ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2024

Assignment 7 - Due date 03/07/24

Sai Powar

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_Sp24.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

```r
library(lubridate)
library(ggplot2)
library(forecast)
library(Kendall)
library(tseries)
library(outliers)
library(tidyverse)
library(smooth)
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.

```
#Importing data
raw_gen <- read.table(file="./Data/Net_generation_United_States_all_sectors_monthly.csv",header=TRUE, d
head(raw_gen)
```

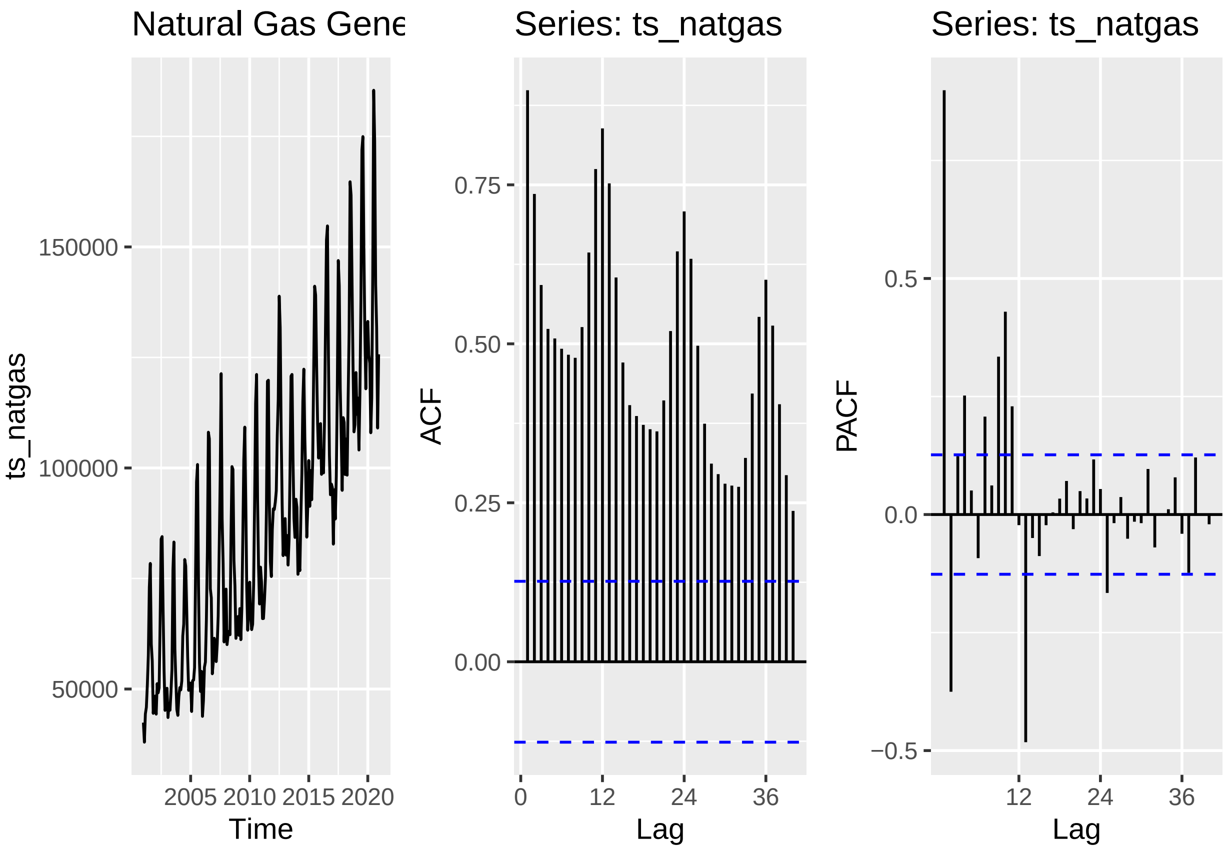
```
##      Month all.fuels..utility.scale..thousand.megawatthours
## 1 Dec 2020                                         344970.4
## 2 Nov 2020                                         302701.8
## 3 Oct 2020                                         313910.0
## 4 Sep 2020                                         334270.1
## 5 Aug 2020                                         399504.2
## 6 Jul 2020                                         414242.5
##   coal.thousand.megawatthours natural.gas.thousand.megawatthours
## 1                    78700.33                           125703.7
## 2                    61332.26                           109037.2
## 3                    59894.57                           131658.2
## 4                    68448.00                           141452.7
## 5                    91252.48                           173926.6
## 6                    89831.36                           185444.8
##   nuclear.thousand.megawatthours
## 1                       69870.98
## 2                       61759.98
## 3                       59362.46
## 4                       65727.32
## 5                       68982.19
## 6                       69385.44
##   conventional.hydroelectric.thousand.megawatthours
## 1                                          23086.37
## 2                                          21831.88
## 3                                          18320.72
## 4                                          19161.97
## 5                                          24081.57
## 6                                          27675.94
```

