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

Assignment 7 - Due date 03/06/25

Jingze Dai

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(forecast)
```

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

```
library(tseries)
library(cowplot)
library(ggplot2)
library(Kendall)
```

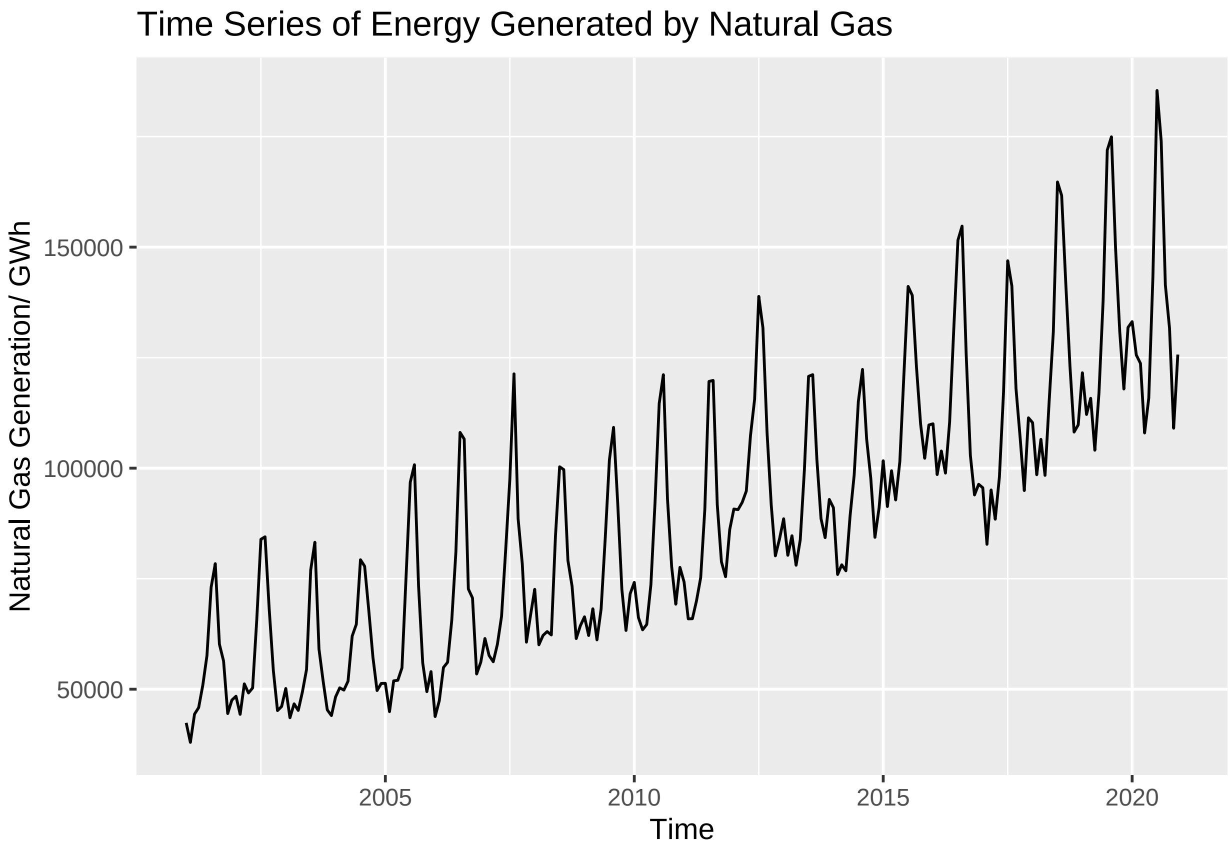
Importing and processing the data set

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**.

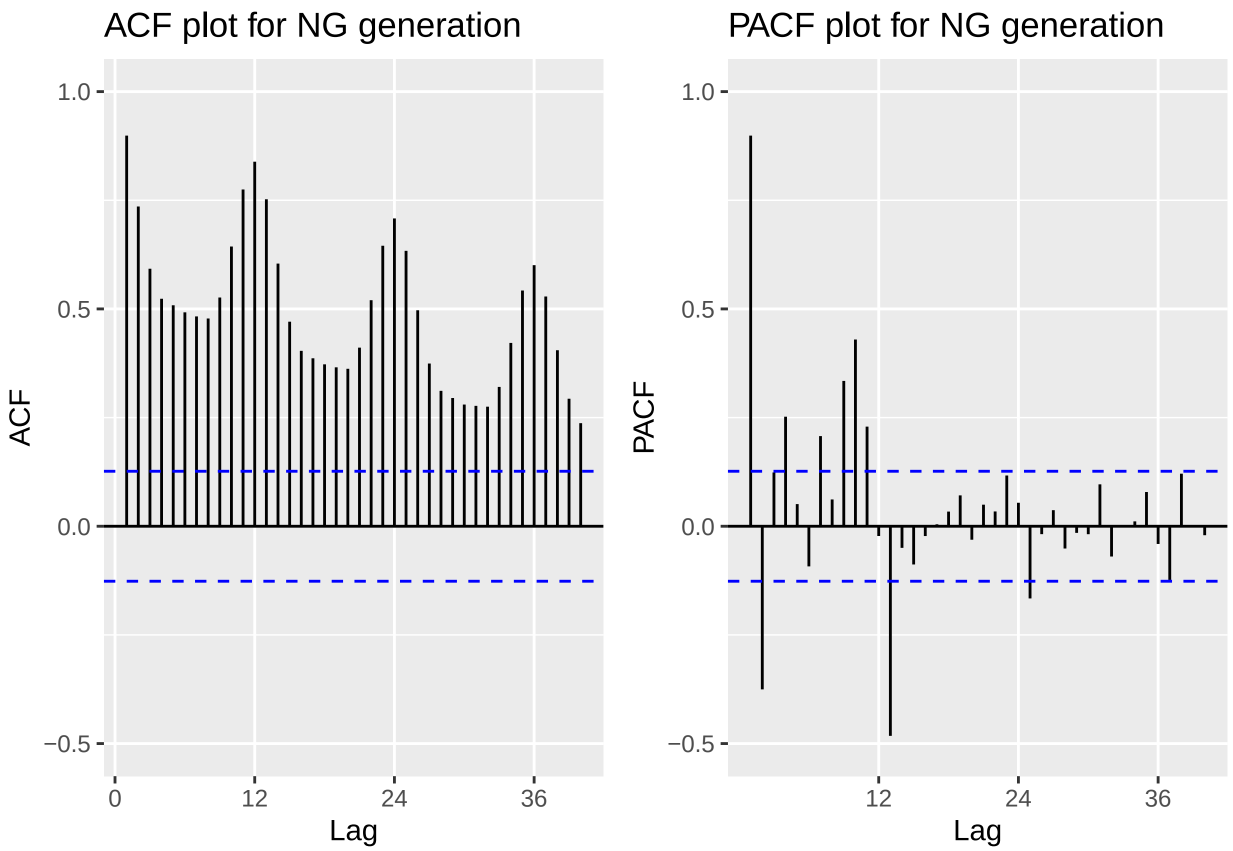
Q1

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
generation_data <- read.csv(
  "../Data/Net_generation_United_States_all_sectors_monthly.csv",
  skip = 4, header = TRUE)

ng_data <- rev(generation_data$natural.gas.thousand.megawatthours)
ng_ts <- ts(ng_data, start = c(2001,1), frequency = 12)


orig <- autoplot(ng_ts,
                 main = "Time Series of Energy Generated by Natural Gas",
                 ylab = "Natural Gas Generation/ GWh")

orig_acf <- autoplot(Acf(ng_ts, lag=40, plot = FALSE), ylim=c(-0.5,1),
                     main="ACF plot for NG generation")

orig_pacf <- autoplot(Pacf(ng_ts, lag=40, plot = FALSE), ylim=c(-0.5,1),
                      main="PACF plot for NG generation")

orig
```



