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

Assignment 7 - Due date 03/25/22

Sarah Mansfield

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

```
#Load/install required package here
library(forecast)
library(tseries)
library(readr)
library(tidyverse)
library(ggfortify)
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**.

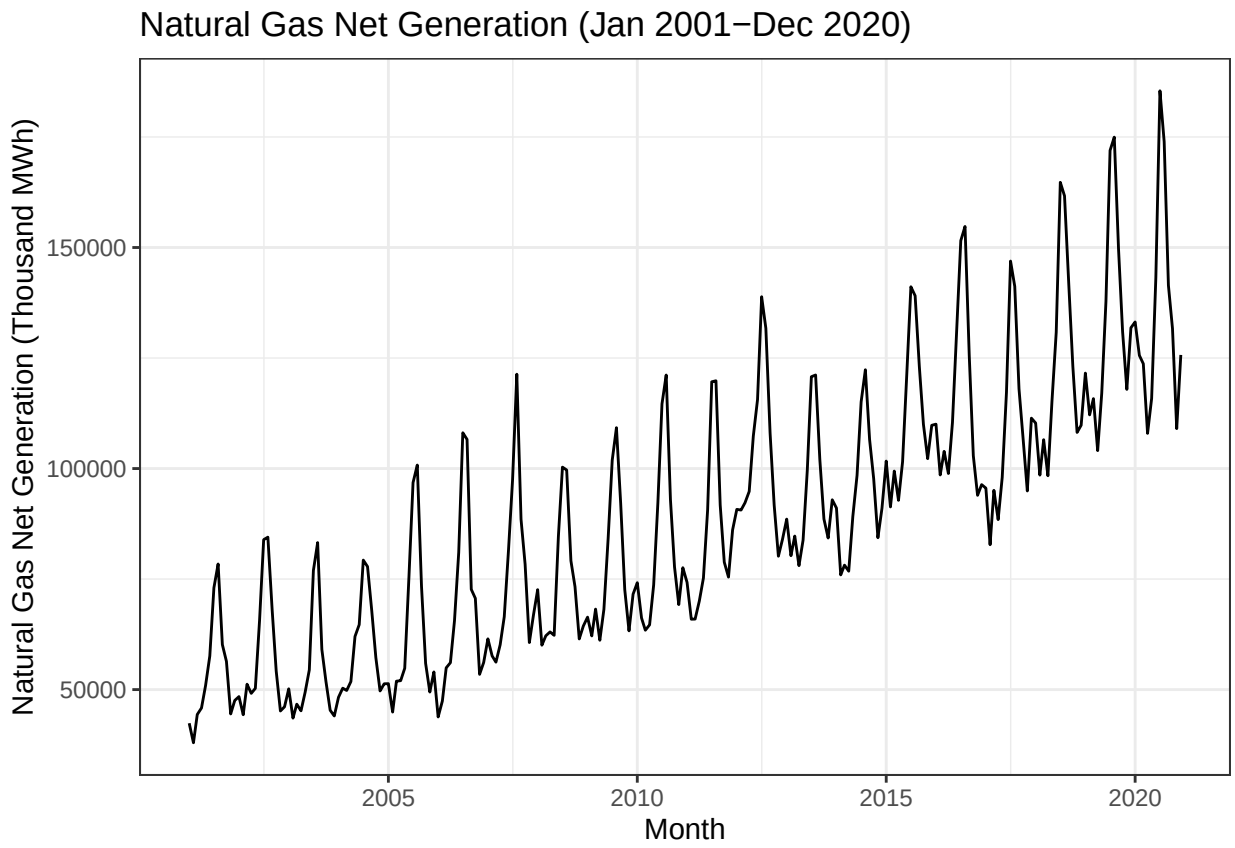
Packages needed for this assignment: "forecast","tseries". Do not forget to load them before running your script, since they are NOT default packages.\

Q1

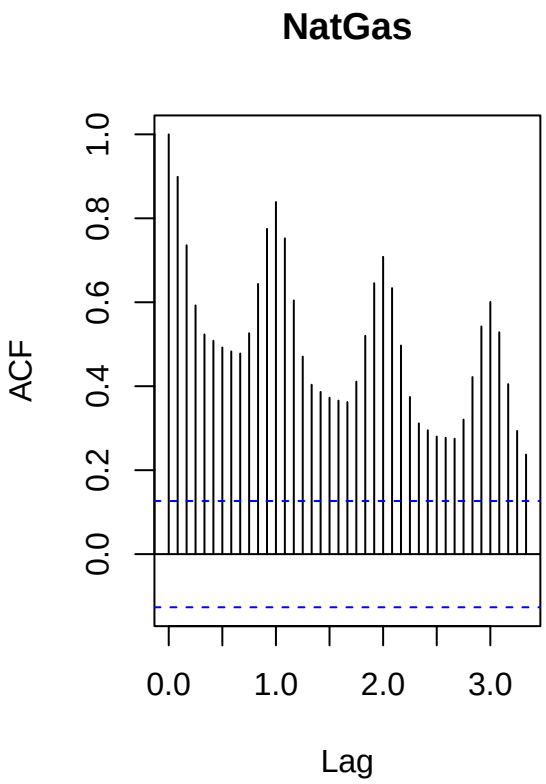
Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

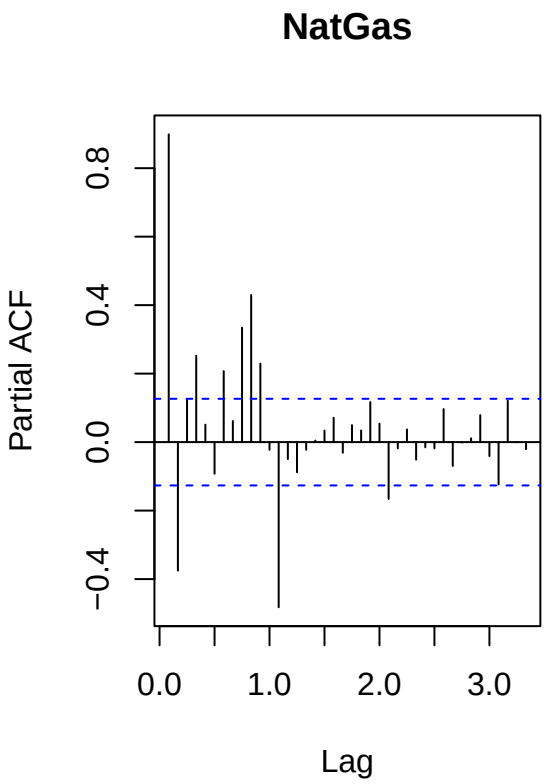
```
netgen <- read_csv("../Data/Net_generation_United_States_all_sectors_monthly.csv",
                   skip = 4, show_col_types = FALSE)
natgas <- netgen %>%
  select(Month, `natural gas thousand megawatthours`) %>%
  rename(`Natural Gas (Thousand MWh)` = `natural gas thousand megawatthours`)