```
#creating usable data frame
natgas <- raw_gen[,c(1,4)]
natgas <-
  natgas %>%
  mutate( Month = my(Month) ) %>%
  rename( NatGas = natural.gas.thousand.megawatthours) %>%
  arrange(Month)

#creating time series object
ts_natgas <- ts(natgas[,2],start = c(2001,1),frequency=12)
head(ts_natgas)
```

```
##           Jan      Feb      Mar      Apr      May      Jun
## 2001 42388.66 37966.93 44364.41 45842.75 50934.21 57603.15
```

```
#plotting the series
plot1 <- plot_grid(
  autoplot(ts_natgas, main = "Natural Gas Generation"),
  autoplot(Acf(ts_natgas,lag.max=40,plot=FALSE)),
  autoplot(Pacf(ts_natgas,lag.max=40,plot=FALSE)),
  nrow=1,ncol=3
)
plot1
```

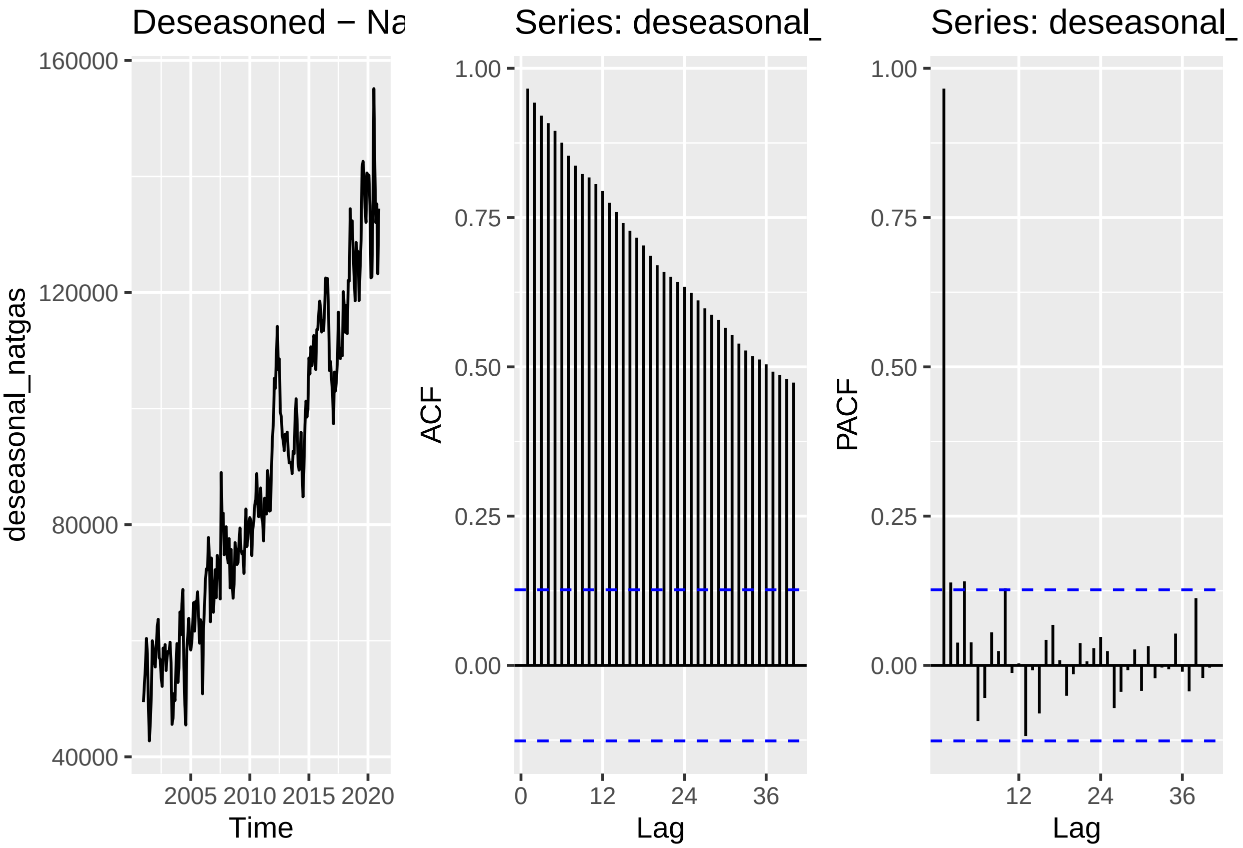


Q2

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.

