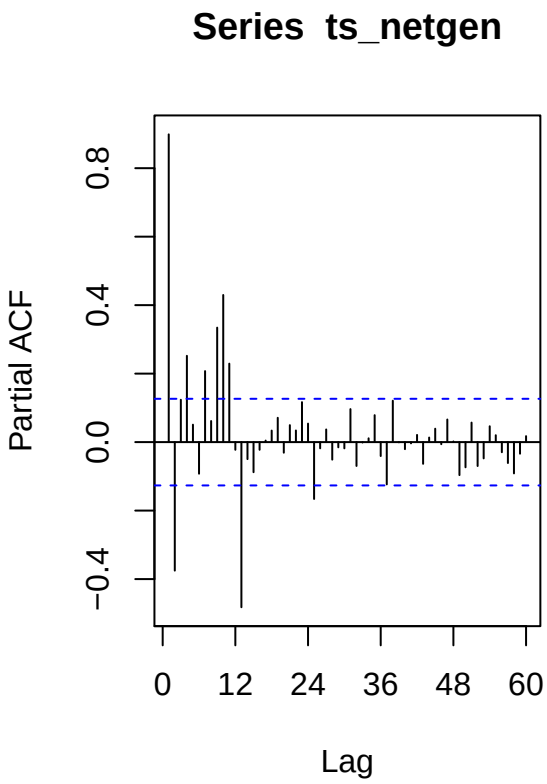
Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

```
# Import data
netgen <- read.csv(
  file = "./Data/Net_generation_United_States_all_sectors_monthly.csv",
  header = TRUE, skip=4)

# Prepare data
netgen_cleaned <- netgen %>%
  mutate(Month = my(Month)) %>%
  select(Month, Natural_gas = "natural.gas.thousand.megawatthours") %>%
  arrange(Month)

# Create time series
ts_netgen <- ts(netgen_cleaned$Natural_gas,
                           start=c(year(netgen_cleaned$Month[1]),
                                   month(netgen_cleaned$Month[1])),
                           frequency=12)

# Plot time series, ACF and PACF
ggplot(netgen_cleaned, aes(x=Month, y=Natural_gas)) +
  geom_line() +
  ylab("Natural Gas (thousand megawatthours)") +
  ggtitle ("Net Generation of Natural Gas over Time")
```




```
par(mfrow=c(1,2))
ACF_Plot <- Acf(ts_netgen, lag = 60, plot = TRUE)
PACF_Plot <- Pacf(ts_netgen, lag = 60)
```

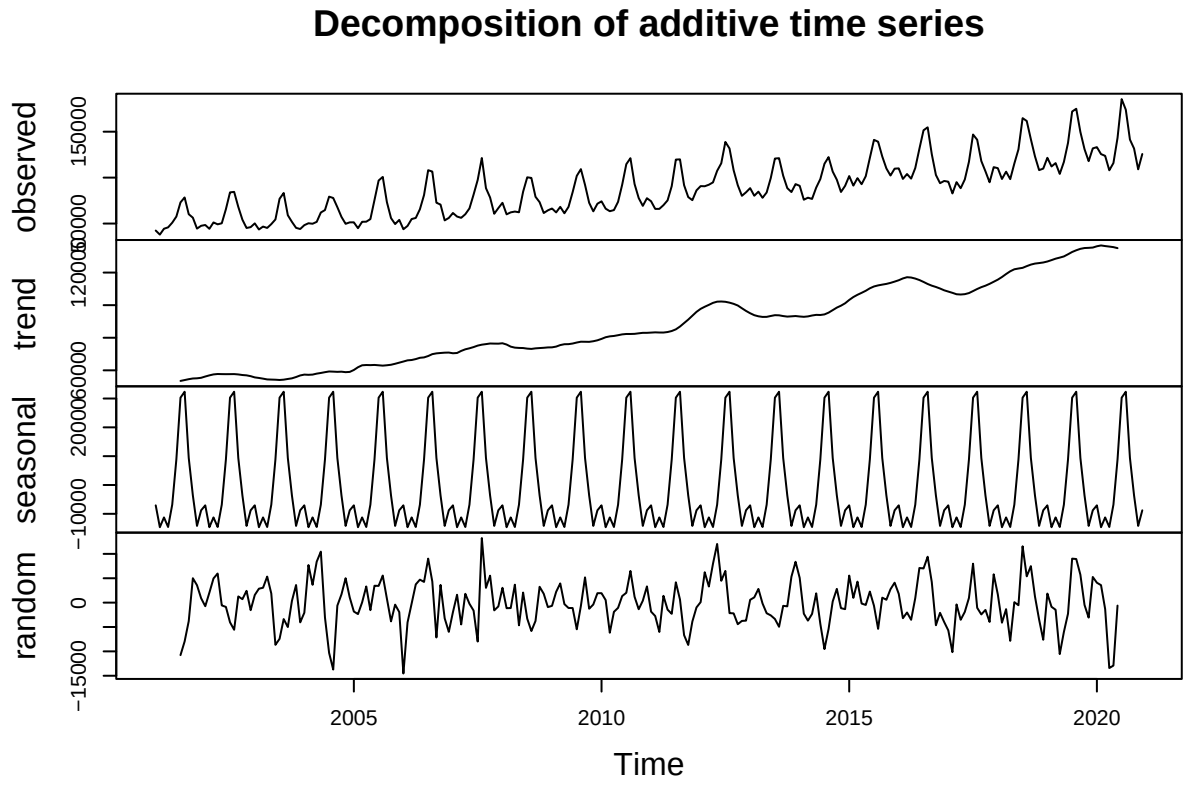


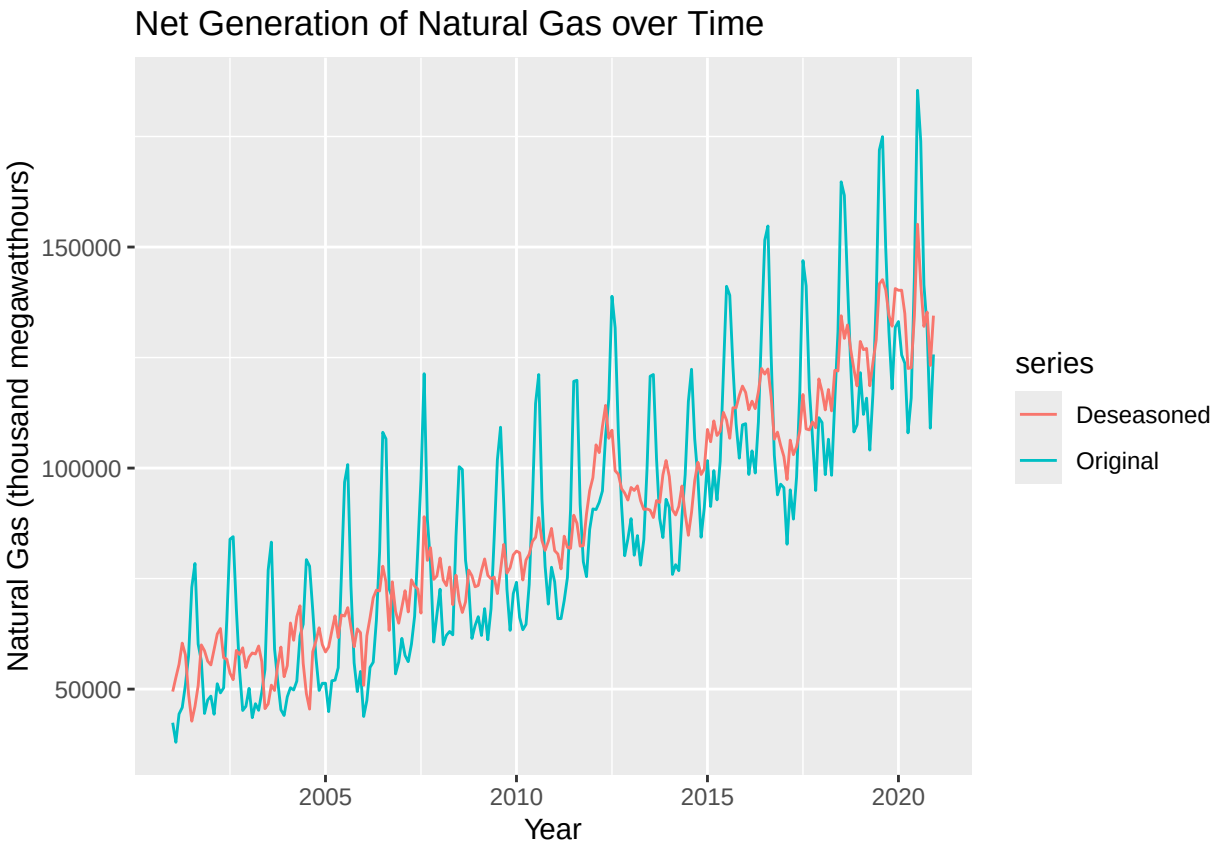

**Q2**

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.