natgas_ts <- ts(rev(natgas$`Natural Gas (Thousand MWh)`),
                start = c(2001, 1), frequency = 12)
```

```
autoplot(natgas_ts) +
  labs(title = "Natural Gas Net Generation (Jan 2001-Dec 2020)",
       x = "Month",
       y = "Natural Gas Net Generation (Thousand MWh)") +
  theme_bw()
```




```
par(mfrow=c(1,2))
acf(natgas_ts, lag.max = 40, main = "NatGas")
pacf(natgas_ts,  lag.max = 40, main = "NatGas")
```

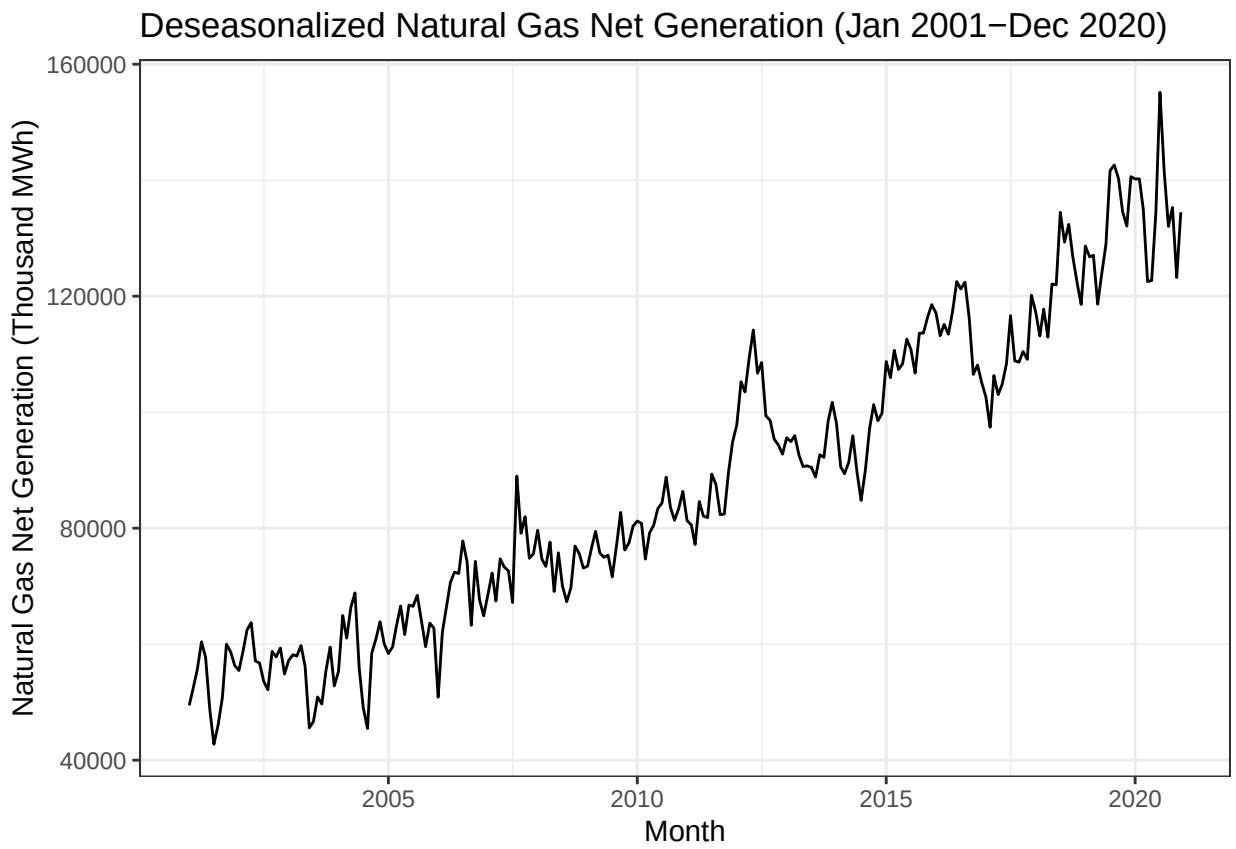




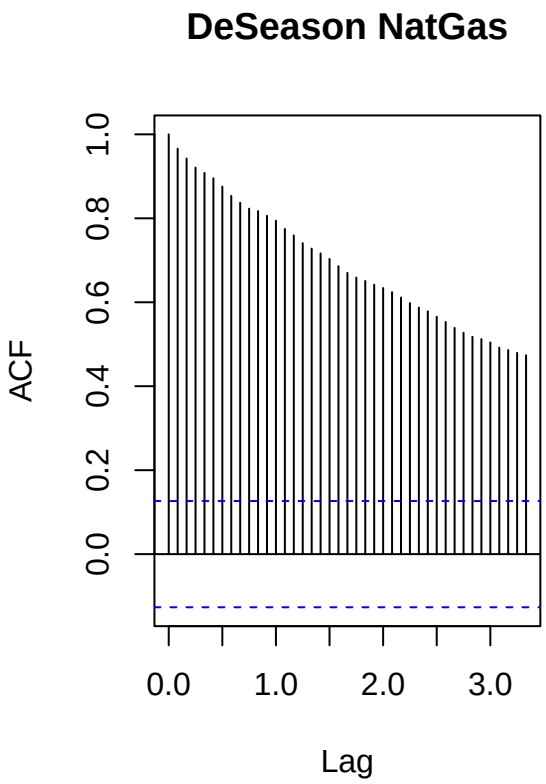
Q2

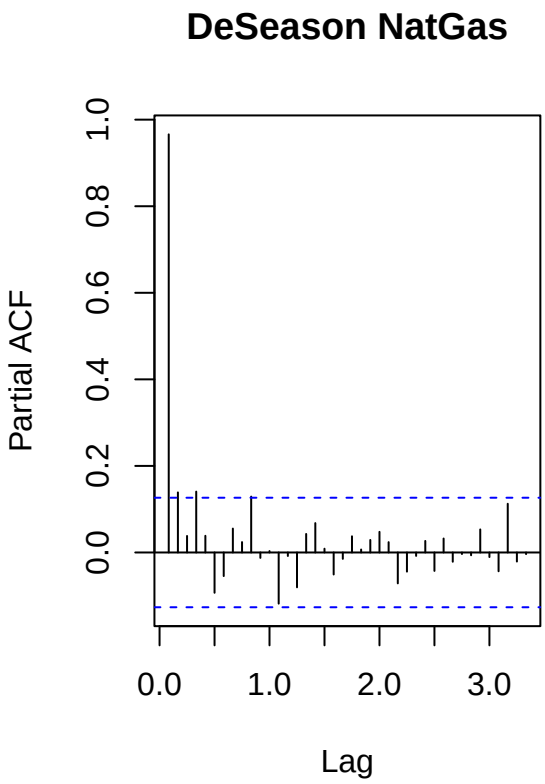
Using the *decompose*() or *stl*() and the *seasadj*() functions create a series without the seasonal component, i.e., a deseasonalized natural gas series. Plot the deseasonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