```
#creating deseasoned series
decompose_natgas <- decompose(ts_natgas,"additive")
deseasonal_natgas <- seasadj(decompose_natgas)

#plotting the series
plot2 <- plot_grid(
  autoplot(deseasonal_natgas, main = "Deseasoned - Natural Gas Generation"),
  autoplot(Acf(deseasonal_natgas,lag.max=40,plot=FALSE)),
  autoplot(Pacf(deseasonal_natgas,lag.max=40,plot=FALSE)),
  nrow=1,ncol=3
)
plot2
```



Answer: After removing the seasonality, the spikes at the seasonal lags are not present in the ACF and the PACF plots. The exponential decay of the ACF indicates a deterministic, linear trend.

Q3

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

```
print("Results of Mann Kendall test/n")
```

```
## [1] "Results of Mann Kendall test/n"
```

```
print(summary(MannKendall(deseasonal_natgas)))
```

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

```r
print("Results for ADF test/n")
```

```
## [1] "Results for ADF test/n"
```

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

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

> Answer: Conclusion from MK test: The S is positive so there is an increasing trend. The p value is <0.05 so the null hypothesis is rejected and we have a significant trend in the series. Conclusion from ADF test: The p-value is <0.05 so we reject the null hypothesis. The data does not have a stochastic trend. Both the tests indicate that the series has an increasing, linear 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.

> Answer: Since the series has a significant trend, d=1. Based on the exponential decay in the ACF and the cutoff after lag 1 in the PACF plot, I predicted the model to have an AR component = p = 1. The MA component is not that obvious from the initial plots so 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()` r `print()` function to print.