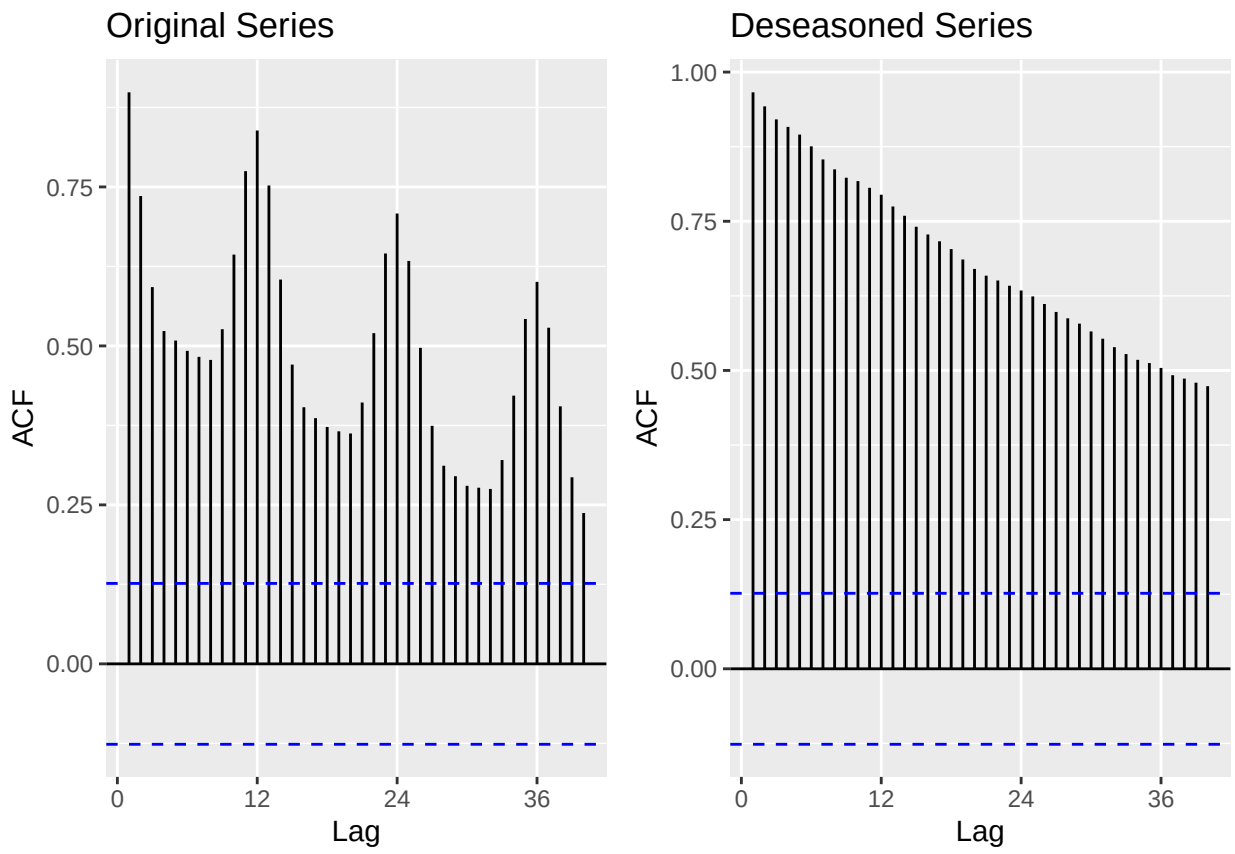
```
# Decompose the series
decompose_netgen <- decompose(ts_netgen, "additive")
plot(decompose_netgen)
```



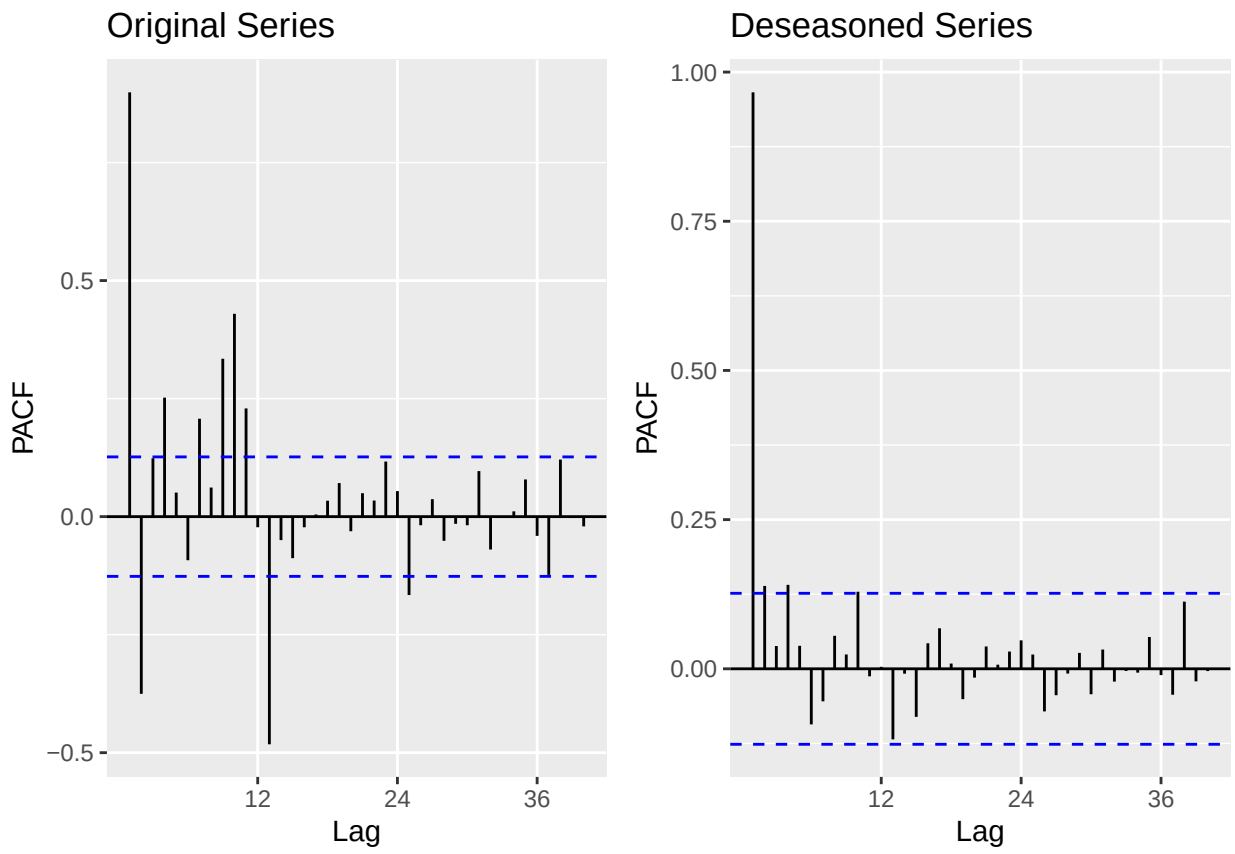

```
# Deseason the series
deseason_netgen <- seasadj(decompose_netgen)

# Plot original and deseasoned series
autoplot(ts_netgen, series="Original") +
  autolayer(deseason_netgen, series="Deseasoned") +
  xlab("Year") + ylab("Natural Gas (thousand megawatthours)") +
  ggtitle("Net Generation of Natural Gas over Time")
```




```
# Compare ACFs
plot_grid(
  autoplot(Acf(ts_netgen, lag = 40, plot=FALSE),
                main = "Original Series") ,
  autoplot(Acf(deseason_netgen, lag = 40, plot=FALSE),
                  main = "Deseasoned Series")
)
```




```
# Compare PACFs
plot_grid(
  autoplot(Pacf(ts_netgen, lag = 40, plot=FALSE),
                main = "Original Series") ,
  autoplot(Pacf(deseason_netgen, lag = 40, plot=FALSE),
                  main = "Deseasoned Series")
)
```




Answer: In the original series, both ACF and PACF plots exhibits the existence of seasonality. The ACF plots has the regular seasonality-like features and spikes in 12, 24 and 36 while PACF also showcases the similar spikes behaviors. After deseasoning the series, the seasonality features have now been removed as there is not regular spikes in both ACF and PACF plots after deasonalization. The ACF plot shows the trend behavior.

## 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 the ADF test
print(adf.test(deseason_netgen, alternative = "stationary"))
```

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

