ENV 790.30 - Time Series Analysis for Energy Data | Spring 2021

Assignment 6 - Due date 03/26/21

Marie McNamara

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_A06_Sp21.Rmd"). Submit this pdf using Sakai.

Set up

```
knitr::opts_chunk$set(echo = TRUE,tidy.opts=list(width.cutoff=80), tidy=FALSE)

#Load/install required package here
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(forecast)
```

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

```
library(Kendall)
library(tseries)
library(outliers)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(sarima)
```

```
## Loading required package: FitAR
## Loading required package: lattice
## Loading required package: leaps
## Loading required package: ltsa
## Loading required package: bestglm
##
## Attaching package: 'FitAR'
## The following object is masked from 'package:forecast':
##
##     BoxCox
## Loading required package: stats4
```

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.

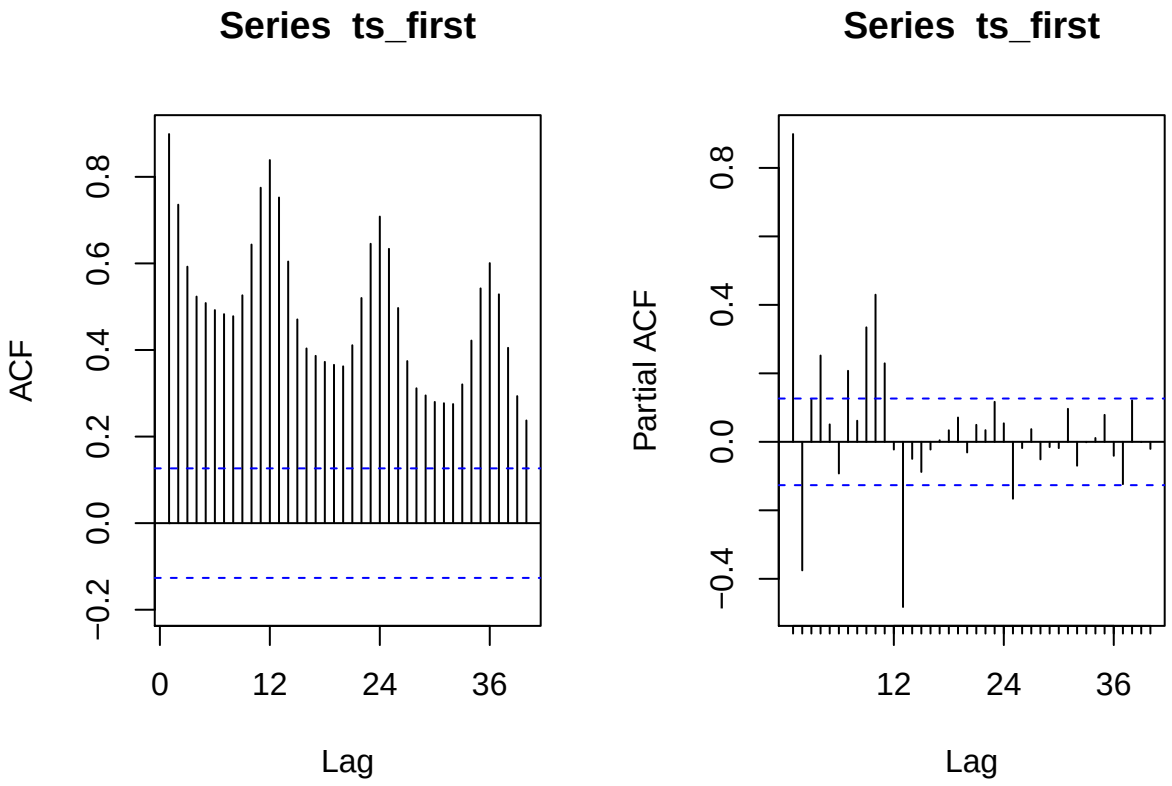
```
natural_gas_new <- read.csv(file="../Data/Net_generation_United_States_all_sectors_monthly.csv" ,header=

first <-
  natural_gas_new %>%
  select("Month","natural.gas.thousand.megawatthours") %>%
  mutate( Month = my(Month)) %>%
  rename( natural_gas = natural.gas.thousand.megawatthours)  %>%
  arrange (Month)

nvar= ncol(first) - 1
ts_first <- ts(first[,2:(nvar+1)],
                           start=c(year(first$Month[1]),month(first$Month[1])),
                           frequency=12)

par(mfrow=c(1,2))
```

```
acf1 = Acf(ts_first, lag.max = 40, plot = TRUE)
pacf1 = Pacf(ts_first, lag.max = 40, plot = TRUE)
```