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

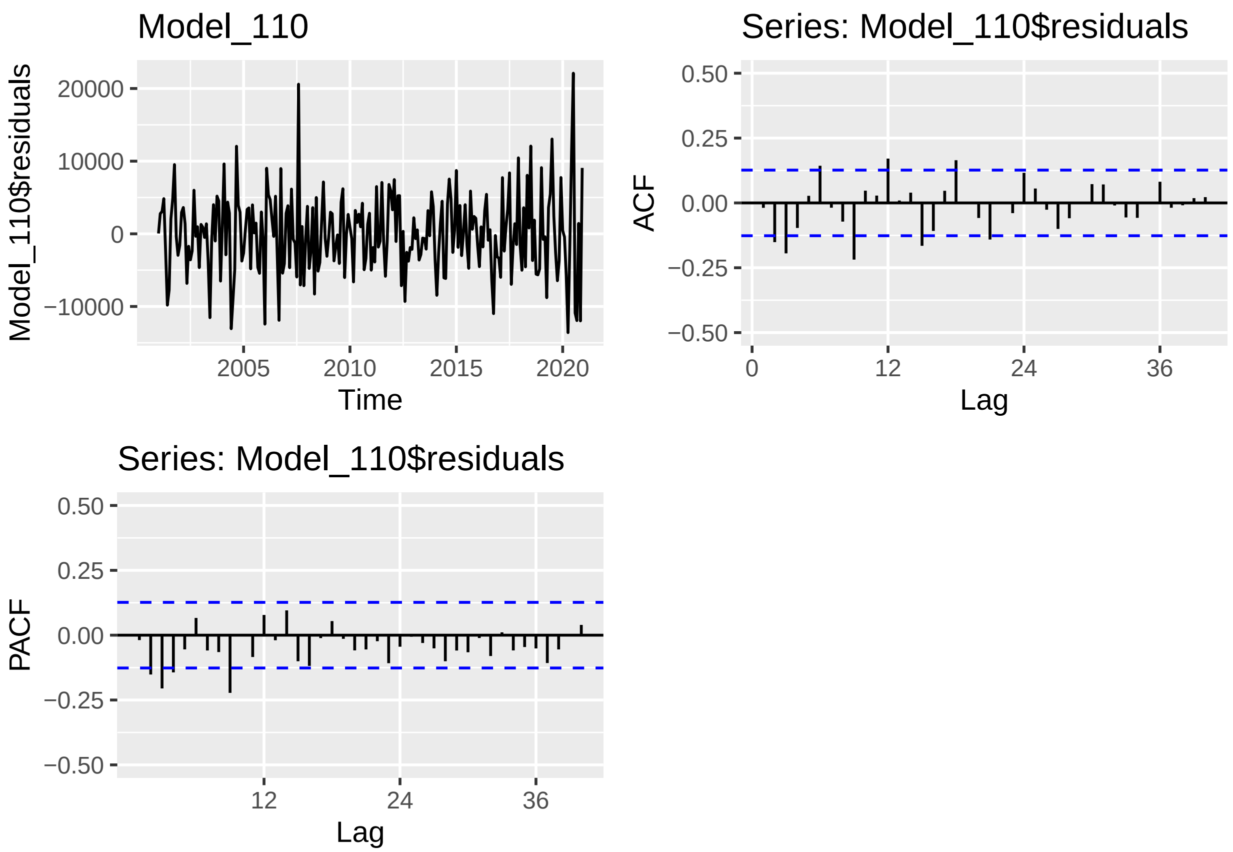
```
## Series: deseasonal_natgas
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1     drift
##       -0.1479  348.3927
## s.e.   0.0644  308.8385
##
## sigma^2 = 30254066:  log likelihood = -2396.54
## AIC=4799.07   AICc=4799.18   BIC=4809.5
```

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?

```
plot_grid(
  autoplot(Model_110$residuals, main = "Model_110"),
  autoplot(Acf(Model_110$residuals,lag.max=40, plot = FALSE),ylim=c(-0.5,0.5)),
  autoplot(Pacf(Model_110$residuals,lag.max=40, plot = FALSE),ylim=c(-0.5,0.5))
)
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘ylim‘
## Ignoring unknown parameters: ‘ylim‘
```



Answer: The residuals plot looks like a white noise series, with the mean centered around 0 and no particular trend. But, there are some significant coefficients in the ACF and PACF plots, so probably the order of the AR component could be increased.

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.

> Answer: By using nsdiff and ndiff, I identified that D=1 because seasonal differencing is needed and d=0 because the seasonal differencing is removing the linear trend as well. By using the ACF and PACF plots of the seeasonal differenced series, I identified that P = 0 and Q = 1 because there are is a negative spike at lag 12 in the ACF plot and negative spikes at lag 12 and 24 in the PACF plot. The residuals plot looks like a white noise series, with the mean centered around 0 and no particular trend. The ACF and PACF values are also non-significant, barring a couple in the initial lags.