```r
# Run the Mann Kendall Test
summary(MannKendall(deseason_netgen))
```

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

> Answer: The Augmented Dickey-Fuller (ADF) test shows that the deseasonalized series has a ADF value at -4.03 and p-value at 0.01. Since the p-value is lower than 0.05, we have enough evidence to reject the null hypothesis that suggests the deseasonalized series is a unit root or stochastic trend. The Mann Kendall test result indicates that the deseasonalized series of net natural gas generation has a very strong increasing trend at tau value of 0.84. As the p-value is very small, the result is statistically significant at 1% level.

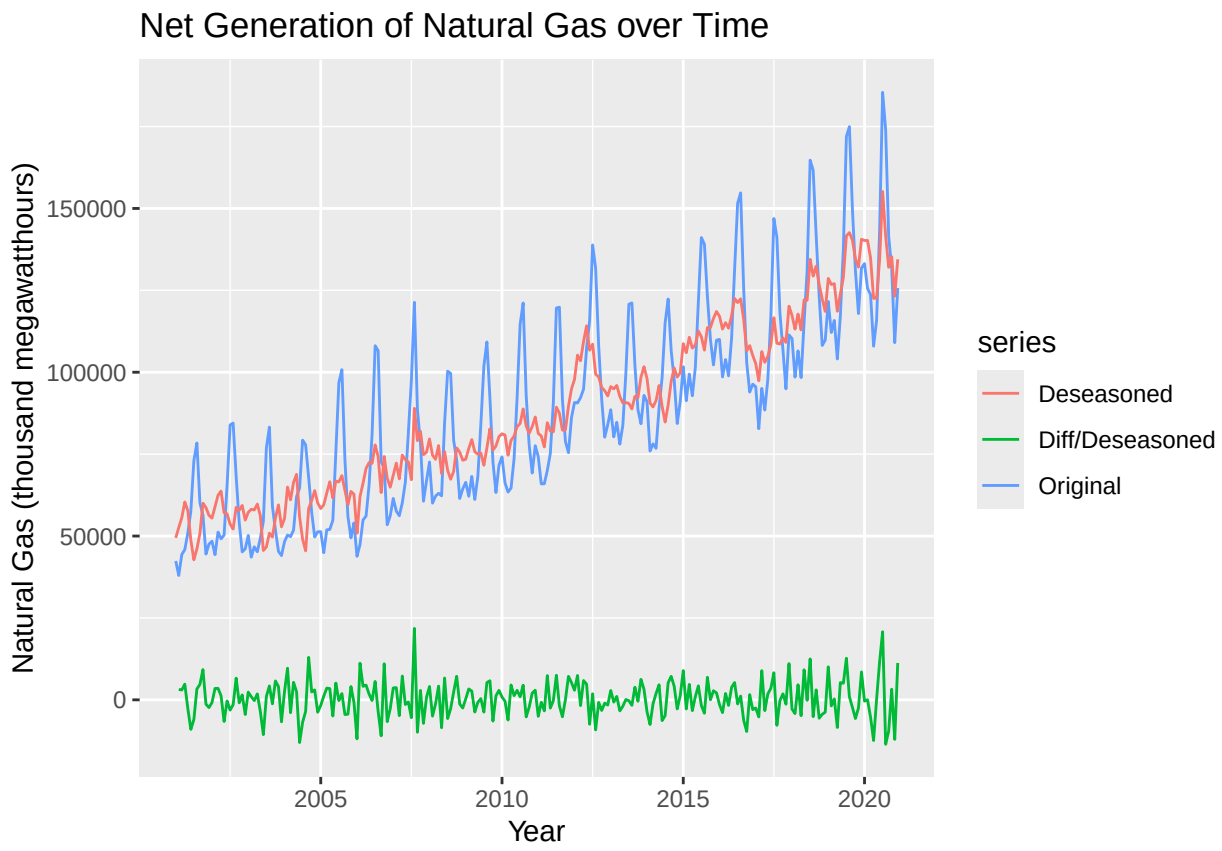
**Q4**

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.

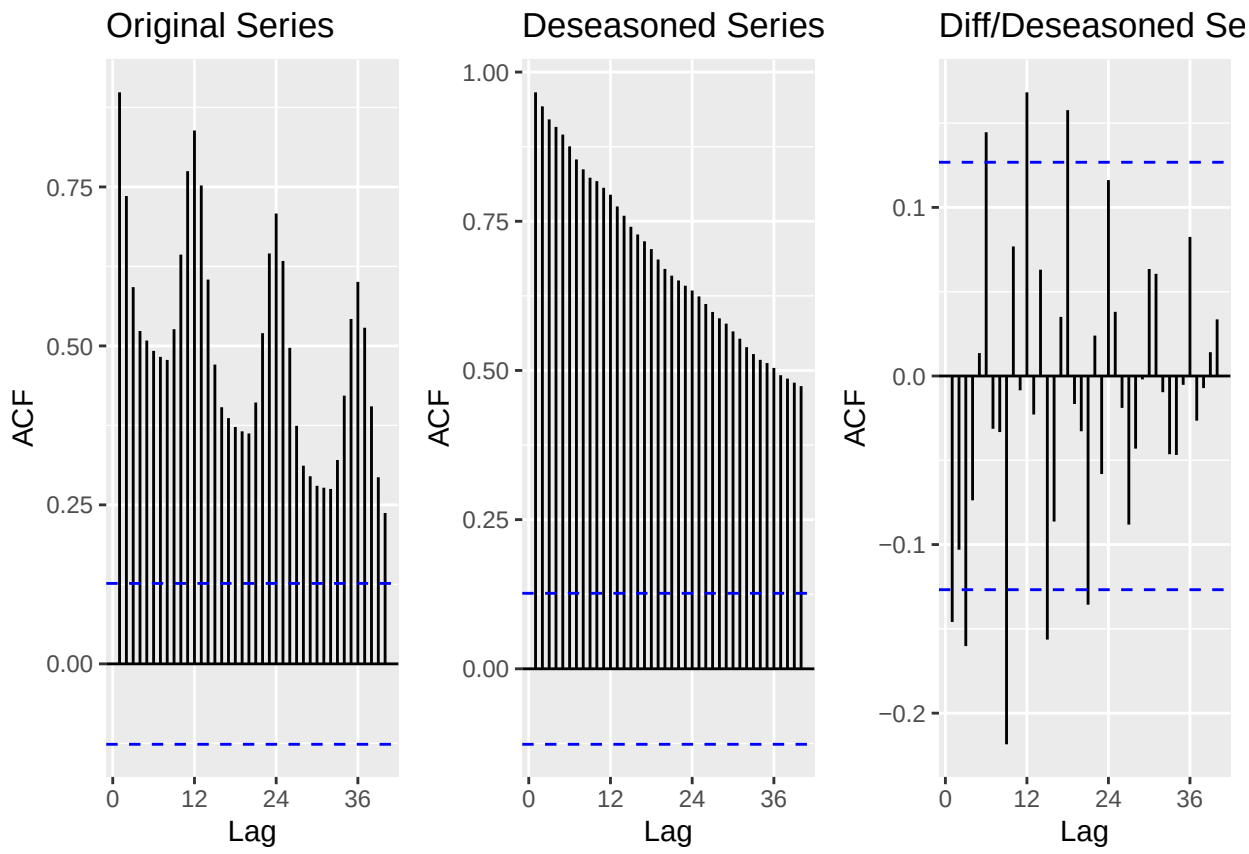
```r
# Find no. of time to difference
n_diff <- ndiffs(deseason_netgen)
cat("Number of non-seasonal differencing needed: ",n_diff)
```

```
## Number of non-seasonal differencing needed:  1
```

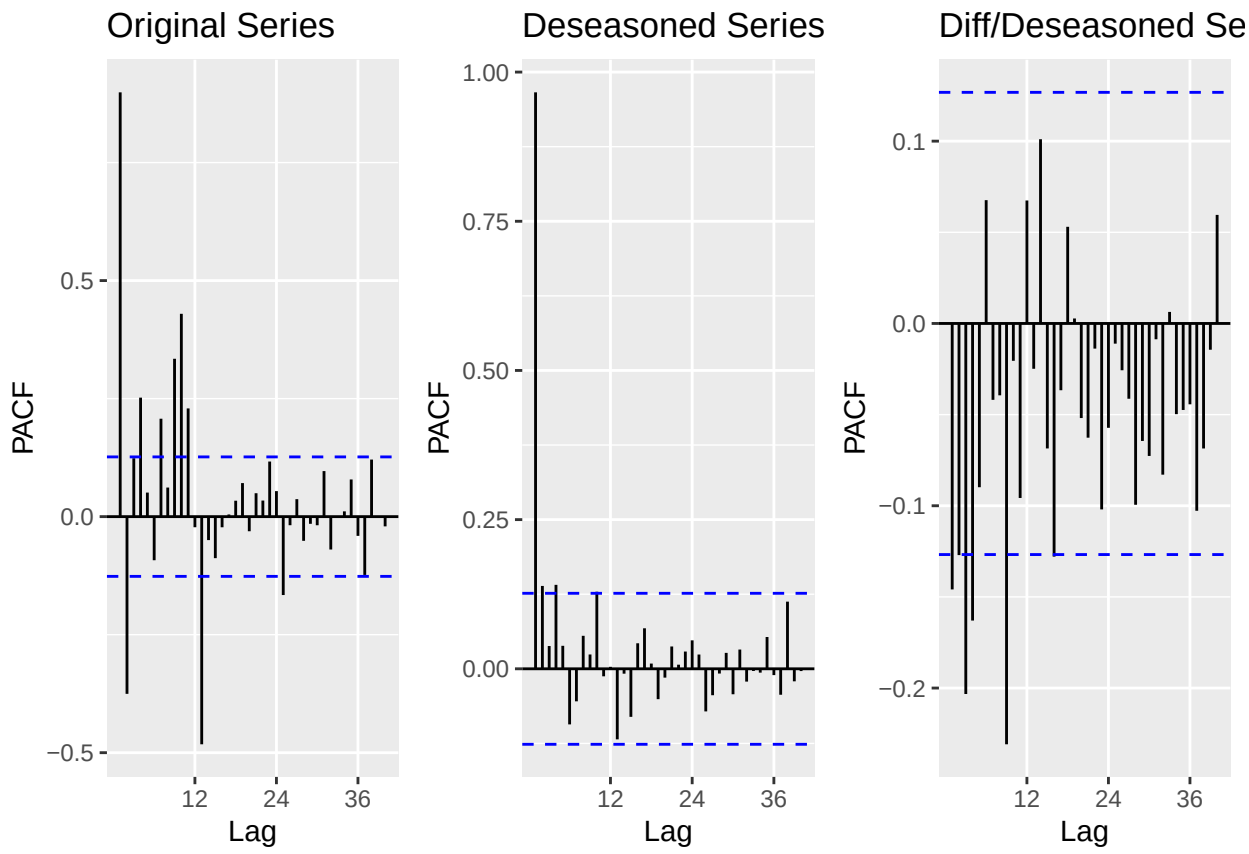
```r
# Therefore, we will assume that d = 1.