```
decompose_natgas <- decompose(natgas_ts, "additive")
deseasonal_natgas <- seasadj(decompose_natgas)

autoplot(deseasonal_natgas) +
  labs(title = "Deseasonalized Natural Gas Net Generation (Jan 2001-Dec 2020)",
       x = "Month",
       y = "Natural Gas Net Generation (Thousand MWh)") +
  theme_bw()
```




```
par(mfrow=c(1,2))
acf(deseasonal_natgas, lag.max = 40, main = "DeSeason NatGas")
pacf(deseasonal_natgas, lag.max = 40, main = "DeSeason NatGas")
```





Looking at the plot of the deseasonalized series over time, we can see that compared to the original series, there is no longer a seasonal trend reflected in the plot. This is also seen in the ACF and PACF plots, as the ACF plot shows smooth decay (compared to the dips and peaks the previous ACF plot has) as lag increases, and the PACF plot cuts off immediately after the first lag, whereas with the previous PACF plot we saw additional spikes at various lags.

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.

```
# null hypothesis is that data has a unit root
print("Results for ADF test")
```

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

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

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

```
##
```

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

```
print("Results for Mann Kendall test")
```

```
## [1] "Results for Mann Kendall test"
```

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

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

For the ADF test, we calculated a p-value of $0.01 < 0.05$, therefore we reject the null hypothesis and conclude that the data does not have a stochastic trend. For the Mann Kendall test, we calculated a p-value of $2.22e-16 < 0.05$, therefore we reject the null hypothesis and conclude that the data follows a deterministic 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 can read the plots and interpret the test results.