```r
# Finding out how many times differencing is needed
ns_diff <- nsdiffs(ts_natgas)
cat("Number of seasonal differencing needed: ",ns_diff)
```

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

```r
#Differencing the series
natgas_seas_diff <- diff(ts_natgas,lag=12, differences=1)
```

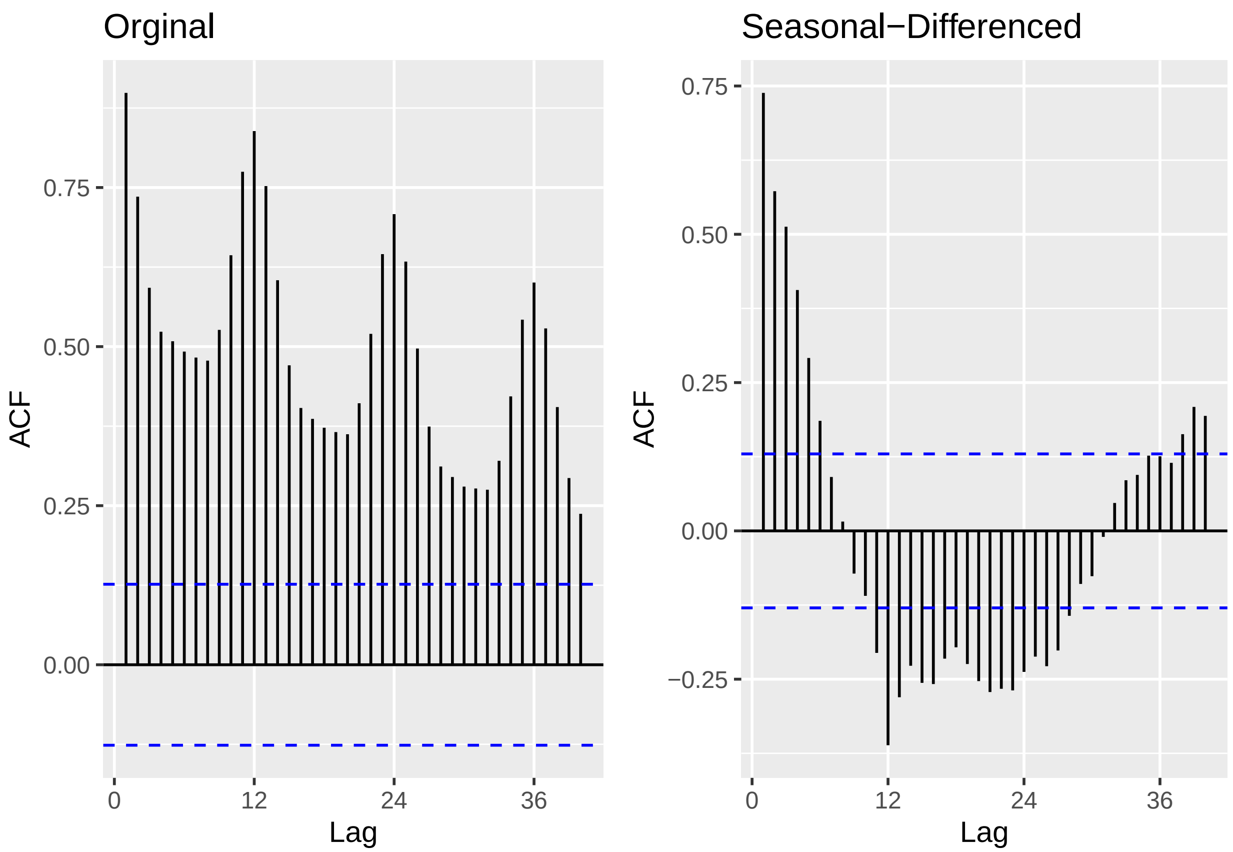
```r
#Checking if trend differencing is needed

n_diff <- ndiffs(natgas_seas_diff)
cat("Number of trend differencing needed: ",n_diff)
```