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 depersonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

Based on the PACF and ACF plots there is a clear seasonal component. There was also an observed decreasing mean trend in the ACF plot (This can be seen by the decreasing mean over time, and the high correlation component between lags)

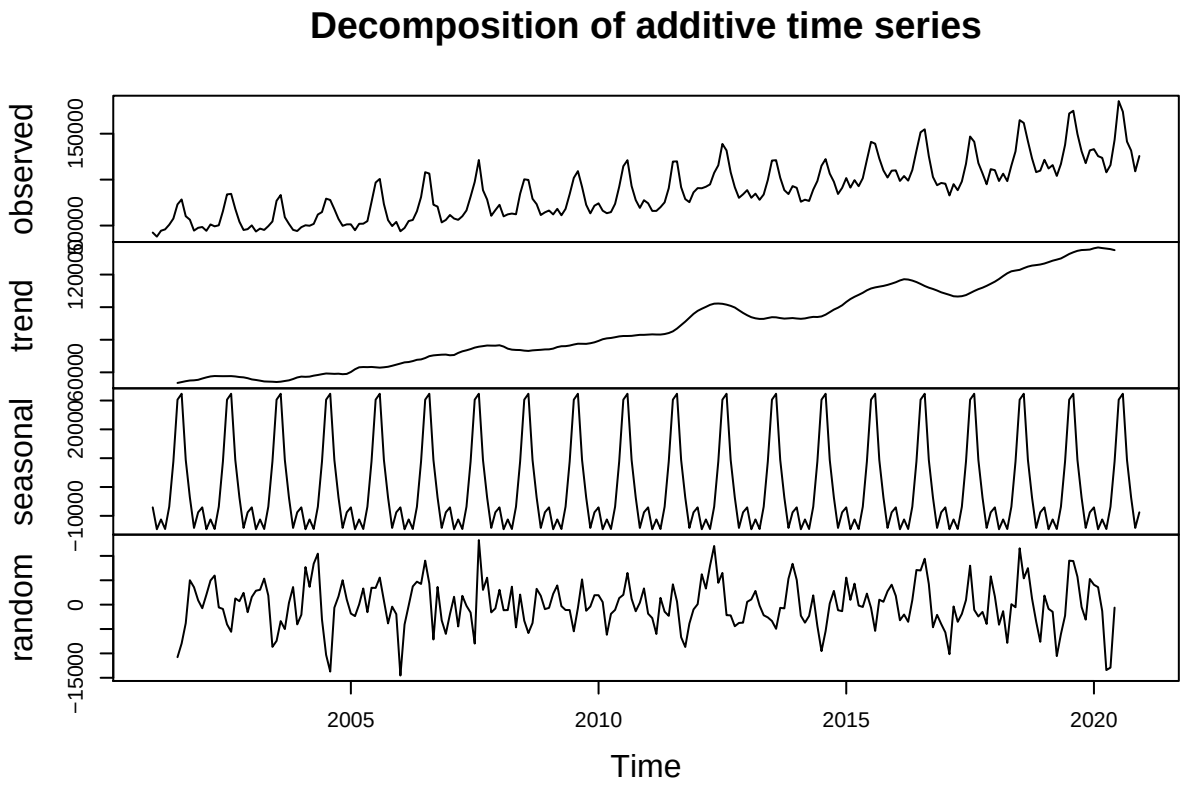
The seasoned plot confirms that there is a seasonal component, as well as a decreasing mean trend.