```
# find out how many time we need to difference
n_diff <- ndiffs(deseasonal_natgas)
cat("Number of differencing needed: ", n_diff)
```

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

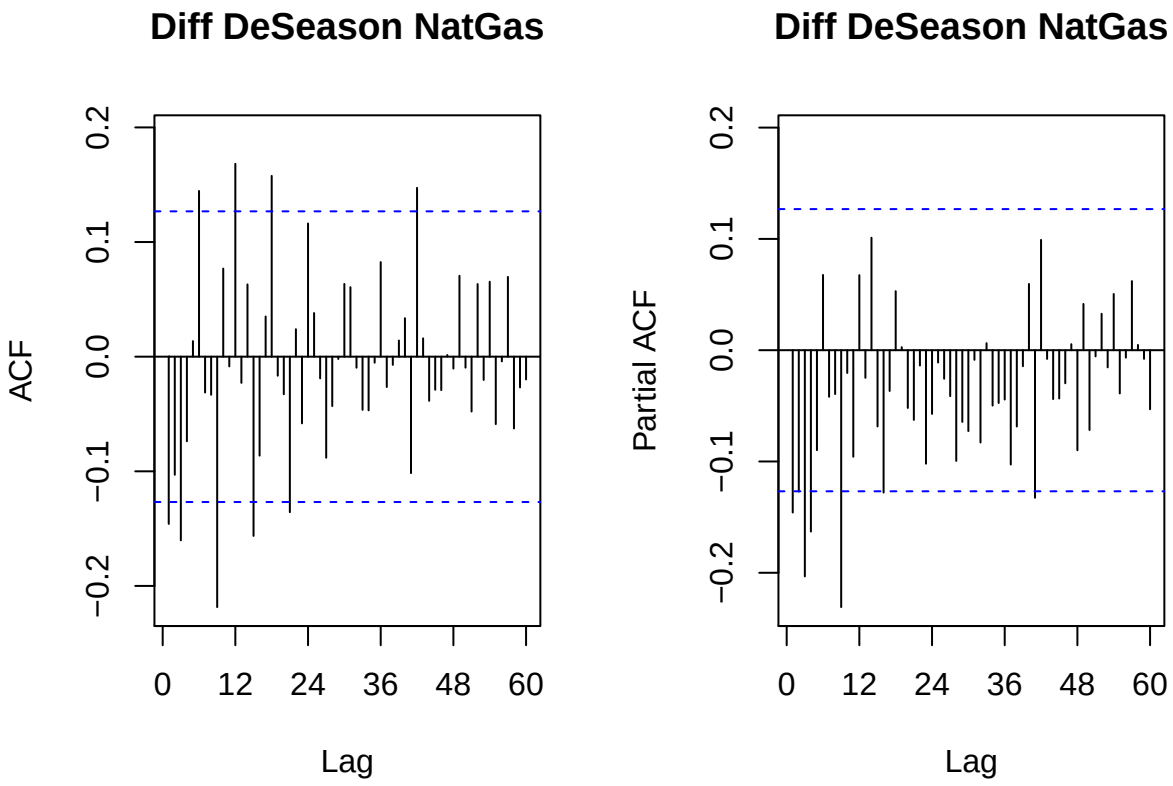
```
natgas_diff <- diff(deseasonal_natgas, differences = 1)

# check Mann Kendall test after differencing
print("Results for Mann Kendall test after differencing")
```

```
## [1] "Results for Mann Kendall test after differencing"
```

```
print(summary(MannKendall(natgas_diff)))
```

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

```
# we look at the differenced series to determine model order
par(mfrow=c(1,2))
Acf(natgas_diff, lag.max = 60, main = "Diff DeSeason NatGas")
Pacf(natgas_diff, lag.max = 60, main = "Diff DeSeason NatGas")
```



Since our test results from Q3 identified the data as having a deterministic trend, it's necessary to difference teh series at least once. Using the `ndiffs` function, we've calculated that $d = 1$. After differencing the deseasonalized series and plotting the ACF and PACF, it looks like both plots cut off after lag 1, so we can assume that $p = 1$ and $q = 1$.

```
mod_111 <- Arima(deseasonal_natgas, order=c(1, 1, 1), include.drift=TRUE)
mod_110 <- Arima(deseasonal_natgas, order=c(1, 1, 0), include.drift=TRUE)
mod_011 <- Arima(deseasonal_natgas, order=c(0, 1, 1), include.drift=TRUE)

compare_aic <- data.frame(mod_111$aic, mod_110$aic, mod_011$aic)
print(compare_aic)
```

```
##   mod_111.aic mod_110.aic mod_011.aic
## 1    4774.213    4799.075    4796.664
```

We can also fit a few different models and compare their AIC values - again, we see that the model with the lowest AIC is the one with $p = 1$ and $q = 1$, so we'll use this model.

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.

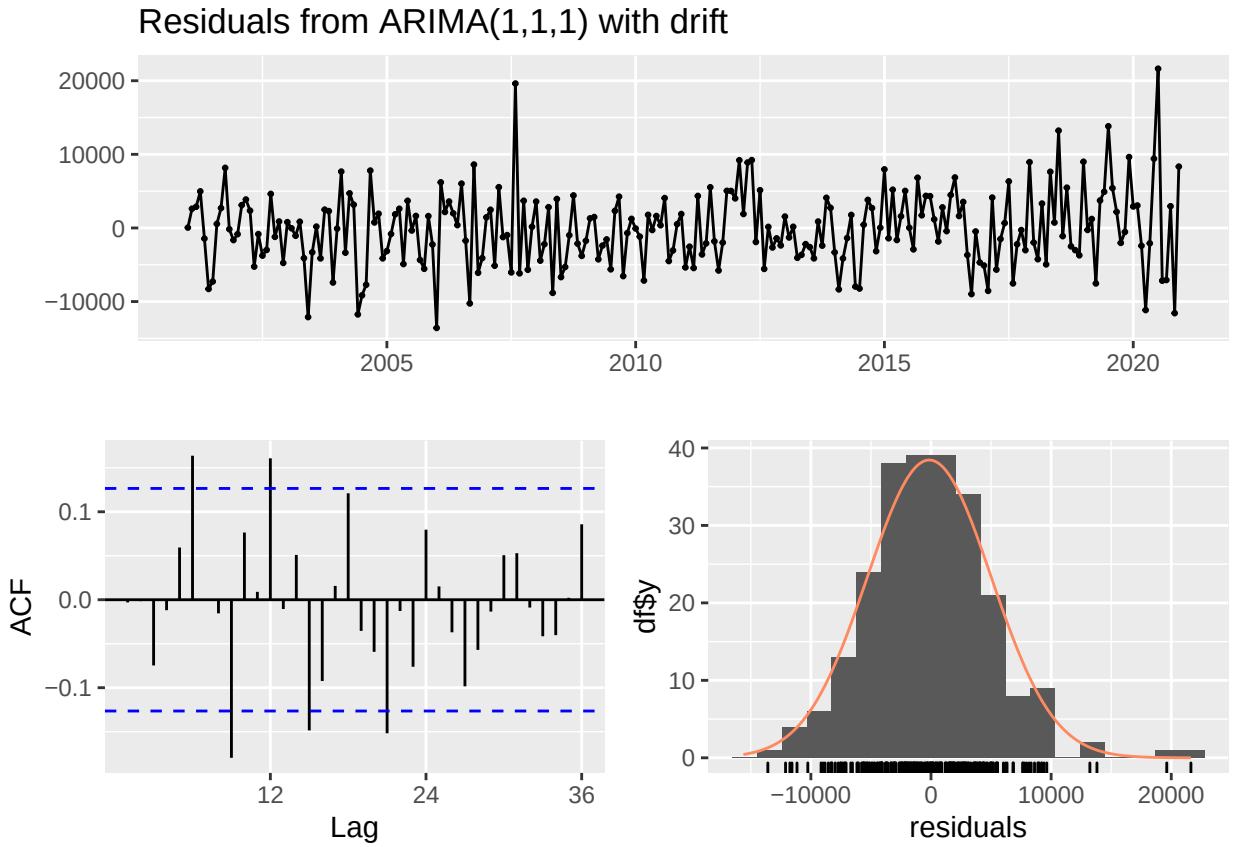
```
ARIMA_deseason <- Arima(deseasonal_natgas, order = c(1, 1, 1), include.drift = TRUE)
print(ARIMA_deseason)
```