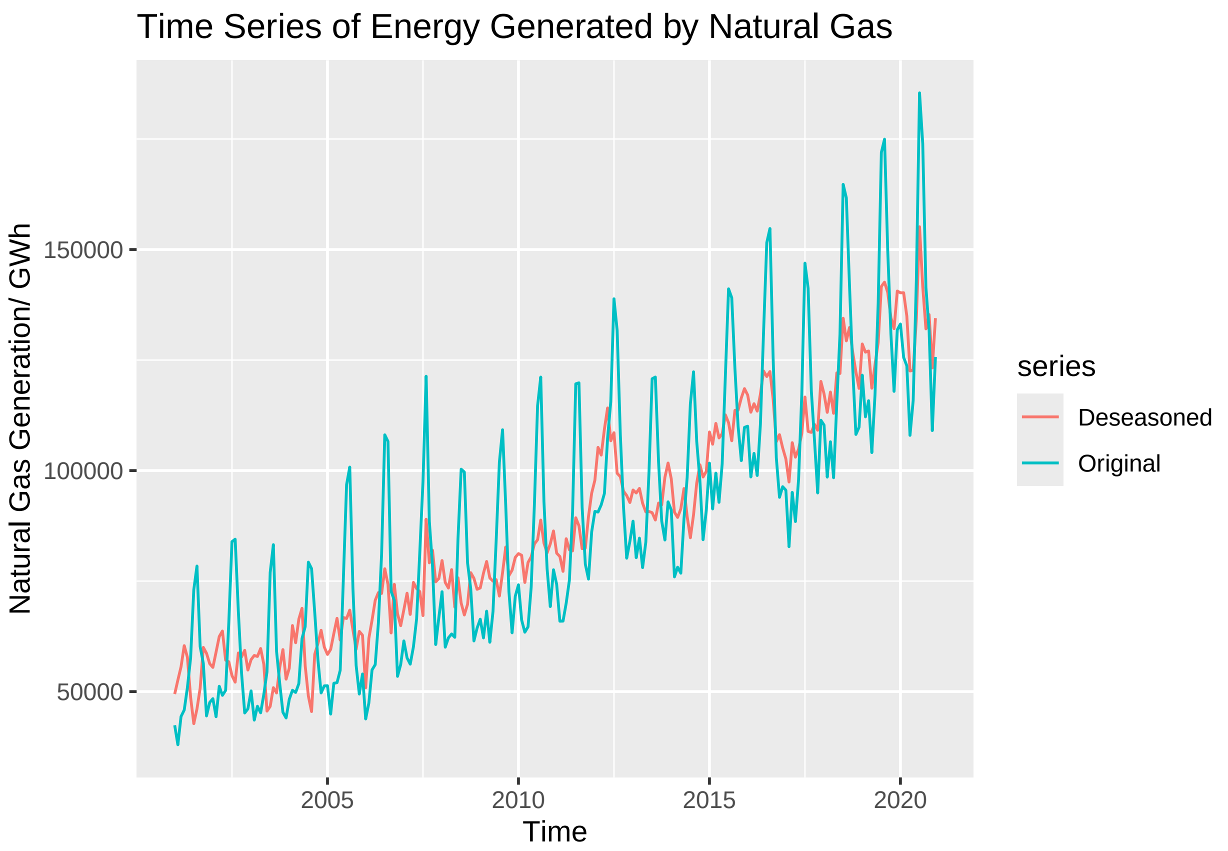
```r
plot_grid(orig_acf, orig_pacf)
```



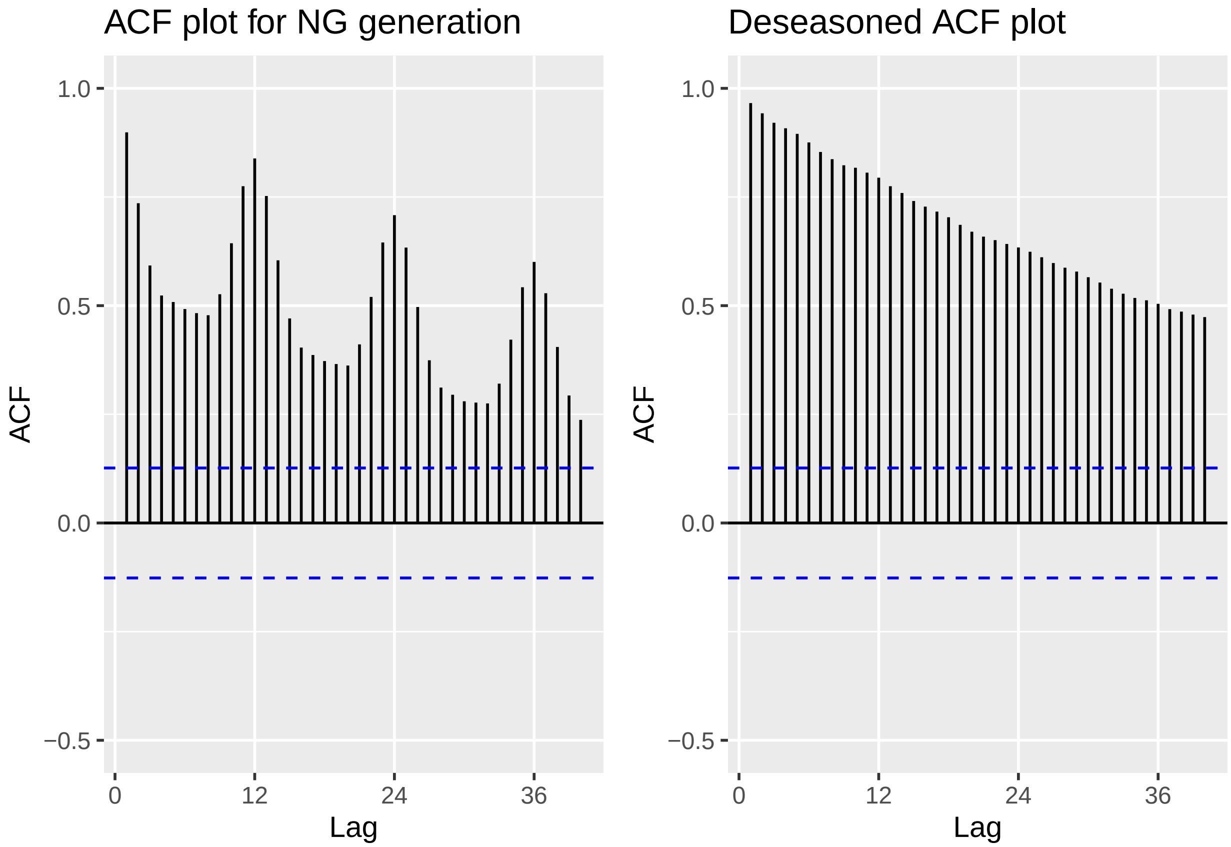
Q2

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.

```r
# decomposing ng_ts
decomposed_ng <- decompose(ng_ts)
# deseasoning decomposed_ng
deseasoned_ng <- seasadj(decomposed_ng)

autoplot(deseasoned_ng, series = "Deseasoned") +
  autolayer(ng_ts, series = "Original") +
  ylab("Natural Gas Generation/ GWh") +
  ggtitle("Time Series of Energy Generated by Natural Gas")
```



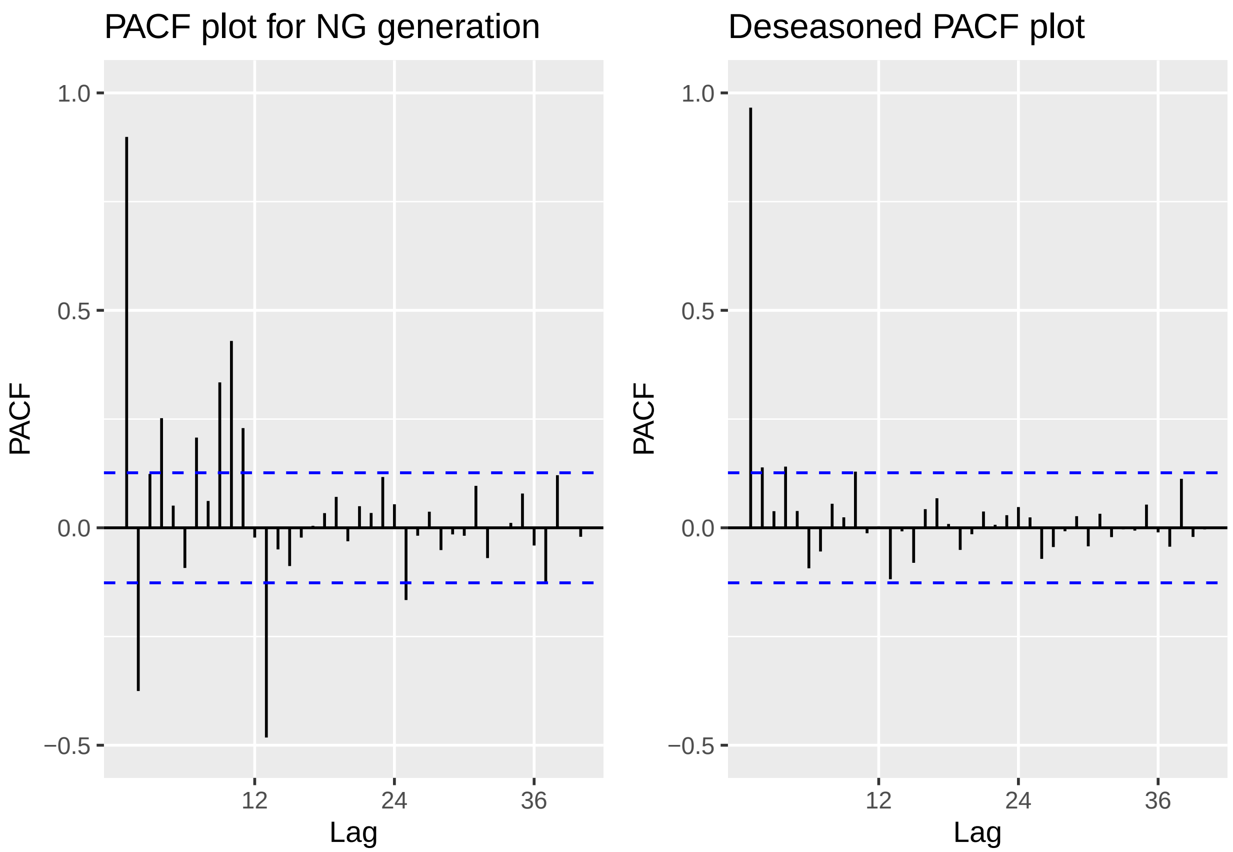
```
desea_acf <- autoplot(Acf(deseasoned_ng, lag=40, plot = FALSE),
                      ylim=c(-0.5,1),
                      main="Deseasoned ACF plot")

desea_pacf <- autoplot(Pacf(deseasoned_ng, lag=40, plot = FALSE),
                       ylim=c(-0.5,1),
                       main="Deseasoned PACF plot")

plot_grid(orig_acf, desea_acf)
```



ACF plot for NG generation

Deseasoned ACF plot

```r
plot_grid(orig_pacf, desea_pacf)
```



> Answer: Based on the time series plots, we can observe that the seasonal pattern has been removed in the deseasoned series, since the up and down fluctuations do not have a unit period. For the ACF plot, it is evident that the seasonal variations are replaced with gradual decrease in its ACF values. For the PACF plot, significant values at the second season (lag=13) has also been removed in the deseasoned PACF plot.