After plotting the desonalized series it can be seen that there is no longer observed seasonality in ACF but there is still some decrease mean trend.

```
n_diff2 <- ndiffs(ts_first)
cat("Number of differencing needed:", n_diff2)
```

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

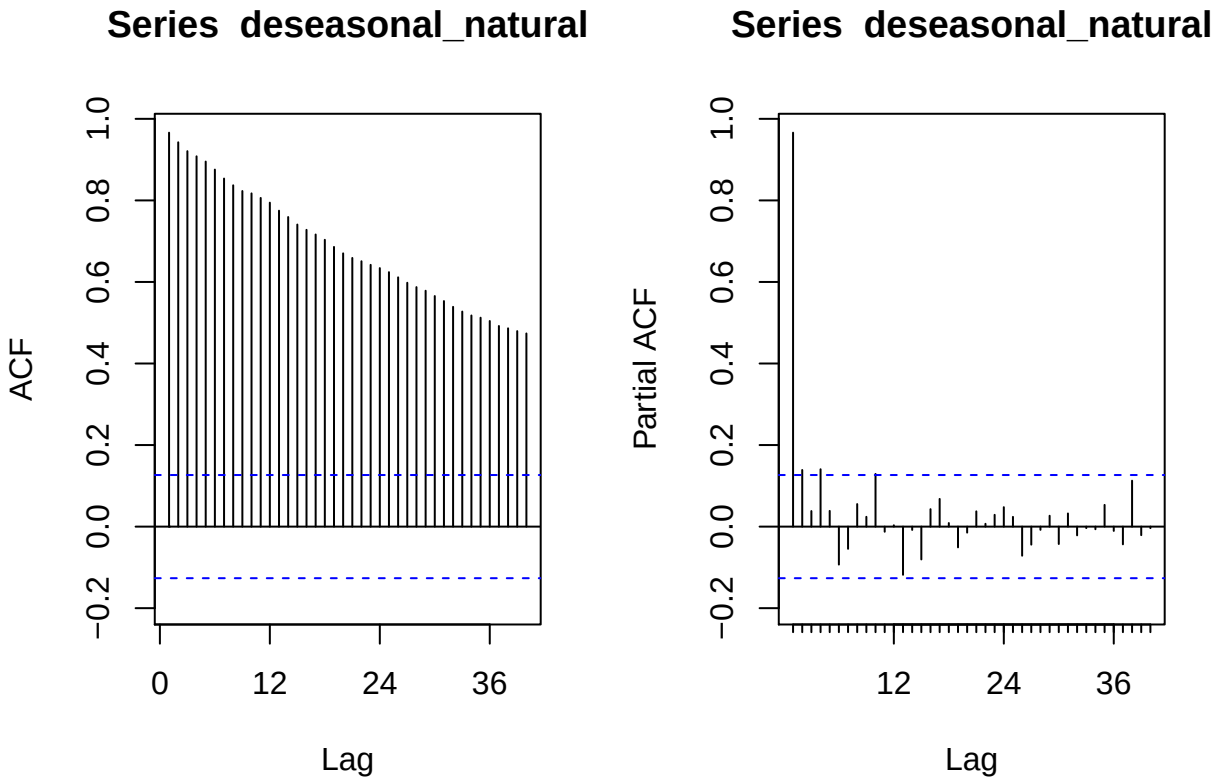
```
decompose_natural_gas <- decompose(ts_first,"additive")
plot(decompose_natural_gas)
```




```
deseasonal_natural <- seasadj(decompose_natural_gas)


#Lets difference the series once at lag 1 to remove the trend.
deseasonal_natural_diff <- diff(deseasonal_natural,differences=1,lag=2)

par(mfrow=c(1,2))
acf1 = Acf(deseasonal_natural, lag.max = 40, plot = TRUE)
pacf1 = Pacf(deseasonal_natural, lag.max = 40, plot = TRUE)
```



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.

The seasonal Mann kendall tests for P value was 2.22e-16, reject the null, that natural gas data is stationary, thus the data is not stationary.

For the ADF test p=0.01 which is less than 0.05 reject the null which is that the series contains a unit root.