```
## 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
```

Q6

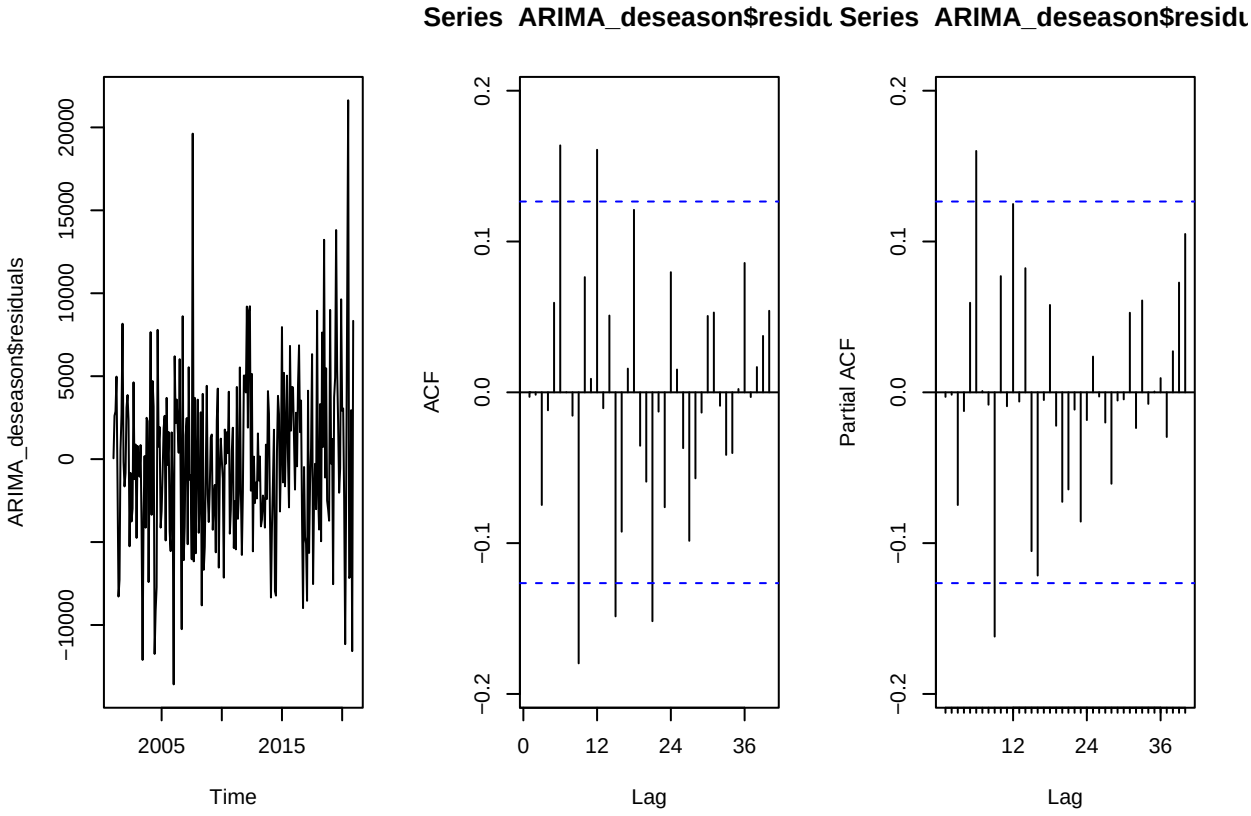
Now plot the residuals of the ARIMA fit from Q5 along with residuals ACF and PACF on the same window. You may use the *checkresiduals*() function to automatically generate the three plots. Do the residual series look like a white noise series? Why?