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 from ADF Test")
```

```
## [1] "Results from ADF Test"
```

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

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

```
# Mann-Kendall test
print("Results from Mann-Kendall Test")
```

```
## [1] "Results from Mann-Kendall Test"
```

```
summary(MannKendall(deseasoned_ng))
```

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

> Answer: For the ADF test, the p-value is 0.01, less than 0.05, meaning that the alternative hypothesis that the time series is stationary (without stochastic trend) is accepted. This indicates that the trend component is stationary and does not change over time. This can be seen from the uniform increase in the time series plot in Q2.
>
> For the Mann-Kendall test, the p-value is less than 0.05, indicating a significant trend. The Mann-Kendall test score is 24186, a positive number, indicating a strong increasing deterministic trend. They agree with the deseasoned plot in Q2, as we indeed see an increasing trend.

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.

```
# differencing the series once
differenced_ng <- diff(deseasoned_ng, lag = 1, differences = 1)

# checking whether the deterministic trend is removed
print("Results from Mann-Kendall Test")
```

```
## [1] "Results from Mann-Kendall Test"
```

```
summary(MannKendall(differenced_ng))
```

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

```
print("Results from ADF Test")
```

```
## [1] "Results from ADF Test"
```

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

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

> Ans: From the PACF and ACF plots in Q2, the deseasoned time series has a clear cut-off value at lag=1 in the PACF plot, this indicates that the order of the AR process is 1. The ACF plot shows a slowly decaying trend, thus the MA process is not present in this time series. As for the differencing order, after performing differencing once, the Mann-Kendall test gave a p-value larger than 0.05, indicating that the deterministic trend has been removed by differencing the series. Moreover, the ADF test still shows negative result for stochastic trend. Thus, the order of differencing is 1. Therefore, p=1, d=1, q=0

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.

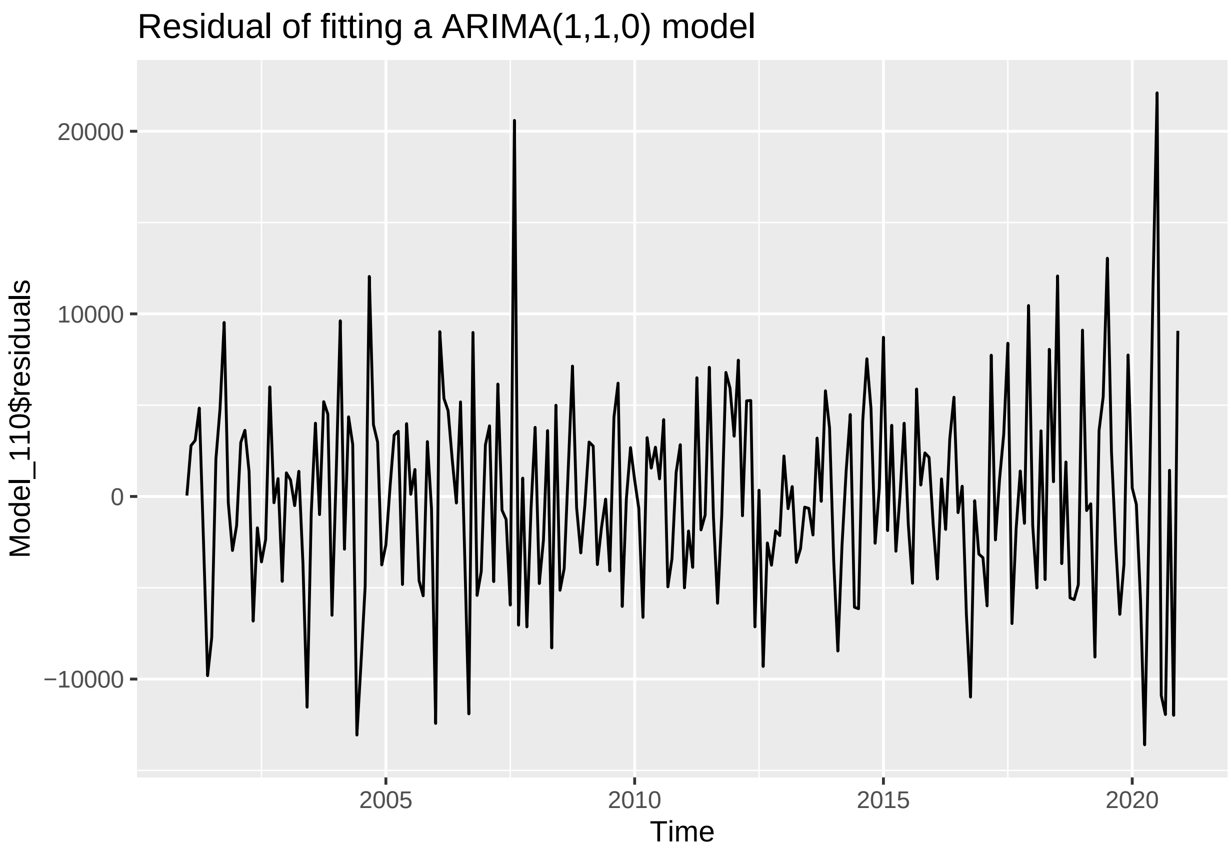
```
Model_110 <- Arima(deseasoned_ng,order=c(1,1,0),include.drift=TRUE)
print(Model_110)
```

```
## Series: deseasoned_ng
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1     drift
##       -0.1479  348.3927
## s.e.   0.0644  308.8385
##
## sigma^2 = 30254066:  log likelihood = -2396.54
## AIC=4799.07   AICc=4799.18   BIC=4809.5
```

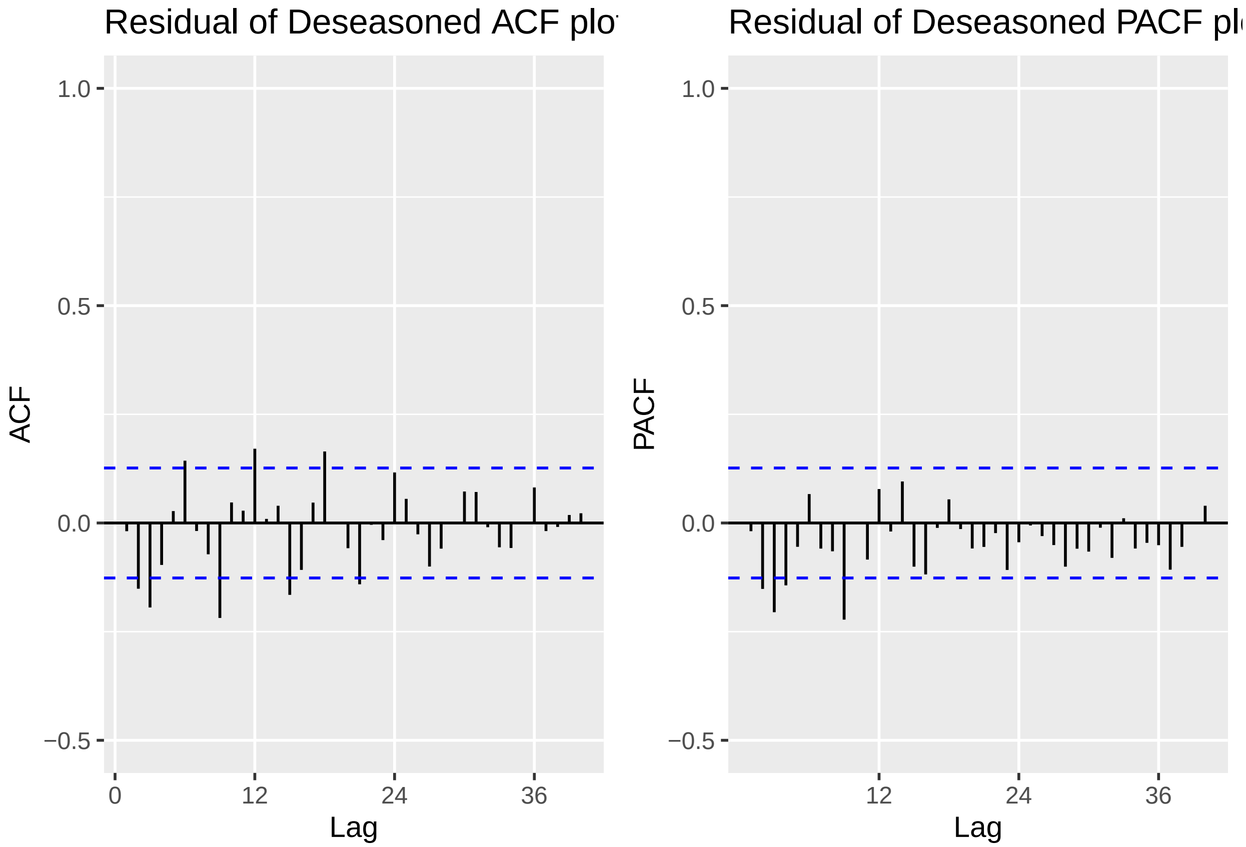
Q6

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?

```
autoplot(Model_110$residuals, main = "Residual of fitting a ARIMA(1,1,0) model")
```



```
plot_grid(
  autoplot(Acf(Model_110$residuals,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Residual of Deseasoned ACF plot"),
  autoplot(Pacf(Model_110$residuals,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Residual of Deseasoned PACF plot"),
  nrow=1)
```



> Ans: Yes, they look like white noise. Based on the plot of the residual of the ARIMA(1,1,0) model, there is no significant trend or period of the residual, thus it appears like a white noise. Moreover, both of the residual's ACF and PACF plots do not have notably significant values and on average they have a value of zero, proving that the residual itself does not have any AR or MA behavior, appearing like white noises.

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.