# Create the differenced series
diff_deseason_netgen <- diff(deseason_netgen,differences=1,lag=1)

# Plot original, deseasoned and differenced/ deaseasoned series
autoplot(ts_netgen, series = "Original") +
  autolayer(deseason_netgen, series = "Deseasoned") +
  autolayer(diff_deseason_netgen, series = "Diff/Deseasoned") +
  xlab("Year") +
  ylab("Natural Gas (thousand megawatthours)") +
  ggtitle("Net Generation of Natural Gas over Time")
```



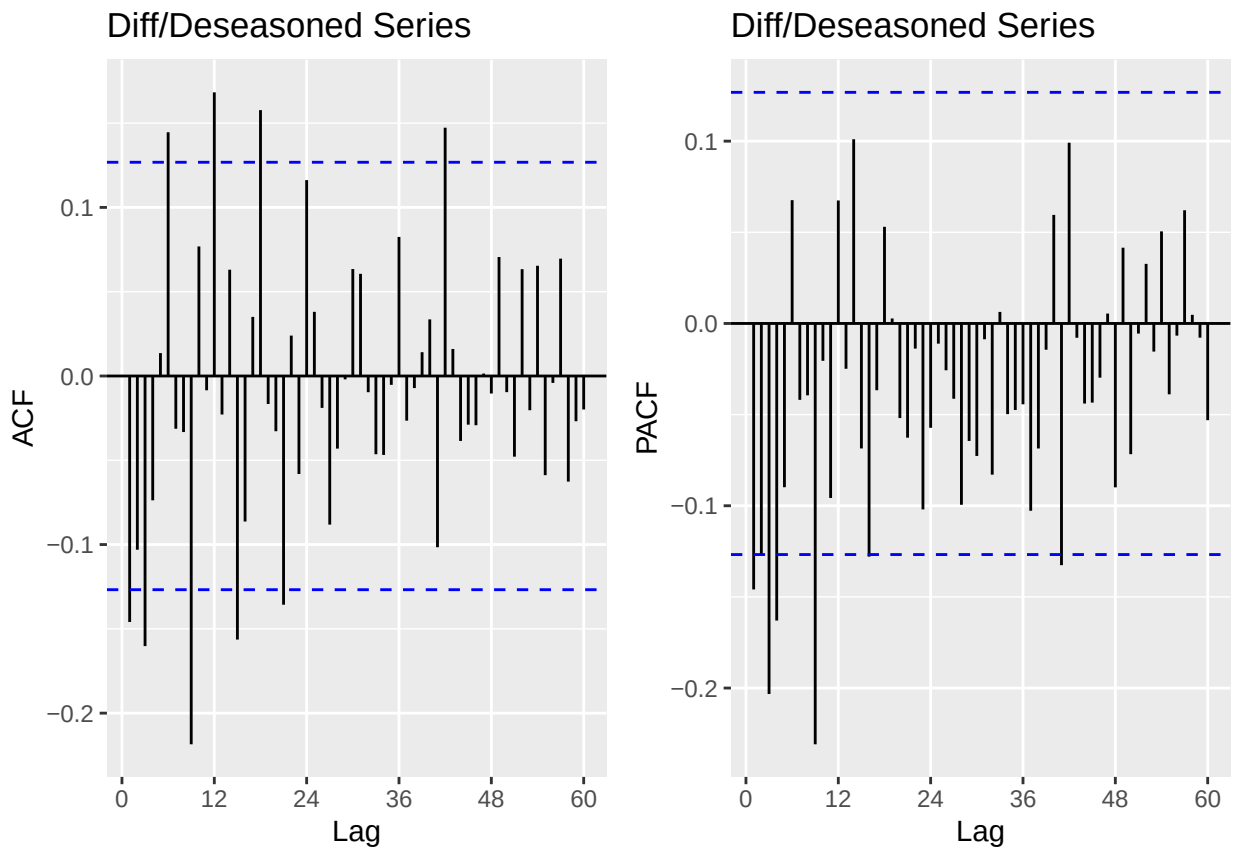

```
# Compare ACFs
plot_grid(
  autoplot(Acf(ts_netgen, lag = 40, plot=FALSE),
                main = "Original Series") ,
  autoplot(Acf(deseason_netgen, lag = 40, plot=FALSE),
                  main = "Deseasoned Series") ,
  autoplot(Acf(diff_deseason_netgen, lag = 40, plot=FALSE),
                  main = "Diff/Deseasoned Series"),
  nrow = 1
)
```




```
# Compare PACFs
plot_grid(
  autoplot(Pacf(ts_netgen, lag = 40, plot=FALSE),
                main = "Original Series") ,
  autoplot(Pacf(deseason_netgen, lag = 40, plot=FALSE),
                  main = "Deseasoned Series") ,
  autoplot(Pacf(diff_deseason_netgen, lag = 40, plot=FALSE),
                  main = "Diff/Deseasoned Series"),
  nrow = 1
)
```




```
# Plot the differenced series together
plot_grid(
  autoplot(Acf(diff_deseason_netgen, lag = 60, plot=FALSE),
                  main = "Diff/Deseasoned Series"),
  autoplot(Pacf(diff_deseason_netgen, lag = 60, plot=FALSE),
                  main = "Diff/Deseasoned Series"),
  nrow = 1
)
```




Comment: Although we know that d = 1 using "diff" function, both ACF and PACF plots do not have a clear and sharp cut-offs. For instance, in the ACF plots, q could also be 1 or 2. At the same time, in the PACF plot, p could also be 1 or 2. Therefore, we will continue with running both models manually.

**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.

```
# Find the possible models
nonseasonal_111 <- Arima(deseason_netgen,order=c(1,1,1))
print(nonseasonal_111)
```

```
## Series: deseason_netgen
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.5720  -0.8167
## s.e.  0.1024   0.0696
##
## sigma^2 = 28492791:  log likelihood = -2389.48
## AIC=4784.95   AICc=4785.06   BIC=4795.38
```

```r
compare_aic <- data.frame(nonseasonal_111$aic)

nonseasonal_112 <- Arima(deseason_netgen,order=c(1,1,2))
print(nonseasonal_112)
```

```
## Series: deseason_netgen
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4770  -0.7003  -0.0712
## s.e.  0.1718   0.1732   0.0914
##
## sigma^2 = 28543172:  log likelihood = -2389.19
## AIC=4786.38   AICc=4786.55   BIC=4800.28
```

```r
compare_aic <- data.frame(compare_aic,nonseasonal_112$aic)

nonseasonal_211 <- Arima(deseason_netgen,order=c(2,1,1))
print(nonseasonal_211)
```

```
## Series: deseason_netgen
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2      ma1
##       0.5565  -0.0689  -0.7732
## s.e.  0.1137   0.0768   0.0967
##
## sigma^2 = 28517663:  log likelihood = -2389.08
## AIC=4786.17   AICc=4786.34   BIC=4800.07
```

```r
compare_aic <- data.frame(compare_aic,nonseasonal_211$aic)

nonseasonal_212 <- Arima(deseason_netgen,order=c(2,1,2))
print(nonseasonal_212)
```

```
## Series: deseason_netgen
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.8906  -0.2961  -1.1020  0.2918
## s.e.  0.3797   0.2215   0.3843  0.3058
##
## sigma^2 = 28567337:  log likelihood = -2388.79
## AIC=4787.57   AICc=4787.83   BIC=4804.95
```

```r
compare_aic <- data.frame(compare_aic,nonseasonal_212$aic)

print(compare_aic)
```

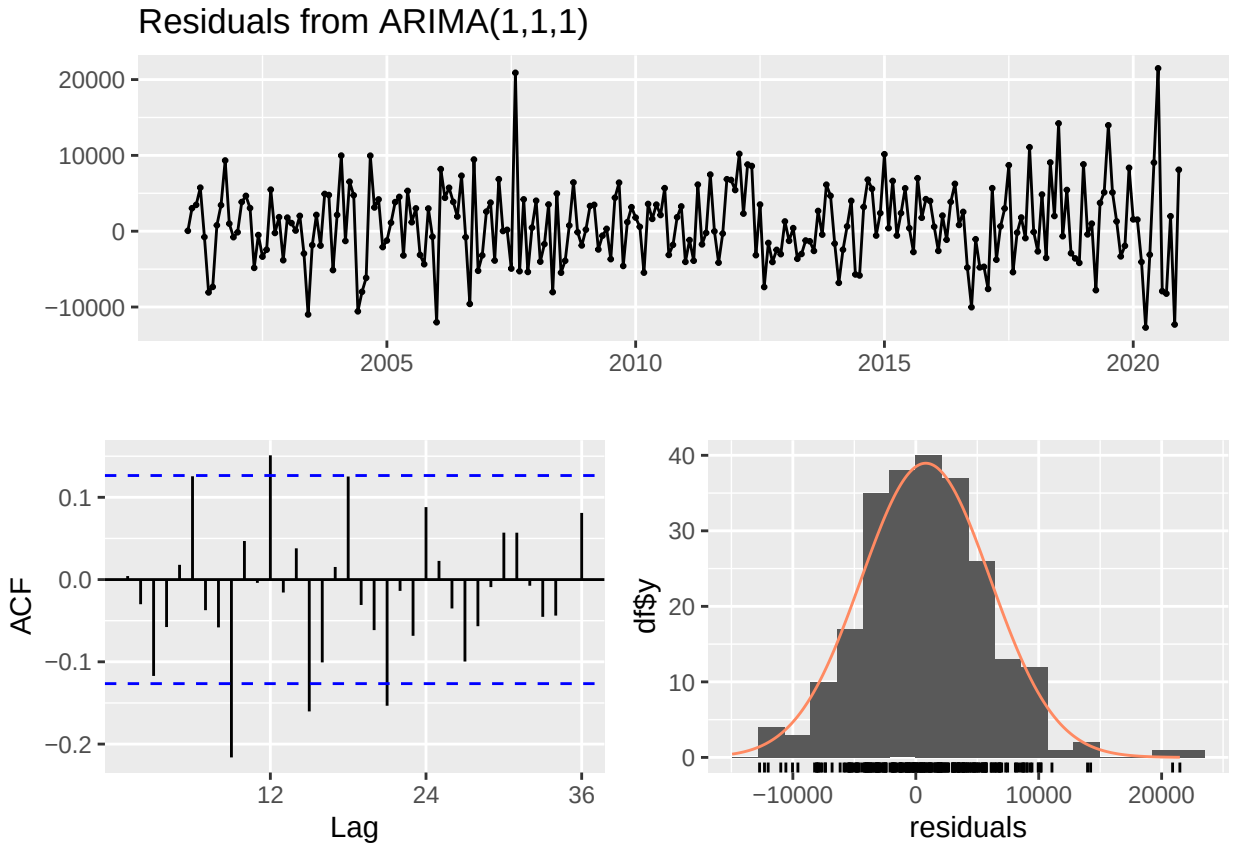
```
##   nonseasonal_111.aic nonseasonal_112.aic nonseasonal_211.aic
## 1            4784.955            4786.376            4786.167
##   nonseasonal_212.aic
## 1            4787.572
```

Answer: As per the AIC results, the best fit model is the ARIMA(1,1,1). Therefore, the combination of stationary tests, ACF and PACF plots and further trial and error model testing indicates that the ARIMA model with p=1, d=1 and q=1 are the best possible alternative.

**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?