```
checkresiduals(ARIMA_deseason)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1) with drift
## Q* = 48.356, df = 21, p-value = 0.000615
##
## Model df: 3.   Total lags used: 24
```

```
par(mfrow=c(1,3))
ts.plot(ARIMA_deseason$residuals)
Acf(ARIMA_deseason$residuals, lag.max = 40)
Pacf(ARIMA_deseason$residuals, lag.max = 40)
```



The residual series does not quite look like a white noise series, as we can see from the ACF plot that there are several spikes that fall outside the bounds. Additionally, we note that the Ljung-Box test resulted in a significant p-value, and therefore we reject the null hypothesis that the model is a good fit/the series isn't autocorrelated and conclude that the model shows lack of fit.

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.

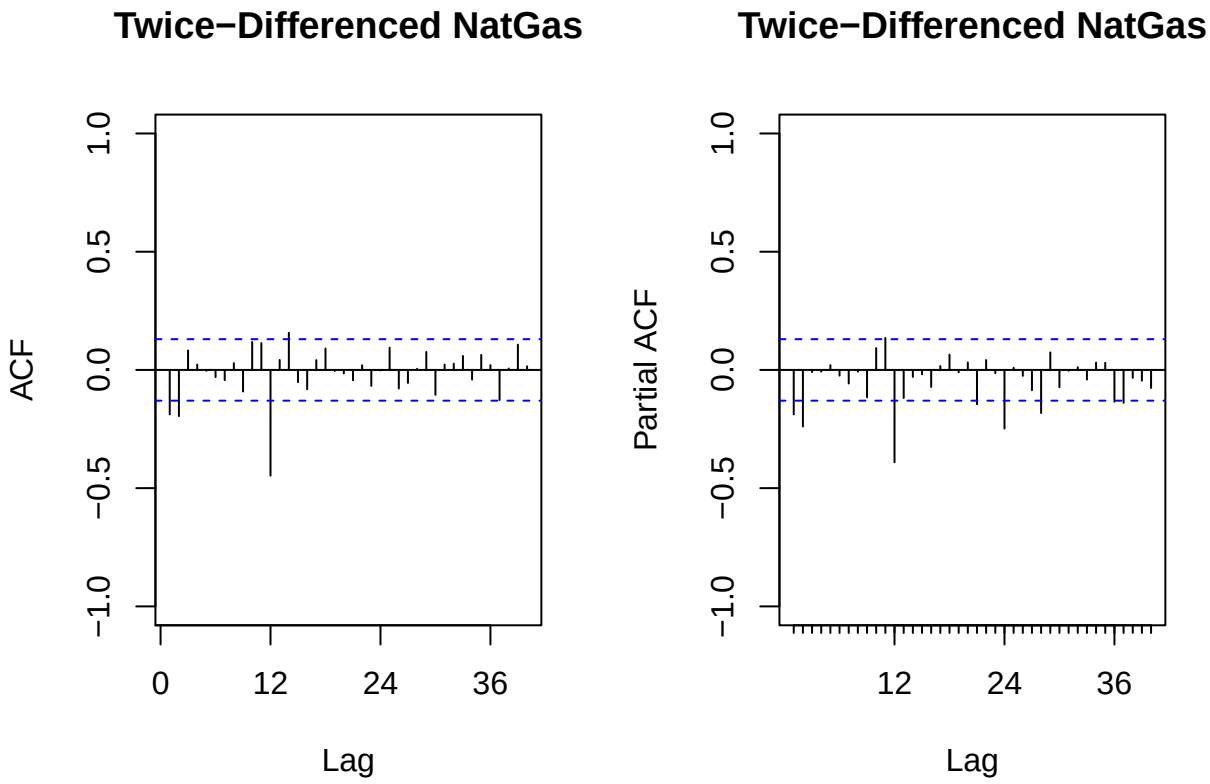
```
# Find out how many time we need to difference
n_diff <- ndiffs(natgas_ts)
cat("Number of differencing needed: ", n_diff)
```

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

```
ns_diff <- nsdiffs(natgas_ts)
cat("Number of seasonal differencing needed: ", ns_diff)
```

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

```r
natgas_trend_diff <- diff(natgas_ts, lag = 1, differences = 1) #diff done on orig series
natgas_both_diff <- diff(natgas_trend_diff, lag = 12, differences = 1)