```
## Number of trend differencing needed:  0
```

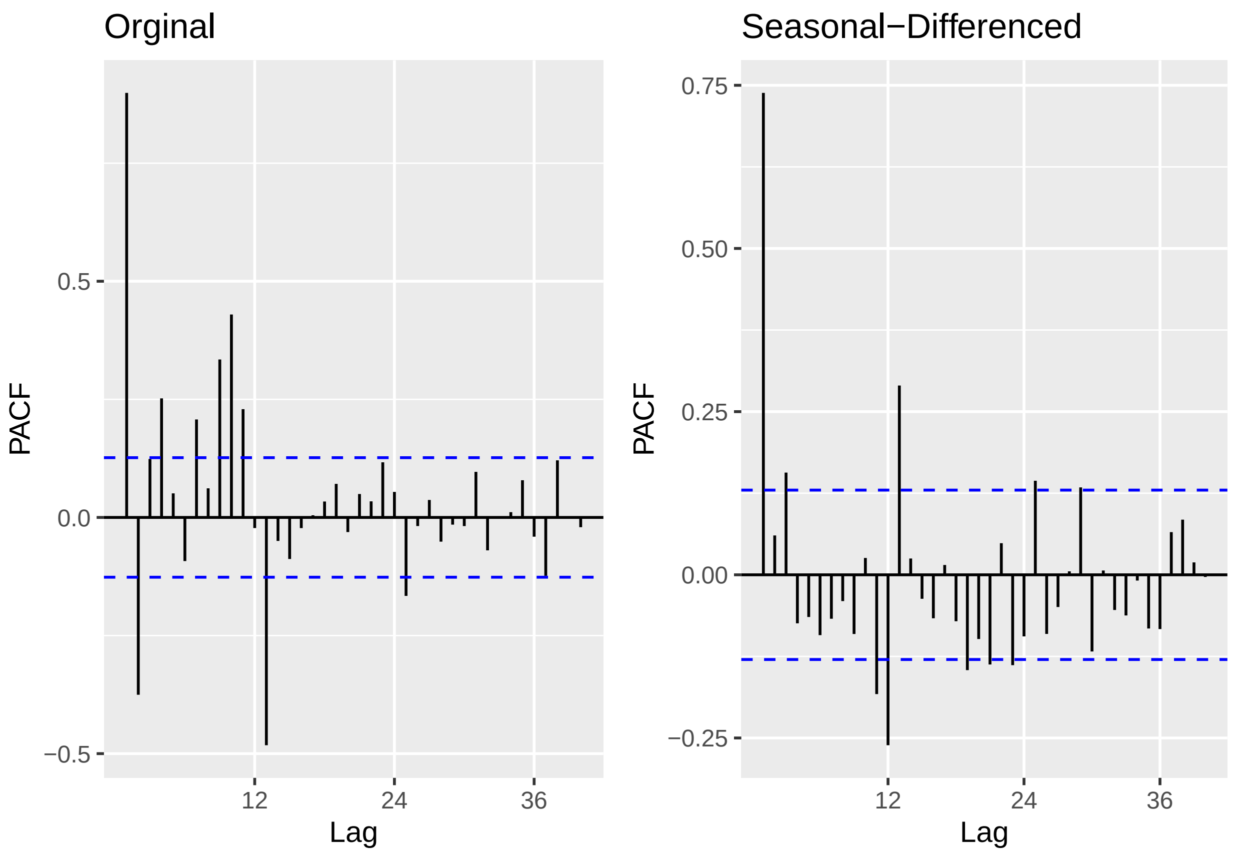
```r
#Comparing ACFs
plot_grid(
  autoplot(Acf(ts_natgas, lag = 40, plot=FALSE),
              main = "Orginal"),
  autoplot(Acf(natgas_seas_diff, lag = 40, plot=FALSE),
                main = "Seasonal-Differenced")
)
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
```