Thus this data is not stationary and follows a trend but it does not have a unit root. The series should be differenced.

```
i=2
SMKtest <- SeasonalMannKendall(deseasonal_natural)
print(summary(SMKtest))
```

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

```
i=2
print(adf.test(deseasonal_natural,alternative = "stationary"))
```

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

```
##
##  Augmented Dickey-Fuller Test
##
```

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

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.

After analyzing the ACF and PACF plots I notice that there is slow decay in the ACF Plot. There is also a PACF cut off at lag two which leads me to conclude that this is a AR process order 2. (With this said i believe this is an ARMA function and there appears to be positive autocorrelations a high number of lags out thus differencing more than once could be needed)

The series needs differencing as the Mankendal test above showed a stochastic trend. After running the n_diff function there is only one differencing needed. The ACF plot determines q the number of moving average terms and since it was idenfited as AR determined order 0 for q. For ARMA there is no systematic process for identification.

p=2 d=1 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.

```
ARIMA_manual <- Arima(deseasonal_natural,order=c(2,1,0), include.mean = TRUE,include.drift=TRUE)
print(ARIMA_manual)
```

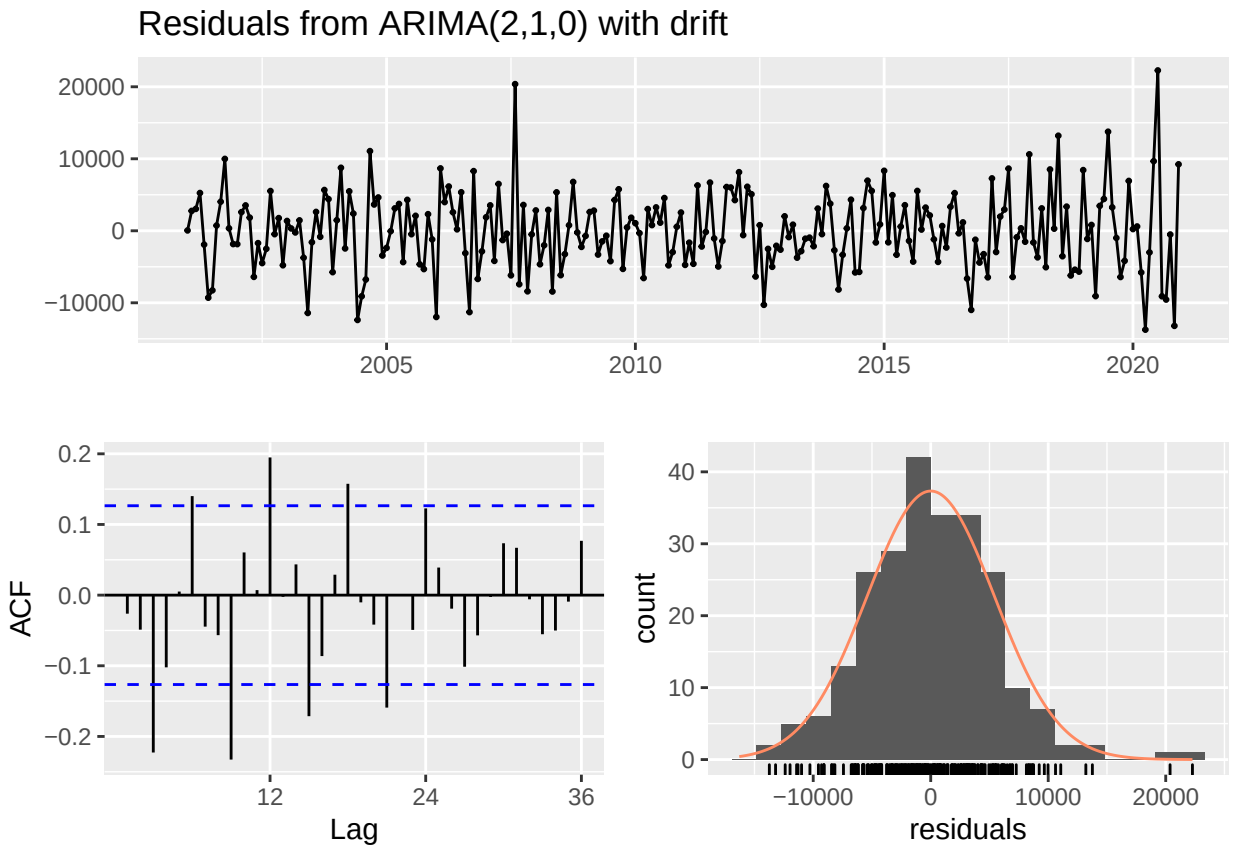
```
## Series: deseasonal_natural
## ARIMA(2,1,0) with drift
##
## Coefficients:
##           ar1      ar2     drift
##       -0.1647  -0.1283  346.8289
## s.e.   0.0645   0.0650  272.1672
##
## sigma^2 estimated as 29891418:  log likelihood=-2394.61
## AIC=4797.21   AICc=4797.39   BIC=4811.12
```