```
# Repeating Q3
# ADF test
print("Results from ADF Test")
```

```
## [1] "Results from ADF Test"
```

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

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

```
# Mann-Kendall test
print("Results from Seasonal Mann-Kendall Test")
```

```
## [1] "Results from Seasonal Mann-Kendall Test"
```

```
summary(SeasonalMannKendall(ng_ts))
```

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

Ans: Both ADF and Seasonal Mann-Kendall test gave the same result as the deseasoned series: ADF test here indicates no stochastic trend, and the Seasonal Mann-Kendall test indicates a significant deterministic trend.

```
# Repeating Q4
# differencing the series once
differenced_ng_orig <- diff(ng_ts, lag = 1, differences = 1)

# checking whether the deterministic trend is removed
print("Results from Seasonal Mann-Kendall Test")
```

```
## [1] "Results from Seasonal Mann-Kendall Test"
```

```
summary(SeasonalMannKendall(differenced_ng_orig))
```

```
## Score =  27 , Var(Score) = 11267
## denominator =  2261
## tau = 0.0119, 2-sided pvalue =0.79921
```

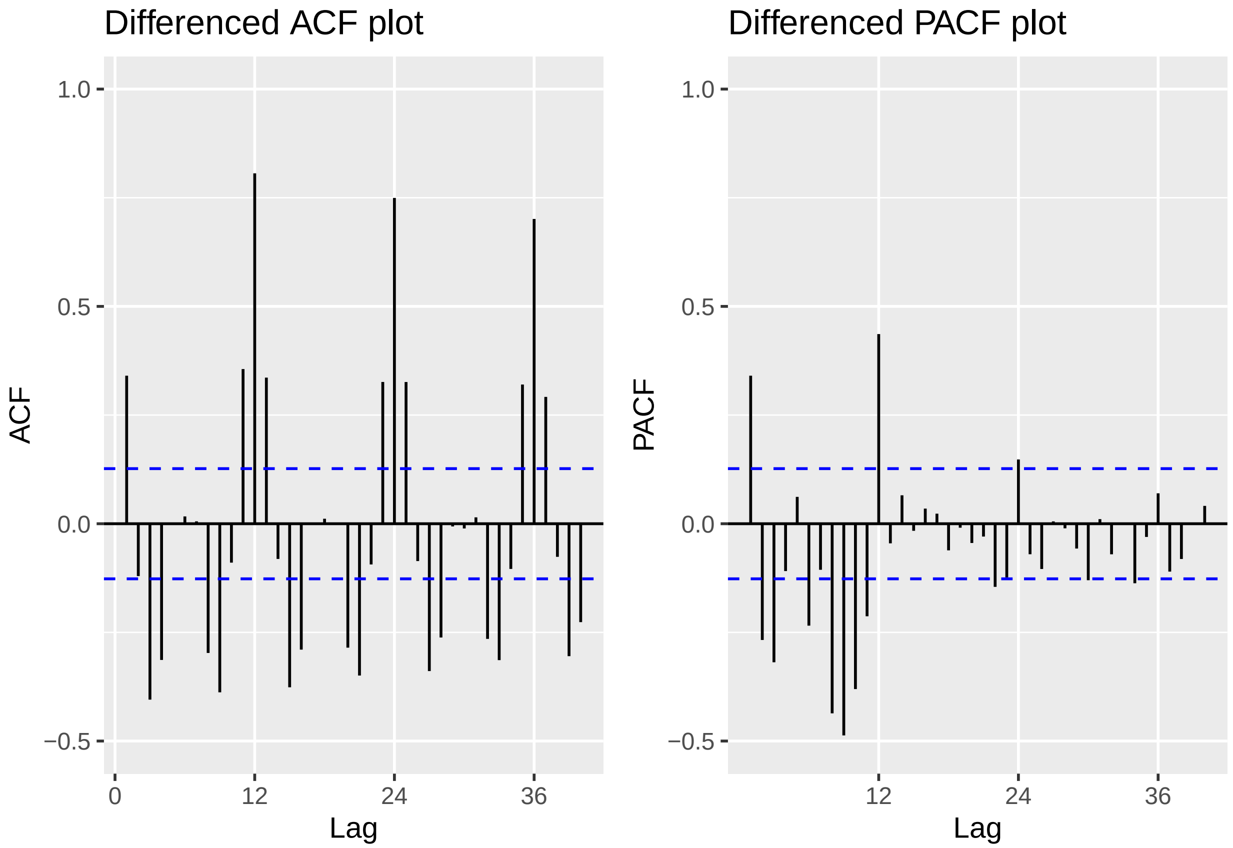
```
print("Results from ADF Test")
```

```
## [1] "Results from ADF Test"
```

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

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

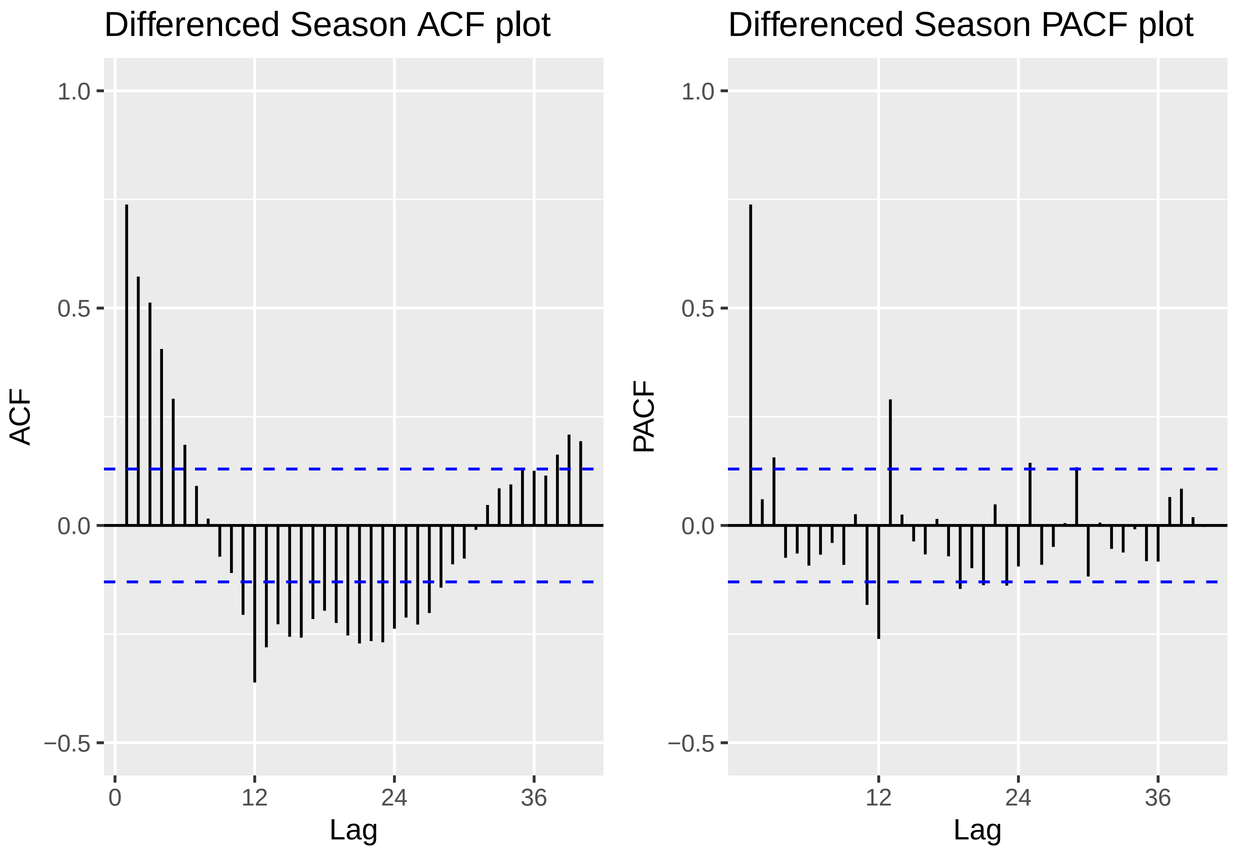
```
plot_grid(
  autoplot(Acf(differenced_ng_orig,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Differenced ACF plot"),
  autoplot(Pacf(differenced_ng_orig,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Differenced PACF plot"),
  nrow=1)
```