```
#Comparing PACFs
plot_grid(
  autoplot(Pacf(ts_natgas, lag = 40, plot=FALSE),
                main = "Orginal"),
  autoplot(Pacf(natgas_seas_diff, lag = 40, plot=FALSE),
                  main = "Seasonal-Differenced")
)
```

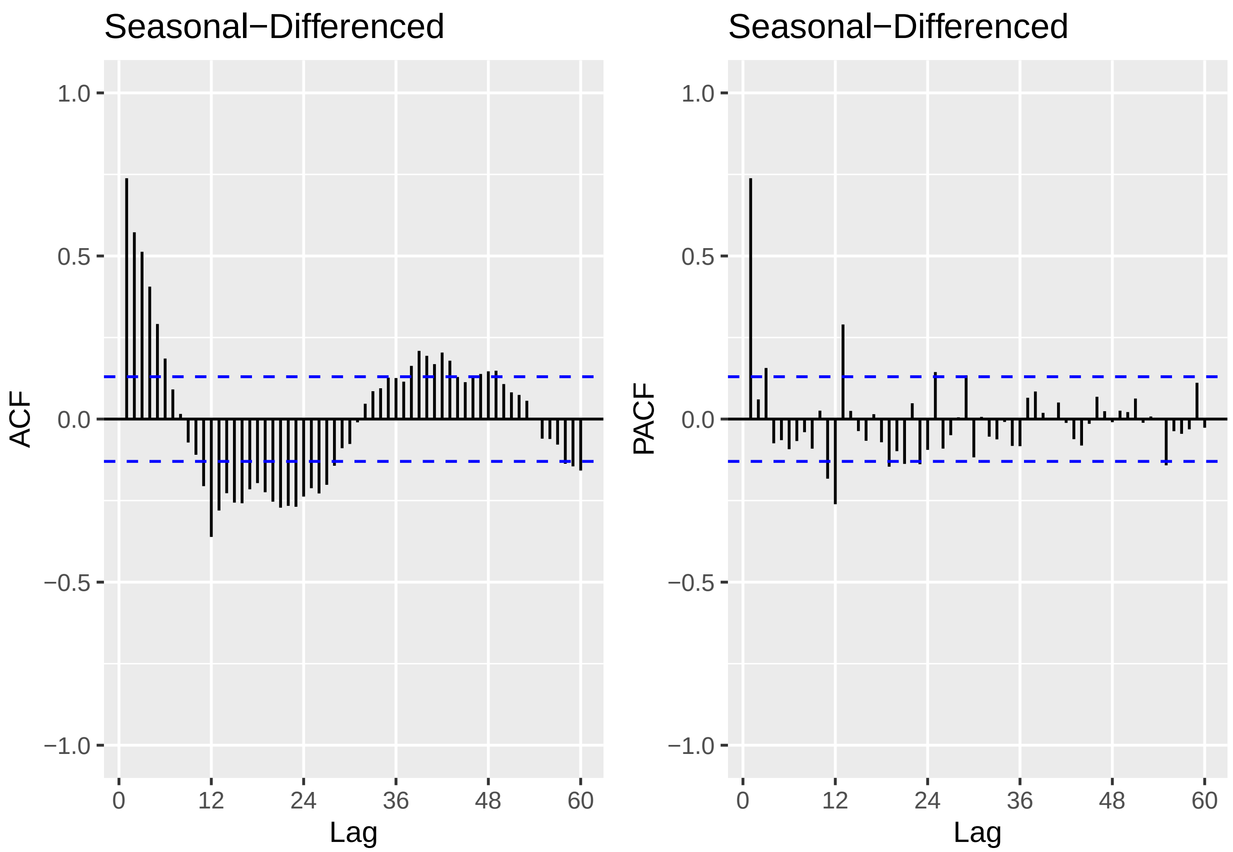
```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
```