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?

The residuals do seem to be random, there is fairly even variation from negative to positive residuals, and no apparent trends. The residuals seem to be evenly spaced leading me to conclude that the residuals do look like a white noise series. The residuals in the ACF appear to have some dependency as several of the residuals are above the blue line indicating some of the residuals have dependency on each other. There is a seasonal correlation with this dependency as the error term likely has some seasonal variation that is not taken into account when the series was deseasoned.

```
checkresiduals(ARIMA_manual)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0) with drift
## Q* = 74.484, df = 21, p-value = 6.6e-08
##
## Model df: 3.   Total lags used: 24
```

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.

There are multiple spikes in the seasonal lags of the ACF plot.There appears to be one seasonal spike around the seasonal lag of the PACF. Conclude that this is an SAR Process.

Based on the auto-correlation the number of of explanatory variables that you need to accurately portray the model is 2 thus the order of Q=2. I Concluded there was no SMA process.

After running the ns_diff function, which is the the seasonal differential function the output returned was 1 D=1

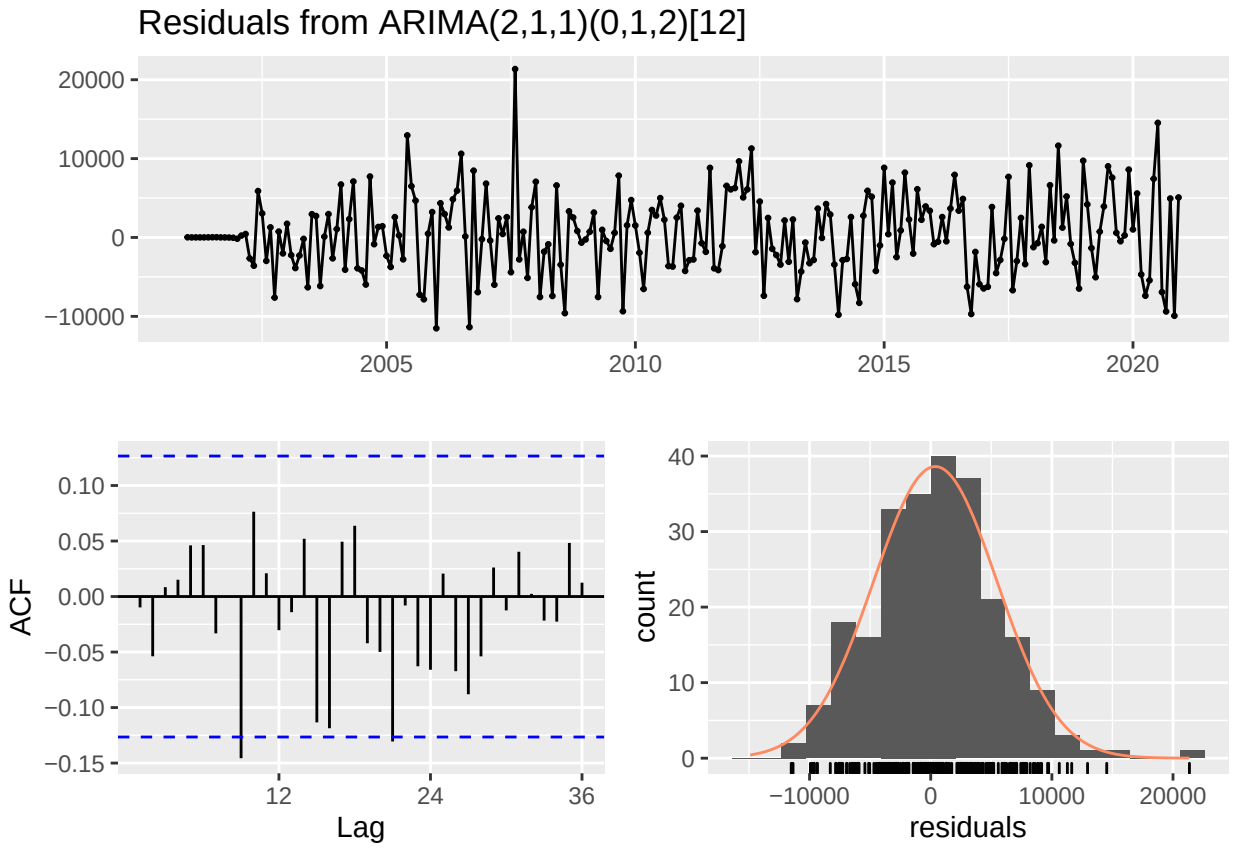
(0,1,2)

```
niff3<- nsdiffs(ts_first)

SARIMA_manual <- Arima(ts_first,order=c(2,1,1),seasonal=c(0,1,2),include.drift=FALSE)
print(SARIMA_manual)
```

```
## Series: ts_first
## ARIMA(2,1,1)(0,1,2)[12]
##
## Coefficients:
##          ar1     ar2      ma1     sma1    sma2
##       0.6939  0.0591  -0.9848  -0.7097  0.0211
## s.e.  0.0673  0.0683   0.0183   0.0747  0.0729
##
## sigma^2 estimated as 28091865:  log likelihood=-2271.79
## AIC=4555.59   AICc=4555.97   BIC=4576.14
```

```
checkresiduals(SARIMA_manual)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)(0,1,2)[12]
## Q* = 26.58, df = 19, p-value = 0.1148
##
## Model df: 5.   Total lags used: 24
```