```
# checking whether need to difference seasonality
ns_diff <- nsdiffs(ng_ts)
cat("Number of seasonal differencing needed: ",ns_diff)
```

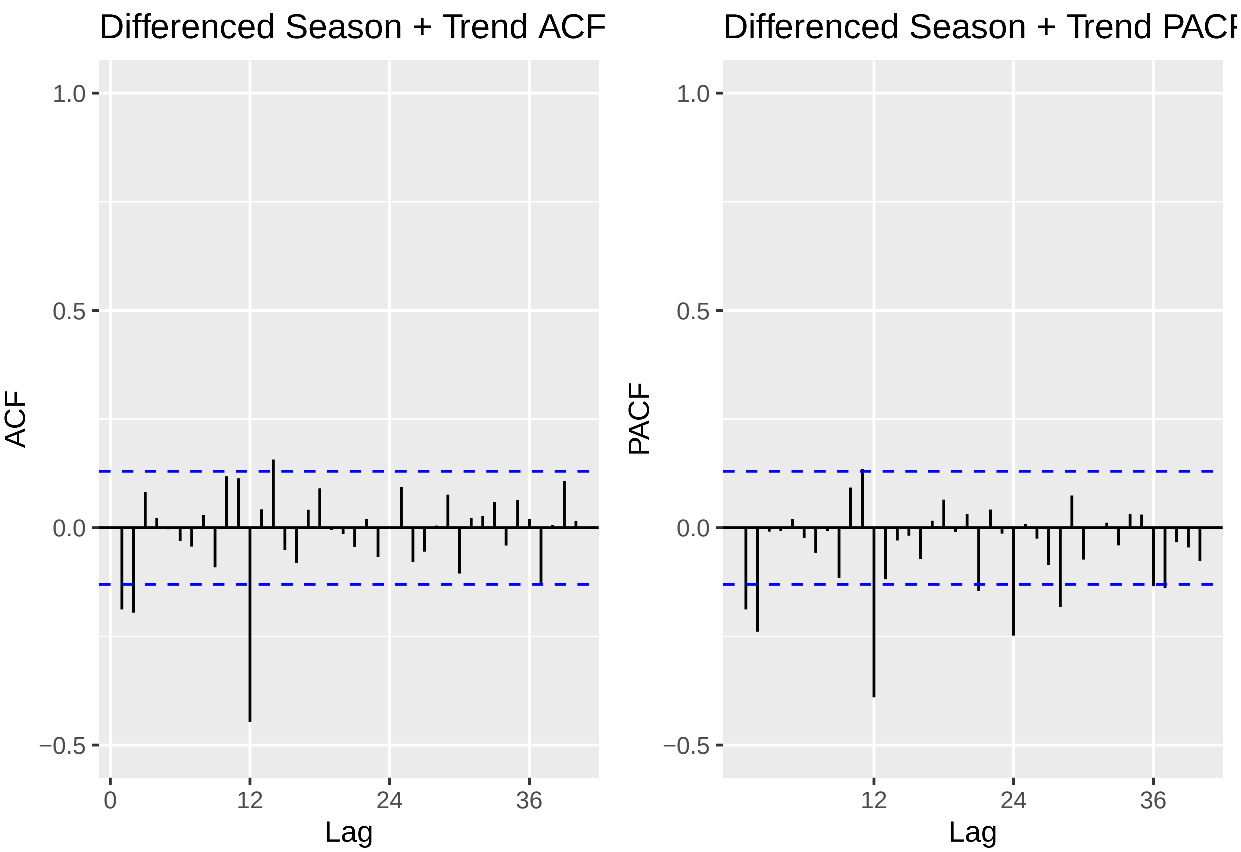
```
## Number of seasonal differencing needed:  1
```

```
# differencing the seasonality once
differenced_ng_season <- diff(ng_ts, lag = 12, differences = 1)

plot_grid(
  autoplot(Acf(differenced_ng_season,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Differenced Season ACF plot"),
  autoplot(Pacf(differenced_ng_season,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
          main="Differenced Season PACF plot"),
  nrow=1)
```


Differenced Season ACF plot

Differenced Season PACF plot

```
# differencing both the series and seasonality
differenced_ng_both <- diff(differenced_ng_orig, lag = 12, differences = 1)

plot_grid(
  autoplot(Acf(differenced_ng_both,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Differenced Season + Trend ACF"),
  autoplot(Pacf(differenced_ng_both,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Differenced Season + Trend PACF"),
  nrow=1)
```

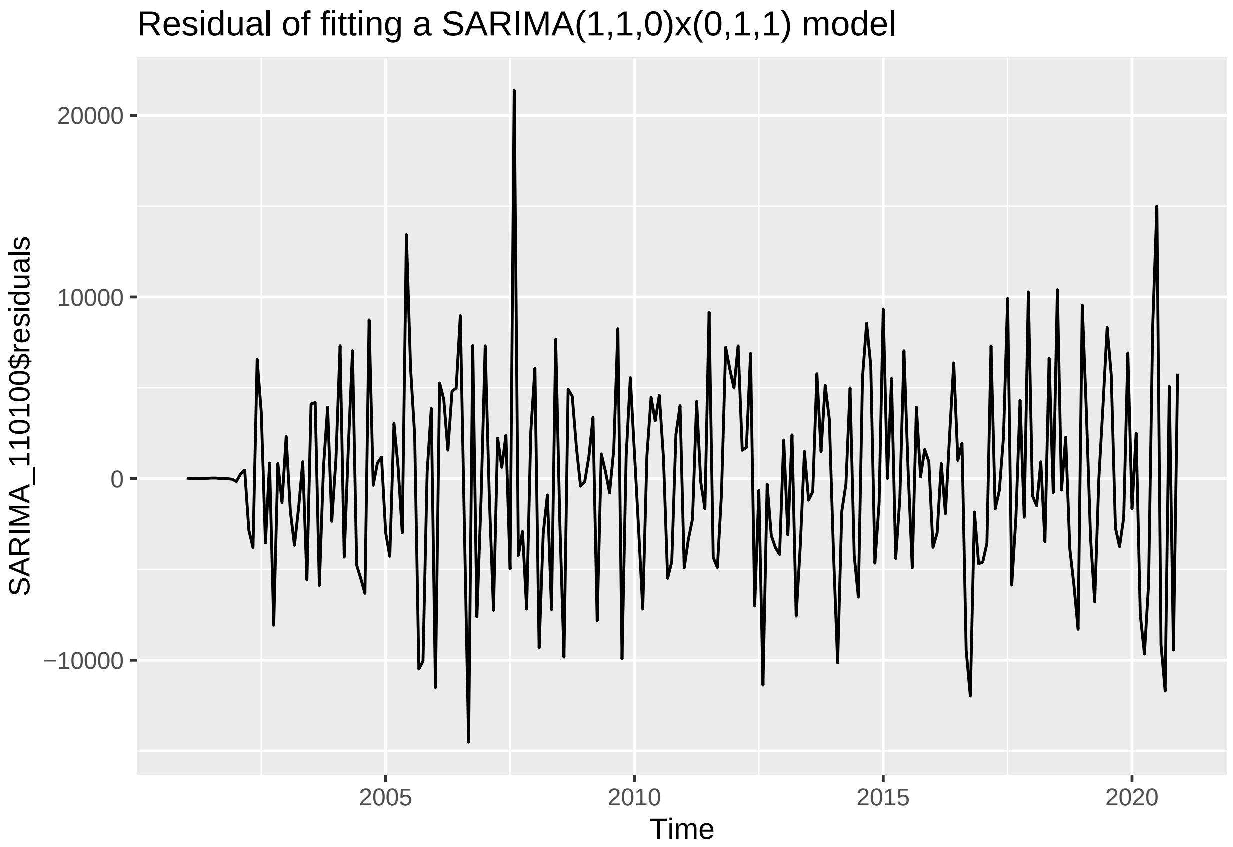


> Ans: After differencing the original series once, the deterministic trend was eliminated as the Seasonal Mann-Kendall test gave a p-value larger than 0.05. The series still do not have stochastic trend. It is thus ready to be fit in an SARIMA model. Based on the ACF plot of the original series, no MA component can be seen. Therefore p=1, d=1, q=0. Based on the ACF and PACF plots of the twice-differenced series, it can be observed that there is only one significant value in ACF at lag=12,, while there are two spikes of PACF at lag=12 and lag=24. This indicates a seasonal MA component, thus P=0 and Q=1. We also found that it is necessary to difference the seasonality once, thus D=1. The final SARIMA model will be SARIMA(1,1,0)x(0,1,1) with seasonality s=12.

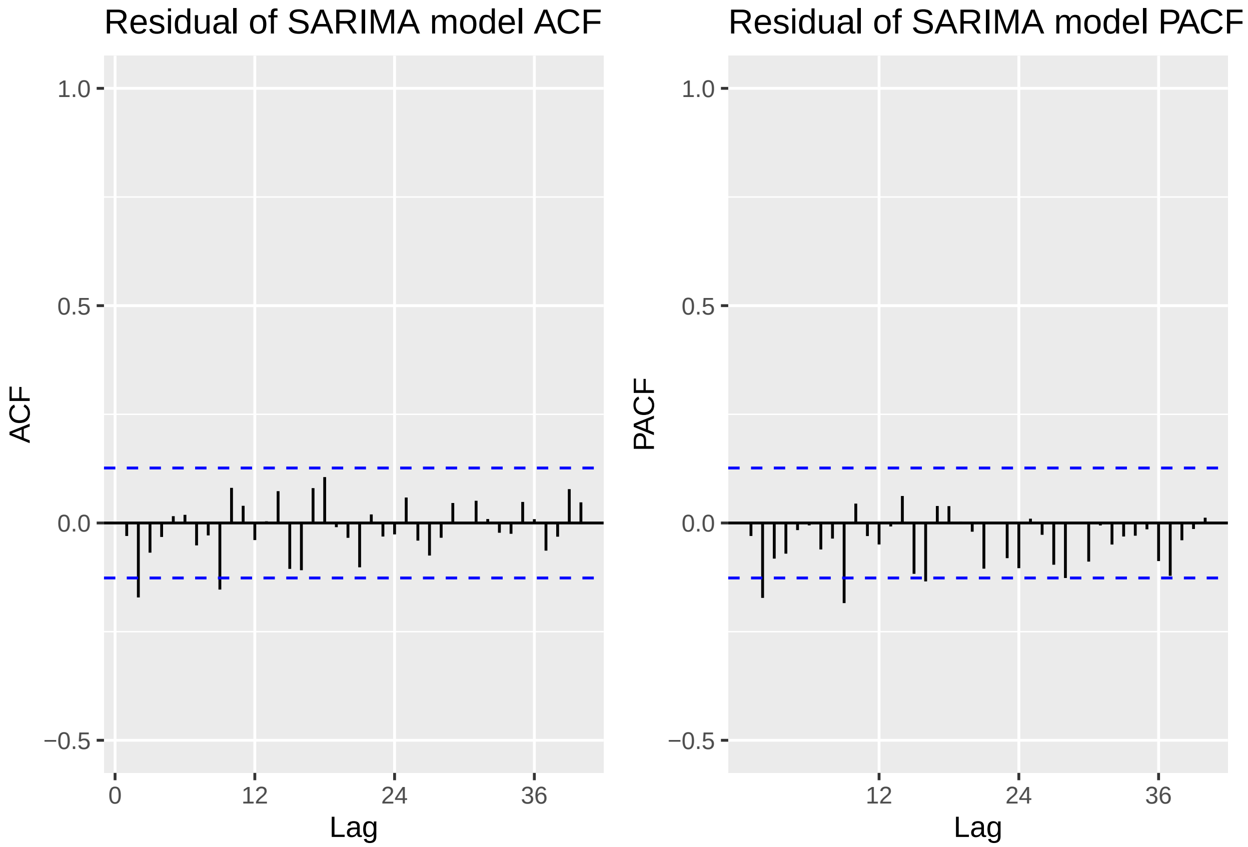
```r
# Repeating Q5 and Q6
SARIMA_110100 <- Arima(ng_ts,
                       order=c(1,1,0),
                       seasonal=c(0,1,1),
                       include.drift=FALSE)