```
#Seasonal differenced only

plot_grid(
  autoplot(Acf(natgas_seas_diff,lag.max=60,plot=FALSE),main="Seasonal-Differenced",ylim=c(-1,1)),
  autoplot(Pacf(natgas_seas_diff,lag.max=60,plot=FALSE),main="Seasonal-Differenced",ylim=c(-1,1)),
  nrow=1
)
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘ and ‘yli
## Ignoring unknown parameters: ‘main‘ and ‘ylim‘
```

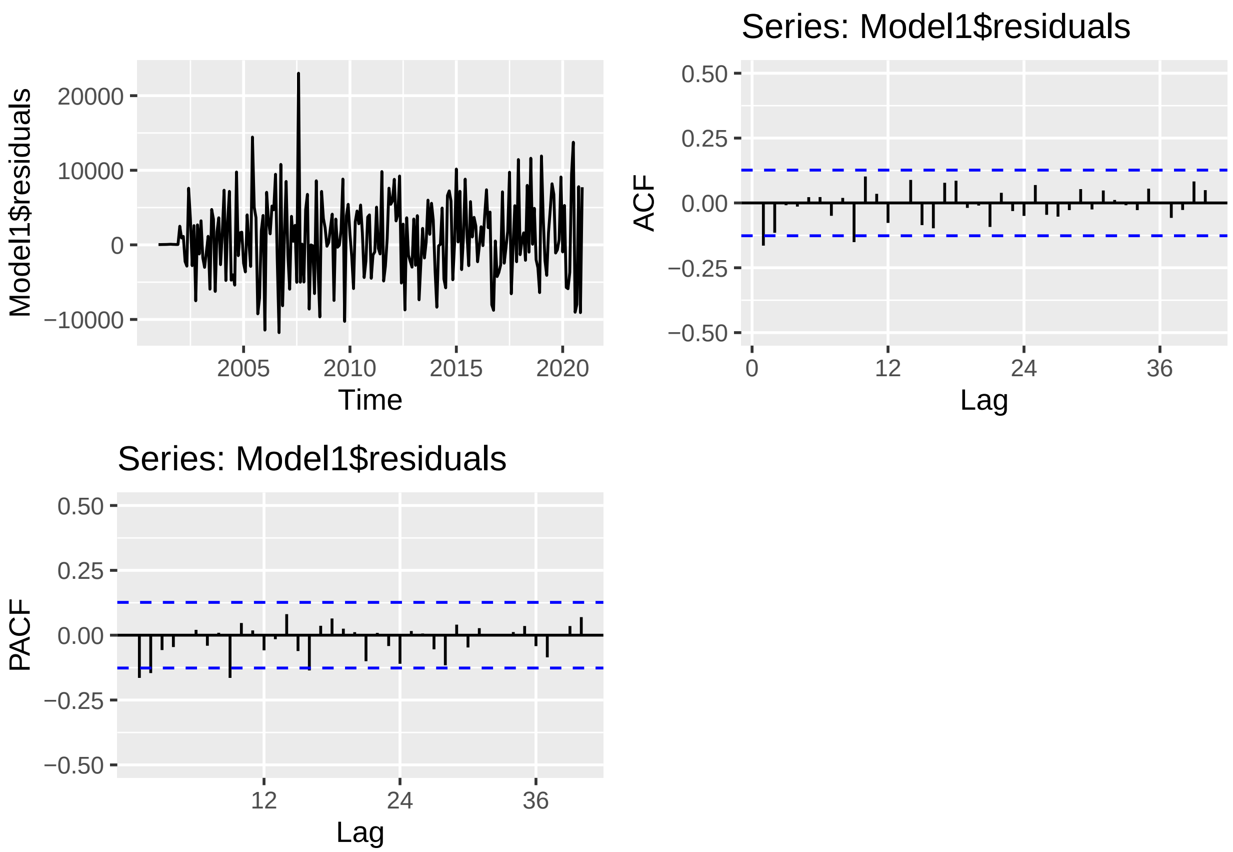


```
#Creating SARIMA model
Model1 <- Arima(ts_natgas,order=c(1,0,0),seasonal=c(0,1,1),include.drift=FALSE)
print(Model1)
```

```
## Series: ts_natgas
## 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
```

```
#Plotting the residuals
plot_grid(
  autoplot(Model1$residuals),
  autoplot(Acf(Model1$residuals,lag.max=40, plot = FALSE),ylim=c(-0.5,0.5)),
  autoplot(Pacf(Model1$residuals,lag.max=40, plot = FALSE),ylim=c(-0.5,0.5))
)
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘ylim‘
## Ignoring unknown parameters: ‘ylim‘
```



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.

> Answer: I think the SARIMA model might be slightly better at representing the natural gas series, but it is not a fair comparison. There are lesser number of ACF and PACF terms that are not significant in the SARIMA model. But, because the ARIMA model does not include the seasonal component, it is not fair to compare the two models because they have different type of data in terms of the trends in the 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 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?

```
Model2 <- auto.arima(deseasonal_natgas, max.D=0,max.P=0,max.Q=0)
print(Model2)
```

```
## Series: deseasonal_natgas
## 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
```

> Answer: The order of the best ARIMA model is 1,1,1. It does not match what I had for Q4.

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?

```r
Model3 <- auto.arima(ts_natgas)
print(Model3)
```

```
## Series: ts_natgas
## 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
```

> Answer: The order of the best SARIMA model is (1,0,0)(0,1,1). It matched what I specified in Q7.