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.

The AIC is the maximum likelihood for parametric model and ARIMA model is a parametric model and the thus the lower the AIC the better fit. The ARIMA AIC is higher in Q6 than Q7. This makes a great deal of sense as Q7 interprets the seasonal component and the model has a clear seasonal component

Less residuals fell outside the ACF significant region for Q7 which makes sense as it appeared like the residuals in Q6 (the ARIMA that did not include a seasonal component showed some seasonal variability)

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?

The ARIMA order is close to the one I selected the order found by R was 3,1,0 and I had selected 2,1,0. I was under the impression that order was typically less than or equal to 2, while lag 3 is nearly significant I had interperated that it was not significant.

```
desasonal <- auto.arima(deseasonal_natural)
print(desasonal)
```

```
## Series: deseasonal_natural
## 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 estimated as 26980609:  log likelihood=-2383.11
## AIC=4774.21   AICc=4774.38   BIC=4788.12
```

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?

The seasonal component was (0,1,1) while I had selected (0,1,2). The difference from the model I had selected and the auto.arima is the P term. I had concluded that this was an SAR process from the multiple spikes in the seasonal lags of the ACF plot. I had selected order 2 becuase of the mltiple spikes in the ACF plot, and the one cut of spike in the PACF plot.I know see why the order is 1.

There appears to be one seasonal spike around the seasonal lag of the PACF, which led me to Conclude that this is an SAR Process. I did not see any indication that this was an SMA process which is why I identified P=0, and this was confirmed running auto.arima

```
orgional<- auto.arima(ts_first)
print(orgional)
```

```
## Series: ts_first
## 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 estimated as 27569123:  log likelihood=-2279.54
## AIC=4567.08   AICc=4567.26   BIC=4580.8
```