```
# Plot the residuals of the manual nonseasonal ARIMA model
checkresiduals(nonseasonal_111)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 52.457, df = 22, p-value = 0.0002702
##
## Model df: 2.   Total lags used: 24
```

Answer: The residuals plot of the deseasonalized series seem to be random as there is no clear trends or regular spikes. Despite a few spikes, most of the correlation lines in the ACF plot are within the blue line and looks random. The residuals distribution also showcases a normal distribution. Therefore, we could assume that it exhibits a white noise series.

## 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 the ADF test
print(adf.test(ts_netgen, alternative = "stationary"))
```

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

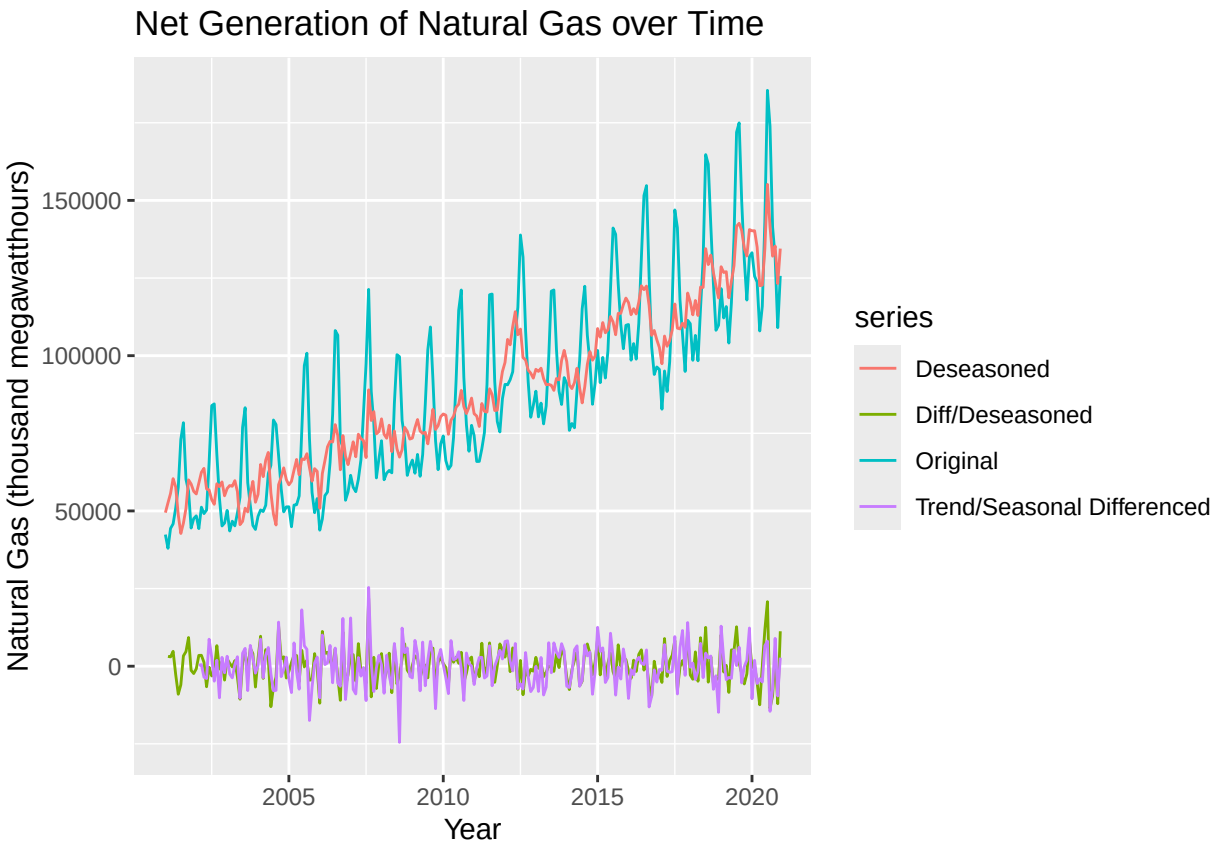
```
# Run the Mann Kendall Test
summary(SeasonalMannKendall(ts_netgen))
```

```
## Score =  2022 , Var(Score) = 11400
## denominator =  2280
## tau = 0.887, 2-sided pvalue =< 2.22e-16
```

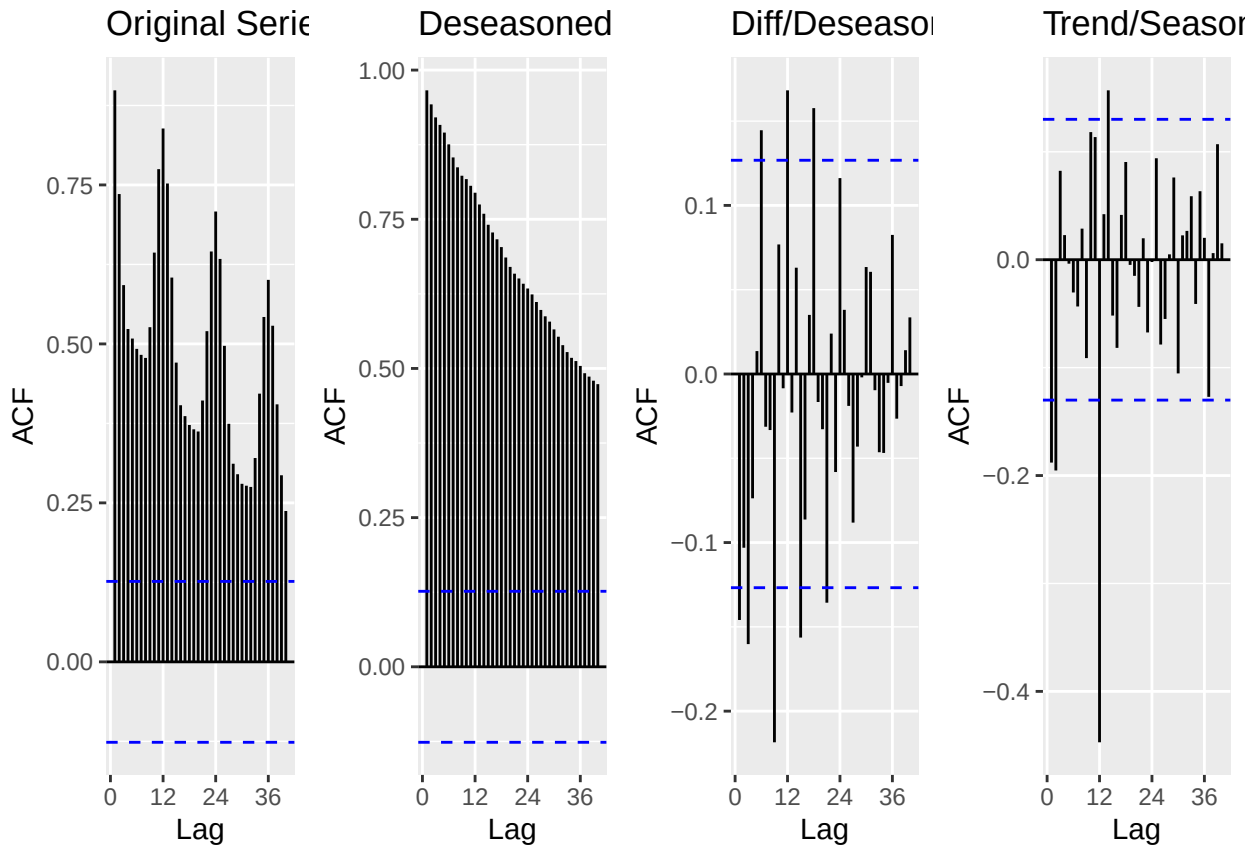
```
# Find no. of time to difference
ns_diff <- ndiffs(ts_netgen)
cat("Number of seasonal differencing needed: ",n_diff)
```