print(SARIMA_110100)
```

```
## Series: ng_ts
## ARIMA(1,1,0)(0,1,1)[12]
##
## Coefficients:
##           ar1     sma1
##       -0.1808  -0.6898
## s.e.   0.0655   0.0557
##
## sigma^2 = 30626308:  log likelihood = -2281.43
## AIC=4568.86   AICc=4568.96   BIC=4579.13
```

```
autoplot(SARIMA_110100$residuals,
         main = "Residual of fitting a SARIMA(1,1,0)x(0,1,1) model")
```



```
plot_grid(
  autoplot(Acf(SARIMA_110100$residuals,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Residual of SARIMA model ACF"),
  autoplot(Pacf(SARIMA_110100$residuals,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Residual of SARIMA model PACF"),
  nrow=1)
```

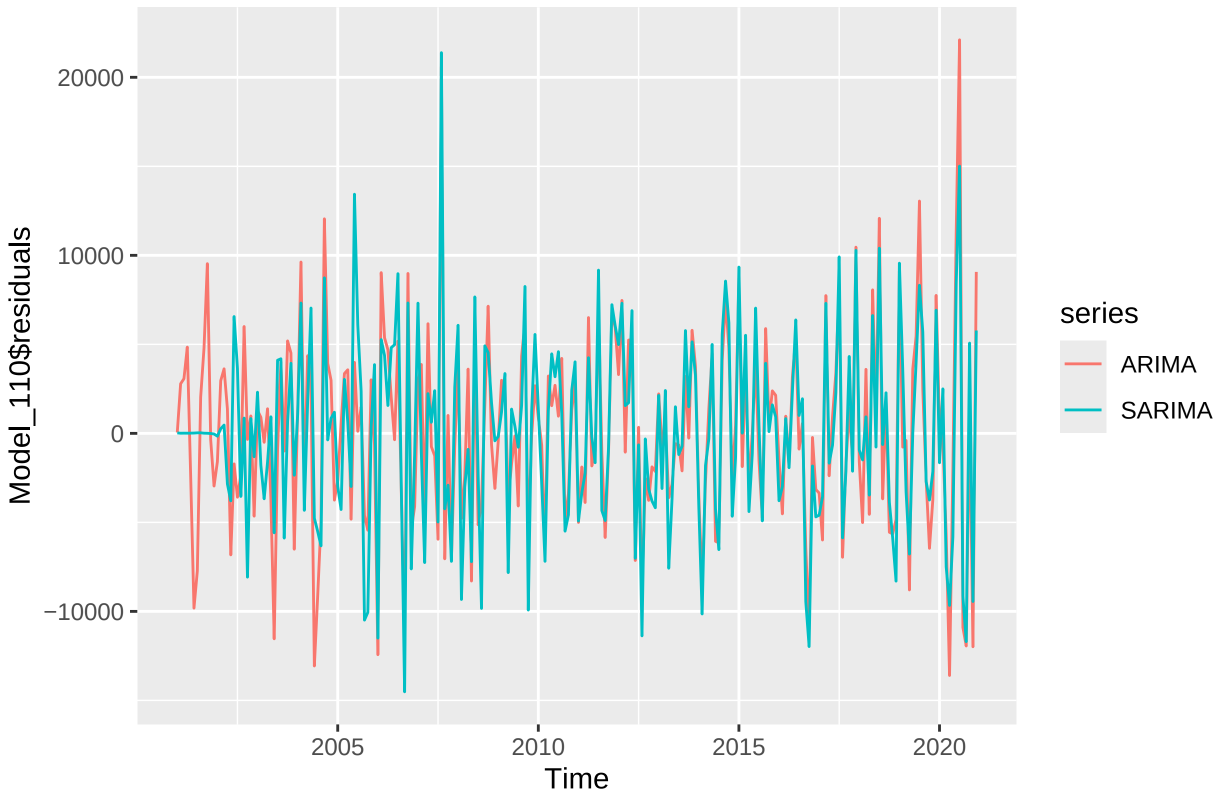


> Ans: Note that drift is unnecessary here since the twice-differenced series already removed the constant. The residual looks like white noise, because the plot does not have any trend or period, and the PACF and ACF do not have notably significant values, with means being close to zero.

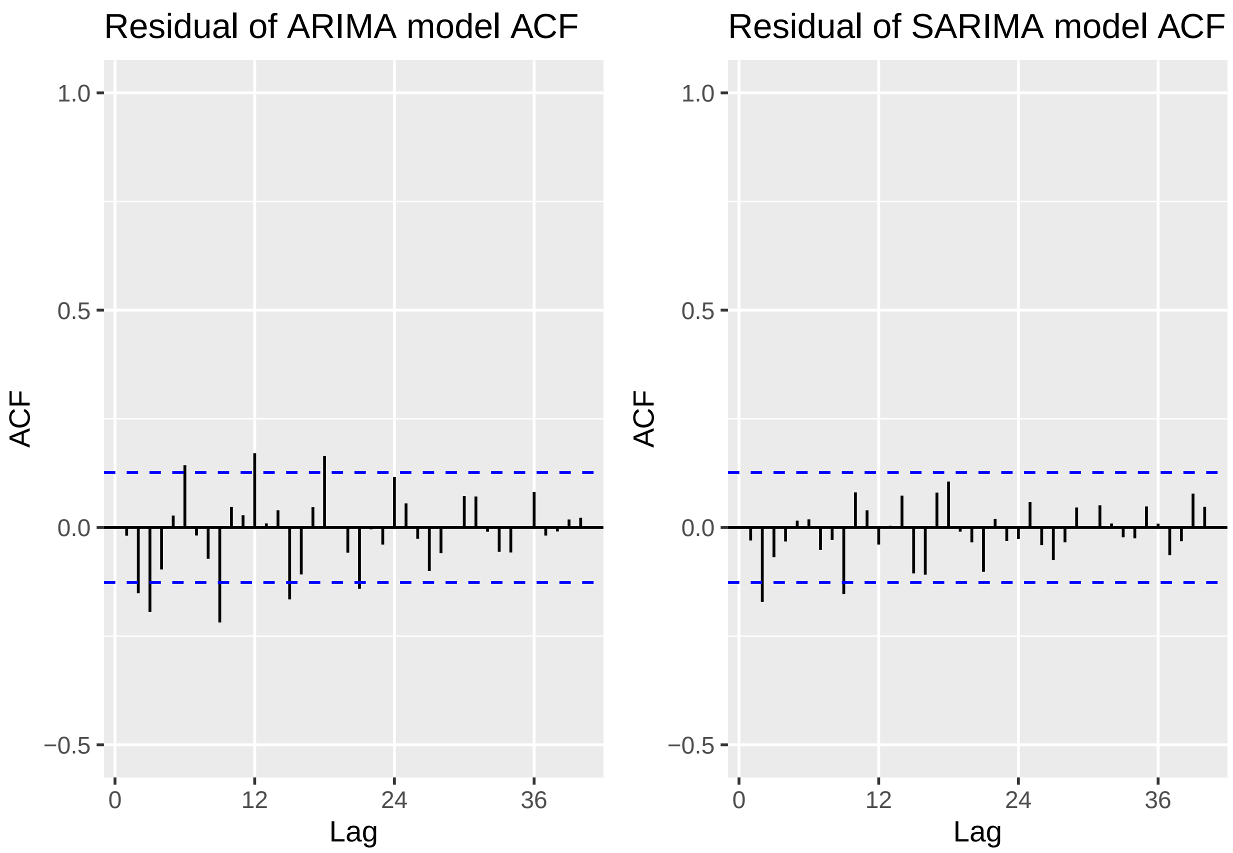
Q8

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.

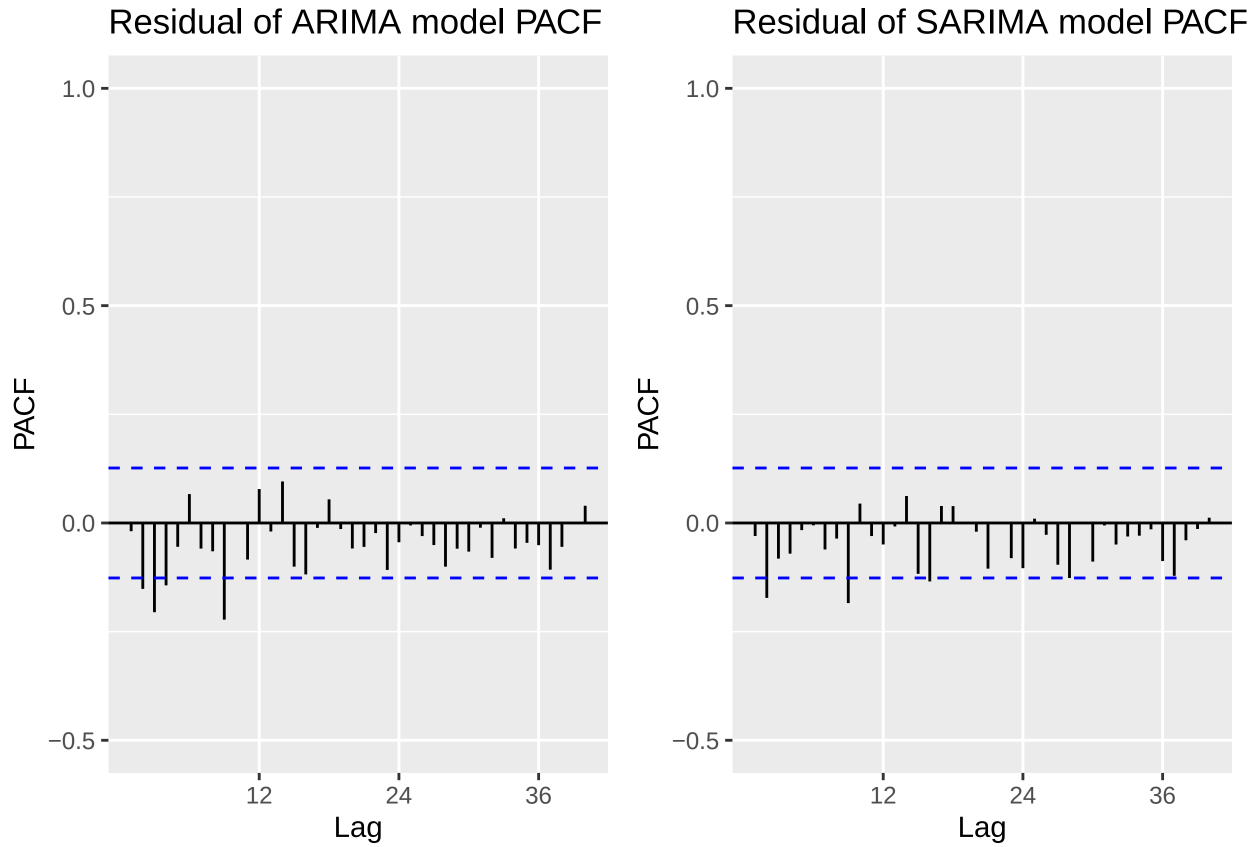
```
autoplot(Model_110$residuals, series = "ARIMA") +
  autolayer(SARIMA_110100$residuals, series = "SARIMA")
```