par(mfrow=c(1,2))
Acf(natgas_both_diff, lag.max = 40, main = "Twice-Differenced NatGas", ylim=c(-1,1))
Pacf(natgas_both_diff, lag.max = 40, main = "Twice-Differenced NatGas", ylim=c(-1,1))
```



When we look at the first 12 lags for ACF & PACF we don't see slow decays but it looks like we have cut offs at lag 2 on both plots, indicating an ARMA(p=2, q=2), and we know from ndiffs that $d = 1$.

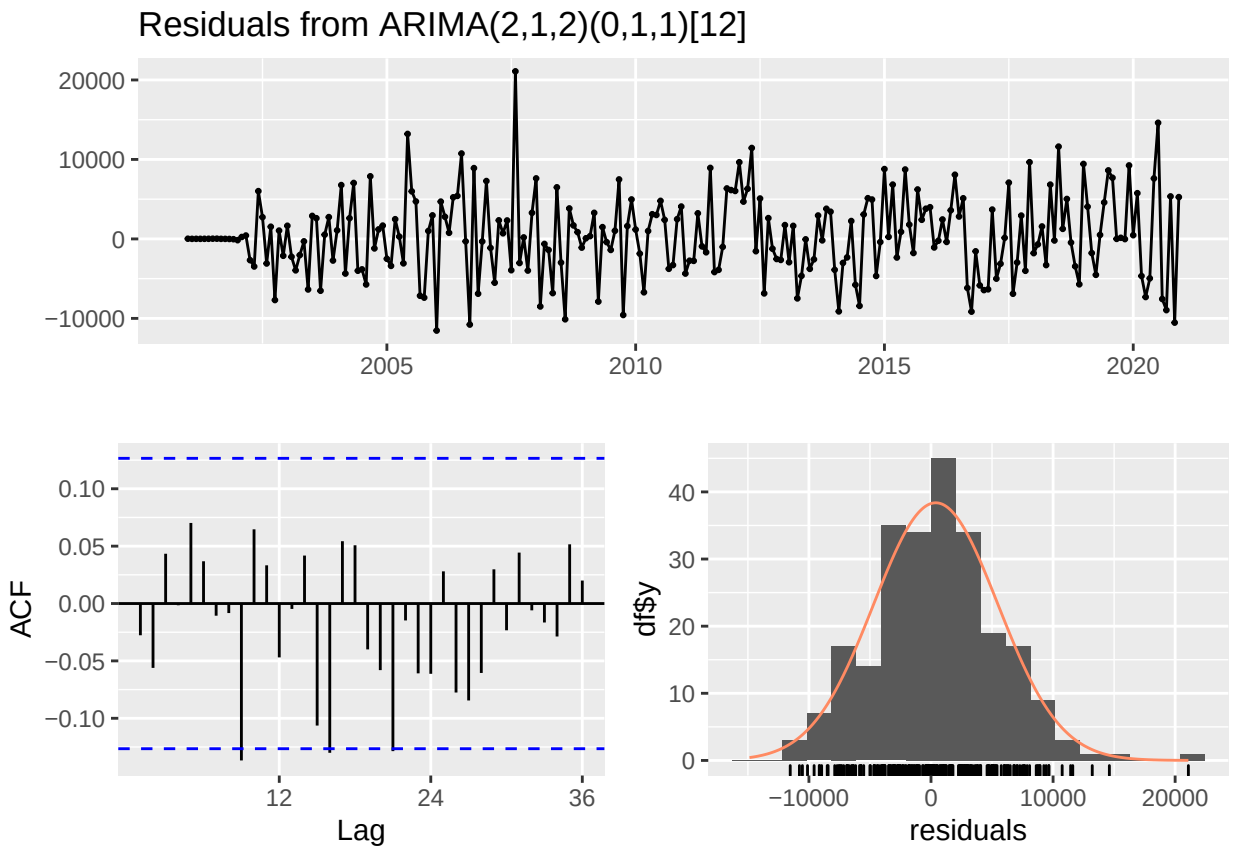
Looking at seasonal lags only (12, 24, 36), we can see that ACF has one spike at 12 and PACF has 2 spikes (one at 12 and one at 24). This is an indication of a seasonal moving average (SMA), and therefore the order of the seasonal component is $P = 0$ and $Q = 1$. We also know from the `nsdiffs` function that $D = 1$.

```r
# manually fitting seasonal ARIMA to the original series
SARIMA_manual <- Arima(natgas_ts, order = c(2, 1, 2), seasonal = c(0, 1, 1), include.drift = FALSE)
print(SARIMA_manual)
```

```
## Series: natgas_ts
## ARIMA(2,1,2)(0,1,1)[12]
##
## Coefficients:
##           ar1     ar2      ma1      ma2     sma1
##       -0.2162  0.7057  -0.0489  -0.9156  -0.7165
## s.e.   0.0834  0.0648   0.0767   0.0737   0.0563
##
```

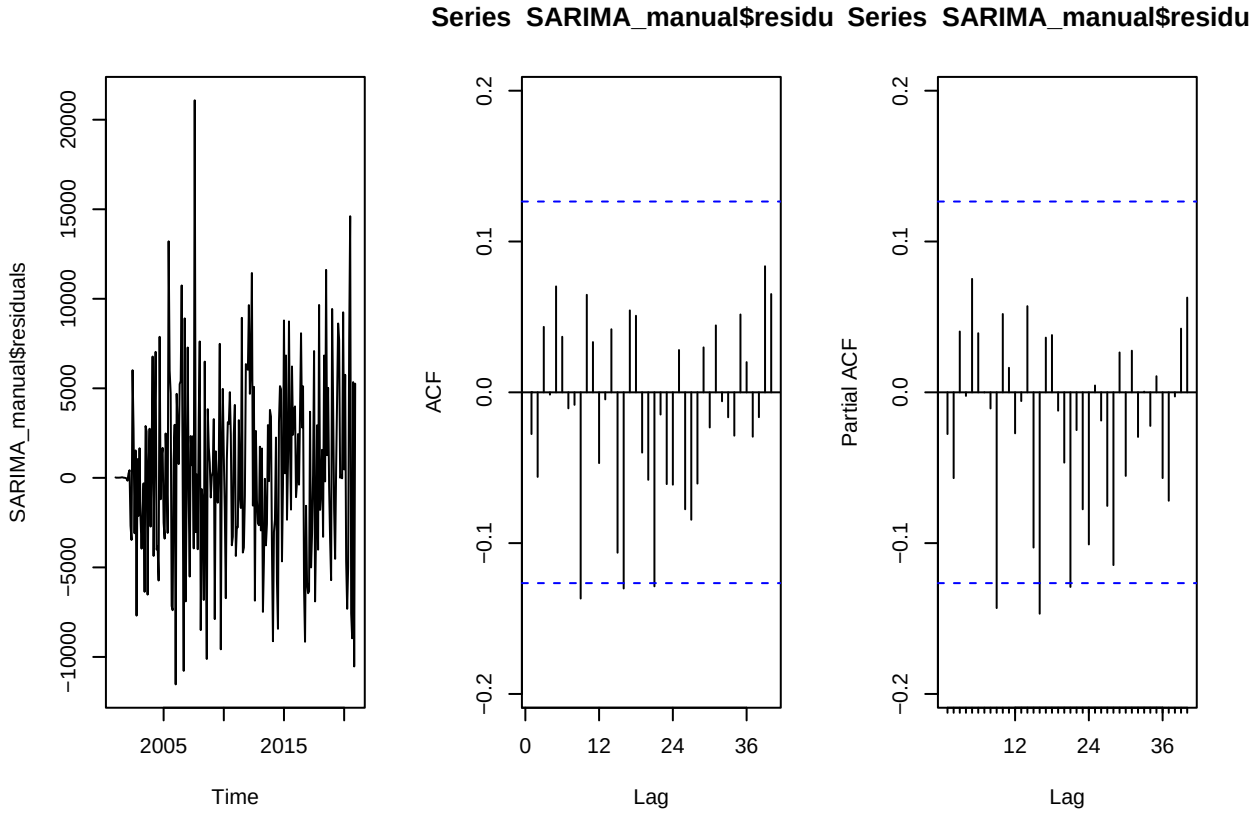
```
## sigma^2 = 28013060:  log likelihood = -2271.66
## AIC=4555.33   AICc=4555.71   BIC=4575.88
```