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

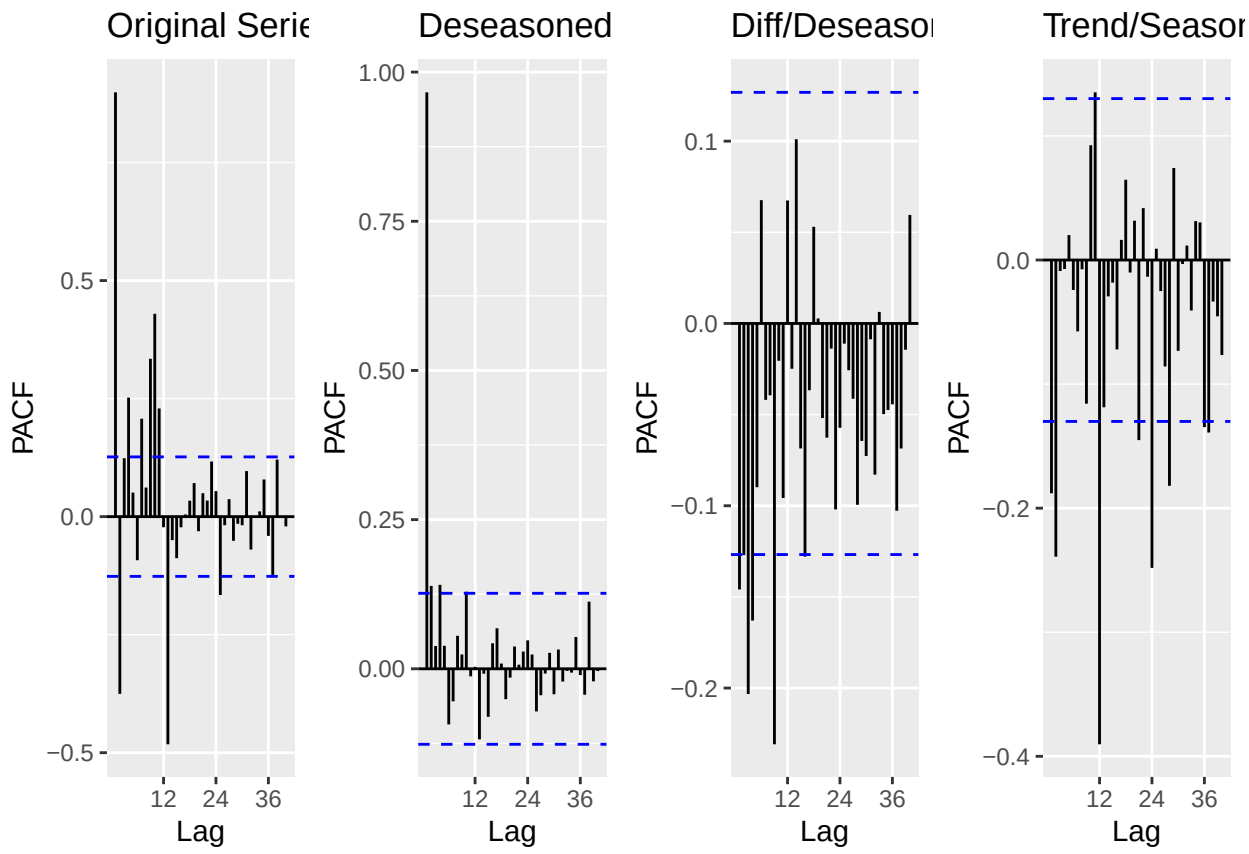
```
# Therefore, we will assume that D = 1.

# Create the differenced series
trend_diff_netgen <- diff(ts_netgen,lag =1, differences=1)
trendseas_diff_netgen <- diff(trend_diff_netgen,lag =12, differences=1)

# Plot original, deseasoned and trend/ seasonal differenced series
autoplot(ts_netgen, series = "Original") +
  autolayer(deseason_netgen, series = "Deseasoned") +
  autolayer(diff_deseason_netgen, series = "Diff/Deseasoned") +
  autolayer(trendseas_diff_netgen, series = "Trend/Seasonal Differenced") +
  xlab("Year") +
  ylab("Natural Gas (thousand megawatthours)") +
  ggtitle("Net Generation of Natural Gas over Time")
```



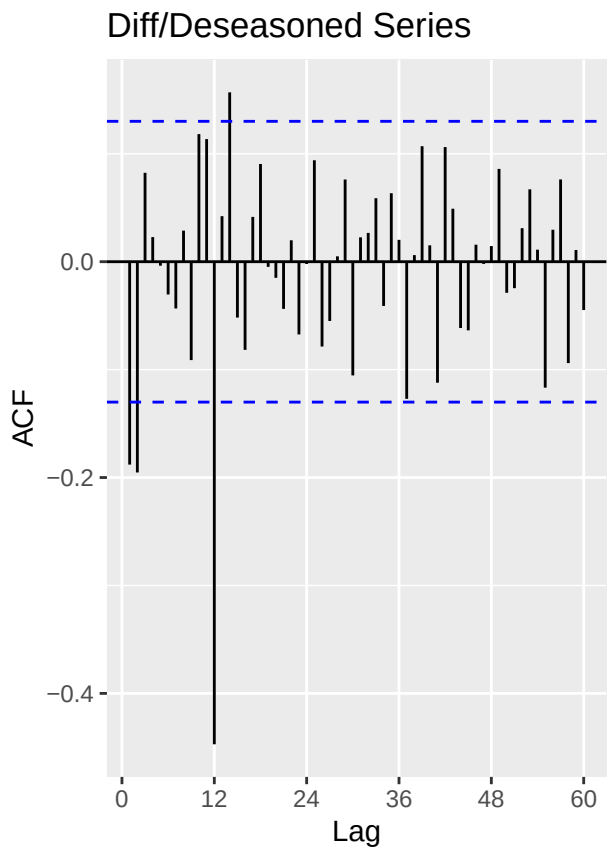

```
# Compare ACFs
plot_grid(
  autoplot(Acf(ts_netgen, lag = 40, plot=FALSE),
                main = "Original Series") ,
  autoplot(Acf(deseason_netgen, lag = 40, plot=FALSE),
                  main = "Deseasoned Series") ,
  autoplot(Acf(diff_deseason_netgen, lag = 40, plot=FALSE),
                  main = "Diff/Deseasoned Series"),
  autoplot(Acf(trendseas_diff_netgen, lag = 40, plot=FALSE),
                  main = "Trend/Seasonal Differenced Series"),
  nrow = 1
)
```

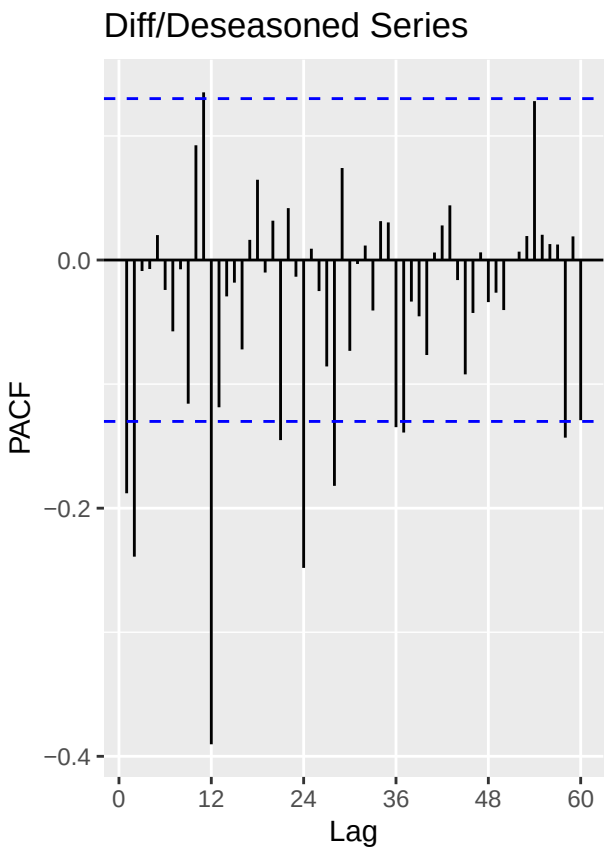



```
# Compare PACFs
plot_grid(
  autoplot(Pacf(ts_netgen, lag = 40, plot=FALSE),
                main = "Original Series") ,
  autoplot(Pacf(deseason_netgen, lag = 40, plot=FALSE),
                  main = "Deseasoned Series") ,
  autoplot(Pacf(diff_deseason_netgen, lag = 40, plot=FALSE),
                  main = "Diff/Deseasoned Series"),
  autoplot(Pacf(trendseas_diff_netgen, lag = 40, plot=FALSE),
                  main = "Trend/Seasonal Differenced Series"),
  nrow = 1
)
```




```
# Plot the differenced series together
plot_grid(
  autoplot(Acf(trendseas_diff_netgen, lag = 60, plot=FALSE),
                  main = "Diff/Deseasoned Series"),
  autoplot(Pacf(trendseas_diff_netgen, lag = 60, plot=FALSE),
                  main = "Diff/Deseasoned Series"),
  nrow = 1
)
```




```
# Find the possible models
seasonal_111011 <- Arima(ts_netgen,
                         order=c(1,1,1),
                         seasonal=c(0,1,1),
                         include.drift=FALSE)
print(seasonal_111011)
```

```
## Series: ts_netgen
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.7323  -0.9819  -0.7017
## s.e.  0.0504   0.0183   0.0563
##
## sigma^2 = 27922085:  log likelihood = -2272.2
## AIC=4552.39   AICc=4552.57   BIC=4566.09
```