```
plot_grid(
  autoplot(Acf(Model_110$residuals,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Residual of ARIMA model ACF"),
   autoplot(Acf(SARIMA_110100$residuals,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Residual of SARIMA model ACF"),
  nrow=1)
```



```
plot_grid(
  autoplot(Pacf(Model_110$residuals,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Residual of ARIMA model PACF"),
  autoplot(Pacf(SARIMA_110100$residuals,lag.max=40, plot = FALSE),
           ylim=c(-0.5,1),
           main="Residual of SARIMA model PACF"),
  nrow=1)
```



> Ans: It is unable to tell which model is better from these comparisons. It is unfair because essentially they are comparing white noises, which does not suggest anything about the quality of the models. We need to compare AIC values instead.

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

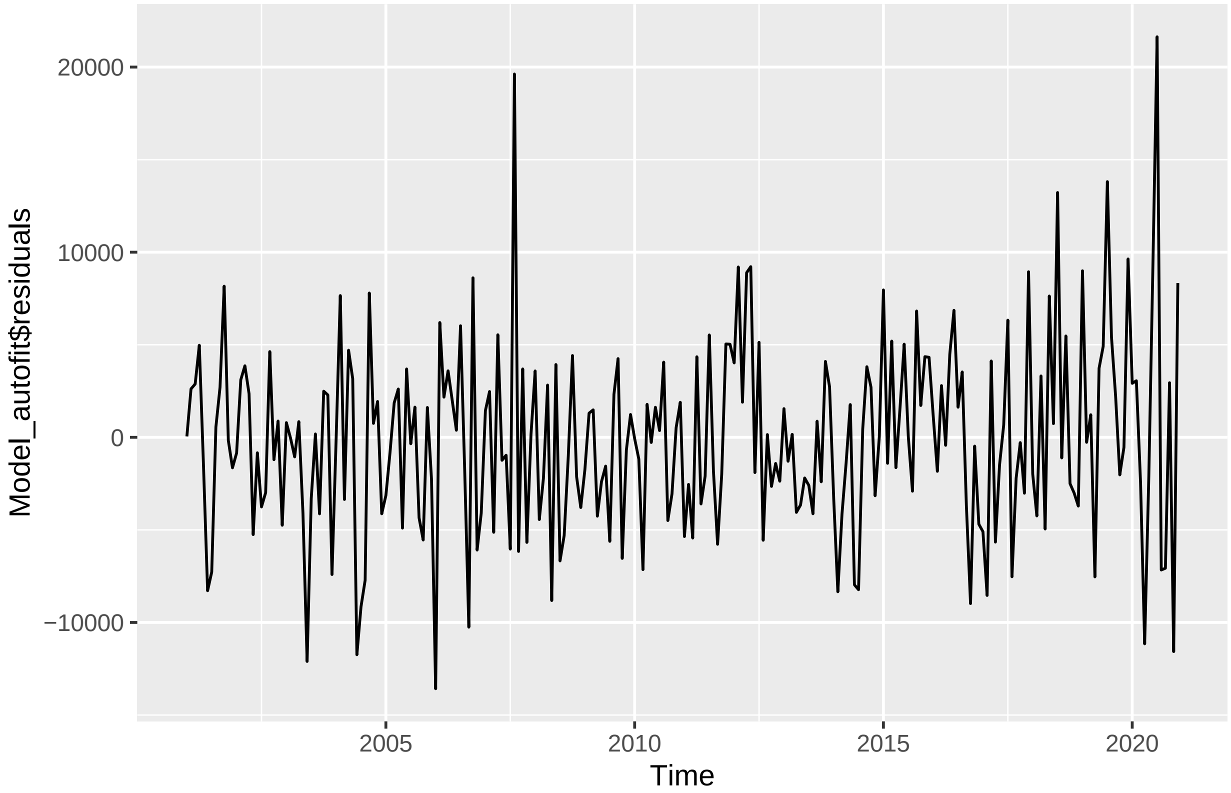
Use the *auto.arima*() command on the **deseasonalized series** to let R choose the model parameter for you. What's the order of the best ARIMA model? Does it match what you specified in Q4?

```
Model_autofit <- auto.arima(deseasoned_ng)
print(Model_autofit)
```

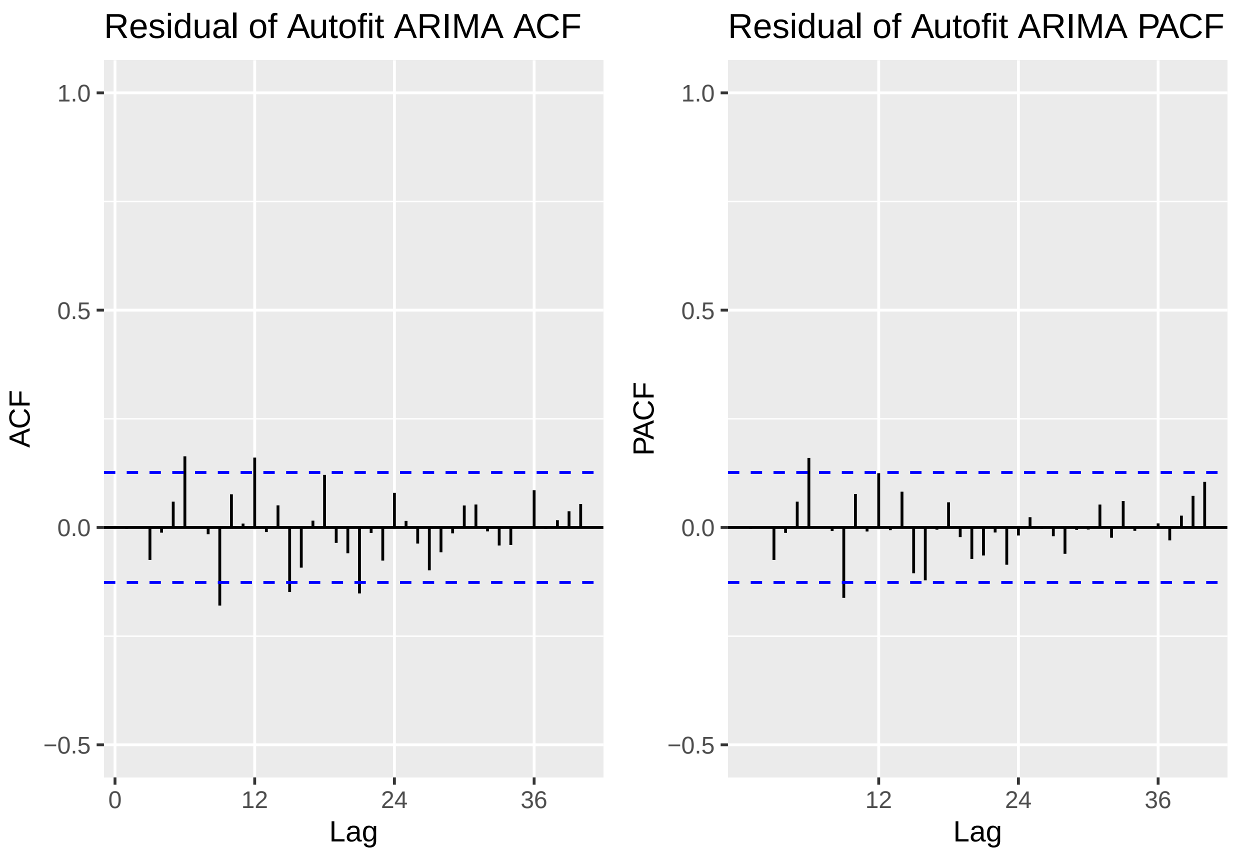
```
## Series: deseasoned_ng
## ARIMA(1,1,1) with drift
##
## Coefficients:
##          ar1      ma1     drift
##       0.7065  -0.9795  359.5052
## s.e.  0.0633   0.0326   29.5277
##
```