```
checkresiduals(SARIMA_manual)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2)(0,1,1)[12]
## Q* = 26.571, df = 19, p-value = 0.115
##
## Model df: 5.   Total lags used: 24
```

```
par(mfrow=c(1,3))
ts.plot(SARIMA_manual$residuals)
Acf(SARIMA_manual$residuals, lag.max = 40)
Pacf(SARIMA_manual$residuals, lag.max = 40)
```

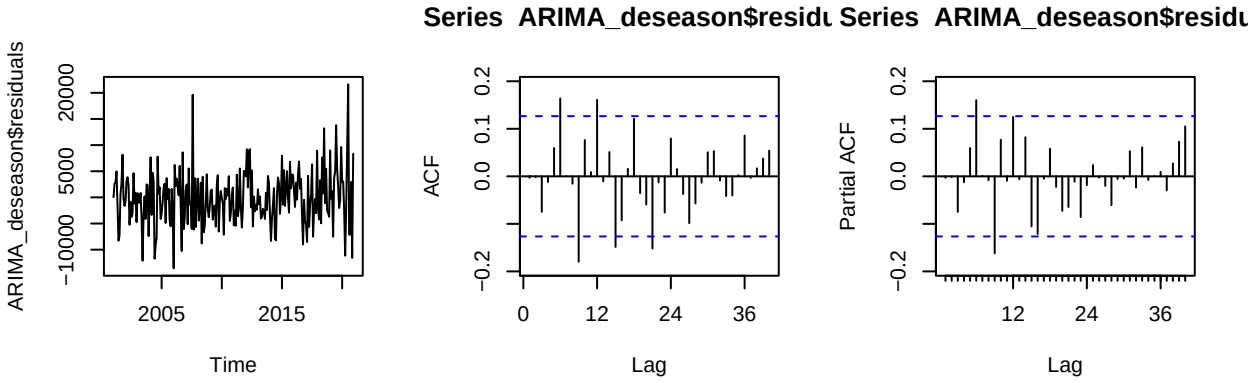


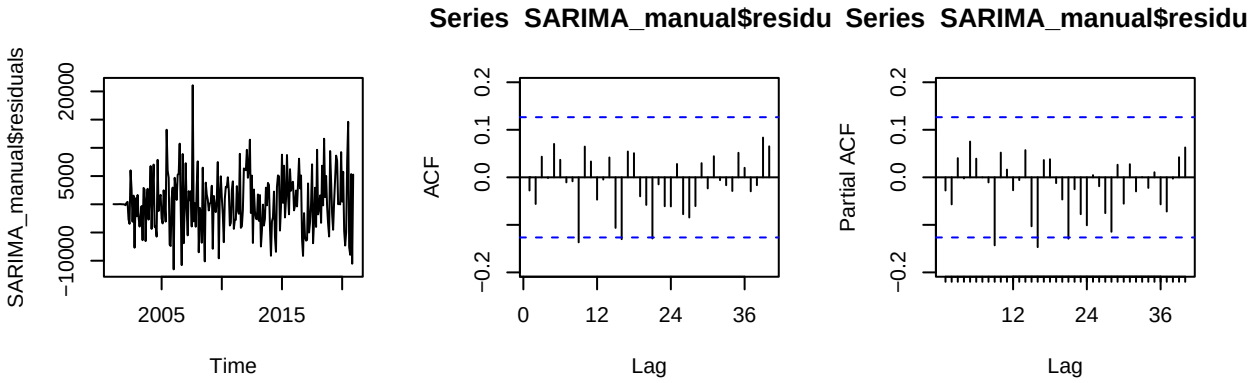
The residual series does seem to look like a white noise series, as we can see from the ACF plot that the majority of ACF values fall within the bounds. Additionally, the Ljung-Box test resulted in a non-significant p-value, which indicates that the model is a good fit for the data.

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.

```
par(mfrow=c(2,3))
ts.plot(ARIMA_deseason$residuals)
Acf(ARIMA_deseason$residuals, lag.max = 40)
Pacf(ARIMA_deseason$residuals, lag.max = 40)

#par(mfrow=c(1,3))
ts.plot(SARIMA_manual$residuals)
Acf(SARIMA_manual$residuals, lag.max = 40)
Pacf(SARIMA_manual$residuals, lag.max = 40)
```





The seasonal ARIMA model seems to be a better representation of the Natural Gas series according to the Ljung-Box test and the corresponding ACF and PACF plots. However, this might not be a fair comparison since the ARIMA model was fit on the deseasonalized series data, whereas the seasonal ARIMA model was fit on the original series data.

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?

```
auto.arima(deseasonal_natgas, max.D = 0, max.P = 0, max.Q = 0)
```

```
## 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
```

Yes, the model chosen with `auto.arima` matches what we specified in 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?

```
auto.arima(natgas_ts)
```

```
## Series: natgas_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
```

No, the model chosen with `auto.arima` does not match what we specified in Q7.