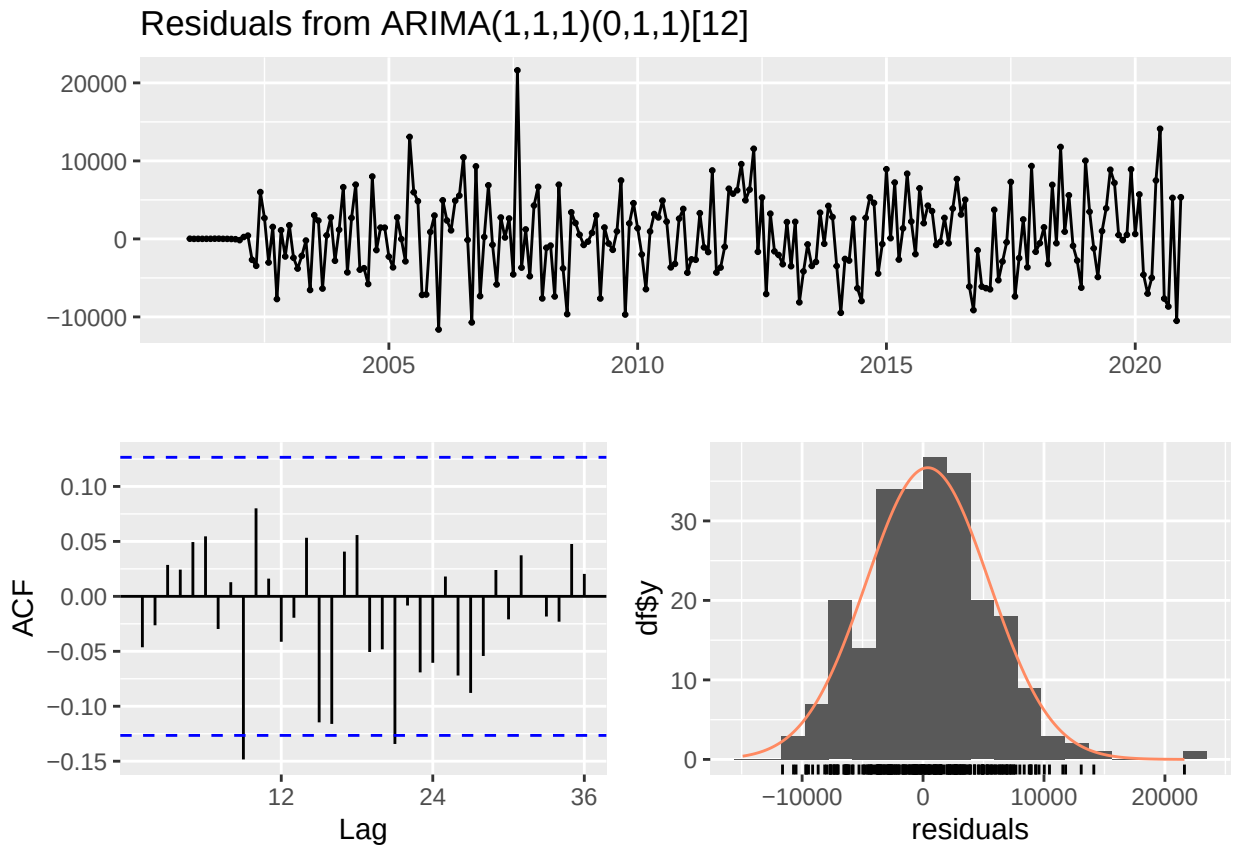
```
compare_aic <- data.frame(compare_aic, seasonal_111011$aic)

seasonal_111110 <- Arima(ts_netgen,
                         order=c(1,1,1),
                         seasonal=c(1,1,0),
                         include.drift=FALSE)
print(seasonal_111110)
```

```
## Series: ts_netgen
## 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 = 32606986:  log likelihood = -2287.56
## AIC=4583.12   AICc=4583.3   BIC=4596.82
```

```
compare_aic <- data.frame(compare_aic, seasonal_111110$aic)

print(compare_aic)
```

```
##   nonseasonal_111.aic nonseasonal_112.aic nonseasonal_211.aic
## 1            4784.955            4786.376            4786.167
##   nonseasonal_212.aic seasonal_111011.aic seasonal_111110.aic
## 1            4787.572            4552.393            4583.117
```

```
# Plot the residuals of the manual nonseasonal ARIMA model
checkresiduals(seasonal_111011)
```




```
##
```

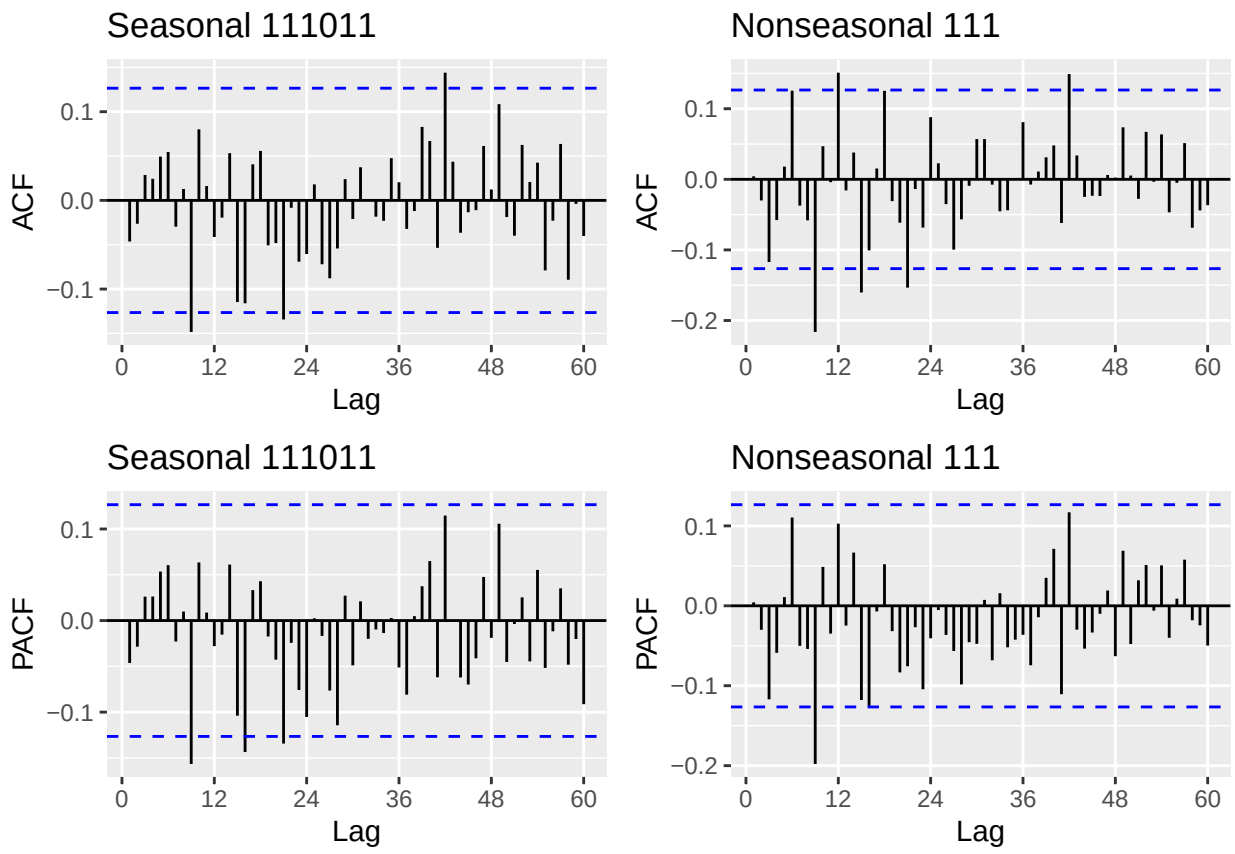
```
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,1,1)[12]
## Q* = 27.607, df = 21, p-value = 0.1516
##
## Model df: 3.   Total lags used: 24
```

Comment: Similar to the non-seasonal parameters identification, D = 1 with the use of "ndiffs" function. As the ACF plot shows only one single spike at lag 12 while PACF plot shows multiple spikes at lag 12, 24, 36, it is more likely to be Seasonal MA (SMA) process. Since we should use P + Q is not more than 1 at a time and the plots show it is more likely to be SMA process, we will assume that P=0, D=1 and Q=1 in the seasonal ARIMA model. We will reuse p=1, d=1 and q=1 from the previous nonseasonal model. But we will also add the option with P=, D=1 and Q=0 to compare.
As per the AIC comparison table, ACF and PACF plots values, SARIMA model(111011) is the best fit with lowest AIC value. The residuals also look random.

### 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.

```
plot_grid(
  autoplot(Acf(seasonal_111011$residuals,lag.max=60,plot=FALSE),main="Seasonal 111011"),
  autoplot(Acf(nonseasonal_111$residuals,lag.max=60,plot=FALSE),main="Nonseasonal 111"),
  autoplot(Pacf(seasonal_111011$residuals,lag.max=60,plot=FALSE),main="Seasonal 111011"),
  autoplot(Pacf(nonseasonal_111$residuals,lag.max=60,plot=FALSE),main="Nonseasonal 111"),
  nrow=2,ncol=2
)
```




Answer: Both of the model seems to present residuals in random nature. The SARIMA model seems to reduce correlation that is outside the blue line.