```
## sigma^2 = 26980609:  log likelihood = -2383.11
## AIC=4774.21   AICc=4774.38   BIC=4788.12
```

```
autoplot(Model_autofit$residuals)
```



```
plot_grid(
  autoplot(Acf(Model_autofit$residuals,lag.max=40,plot=FALSE),
           main="Residual of Autofit ARIMA ACF",
           ylim=c(-0.5,1)),
  autoplot(Pacf(Model_autofit$residuals,lag.max=40,plot=FALSE),
           main="Residual of Autofit ARIMA PACF",
           ylim=c(-0.5,1)),
  nrow=1)
```



No, the result does not match with what I have identified in Q4. The autofit function suggests that ARIMA(1,1,1) model is better in this case, with an AIC value of 4774.21, which is lower than my ARIMA(1,1,0) model's AIC value of 4799.07.

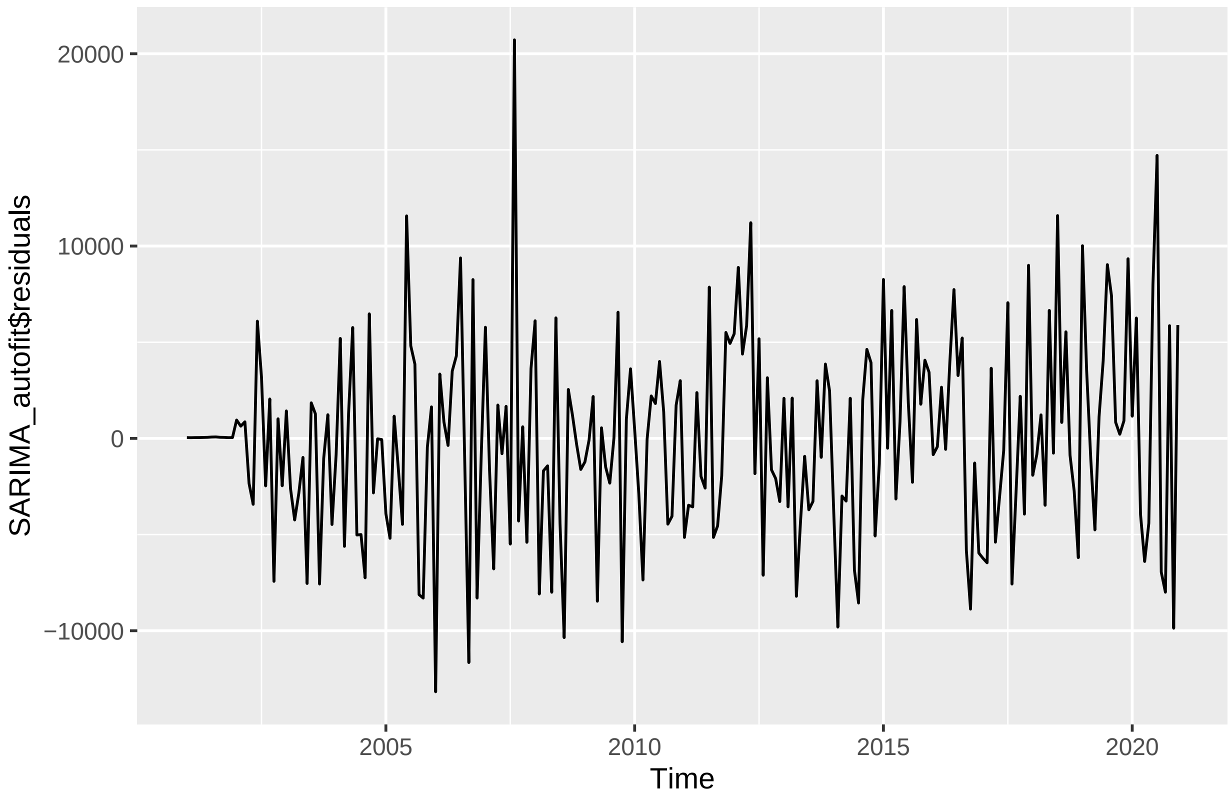
Q10

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?

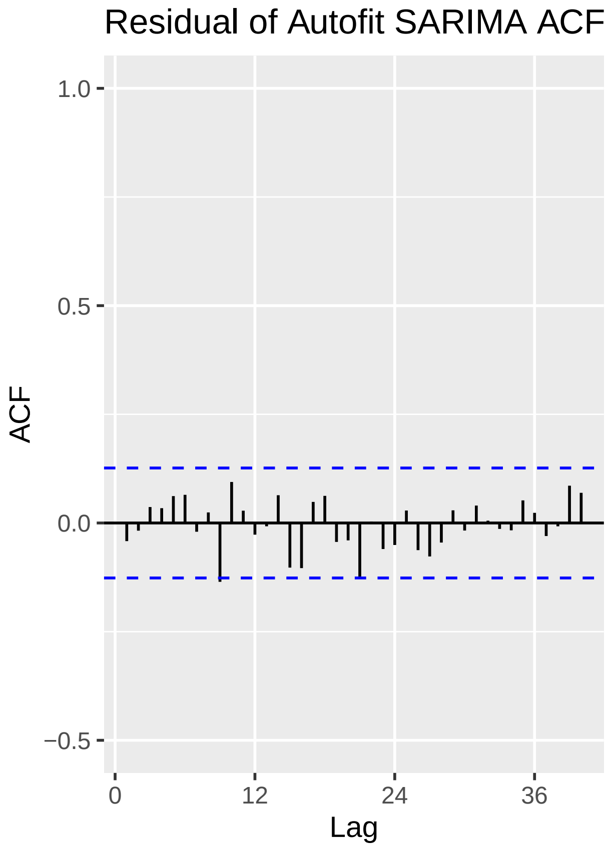
```
SARIMA_autofit <- auto.arima(ng_ts)
print(SARIMA_autofit)
```

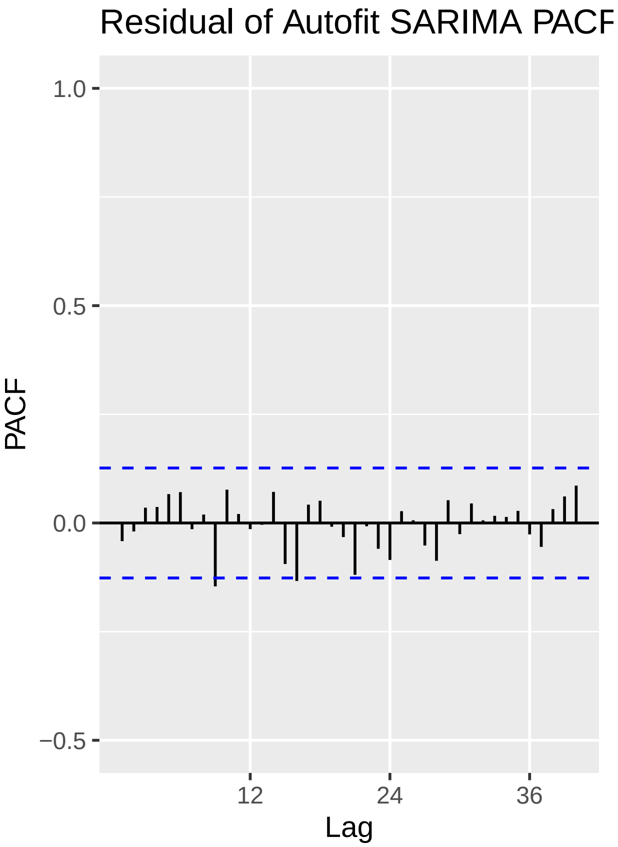
```
## Series: ng_ts
## ARIMA(1,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     sma1     drift
##       0.7416  -0.7026  358.7988
## s.e.  0.0442   0.0557   37.5875
##
## sigma^2 = 27569124:  log likelihood = -2279.54
## AIC=4567.08   AICc=4567.26   BIC=4580.8
```

```
autoplot(SARIMA_autofit$residuals)
```



```
plot_grid(
  autoplot(Acf(SARIMA_autofit$residuals,lag.max=40,plot=FALSE),
           main="Residual of Autofit SARIMA ACF",
           ylim=c(-0.5,1)),
  autoplot(Pacf(SARIMA_autofit$residuals,lag.max=40,plot=FALSE),
           main="Residual of Autofit SARIMA PACF",
           ylim=c(-0.5,1)),
  nrow=1)
```





No, the overall result for the best SARIMA model does not match with what I have identified in Q7. The autofit function suggests that SARIMA(1,0,0)(0,1,1) model with drift is better in this case, with an AIC value of 4567.08, which is lower than my SARIMA(1,1,0)(0,1,1) model's AIC value of 4568.86. The seasonal part, however, matched with what I have suggested in Q7.