## 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

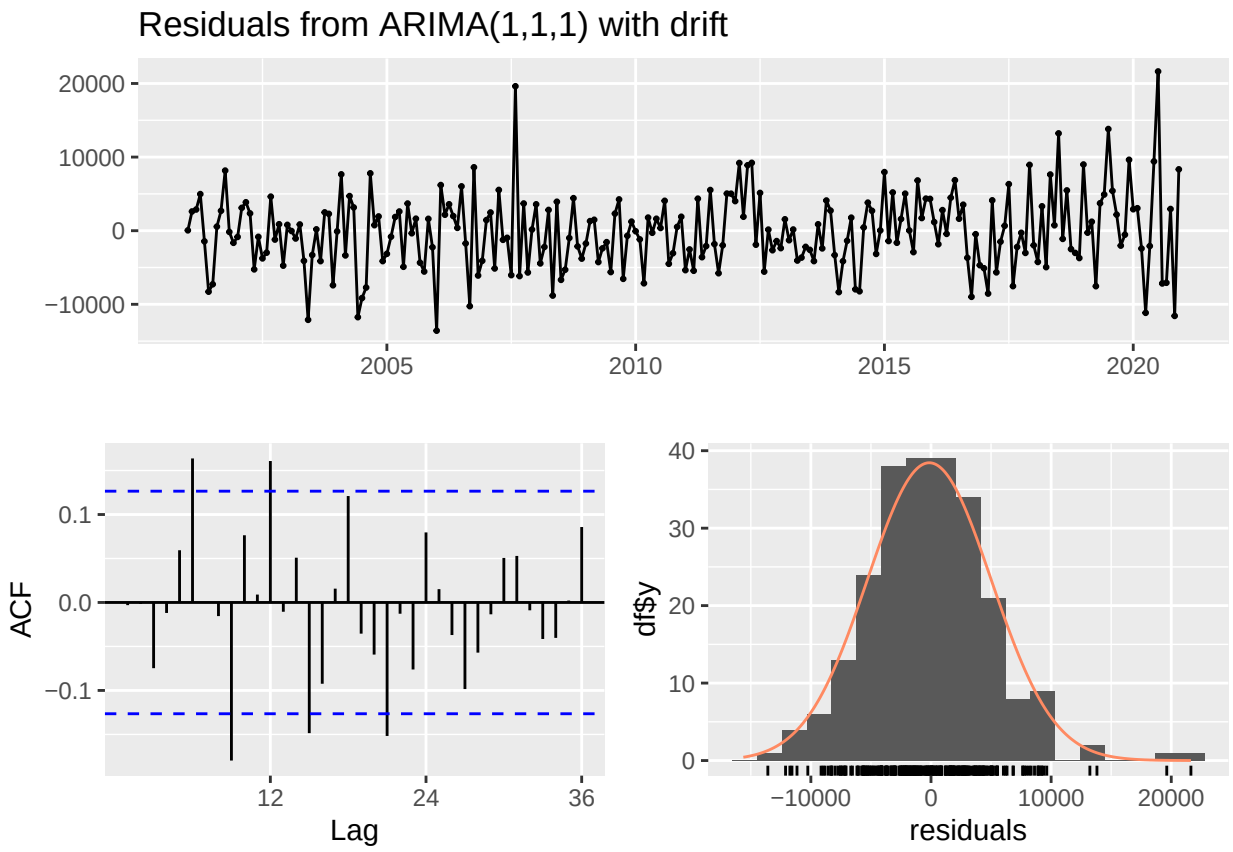
Use the *auto.arima*() command on the **deseasonalized series** to let R choose the model parameter for you. What's the order of the best ARIMA model? Does it match what you specified in Q4?

```
# Run the auto ARIMA model on the deseasonalized series
auto_ARIMA_deseason <- auto.arima(deseason_netgen)
print(auto_ARIMA_deseason)
```

```
## Series: deseason_netgen
## 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
```

```
# Plot the residuals of the auto ARIMA model on the deseasonalized series
checkresiduals(auto_ARIMA_deseason)
```




```
##
##  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
```

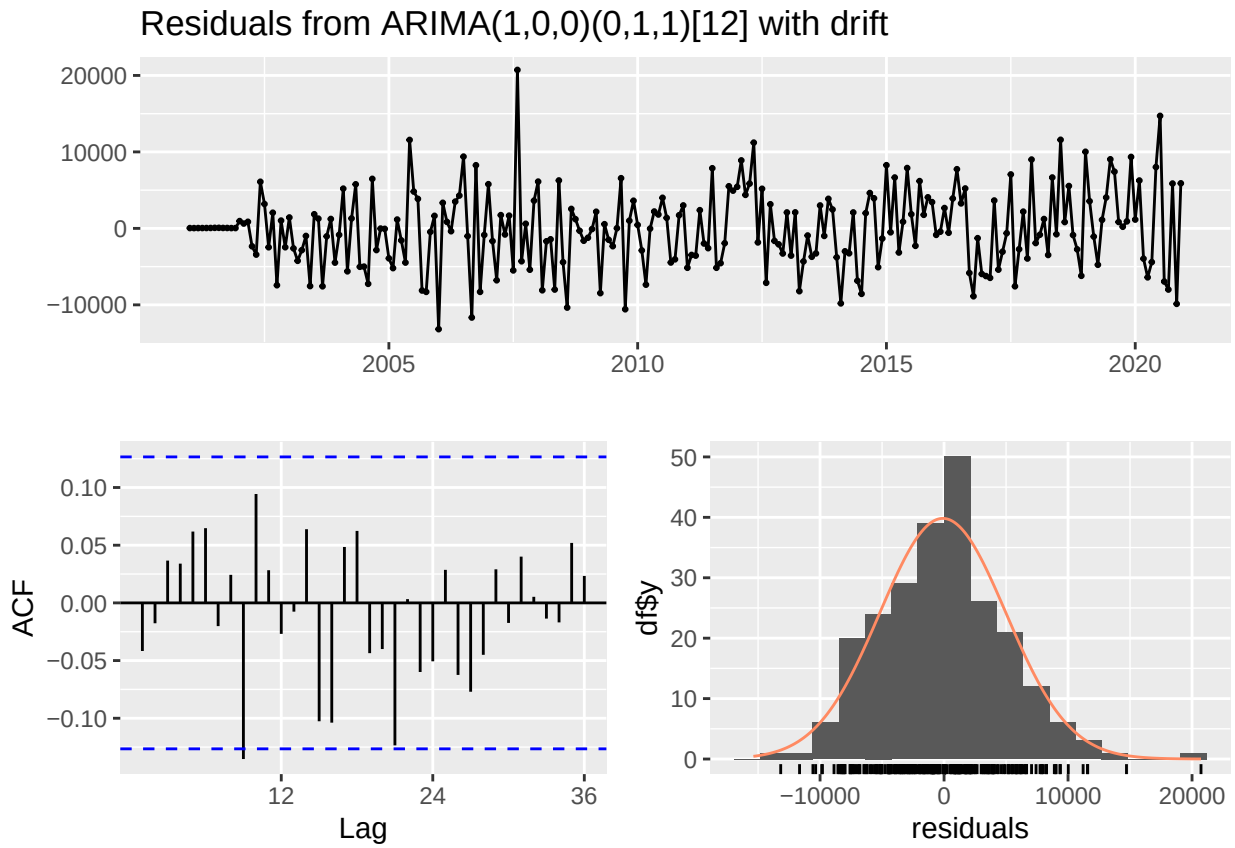
**Q10**

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?

```
# Run the auto ARIMA model on the deseasonalized series
auto_ARIMA_original <- auto.arima(ts_netgen)
print(auto_ARIMA_original)
```

```
## Series: ts_netgen
## 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
```

```
# Plot the residuals of the auto SARIMA model
checkresiduals(auto_ARIMA_original)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,1)[12] with drift
## Q* = 25.414, df = 22, p-value = 0.2777
##
## Model df: 2.   Total lags used: 24
```

```
# Compare AIC
seasonal_100011 <- Arima(ts_netgen,
                         order=c(1,0,0),
```

```
                      seasonal=c(0,1,1),
                      include.drift=FALSE)
print(seasonal_100011)
```

```
## Series: ts_netgen
## ARIMA(1,0,0)(0,1,1)[12]
##
## Coefficients:
##          ar1     sma1
##       0.9112  -0.6346
## s.e.  0.0333   0.0650
##
## sigma^2 = 30545109:  log likelihood = -2291.02
## AIC=4588.03   AICc=4588.14   BIC=4598.32
```

```
compare_aic <- data.frame(compare_aic, seasonal_100011$aic)

print(compare_aic)
```

```
##   nonseasonal_111.aic nonseasonal_112.aic nonseasonal_211.aic
## 1            4784.955            4786.376            4786.167
##   nonseasonal_212.aic seasonal_111011.aic seasonal_111110.aic
## 1            4787.572            4552.393            4583.117
##   seasonal_100011.aic
## 1            4588.032
```

Answer: For the non-seasonal ARIMA model, what I specified in Q7 and what R auto-calculates match. However, for the seasonal ARIMA model, it does not match. I consider that for the non-seasonal part, I don't need to respecify and just focus on the seasonal part, thereby it ends up with SARIMA(111011). When R auto calculates, it produces (100011) model. I consider that the model I specify could also work as it has a low AIC value and the residuals also look random.